



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería Eléctrica y Computación

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DOMÓTICO USANDO
LABVIEW Y MICROCONTROLADORES”**

INFORME DE PROYECTO DE GRADUACIÓN

Previo a la obtención del Título de:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y
AUTOMATIZACIÓN INDUSTRIAL**

Presentado por:

Carlos Roberto Luo Xu
Marianela Del Cisne Ochoa Piña

GUAYAQUIL – ECUADOR

2015

AGRADECIMIENTO

A Dios, por ser la guía de nuestro camino y fuente de bendiciones, por alentarnos y brindarnos esperanza.

A nuestros padres, por su apoyo y consejos para nuestro diario vivir, por inculcarnos metas y por sus deseos de vernos triunfantes.

A nuestros familiares y amigos, por su apoyo para que fuese posible la realización del proyecto.

Al Ing. Holger Cevallos por su guía a través del desarrollo de nuestro proyecto, por sus consejos y ejemplo de profesionalismo.

A Patricia, Martín y César por su predisposición a orientación y ayuda incondicional.

DEDICATORIA

A nuestros padres y demás familiares,
pilares fundamentales de nuestras vidas,
presentes durante el tiempo de
aprendizaje y formación profesional.

TRIBUNAL DE SUSTENTACIÓN

ING. SARA RIOS O.

SUBDECANA FIEC

ING. DAMIAN LARCO GOMEZ

PROFESOR DELEGADO POR LA

UNIDAD ACADÉMICA

ING. HOLGER CEVALLOS

PROFESOR DEL PROYECTO DE

GRADUACIÓN

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de este informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma, a la Escuela Superior Politécnica del Litoral”.

(Reglamento de Graduación de la ESPOL)

Carlos Roberto Luo Xu

Marianela Del Cisne Ochoa Piña

RESUMEN

El presente proyecto consiste en crear un sistema domótico usando como interfaz de programación gráfica Labview (Laboratory Virtual Instrument Engineering Workbench) e incorporando microcontroladores en marca MICROCHIP Y ARDUINO que aportarán al funcionamiento del mismo.

La tarjeta que funcionará como adquisición de datos es la Arduino SainSmart mega 2560 el cual previamente para que establezca comunicación con Labview se cargó el LIFA BASE haciendo uso del entorno IDE de Arduino.

Inicialmente se realizó una investigación de los componentes domóticos presentes en el mercado actual de las empresas como Bticino, SimonVis y Domobak; y a la vez se realiza un estudio de instalación domótica.

Luego se explica el funcionamiento tanto de la programación en Labview 2013 como de la programación en MICROCHIP (PIC16F887) y Arduino (Uno y MEGA 2560), para uso de los mismos en nuestro proyecto.

Finalmente se presentará el resultado de las pruebas realizadas en la maqueta en tiempo real.

INDICE GENERAL

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE SUSTENTACIÓN	IV
DECLARACIÓN EXPRESA	V
RESUMEN.....	VI
ABREVIATURAS	X
INDICE DE FIGURAS	XII
INDICE DE TABLAS	XVIII
INTRODUCCION	XIX
CAPÍTULO 1.....	1
DESCRIPCIÓN GENERAL DEL PROYECTO	1
1.1 DESCRIPCIÓN DEL PROBLEMA	1
1.2 SOLUCIÓN DEL PROBLEMA	3
1.3 JUSTIFICACIÓN.....	3
CAPÍTULO 2.....	6
FUNDAMENTO TEÓRICO	6
2.1 DEFINICIONES DE LA DOMÓTICA.....	6
2.2 TIPOS DE GESTIONES DE LA DOMÓTICA	8
2.3 TIPOS DE DISPOSITIVOS USADOS	11
2.4 DIAGRAMA DE FLUJO DE LAS GESTIONES DOMOTICAS Y LOS DISPOSITIVOS A USAR	12
CAPÍTULO 3.....	15

METODOLOGÍA	15
3.1 ANÁLISIS DE COSTOS DE LOS DISPOSITIVOS DEL MERCADO ..	15
3.2 ESTUDIOS DE CAMPOS DE INSTALACIONES DOMÓTICAS	29
3.3 DESCRIPCIÓN DE LOS DISPOSITIVOS A UTILIZAR	31
CAPÍTULO 4	33
ANÁLISIS Y DISEÑOS DE LA MAQUETA	33
4.1 PLANOS DE LA VIVIENDA	33
4.2 PLANO ELÉCTRICO	38
4.3 PLANOS DE UBICACIÓN DE LOS DISPOSITIVOS	40
CAPÍTULO 5	44
PROGRAMACIÓN DE LAS GESTIONES DOMÓTICAS EN LABVIEW	44
5.1 SISTEMAS DE CONTROL DE ILUMINACIÓN	44
5.1.1 ESCENARIO POR PRESENCIA	44
5.1.2 DIMERIZACIÓN POR AMBIENTE	55
5.1.3 CONTROL POR TIEMPO	64
5.1.4 CONTROL MANUAL	68
5.1.5 AMBIENTACIÓN	70
5.2 SISTEMAS DE PERSIANAS	74
5.3 SISTEMA DE SEGURIDAD	81
5.3.1 MEMORIAS DE RUTINA PARA SIMULAR PRESENCIA DE UNA PERSONA EN LA CASA	81
5.3.2 SISTEMA POR VOZ Y TARJETA PARA PUERTA	84
5.4 SISTEMA DE RIEGO	97

5.5 SISTEMA DE GARAGE	99
5.6 DISEÑO DE PRUEBAS	103
CAPÍTULO 6.....	117
MONTAJE Y ACOPLAMIENTO DE LOS DISPOSITIVOS ELECTRÓNICOS.....	117
6.1 SOFTWARE Y DISPOSITIVOS ELECTRÓNICOS	117
6.2 ACOPLAMIENTO GENERAL	159
CAPÍTULO 7.....	163
ANÁLISIS DE RESULTADOS.....	163
7.1 ELEMENTOS UTILIZADOS Y COSTOS.....	163
7.2 PRESUPUESTOS	166
7.3 PRUEBAS Y RESULTADOS DE PRUEBAS	167
CONCLUSIONES Y RECOMENDACIONES ..	¡Error! Marcador no definido.
BIBLIOGRAFÍA.....	176
ANEXOS.....	179

ABREVIATURAS

ASK	Amplitude-Shift Keying
DAQ	Data Acquisition - Adquisición de Datos
HIGH	Alto
ID	Identidad
LABVIEW	Laboratory Virtual Instrument Engineering Workbench - Laboratorio Virtual banco Instrumento de Trabajo de Ingeniería
LED	Light-Emitting Diode - Diodo Emisor de Luz
LDR	Light-Dependent Resistor - Resistencia Dependiente de la Luz
LOW	Bajo
MISO	Master Input, Slave Output - Entrada del Maestro y Salida del Esclavo
MOSI	Master Output, Slave Input - Salida del Maestro y Entrada del Esclavo
NFC	Near Field Communication - Comunicación de Campo Cercano
OFF	Apagado
ON	Encendido
PIC	Peripheral Interface Controller - Controlador de Interfaz Periférico
PWM	Pulse-Width Modulation - Modulación de Ancho de Pulso
RFID	Radio Frequency Identification - Identificación por Radiofrecuencia

RX	Recepción
SCK	Serial Clock
SPI	Serial Peripheral Interface
SS	Slave Select - Selección de Esclavo
TX	Transmisión
UART	Universal Asynchronous Receiver-Transmitter - Transmisor- Receptor Asíncrono Universal

INDICE DE FIGURAS

Fig. 2.1 Gestiones de la domótica.....	12
Fig. 2.2 Organizador gráfico de las gestiones domóticas	13
Fig. 3.1 Plano eléctrico-domótico planta alta.....	29
Fig. 3.2 Plano eléctrico-domótico planta baja.....	30
Fig. 4.1 Vista frontal sur	34
Fig. 4.2 Vista lateral sur	35
Fig. 4.3 Perspectiva oeste.....	35
Fig. 4.4 Perspectiva este.....	35
Fig. 4.5 Plano de la planta baja.....	36
Fig. 4.6 Plano de la planta alta.....	37
Fig. 4.7 Plano eléctrico de la planta baja.....	38
Fig. 4.8 Plano eléctrico de la planta alta.....	39
Fig. 4.9 Simbología usada en los planos eléctrico.....	40
Fig. 4.10 Plano del sistema domótico planta baja	41
Fig. 4.11 Plano del sistema domótico planta alta.	42
Fig. 4.12 Simbología empleada para el sistema domótico	43
Fig. 5.1 Panel frontal del control por escenario	45
Fig. 5.2 Programación del sistema de escenario.....	47
Fig. 5.3 Escenario#1 uso de variable local	48
Fig. 5.4 Escenario#1 variable local de planta baja y alta	49
Fig. 5.5 Escenario#2 secuencia de encendido- garaje	50

Fig. 5.6 Escenario#2 secuencia de encendido- cocina.....	51
Fig. 5.7 Escenario#2 secuencia de apagado- dormitorio.....	51
Fig. 5.8 Escenario#2 secuencia de apagado- estudio	52
Fig. 5.9 Escenario#3 ambientación y persiana	53
Fig. 5.10 Programación de ambientación-escenario 3	53
Fig. 5.11 Programación de persiana-escenario 3	54
Fig. 5.12 Control programable del sistema de iluminación	55
Fig. 5.13 Control del sensor lumínico pasillo 1	57
Fig. 5.14 Control del sensor lumínico pasillo 2	58
Fig. 5.15 Control del LDR.....	59
Fig. 5.16 Control de horas para el sensor	60
Fig. 5.17 Control del pulsador ON/OFF	61
Fig. 5.18 Control manual planta baja del panel frontal	61
Fig. 5.19 Control programable y dimmer para la sala	62
Fig. 5.20 Control manual planta alta del panel frontal	63
Fig. 5.21 Control programable.....	64
Fig. 5. 22 Control de encendido de luminarias por tiempo.....	65
Fig. 5.23 Control programable para la cocina.....	67
Fig. 5.24 Control manual diagrama de bloques.....	68
Fig. 5.25 Programación para control manual planta alta	69
Fig. 5.26 Código ambientación en IDE Arduino.....	73
Fig. 5.27 Controles para ambientación.....	73

Fig. 5.28 Control programable de ambientación.....	74
Fig. 5.29 Control del sistema de persiana	80
Fig. 5.30 Indicadores string–rutina de seguridad.....	81
Fig. 5.31 Programación rutina de seguridad.....	82
Fig. 5.32 Programación de captura de imagen.....	84
Fig. 5.33 Programación de cierre de captura de imagen	86
Fig. 5.34 Programación de control para los módulos	95
Fig. 5.35 Bloques para configuración de servo	96
Fig. 5.36 Panel frontal del sistema de seguridad.....	97
Fig. 5.37 Control del sistema de riego.....	98
Fig. 5.38 Control remoto para el garaje.....	102
Fig. 5.39 Prueba uno para escenarios	104
Fig. 5.40 Prueba dos para dimerización.....	105
Fig. 5.41 Prueba tres dimerización por hora.....	106
Fig. 5.42 Prueba cuatro para el control por tiempo.....	107
Fig. 5.43 Prueba cinco-panel frontal	108
Fig. 5.44 Prueba seis del control para ambientación.....	109
Fig. 5.45 Prueba siete para la captura de imágenes	110
Fig. 5.46 Prueba ocho para el sistema de riego.....	111
Fig. 5.47 Prueba nueve para la sección dos	112
Fig. 5.48 Código para la prueba nueve	115
Fig. 5.49 Código de la prueba final	116

Fig. 6. 1 Organizador gráficos de los controles implementados	119
Fig. 6.2 Comunicación entre pc y tarjeta.....	120
Fig. 6.3 Modelo de la trama de bits	121
Fig. 6.4 Esquema de puente UART a USB	122
Fig. 6.5 Pin1 Tx.....	123
Fig. 6.6 Pin 0 RX.....	124
Fig. 6.7 Data -	124
Fig. 6.8 Data +	125
Fig. 6.9 Arduino Mega 2560	125
Fig. 6.10 Descripción de los pines de la interfaz USB	127
Fig. 6.11 Administrador de dispositivos.....	128
Fig. 6.12 Bloque Init- Labview.....	129
Fig. 6.13 Configuración del bloque Init	130
Fig. 6.14 Programación interna del bloque Init.....	131
Fig. 6.15 Placa con Relays	133
Fig. 6.16 Placa electrónica Dimmer.....	134
Fig. 6.17 Placa con los pics PIC16F887	136
Fig. 6.18 ULN2003APG Driver.....	137
Fig. 6.19 Motor de pasos 28BYJ-48.....	138
Fig. 6. 20 Placa Arduino UNO.....	140
Fig. 6.21 Arduino UNO y NFC PN532	141
Fig. 6. 22 Conexión entre la Arduino y la tarjeta.....	142

Fig. 6.23 Comunicación serial SPI.....	143
Fig. 6.24 Módulo EASYVR.....	147
Fig. 6. 25 Trama de bits.....	147
Fig. 6. 26 Diagrama de conexión.....	149
Fig. 6. 27 Ancho de pulso y ángulo de giro del motor.....	150
Fig. 6.28 Modo de envío de la trama para la comunicación.....	152
Fig. 6.29 Módulo de Transmisión.....	152
Fig. 6.30 Módulo de recepción.....	153
Fig. 6.31 Obtención de los datos.....	154
Fig. 6.32 Tiempos de los códigos.....	155
Fig. 6.33 Trama de datos de los LEDS.....	156
Fig. 6.34 Imagen de la Tira de LEDS WS2812B.....	157
Fig. 6.35 Imagen de los conectores de la tira de LEDS.....	157
Fig. 6.36 Comunicación al Conjunto de LEDS.....	159
Fig. 6.37 Acoplamiento del sistema.....	160
Fig. 6.38 Dimmer baquelita.....	161
Fig. 6.39 Control Remoto y Receptor.....	161
Fig. 6.40 PIC para los motores de paso.....	162
Fig. 6.41 Relays.....	162
Fig. 7.1 Resultado uno en el cual se visualizan los LEDS encendidos..	167
Fig. 7.2 Resultado de escenario.....	168
Fig. 7.3 Resultado del control lumínico.....	168

Fig. 7.4 Resultado del control.....	169
Fig. 7. 5 Resultado de la maqueta	170
Fig. 7. 6 Resultado del control de persianas.....	172
Fig. 7. 7 Resultado del control de tira de LEDS.....	172

INDICE DE TABLAS

Tabla # 1 Módulos utilizados.....	11
Tabla # 2 Precios de productos Bticino.....	21
Tabla # 3 Precios de productos SimonVis.....	26
Tabla # 4 Precios de productos Domobak.....	28
Tabla # 5 Costos de software de empresas.....	28
Tabla # 6 Descripción de los módulos.....	32
Tabla # 8 Descripción de puertos.....	126
Tabla # 9 Descripción del cable.....	126
Tabla # 10 Descripción de los bloques de configuración.....	130
Tabla # 11 Funciones del SPI.....	142
Tabla # 12 Precios de elementos adquiridos.....	165
Tabla # 13 Precios de materiales adquiridos.....	166
Tabla # 14 Inversión total.....	166

INTRODUCCION

El presente proyecto de graduación tiene el fin de crear un sistema domótico que será demostrativo para la materia Domótica, que permita el diseño de automatizar una vivienda usando microcontroladores y el software Labview como interfaz gráfica para el usuario, ampliando los conocimientos de programación en Labview, microcontroladores y automatización, cuyo fin es la demostración que es posible implementar este sistema domótico utilizando los conocimientos académicos adquiridos sin necesidad de contratar empresas dedicadas a la automatización de viviendas.

Los capítulos se encuentran estructurados de la siguiente manera:

En el capítulo uno se describe el problema, la solución presentada y la justificación del desarrollo del presente proyecto.

En el capítulo dos se detalla las definiciones de la domótica así como las gestiones que presenta, se indica los dispositivos más comunes a usar en el momento de cotizar una instalación domótica y un diagrama de flujo que permite visualizar las gestiones y los dispositivos domóticos.

En el Capítulo tres se describe los dispositivos domóticos de las marcas Myhome, Insteon y Simonvis presentes en el mercado y el costo de adquisición de cada uno ellos y se presenta un estudio de una posible instalación domótica.

En el Capítulo cuatro se puede visualizar los planos de la vivienda que se tomó como modelo para la construcción de la maqueta.

En el Capítulo cinco se detalla la programación en Labview realizada; y se incluye las programaciones en Arduino y Mikroc PRO for Pic

En el Capítulo seis se hace referencia al software utilizado, a la programación y el modo de comunicación para verificar que los dispositivos electrónicos seleccionados funcionen de acuerdo a lo esperado.

En el Capítulo siete se presenta los elementos electrónicos y módulos usados para la demostración en la maqueta, así como el costo de cada uno de ellos, presupuesto para la construcción e implementación de la maqueta domótica, pruebas y el resultado que se obtuvo de cada una de las pruebas.

CAPÍTULO 1

DESCRIPCIÓN GENERAL DEL PROYECTO

1.1 DESCRIPCIÓN DEL PROBLEMA

La tecnología y la ciencia con el pasar de los años se han desarrollado notoriamente evolucionando e innovando para satisfacer las crecientes necesidades no solo en el área industrial sino también en el hogar, originando avances domóticos e inmóticos cuyo fin es aportar servicios de gestión energética, seguridad, bienestar y comunicación.

Obtener todos estos servicios que la línea domótica ofrece, requiere tener en consideración sistemas a gestionar en el dimensionado de la casa que se posea, los elementos a implementar para dar el servicio,

el costo de mano de obra y el costo de cada elemento incorporado, todo esto implica grandes costos¹ de inversión lo que significa que solo lo podrá obtener personas de un nivel económico alto.

Por tal motivo da como resultado a que surja la necesidad de desarrollar un sistema alternativo para domotizar la vivienda usando el entorno de programación gráfica Labview como interfaz para el usuario y la incorporación de microcontroladores; cuya finalidad es brindar los mismos beneficios de seguridad, control y confort que la domótica ofrece.

Además, la materia de Domótica no cuenta con un laboratorio para que sus estudiantes realicen prácticas relacionadas a la automatización de viviendas o creación de proyectos, que permita reforzar los conocimientos académicos adquiridos.

Con esto aspiramos a desarrollar un sistema domótico que pueda ser más accesible y cómodo para el usuario; y, que pueda ser demostrado en las clases de Domótica e Inmótica.

¹ Véase el anexo3 en el cual se adjunta la proforma dada por Bticino.

1.2 SOLUCIÓN DEL PROBLEMA

Desarrollar una aplicación domótica usando el software de ingeniería Labview y dispositivos que se puedan obtener en el mercado nacional como son los módulos Arduino, microcontroladores o PIC16F887, etc.

La maqueta fue creada con el fin de realizar las demostraciones del sistema domótico implementado para verificar que se lo puede aplicar a una vivienda real, y además; podrá ser usado en alguna clase demostrativa en la materia de Domótica e Inmótica ya que integra los conceptos de automatización en vivienda y controles usando la interfaz gráfica Labview.

1.3 JUSTIFICACIÓN

Los estudiantes de la materia de Domotica e Inmotica necesitan reforzar los conocimientos teóricos adquiridos realizando prácticas, proyectos demostrativos y exposiciones, manipulando el sistema domótico desarrollado o creando nuevos sistemas que puedan ser demostrados en la maqueta.

Además, se considera que el sistema propuesto es más económico² comparado con los existentes en el mercado ya que empresas como Bticino utilizan módulos eléctricos compatibles con el software que usan, propios de la línea de ellos y sus equipos³ domóticos son importados haciendo que el costo total de instalación sea extremadamente elevado.

Por tal motivo se implementará un sistema que permita realizar los servicios domóticos que ofrecen otras empresas, incorporando tarjetas Arduino, microcontroladores y demás módulos que permitan acoplarse al Arduino para acceder a las gestiones que la domótica ofrece mejorando el estilo de vida de todas las personas, brindando seguridad, confianza, ahorro energético, teniendo un mayor control del entorno, y todo esto ayuda a mejorar el estado emocional de la persona haciendo que el sistema sea accesible y cómodo para el usuario, teniendo un servicio de calidad.

Con el fin de cubrir lo anteriormente mencionado es necesario utilizar un software de ingeniería que permita controlar todas las gestiones por computadora.

² Véase la tabla# 3 que es el cuadro comparativo de costos de software.

³ Véase la tabla# 2 en la que se describen algunos de los módulos eléctricos y domóticos de Bticino.

Se empleará Labview el cual posee una gran variedad de herramientas como Vision⁴ development y Labview⁵ Interface for Arduino; bloques gráficos que facilitan la programación.

Para el desarrollo del proyecto se usará microcontroladores, circuitos electrónicos y el software Labview que será la interfaz de usuario.

⁴ Es un toolkit o herramienta que permite la interface entre Labview y la webcam

⁵ Es un toolkit o herramienta que permite la interface entre Labview y la tarjeta Arduino mega2560

CAPÍTULO 2

FUNDAMENTO TEÓRICO

2.1 DEFINICIONES DE LA DOMÓTICA

El término domótica viene del Latín DOMUS, “casa” y del término, “TICA” de robótica; también conocida en inglés como home automation, concretamente es la automatización del hogar. [1]

En la actualidad se considera a la domótica como la integración de nuevas tecnologías al espacio arquitectónico; cabe recalcar que la domótica busca aportar en el usuario una mejor calidad de vida con la incorporación de equipos que permitan gestionar de forma eficiente,

segura y confortable los distintos sistemas domésticos que integra una vivienda.

Un sistema domótico dispone de una red de comunicación y diálogo con lo cual se interconecta con los equipos a fin de obtener información sobre el entorno doméstico; y, permite que la vivienda posea un entorno más funcional y acorde a cada usuario. [2]

Las técnicas empleadas para automatizar hogares, son las incluidas en la automatización de construcciones para el control de actividades domésticas, tales controles incluyen los sistemas de entretenimiento, sistemas de riego, alimentación de mascotas, utilización de escenas para cambiar el ambiente según el evento, etc.

También se lo denomina al término domótica como gestión técnica de edificios y viviendas ya que incluye de manera simultánea la utilización de electricidad, electrónica e informática, gestionando la comunicación entre el usuario y la vivienda. [3]

Al hablar de este concepto encontramos que la domótica presenta dos tipos de filosofías principales:

La primera filosofía corresponde a los sistemas domóticos restringidos, se llaman así porque no permiten utilizar mecanismos (interruptores, pulsadores, etc.) que desee el usuario, sino que limitan a los modelos y marcas que son compatibles con sus sistemas y estándares de control.[4]

La segunda filosofía se refiere a la gestión, con esta filosofía trabajan los fabricantes del país de la domótica (home automation), estos sistemas controlan todo lo que se instala en las viviendas como por ejemplo puertas automáticas, electroválvulas, toldos, televisores, persianas, luces, etc. [4]

2.2 TIPOS DE GESTIONES DE LA DOMÓTICA

La domótica persigue la integración de 4 aspectos básicos en el hogar; siendo estos: confort, seguridad, comunicaciones y gestión energética. [2]

GESTIÓN DEL CONFORT:

Proporciona comodidades ya que se ejerce control en accesos como persianas, riego, toldos, ventanas, puertas, etc.

Permite establecer controles para la climatización que puede ser manejado mediante control remoto para tener diferentes temperaturas en el día o programarlo según la zona climática cuya finalidad es tener temperaturas ideales al llegar a la vivienda.

También es posible tener el control de la iluminación regulando los niveles de intensidad a diferentes horas del día, o estableciendo controles de encendido y apagado usando sensores de presencia. [5]

GESTIÓN DE LA SEGURIDAD:

Evita riesgos y accidentes domésticos con la integración de detectores de fugas que pueden ser de gas, agua y fuego.

Además permite la protección de bienes gestionando el control de acceso a la vivienda, incorporando controles para la simulación de presencia cuya finalidad es repetir patrones de comportamientos de las personas en el momento que la casa se encuentre deshabitada, incluye también alarmas contra intrusiones.

Con la gestión de seguridad es posible brindar auxilio a las personas especialmente de edades mayores con la realización de llamadas telefónicas y accedendo a los servicios de ambulancias, policía, etc.

GESTIÓN DE LA COMUNICACIÓN:

Está íntimamente relacionada con las funcionalidades que tengamos definidas en el hogar puesto que se debe dar la posibilidad de expandir el sistema teniendo controles externos e internos, visualizando aquellos parámetros de interés, permitiendo la entrada de datos, transmisión de alarmas, y las intercomunicaciones.

Incorpora la tele asistencia, tele mantenimiento, informes de consumo y costos.

GESTION ENERGÉTICA:

Permite gestionar el accionamiento automático de luces usando detectores de presencia, de esta manera se controlará el consumo energético manteniendo apagadas las demás luminarias que han sido instaladas en la vivienda y regula la intensidad de iluminación censando la luminosidad del día.

También controla las conexiones de cargas a distintas horas para reducir la potencia contratada y ejercer el control de climatización.

La gestión energética permite la subida automática de persianas al detectar luz solar para aprovechar la iluminación natural. [6]

2.3 TIPOS DE DISPOSITIVOS USADOS

DISPOSITIVO	DESCRIPCION
Arduino mega 2560	Será conectado directamente al computador y a los diferentes módulos.
Arduino UNO	Para la programación del módulo Shield NFC PN532.
Arduino UNO	Usado para la programación del módulo de reconocimiento de voz.
Arduino UNO	Usado para la programación de la tira de LEDS.
Módulo Shield NFC pn532	Módulo lector de tarjeta magnética compatible con Arduino.
Módulo EasyVR Shield 2.0	Módulo de reconocimiento de voz adaptable a la programación con Arduino.
Cámara zen	Usado para la captura de imágenes.
Tira led strip	Incluye el integrado WS2812B ⁶ , tira programable para el control de los colores en los LEDS RGB. [7]

Tabla # 1 Módulos utilizados

⁶ Para conocer con más detalle el integrado véase el anexo en el cual se adjunta el datasheet.

Se ha realizado un breve detalle de los módulos que ayudarán al funcionamiento del sistema domótico que será demostrado en la maqueta y se ha omitido todos los componentes electrónicos debido a que ayudarán a los diseños de fuerza y porque están relacionados a la electrónica que se ha estudiado.

2.4 DIAGRAMA DE FLUJO DE LAS GESTIONES DOMOTICAS Y LOS DISPOSITIVOS A USAR

Se presenta la Fig. 2.1 en la cual podemos visualizar la distribución de las aplicaciones domóticas que se ha implementado en la maqueta.

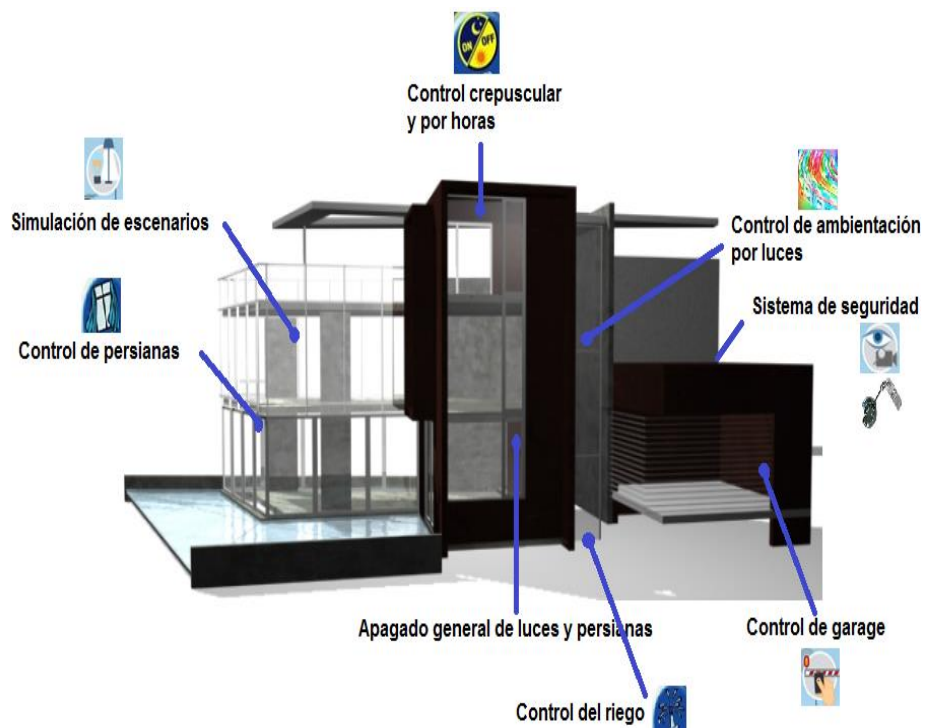


Fig. 2.1 Gestiones de la domótica.

El modelo arquitectónico que se visualiza en la Fig. 2.1 fue creado por el Arq. Segal Faia ⁷.

Para especificar los controles creados por gestión se ha elaborado un organizador gráfico; el mismo que se encuentra representado en la Fig. 2.2:

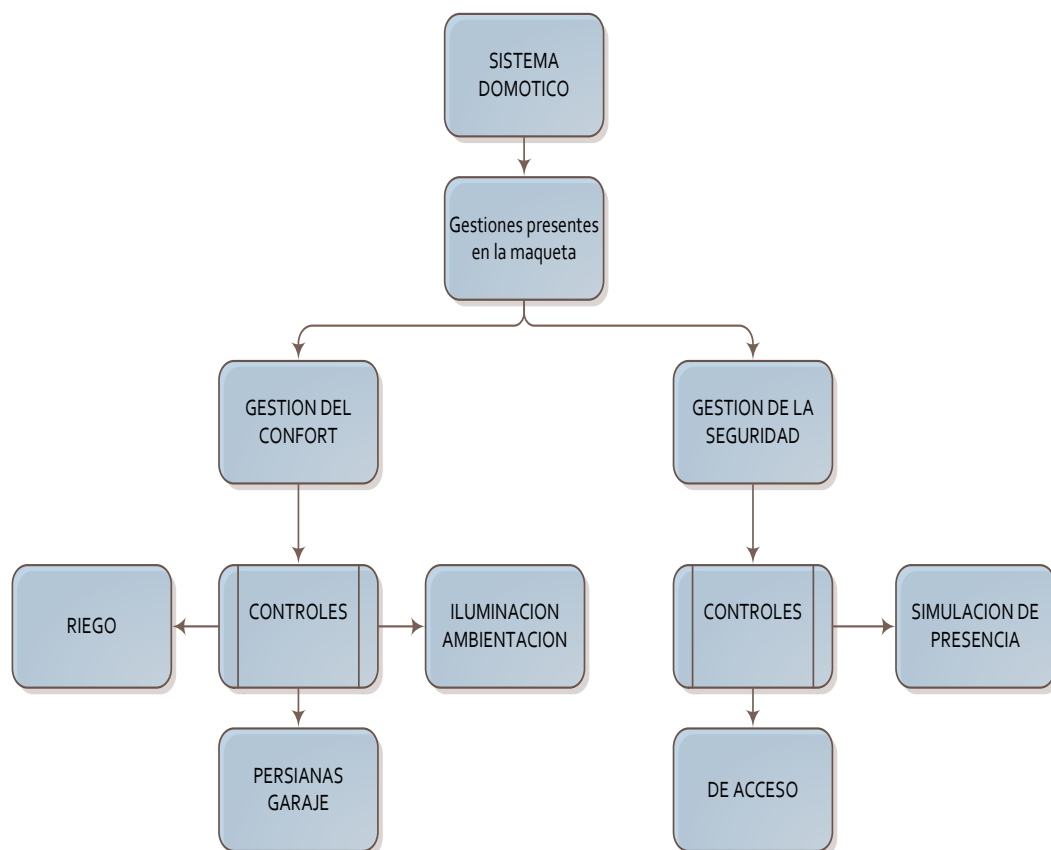


Fig. 2.2 Organizador gráfico de las gestiones domóticas

⁷ Arquitecto de la ciudad de San Diego California, premiado por su desarrollo y tendencia única en arquitectura; para más información visite el sitio web: <http://www.jonathansegalarchitect.com/index.html>

Los sensores que se ha implementado en la maqueta corresponden a los módulos EasyVR 2.0⁸, shield NFC PN532⁹, y el sensor lumínico LDR.

Los actuadores presentes en la maqueta son: caja de relays, caja de dimmers, WS2812B¹⁰, ULN2003APG¹¹, servomotor.

La descripción de los módulos usados como sensores están detallados en el capítulo seis y de la misma manera para los actuadores.

⁸ Véase el capítulo seis en el cual se describe el funcionamiento del mismo.

⁹ Examínese el capítulo seis en el cual se describe el funcionamiento y véase los anexos en el cual se adjunta el datasheet.

¹⁰ Véase los anexos en el cual se adjunta el datasheet.


¹¹ Obsérvese los anexos en el cual se adjunta el datasheet.

CAPÍTULO 3

METODOLOGÍA

3.1 ANÁLISIS DE COSTOS DE LOS DISPOSITIVOS DEL MERCADO









En la tabla que se presenta a continuación se detalla el costo y el modelo de algunos de los dispositivos que se pueden encontrar en la marca Bticino para la automatización de persianas, luces, seguridad así como el software para el funcionamiento de los mismos.

ARTICULO	MODELO	DESCRIPCION	CANTIDAD	COSTOS
AM5704		Dimmer electrónico, 500 W, 127 Vac, 2 mod.	6	1198,79

AM4406		Dimmer resistivo, 300 W, 127Vac.	30	554
AM5356R*		Dimmer deslizable, 1000W, 127Vac.	10	961,41
AM5377R*		Detector de movimiento de pared 180° de cobertura 800/1200W, 127/277V con placa.	10	1578,88
316311		Sistema de videointerfono 4 hilos c/monitor manos libres.	1	3567,42
316411		Pantalla LCD color 3.5" cámara oculta con visión nocturna permite crecimiento hasta dos monitores adicionales.	1	4338,6
316611		Sistema de videointerfono a color con monitor lcd 7"	1	6486,22
316811		Sistema de videointerfono a color con monitor lcd 7"	1	8432,08
316211		Sistema de videointerfono 4 hilos	1	3488,75
331551		Videoportero con auricular	1	2229,64
331651		Videoportero sin auricular	1	2308,34

331751		Videoportero con auricular en color	1	2774,82
369411		Videoportero con controles domóticos y video para la puerta de la casa, kit de manos libres con memoria de llamada e imágenes.	1	1075,82
HD4690 HC4690 HS4690		Funciones de videoporteros y videocontrol, gestión de contenidos multimedia presentes en fuentes externas (usb y SDCard), en red LAN, IP.	1	2271,92
F425		Módulo de memoria para el estado de los actuadores	1	51,1
369521		Videoportero con funciones de automatización Myhome	1	2089,37
F420		Dispositivo para memorizar 16 escenarios	1	42,86
MH200N		Dispositivo para la ejecución de escenarios programados con el software TIMH200N.	1	570,41
346020		Alimentador compacto para alimentar el programador de escenarios MH200N-2 módulos DIN	1	82,92
H4651M2		Comando para funciones especiales tales como activación de 4 escenarios memorizados con el módulo F420, comandos, dimmer activación de cerraduras	1	80,104

HA4596 HB4596		Mando integrable con una cubretecla de 1 módulo para el control on/off de grupo de actuadores.	1	35,56
HA4597 HB4597		Mando integrable con 2 cubreteclas de 1 módulo para el control on/off de 2 grupos de actuadores.	1	37,81
HA4598 HB4598		Mando integrable con 2 cubreteclas de 1 módulo para el control de 1 grupo de actuadores.	1	43,27
H4590		Actuador integrable con cubretecla de 1 módulo para accionamiento de lámparas.	1	103,08
H4591		Actuador integrable con cubretecla de 1 módulo para accionamiento de lámparas.	1	193,15
H4592		Actuador integrable con cubretecla de 1 módulo para accionamiento de lámparas.	1	67,54
HA4599 HB4599	-----	Modulo radio para automatización de persianas.	1	55,69
HA4589 HB4589		Mando radio para la gestión de escenarios.	1	32
H4593		Actuador radio dimmer para automatización de luces.	1	69,3

H4594		Actuador radio dimmer con 2 cubreteclas de 1 módulo.	1	71,1
H4595		Actuadores radio para automatización de persianas.	1	71,1
N4578N NT4578N L4578N	-----	Interfaz de radio para la integración de los sistemas de radio.	1	79,92
L4599N N4599N		Mando radio para automatización de persianas.	1	42,83
N4589N L4589N		Mando radio para la gestión de escenarios.	1	24,608
N4596N L4596N		Mando radio para la automatización de luces con 1 cubretecla de 1 módulo.	1	27,34
N4597N L4597N		Mando radio para la automatización de luces con 2 cubreteclas de 1 módulo.	1	29,158
LN4595		Actuadores radio para automatización de persianas.	1	54,685
3527N		Mando a distancia radio	1	53,23

3528N		Mando a distancia radio	1	39,27
3579		Mando radio por infrarojo para cargas generales, dotado de detector de movimiento.	1	36,453
3576		Actuadores radio para automatización de persianas.	1	42,83
HD4607 HC4607 HS4607		Mando protegido con código, memoriza hasta 30 identificativos.	1	91,77
HD4607Z4		Comando escenarios, mando protegido con código hasta 30 identificativos.	1	117,85
3530S		Identificativo portátil	1	5,41
BMSE1001		Sensor de iluminación/ movimiento.	1	81,53
MHKIT10N		Kit automatización para 5 persianas o cortinas motorizadas.	1	364,5
BMSE2002	-----	Sensor de iluminación/ movimiento.	1	44,98
BMSE2005		Sensor de iluminación/ movimiento.	1	65,85














F411/1N	 F411/1N	Actuadores para centralizaciones con 1 relé de dos vías.	1	36,18
BMSW1003		Actuador para centralizaciones de 4 salidas.	1	109,29
391661 391662 391663		Telecámara de interior para instalaciones de 2 hilos	1	4331,69
H4687		Panel pc de empotrar con pantalla táctil para el control del sistema MY HOME mediante el software MHVisual	1	3492,41
MHVisual		Software MHVisual de supervisión de las funciones de iluminación, automatización, escenarios, videocontrol, gestión energética y sonora.	1	878,18
N4680		Comando escenarios personalizado para accionar 4 situaciones ambientales de automatización	1	50,93
3477		Interfaz de contactos para accionamientos con funciones dobles para persianas, interruptores.	1	24,1
F425		Módulo de memoria para el estado de los actuadores para restablecer el sistema de luces anti black out.	1	51,1

Tabla # 2 Precios de productos Bticino

La lista de precios que se muestran a continuación corresponde a los módulos simonvis y hay que considerar que todos los productos domóticos son importados.

ARTÍCULO	MODELO	DESCRIPCIÓN	CANTIDAD	COSTO
Touch LightManager		Touch LightManager permite programar y controlar todos los elementos del sistema a través de pantalla táctil.	1	1047,8
Touch Light Manager Multi-Scenas		Controla los recursos naturales para mantener un nivel constante de luz y maximizar las fuentes de luz natural.	1	1047,8
Touch LightKeypad		Touch LightKeypad permite el control desde otro punto. Activa por defecto: 6 escenas + subida/bajada de persianas y apagado.	1	200,60
Touch LightKeypad (serigrafía 4 canales / 4 escenas)	-	Escenarios de iluminación y secuencias aplicados en las habitaciones para que sean más acogedoras.	1	200,60
Módulo Control persianas	-	Módulo control de persianas para el control de motores con finales de carrera.	1	179,30
Convertor señales Digitales	-	Módulo convertor de señales digitales para la vinculación de una señal 230V en función escena.	1	134,48

Convertor señales Analógicas	--	Módulo conversor de señales analógicas para la vinculación de una señal tipo 1-10V en función escena detector para el control de luz.	1	151,29
Sensor Daylight 1-10V standalone		Escenarios de iluminación y ahorro de energía de hasta un 40 %. Compensa el 100% de la luz entrante	1	-
Central SimonVox basic		Central de telecontrol de servicios domésticos a través del teléfono, mediante la previa introducción de un código personal.	1	172,52
Central telecontrol 230V~ 50/60 Hz CRA		5 entradas (NA-NC). 5 salidas de relé	1	723,39
Detector de movimiento de techo		El detector de movimiento debe colocarse en el techo a una altura ideal de 2,4 mt. Dispone de un ángulo de cobertura de 360°.	1	93,92
Detector de presencia regulador de luz techo empotrar		Regula la intensidad de luz de las luminarias.	1	116,10
Electroválvula de agua		Usada para el control de los sistemas de riego.	1	116,10
Receptor IR para persianas		Receptor para persianas IR, Potencia máxima 700W/VA 230V.	1	106,09
Detector de luminosidad		Se utiliza para gestionar el nivel de iluminación de una estancia.	1	57,87

Módulo Dimmer universal		Módulo Dimmer Universal para regulación de circuitos de iluminación para cualquier tipo de carga.	1	475,22
Módulo Dimmer de 4 Canales	-	Módulo dimmer 4 canales, regula 4 circuitos de carga independientes entre sí.	1	782,89
Módulo terminador de red		Permite la transmisión de datos de manera fiable y sin errores.	1	68,82
Detector de infrarrojos pasivo PIR		Detector pasivo de infrarrojos con alimentación a 24 V c.c. desde la fuente, ángulo de cobertura 110 vertical y 80 horizontal, alcance de 6 a 8m.	1	115,57
Detector crepuscular		Detector crepuscular con alimentación a 24V c.c. desde la fuente, regulable de 5 a 100 lux con led indicador de contacto cerrado.	1	137,11
Software de instalación SIMON Vit@		Protocolo de comunicaciones, software y sistema operativo común para la tecnología Lonworks.	1	765,12
Software formación SIMON Vit@		Software de programación con 64 créditos	1	467,50
Módulo conexión a PC		Módulo Conexión a PC: conexión al bus de datos a través del PC mediante puerto USB.	1	619,72

Interfaz de conexión remota		Conexión al bus de datos, conexión a la red LAN a través de un puerto Ethernet; se utiliza para acceder de forma remota a instalaciones sobre bus SimonVIT@.		627,79
Kit de control de acceso autónomo por tarjeta RFID		El kit controla los accesos utilizando tarjetas RFID.		240,04
Lector biométrico y RFID. Incluye Software		Registro de usuarios, restricción de acceso, posibilidad de control y monitorización a través de Internet. Combina el acceso con cámaras de seguridad y ahorro energético.		843,43
Abre puertas eléctrico 12 Vdc		Terminal de bloqueo. Simétrico. Funcionamiento normal o autónomo. Funcionamiento normal o invertido.	1	28,79
Controlador de puerta 2 relés		Tiene la función de un lector de control de acceso o de un interruptor de llave.	1	739,81
Fuente de alimentación 12 Vdc 2A		Fuente de alimentación switching 12VDC - 2A Ideal para Alimentar Varias Cámaras (Según Modelo) o Cámaras con Movimiento	1	58,02
Pack 10 tarjetas RFID		Pack 10 tarjetas RFID para sistema de control de acceso en red	1	50,96







Cámara Domo Día/Noche con IR		Movilidad en los 3 ejes que permite orientar el visor a cualquier objetivo.	1	89,65
Cámara Domo Día/Noche Antivandálica Exterior IP66		Sensor CMOS de 2-megapixel Filtro de corte IR con función Día / Noche, recorte de vídeo para ahorrar ancho de banda, detección de manipulación y sabotaje	1	207,32
Juego 4 Conectores DC alimentación		Juego de 4 Conectores para alimentación DC	1	13,85
Alimentador de cámaras		Alimentador para cámaras SIMON.	1	18,49
Monitor de video vigilancia profesional		Monitor profesional de alto rendimiento, especial para funcionar de forma ininterrumpida.	1	516,90

Tabla # 3 Precios de productos SimonVis.

A continuación se presenta algunos de los módulos que ofrece Domobak:

ARTÍCULO	MODELO	DESCRIPCIÓN	CANTIDAD	COSTO
Switchlinc- INSTEON Dimmer (Dual- Band)		Interruptor dimmer para aumentar o disminuir la luz de un espacio.	1	59,99

Switchline INSTEON Relay (Dual- Band)		Interruptor para encender o apagar la luz de un espacio no dimerizable o artefactos eléctricos.	1	54,99
INSTEON HUB Cerebro de Toda la casa		Configuración de horarios , escenas, se lo controla mediante smartphones	1	129,99
INSTEON Micro On/Off Module		Automatiza lo que el usuario desee.	1	49,99
Micro Open/Close control module		Módulo para controlar motores. Especial para instalarlo "in situ", conjuntamente con las cortinas y la motorización de toldos.	1	50
I/O Linc Contact Closure Interface		Modulo Universal IOLinc para automatizar la puerta eléctrica o cargas más fuertes, e integrarlas al sistema.	1	45,99
Cámara IP Inalámbrica Blanca		Funciona con smartphones y equipos móviles. Amplio rango de movimientos con 300° pan y 120° tilt (pan: mov. horizontal, tilt: mov. vertical).	1	79,9
Controladores NetLinx NX Integrados		El NXS - MHS acomoda una tarjeta y se completa con placas frontales para el Maestro y el Hub	1	49,32
NXS-NMS		El NXS - NMS es una opción sencilla y económica para la integración de una funcionalidad adicional en los sistemas de control.	1	59,99
Controles de iluminación		Los LED emiten una señal de advertencia por un minuto cuando las luces o un ventilador están a punto de apagarse.	1	50,4






Pico Wireless Control & Pedestal		Ajuste de luces desde cualquier lugar en un espacio con control inalámbrico Pico.	1	56
ENV-VST-C		La ENV- VST -C es un termostato de comunicación programable para la programación automática de la temperatura y los ajustes de humedad relativa y la protección en corte de luz.	1	400,25
NMX-VCC-1000-WM Sereno™ Camera Wall Mount		La cámara Sereno puede ser montado en variedad de superficies.	1	163
Motion sensor		Se apaga automáticamente después de que se detenga el movimiento (ajustable entre 30 segundos y 2 horas)	1	34,99
INSTEON 2672-222 LED Bulb, 8-wat		Bombilla regulable en red.	1	29,99

Tabla # 4 Precios de productos Domobak.

Software MHVisual de Bticino	\$ 879
Software formación y software de instalación SIMON Vit@	\$ 1232.62

Tabla # 5 Costos de software de empresas

En tecnología no se puede comparar con la marca Bticino conociendo que los costos de adquisición de sus módulos son muy elevados, en el

mercado hay módulos similares; la diferencia está en el tiempo de vida útil, la calidad de los mismos y las certificaciones que poseen.

3.2 ESTUDIOS DE CAMPOS DE INSTALACIONES DOMÓTICAS

Se presenta a continuación el plano eléctrico¹² de la vivienda con la incorporación de la distribución domótica:

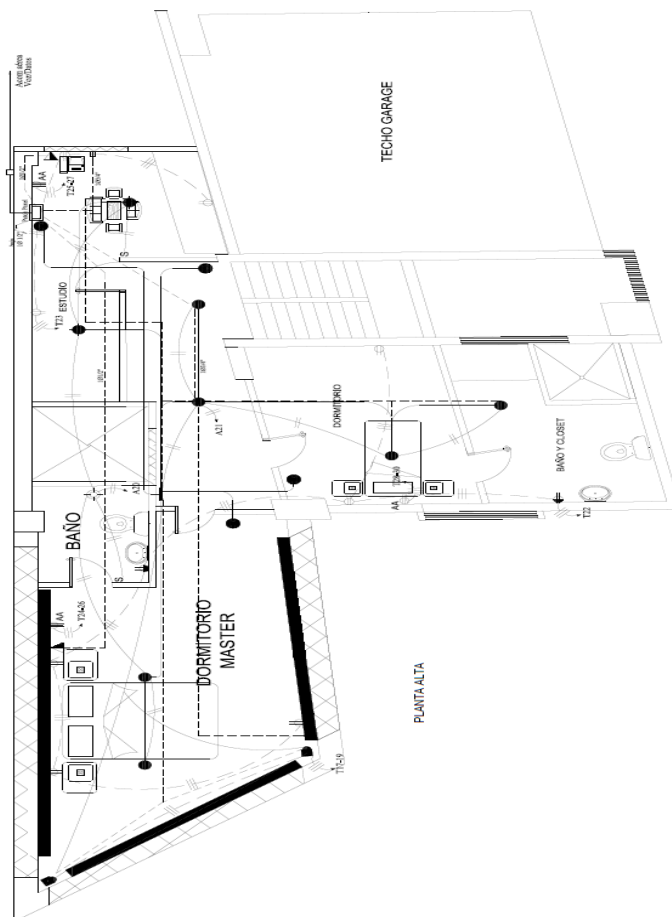


Fig. 3.1 Plano eléctrico-domótico planta alta

¹² Véase los anexos ya que se adjunta el plano en formato A3 para las figuras 3.1 y 3.2.

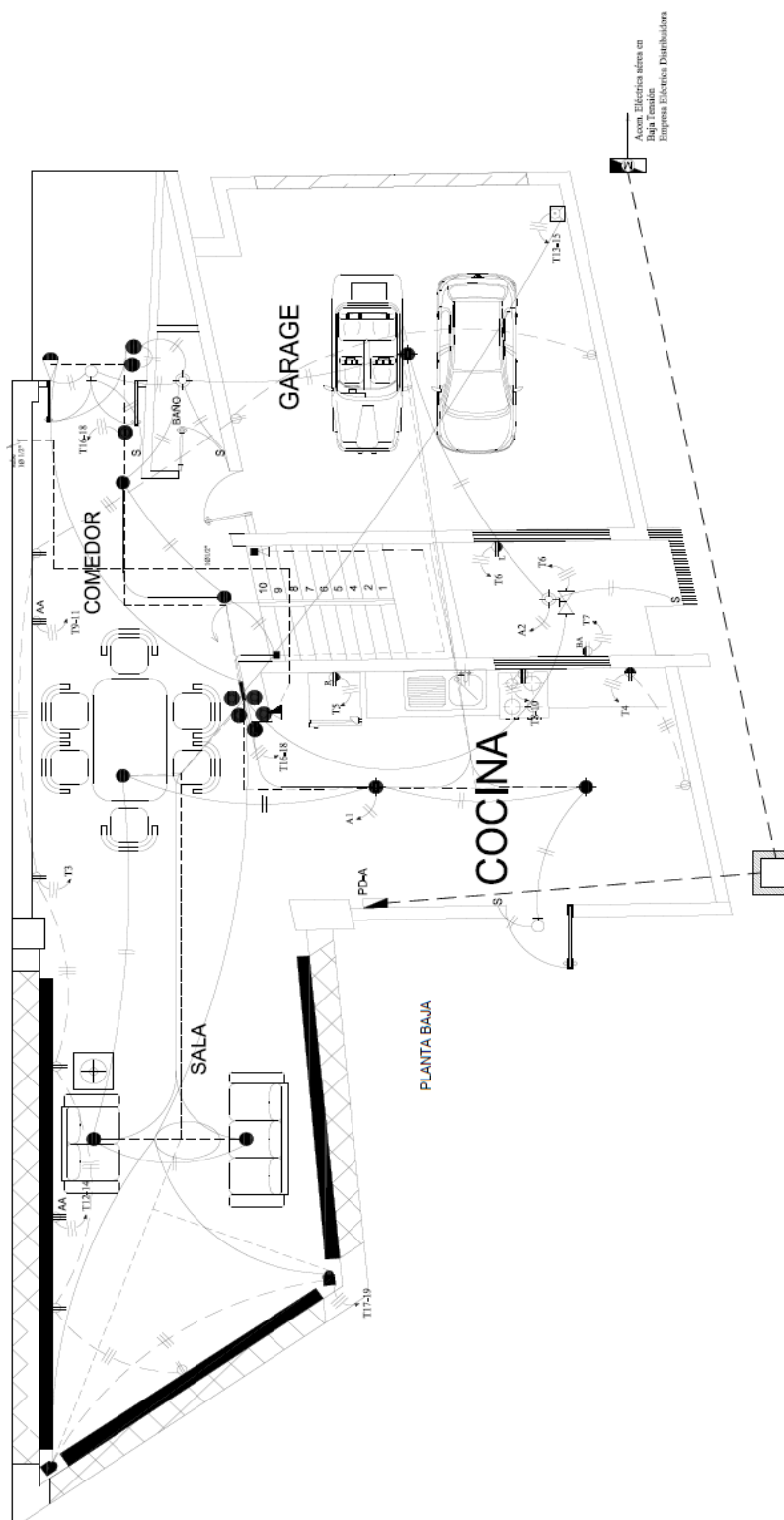


Fig. 3.2 Plano eléctrico-domótico planta baja

3.3 DESCRIPCIÓN DE LOS DISPOSITIVOS A UTILIZAR

Se presenta a continuación la tabla con los módulos a considerar para el sistema domótico, es importante analizar la fuente de alimentación para cada módulo ya que se puede requerir adaptadores o de ser necesario la alimentación directa.

Se ha considerado la potencia de cada módulo para estimar el amperaje necesario que la fuente de alimentación debe suplir.

Nombre	Voltaje [V]	Corriente Inicial [A]	Corriente Nominal con Carga [A]	Función
Servomotor	5	0,13	0,075	Movimiento de apertura de la puerta del hogar
Arduino - Pololu	9		0,05	Módulo de control de la tira de LEDS
Arduino-NFC	9		0,13	Lector de la tarjeta magnética
Tira de LEDS	5	0,05	0,7	Tira de LEDS para a ambientación
7805	12		0,18	Reductor de voltaje para los Arduinos
Arduino Mega	5		0,3	Procesador central
Placa Dimmer	5		0,055	Módulo que puede dimerizar 11 LEDS
Placa Relay	5		0,38	Módulo de control de apertura de 5 LEDS y el riego
Placa PIC	5		0,003	Módulo de control de persianas

Motor de Paso	5		0,16	Motor que permite el movimiento de persiana y garaje
Total		0,18	2,033	

Tabla # 6 Descripción de los módulos

CAPÍTULO 4

ANÁLISIS Y DISEÑOS DE LA MAQUETA

4.1 PLANOS DE LA VIVIENDA

El modelo de la vivienda fue tomada de una residencia ya edificada, fue considerada por cómo está estructurada; es decir, los grandes ventanales que posee facilitan la demostración del sistema domótico con respecto a las persianas. Los detalles de cómo ha sido construida las escaleras hace posible la visualización del sistema de ambientación.

Toda la residencia presenta ventanales lo que permite la demostración del sistema de luces y el diseño a la entrada de la vivienda nos facilita

la incorporación del sistema de seguridad y realizar las demostraciones respectivas.

Cabe indicar que la edificación de ésta vivienda (maqueta), hace posible que todo el sistema domótico sea demostrativo y se puede seguir manipulándola para fines didácticos.

A continuación se presenta las perspectivas de la vivienda real y los planos:



Fig. 4.1 Vista frontal sur



Fig. 4.2 Vista lateral sur

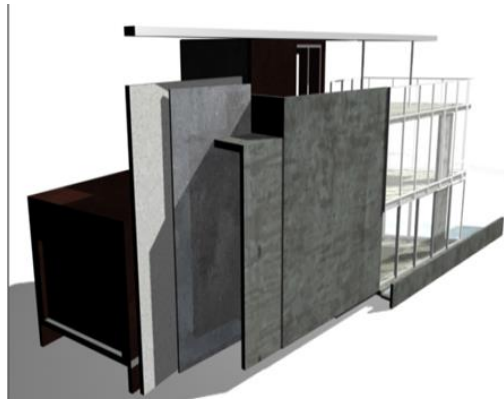


Fig. 4.3 Perspectiva oeste



Fig. 4.4 Perspectiva este

PLANO 1: PLANTA BAJA

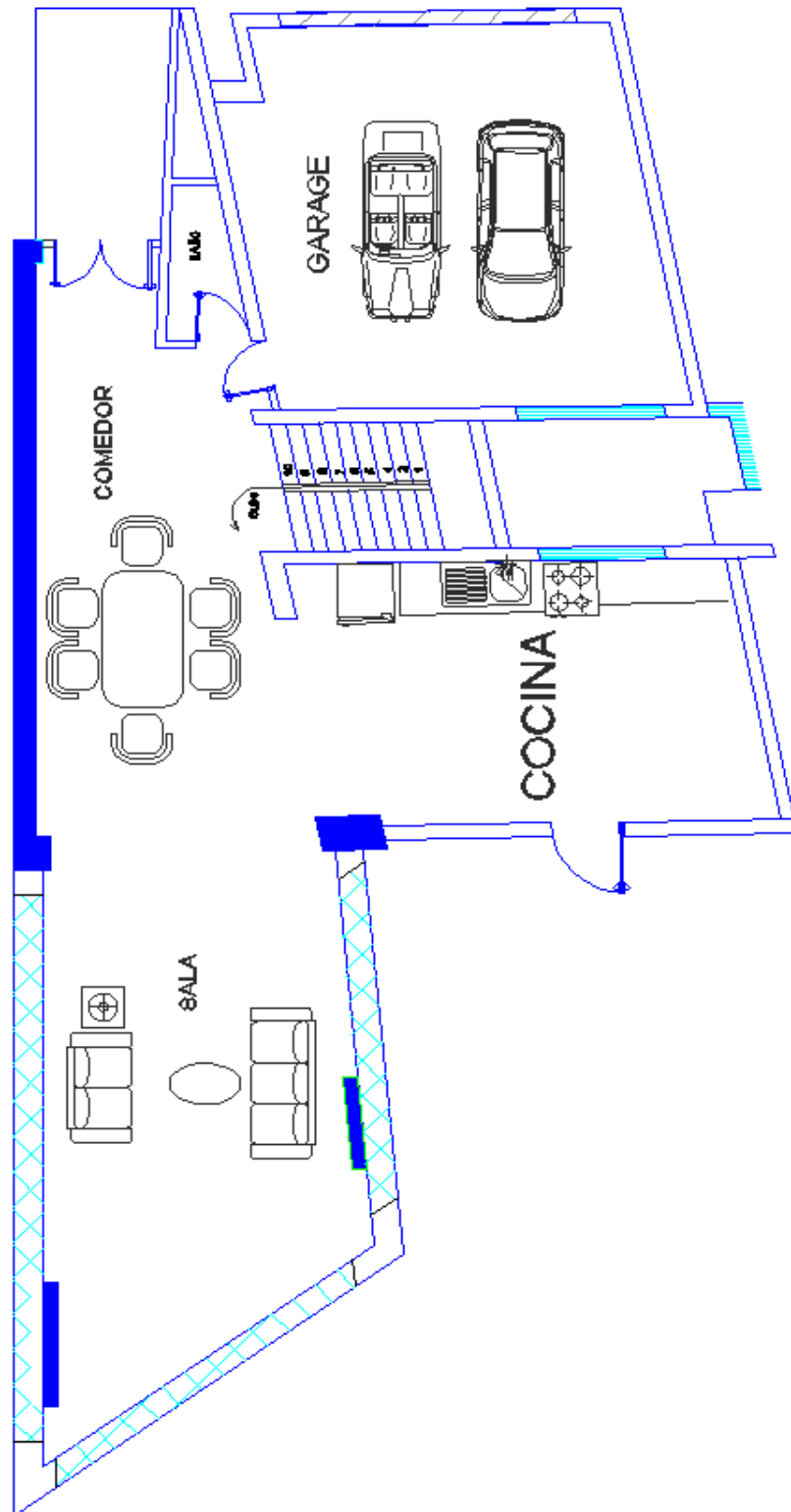


Fig. 4.5 Plano de la planta baja

PLANO 2: PLANTA ALTA

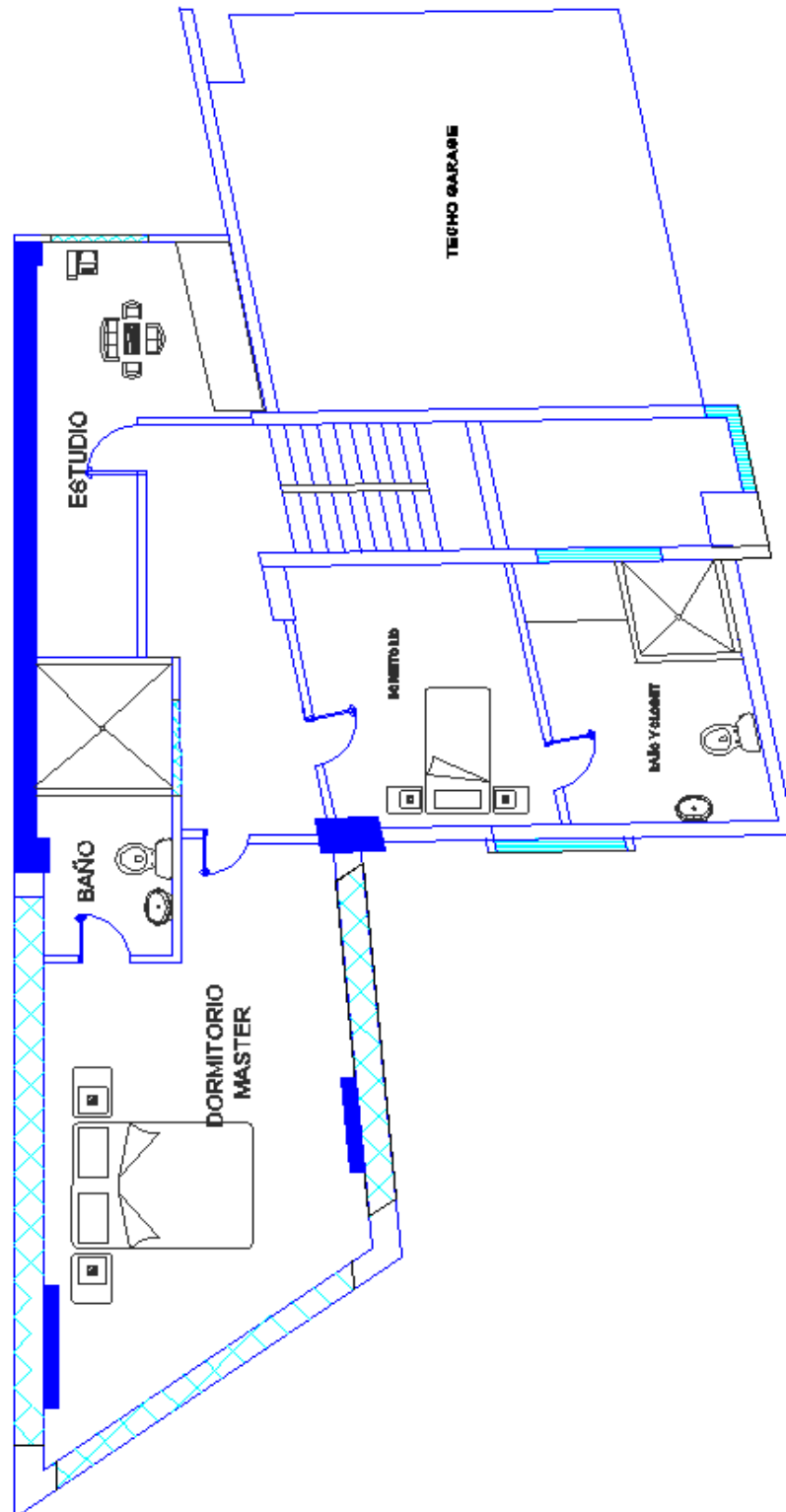


Fig. 4.6 Plano de la planta alta

4.2 PLANO ELÉCTRICO

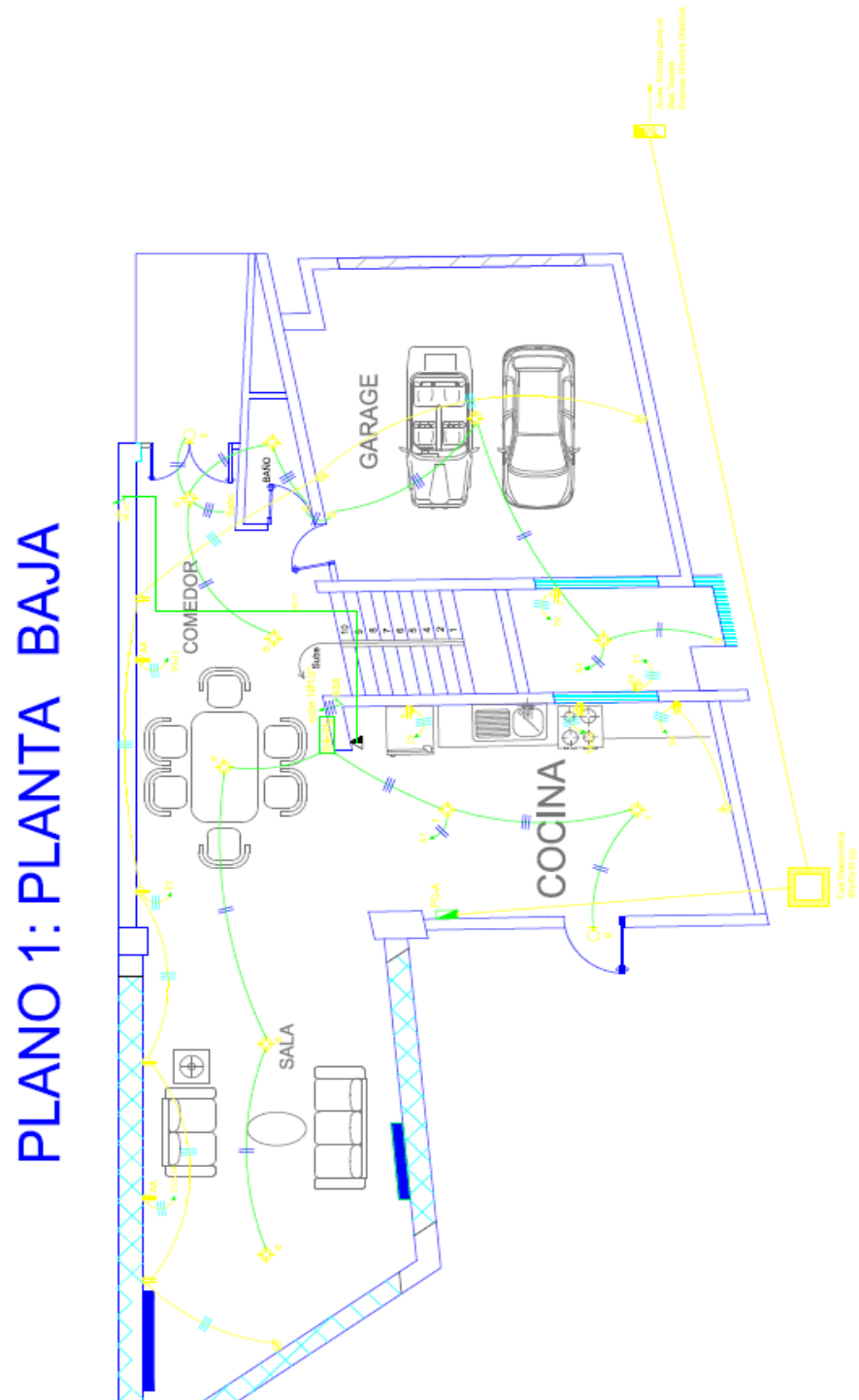


Fig. 4.7 Plano eléctrico de la planta baja

PLANO 2: PLANTA ALTA

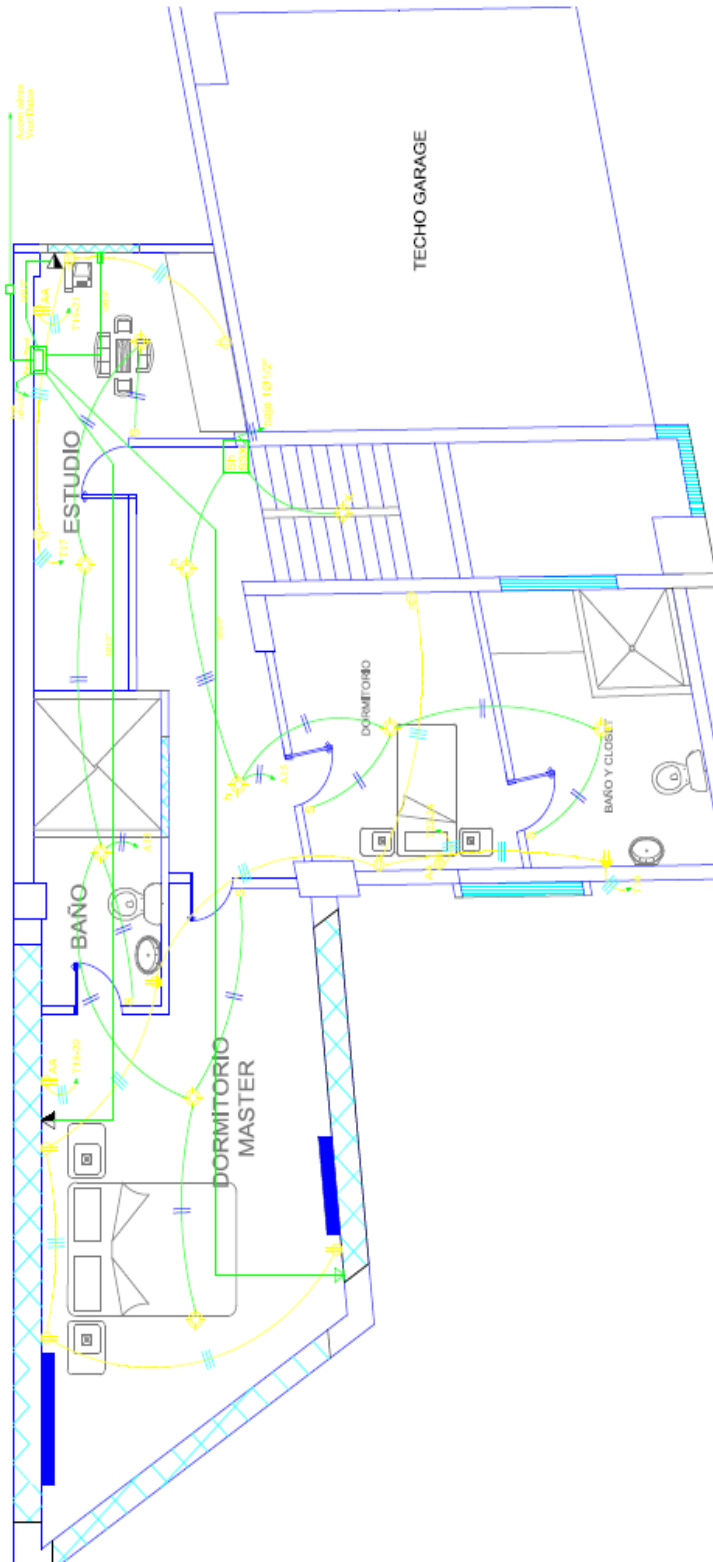


Fig. 4.8 Plano eléctrico de la planta alta

SIMBOLOGIA

























	PANEL DE DISTRIBUCION
	TABLERO DE MEDIDOR
	PUNTO ALUMBRADO 120V
	APLIQUE DE PARED 120 V
	PUNTO DE ALUMBRADO 120V. (OJO DE BUEY)
	INTERRUPTOR SENCILLO
	INTERRUPTOR DOBLE
	INTERRUPTOR TRES VÍAS
	TOMACORRIENTE DOBLE 120 V h = 0.40 m.
	TOMACORRIENTE MESON POLARIZADO (h= 1.20 m)
	TC. PROPÓSITO ESPECIAL
	TOMACORRIENTE MIXTO (h=1.20 m)
	PUNTO TOMA AA h=1.80 m
	CAJA DE PASO ELÉCTRICA 10*10 cms
	INTERCOMUNICADOR CON PORTERO ELÉCTRICO
	CAMPANA DE TIMBRE
	PORTERO ELÉCTRICO
	TUBERIA EMPOTRADA POR PISO.
	TUBERIA EMPOTRADA POR PAREDES Y TUMBADO.
	TERMINAL DIRECTA TELEFÓNICA
	EXTENSIÓN TELEFÓNICA
	PUNTO DE TV
	PUNTO DE RED/INTERNET
	TUBERÍA QUE SUBE

Fig. 4.9 Simbología usada en los planos eléctrico

4.3 PLANOS DE UBICACIÓN DE LOS DISPOSITIVOS

Se presenta en la página siguiente el plano de ubicación de los componentes que fueron usados para domotizar la maqueta.

PLANO 1: PLANTA BAJA

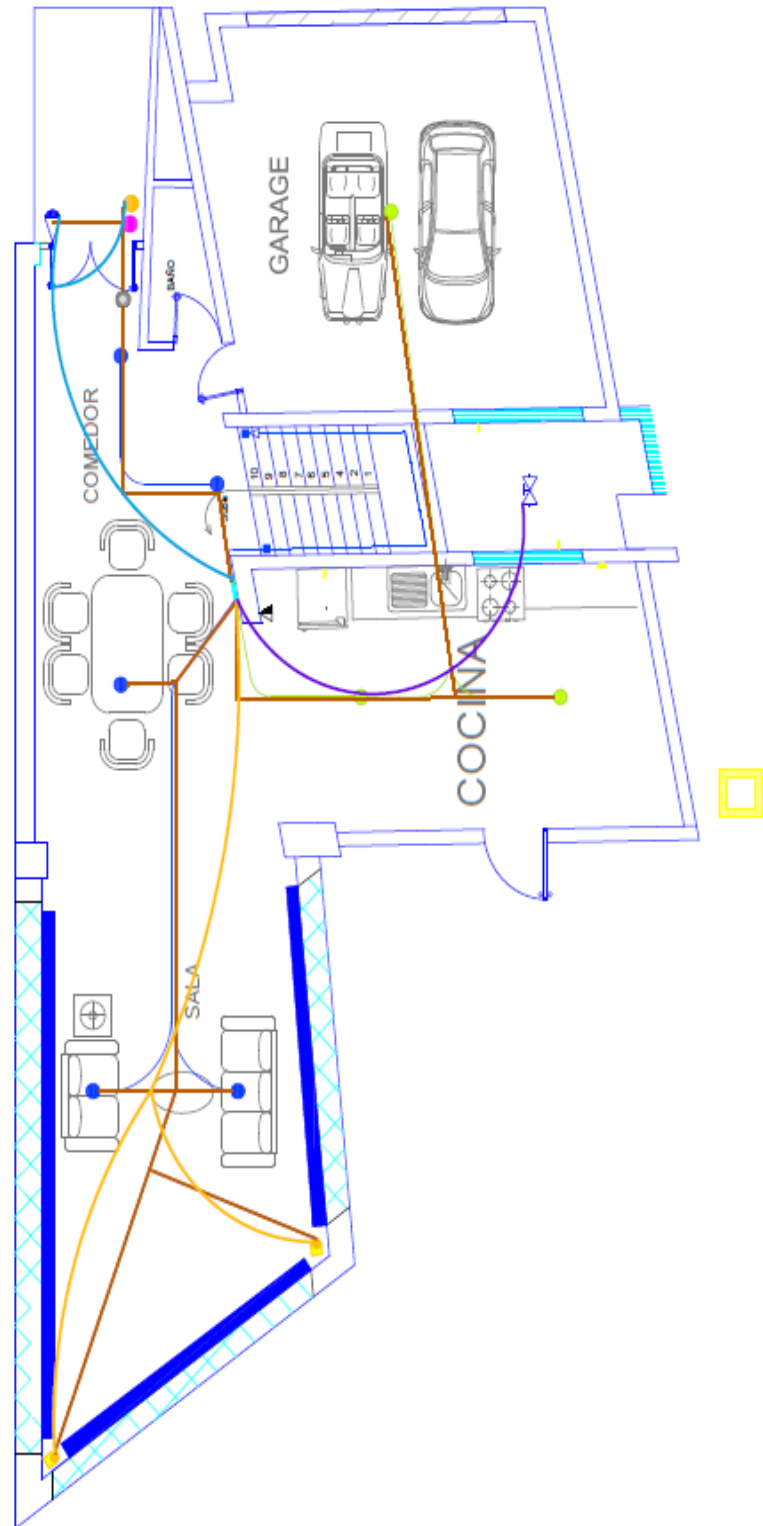


Fig. 4.10 Plano del sistema domótico planta baja

PLANO 2: PLANTA ALTA

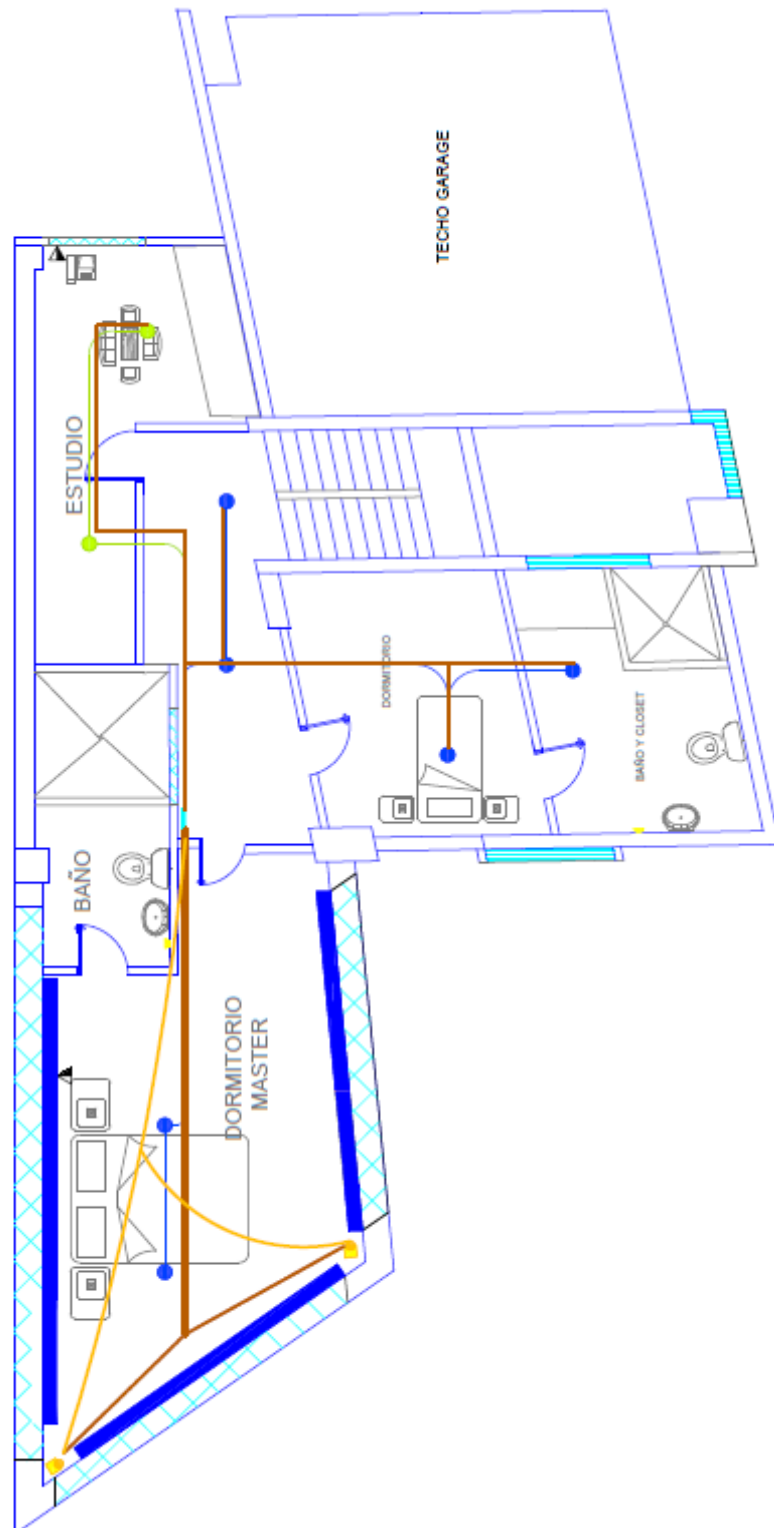


Fig. 4.11 Plano del sistema domótico planta alta.

SIMBOLOGIA DOMOTICA





	PUNTO DE LUZ LED (DIMMER) EMPOTRABLE	
	PUNTO DE LUZ HALOGENO EMPOTRABLE	
	ARMARIO DE DISTRIBUCION DOMOTICA	
	TIRA DE LEDS PARA LA AMBIENTACION	
	MOTOR DE PERSIANAS	
	PERSIANAS	
	ELECTROVALVULA DE AGUA	
	CAMARA DE VIGILANCIA	
	MODULO RFID PARA LA TARJETA MAGNETICA	
	MODULO RECONOCEDOR DE VOZ	
	CERRADURA ELECTRONICA	
	CANAL DE CONDUCCION ELECTRICA DOMOTICA	
	CANAL DE COMUNICACION DE MOTORIZACION DOMOTICO	
	CANAL DE SISTEMA DE COMUNICACION DEL SISTEMA DE SEGURIDAD	
	CANAL DE CONTROL DE ELECTROVALVULA	

Fig. 4.12 Simbología empleada para el sistema doméstico

CAPÍTULO 5

PROGRAMACIÓN DE LAS GESTIONES DOMÓTICAS EN LABVIEW

5.1 SISTEMAS DE CONTROL DE ILUMINACIÓN

5.1.1 ESCENARIO POR PRESENCIA

Previo a la programación se ha analizado los espacios más habituales dentro de la vivienda por tal motivo se ha incorporado tres escenarios los cuales incluyen lo siguiente:

Escenario 1: luminaria del comedor, luminaria del pasillo de la planta baja y alta, luminaria del dormitorio general.

Escenario 2: luminaria de garaje, cocina, planta baja y alta, luminaria del estudio y dormitorio general.

Escenario 3: incluye la ambientación y persianas.

A continuación en la Fig. 5.1 se presenta el panel frontal del sistema, en los cuales se visualiza el control por escenario:

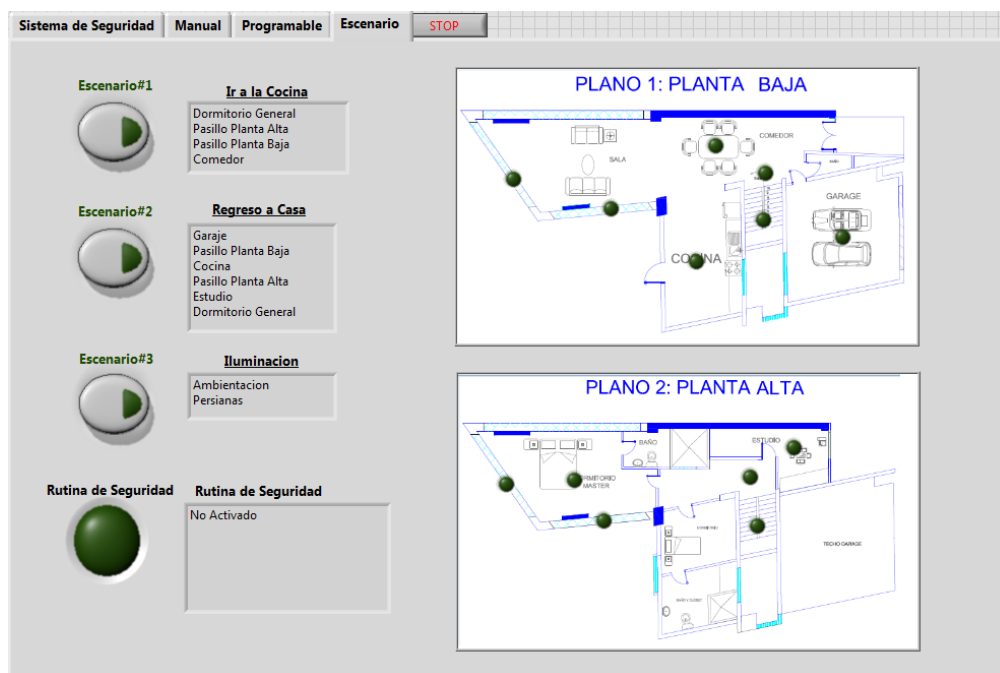


Fig. 5.1 Panel frontal del control por escenario

Para una mejor visualización se ha incorporado las capturas de los planos de distribución de la vivienda. Tanto la planta baja como la alta

presenta indicadores (LEDS) con los que se puede vigilar y asegurar que el escenario seleccionado haya sido activado.

Los led indicadores de cocina, estudio, garaje y riego visualizados en el panel frontal tienen salida a relays permitiendo el control y a la vez la energización de las luminarias; los demás LEDS indicadores presentan salidas de PWM que han sido establecidos al valor de 250 (unidad de 8 bits de 0 a 255 valor decimal) al ser activados.

Como se ha integrado todas las programaciones al sistema domótico general se presentará solo los códigos que corresponden a cada subsistema para una mejor comprensión y desglose de los mismos.

En la Fig. 5.2 se aprecia la programación realizada para el control de escenarios.

Observando la imagen podremos identificar los push button con los nombres Escenario#1, Escenario#2 y Escenario#3; los cuales activarán a cada distribución de LEDS asignada.

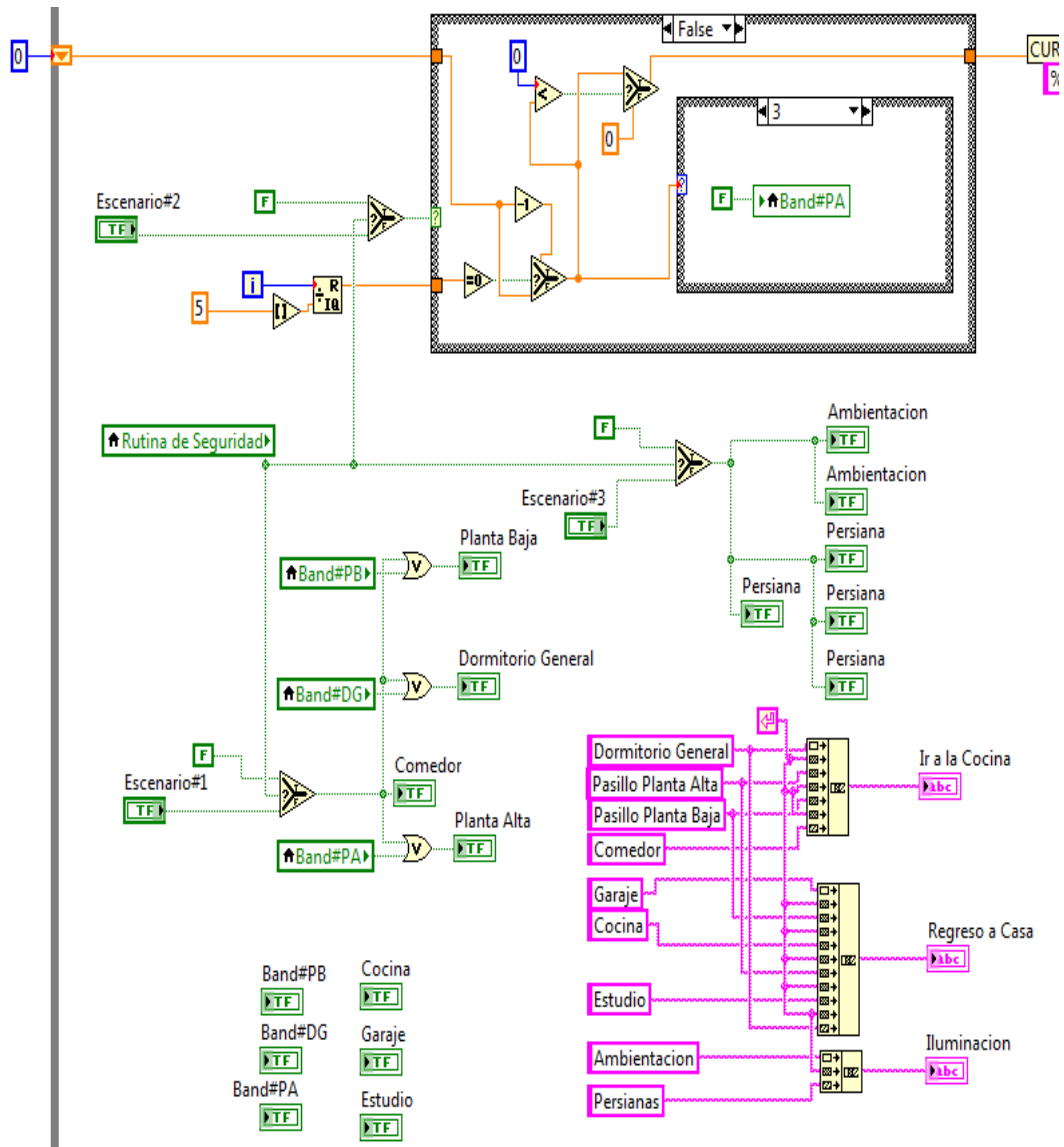


Fig. 5.2 Programación del sistema de escenario

Si analizamos la distribución correspondiente a Escenario# 1 se observa que al activar el push button, se encienden los LEDS de planta baja, dormitorio general, planta alta y comedor. Al ser activados éstos y usando el comando de variable local se asigna a la salida de

los pines el valor máximo PWM que es 255 de acuerdo al número de pin asignado.

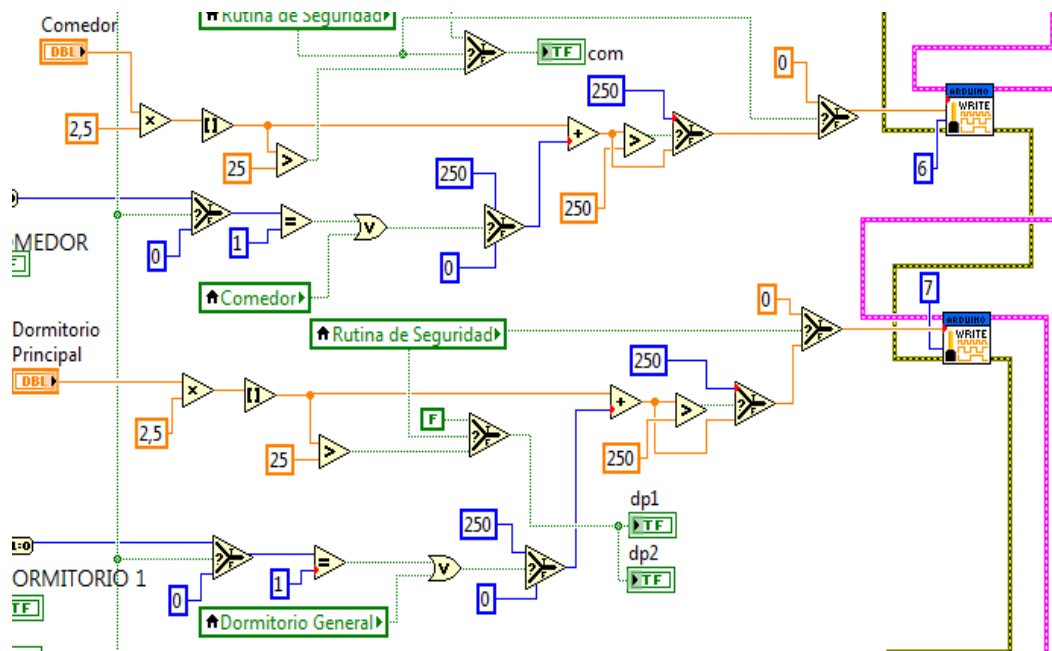


Fig. 5.3 Escenario#1 uso de variable local

Enfocándonos solo en las variables locales *comedor* y *dormitorio* general de la Fig. 5.3 podemos comprender que al estar en estado ON, se enviará el valor máximo PWM a la salida del puerto digital que será en los pines 6 y 7.

Se cumple lo anteriormente descrito siempre y cuando no se haya usado el slide para dimerización el cual manipula la señal PWM controlando la intensidad luminosa y no se haya activado la rutina de seguridad; la cual impide la activación de cualquier escenario.

La programación será exactamente igual para el control de los pasillos de la planta alta y baja, la diferencia está en que al activar la rutina de seguridad los LEDS indicadores que se encuentran en la sección manual son bloqueados; impidiendo su ejecución.

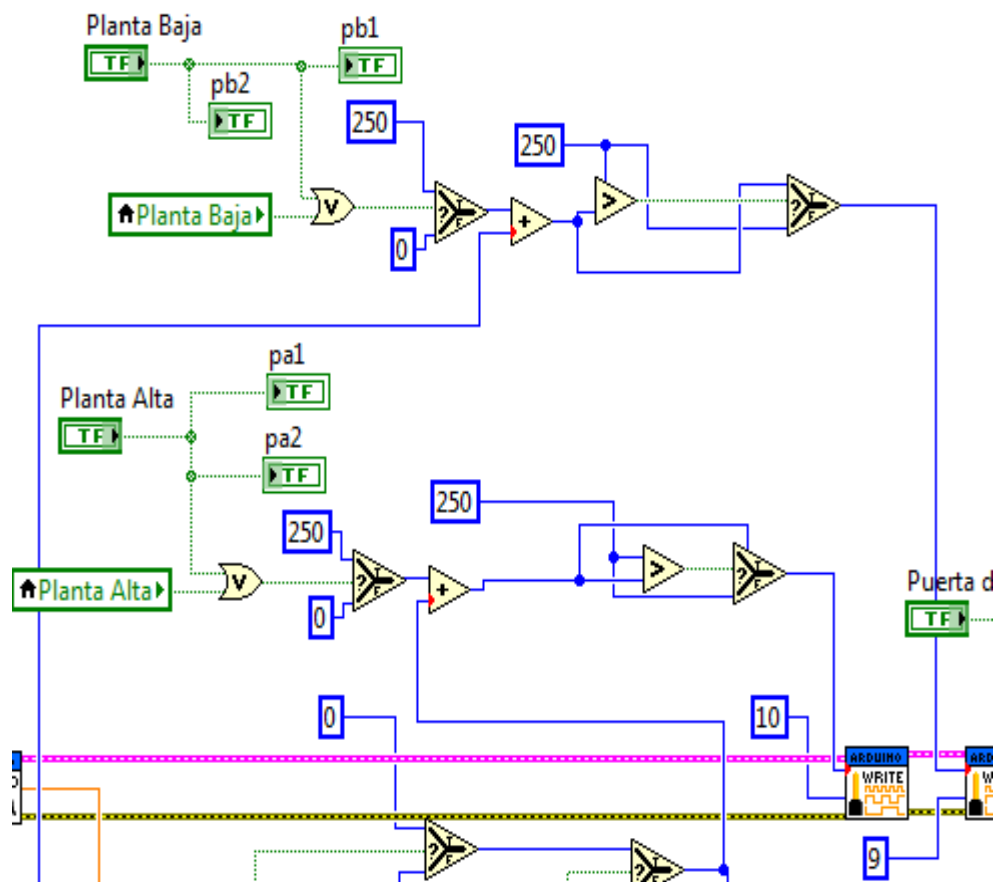


Fig. 5.4 Escenario#1 variable local de planta baja y alta

Las luminarias de los pasillos de la planta alta y baja se han establecido en los códigos posteriores para que funcionen mediante dimerización y control lumínico, por tal motivo para que los escenarios

y las variables locales se ejecuten; se envía el valor de 250 al controlador PWM del Arduino como se puede apreciar en la Fig. 5.4. Si otra programación (dimmer y lumínico) le envía otro valor los datos son sumados y el resultado es comparado, si el valor se excede de 250 el selector lo setea a 250 para el PWM; y, de esta manera ningún control presenta fallas y se ejecuta el correcto funcionamiento.

Analizando la Fig. 5.5, en el escenario 2 existe dos consideraciones: en la primera se ha añadido el shift register con la finalidad de ejercer un control durante el encendido de los LEDS de manera pausada y secuencial haciendo uso de dos case y de bloques comparadores. En la segunda consideración los valores que el shift register presente serán decrementados dentro de la estructura case con la finalidad de dar inicio al apagado de LEDS de manera ordenada y secuencial.

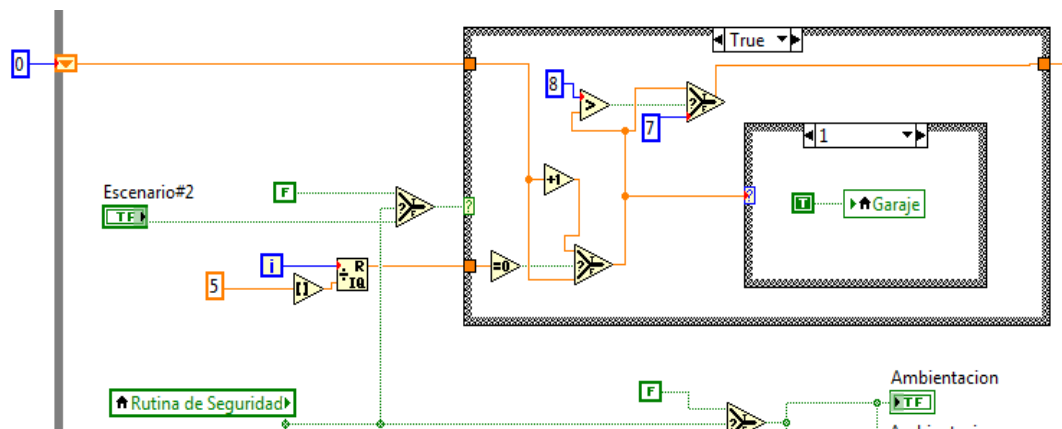


Fig. 5.5 Escenario#2 secuencia de encendido- garaje

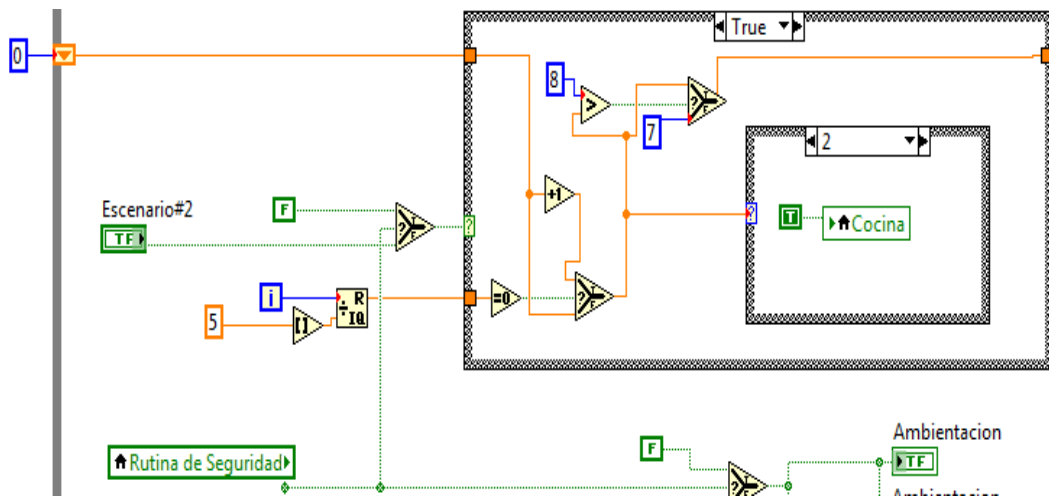


Fig. 5.6 Escenario#2 secuencia de encendido- cocina

A continuación se presenta la secuencia de apagado:

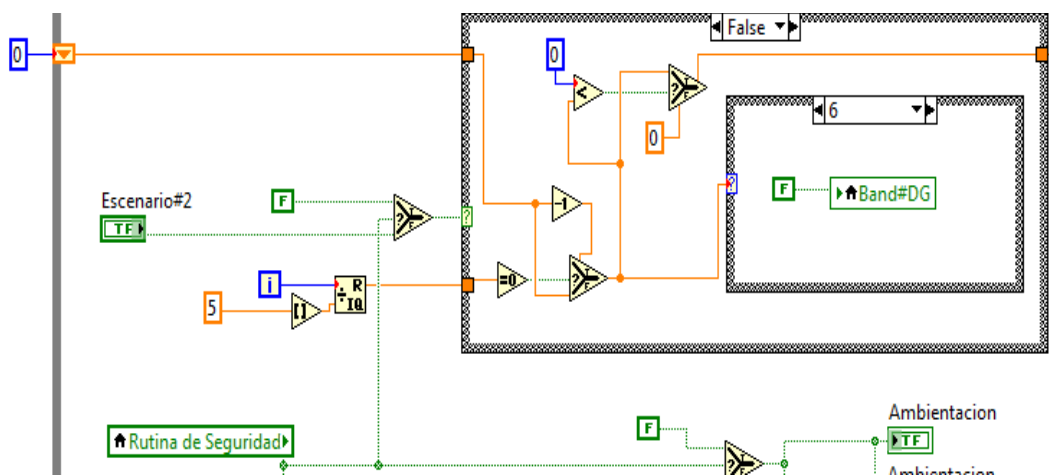


Fig. 5.7 Escenario#2 secuencia de apagado- dormitorio

Dentro del primer caso se ha establecido límites numéricos para el shift register siendo el número máximo el siete como se visualiza en la

Fig. 5.6, de esta manera se establece seis secuencias en el case interno y el último número corresponde al estado vacío; el cual al no presentar código preguntará constantemente por el estado del push button escenario#2 para dar inicio a la secuencia de apagado.

Si asumimos por un instante que el valor del shift register es siete, al ejecutarse la división para uno tenemos un residuo de cero; esto permitirá que el valor del registro sea decrementado en uno teniendo como resultado el seis y de esta manera se permite la secuencia de apagado del escenario dos; como se aprecia en la Fig. 5.7.

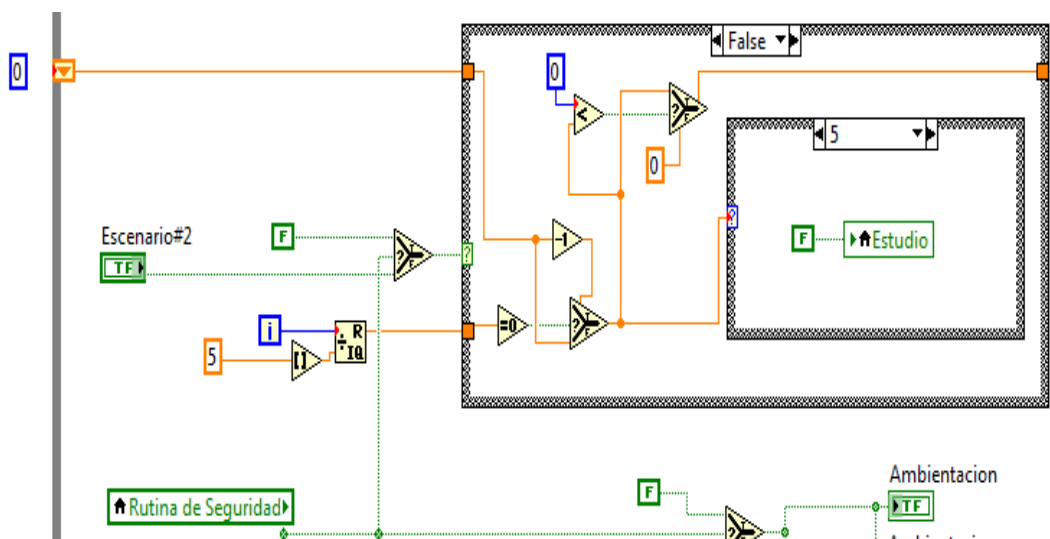


Fig. 5.8 Escenario#2 secuencia de apagado- estudio

La programación del escenario tres incluye el control sobre la ambientación y las persianas. En el caso de ambientación se ha

distribuido para que tanto la planta baja como alta presenten la secuencia de juegos de luces y la variable local *ambientación* activará la secuencia en el pin correspondiente; véase la Fig. 5.9 y 5.10.

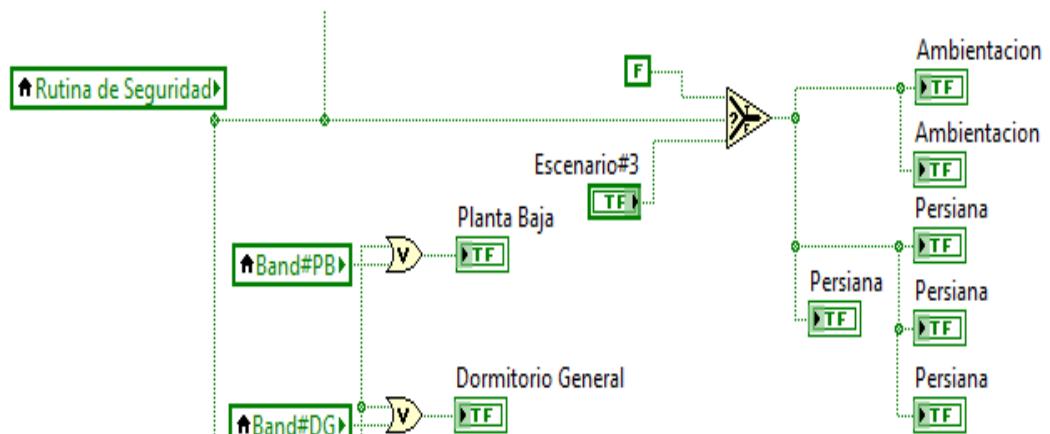


Fig. 5.9 Escenario#3 ambientación y persiana

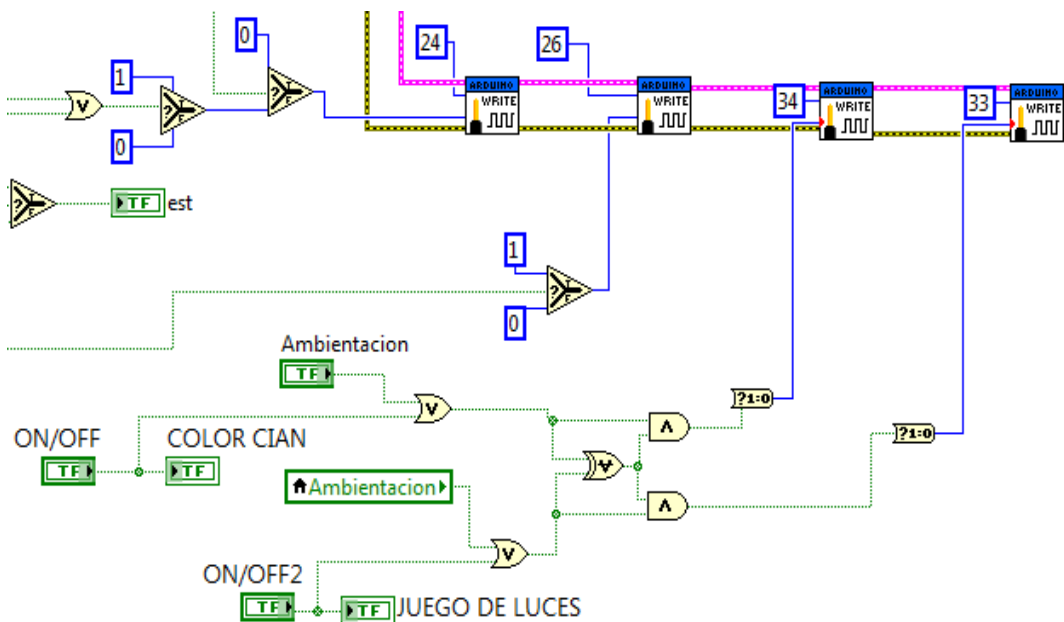


Fig. 5.10 Programación de ambientación-escenario 3

Hay que indicar que mientras esté activada la rutina de seguridad ningún escenario se ejecutará y se lo puede visualizar en la Fig. 5.11.

De una manera muy similar se realiza el control sobre las persianas, es decir si la variable local *Persiana* es true, enviará un valor lógico 1 a la salida de los pines 28, 29, 30 y 31.

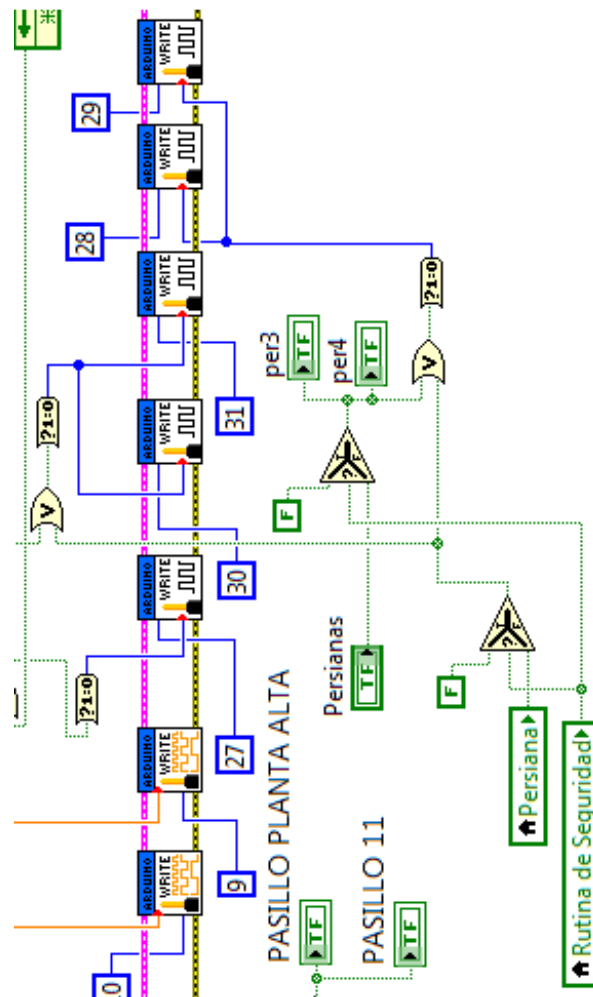


Fig. 5.11 Programación de persiana-escenario 3

5.1.2 DIMERIZACIÓN POR AMBIENTE



Fig. 5.12 Control programable del sistema de iluminación

En la Fig. 5.12 se puede apreciar que existen tres tipos de controles, el primero es el lumínico, el segundo es controlado por la hora del sistema dejando a disposición del usuario el uso del mismo y el tercero es el control de ambientación.

Previo a introducirnos a la programación del primer control debemos conocer los conceptos básicos para el funcionamiento del mismo.

Los interruptores lumínicos encienden y apagan la luminaria en función del valor de luminosidad que se haya ajustado, es decir cuando el dispositivo detecte que no hay luz solar suficiente se encenderá la bombilla instantáneamente.

El dimmer es un regulador o atenuador de la energía variando la intensidad de luz que emite cierta luminaria.

El LDR es un fotoresistor es decir varia su resistencia en función de la intensidad lumínica, cuando hay incidencia de luz el valor de resistencia puede oscilar entre los 50 a 1000 ohmios y de 50k a varios mega ohmios cuando está a oscuras.

Teniendo bajo consideración los conceptos mencionados se ha programado de tal manera que el control con sensor lumínico permita dimerizar las luminarias de los pasillos a determinada hora y oscuridad.

Debido a que la programación es extensa y para una mejor comprensión y visualización se ha optado por presentar el código seccionado; es decir, el código correspondiente por pasillo.

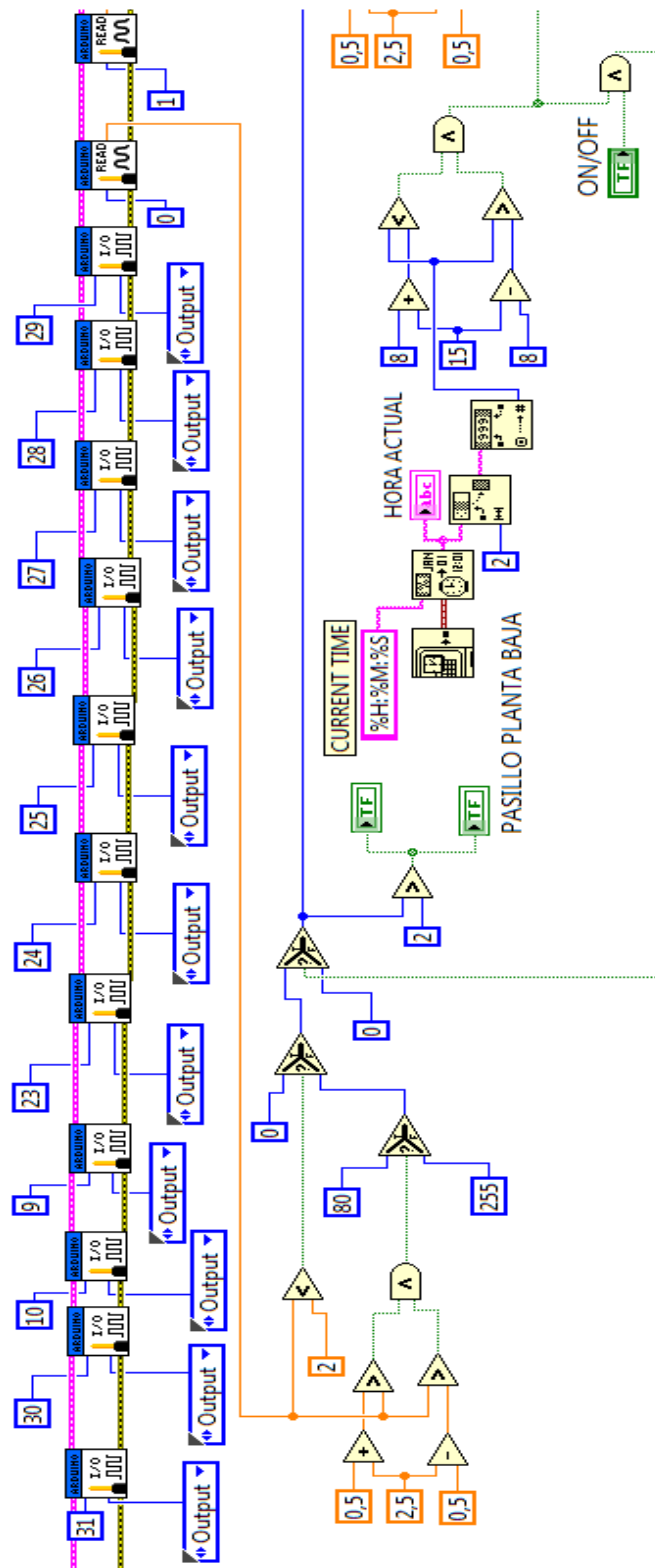


Fig. 5.13 Control del sensor lumínico pasillo 1

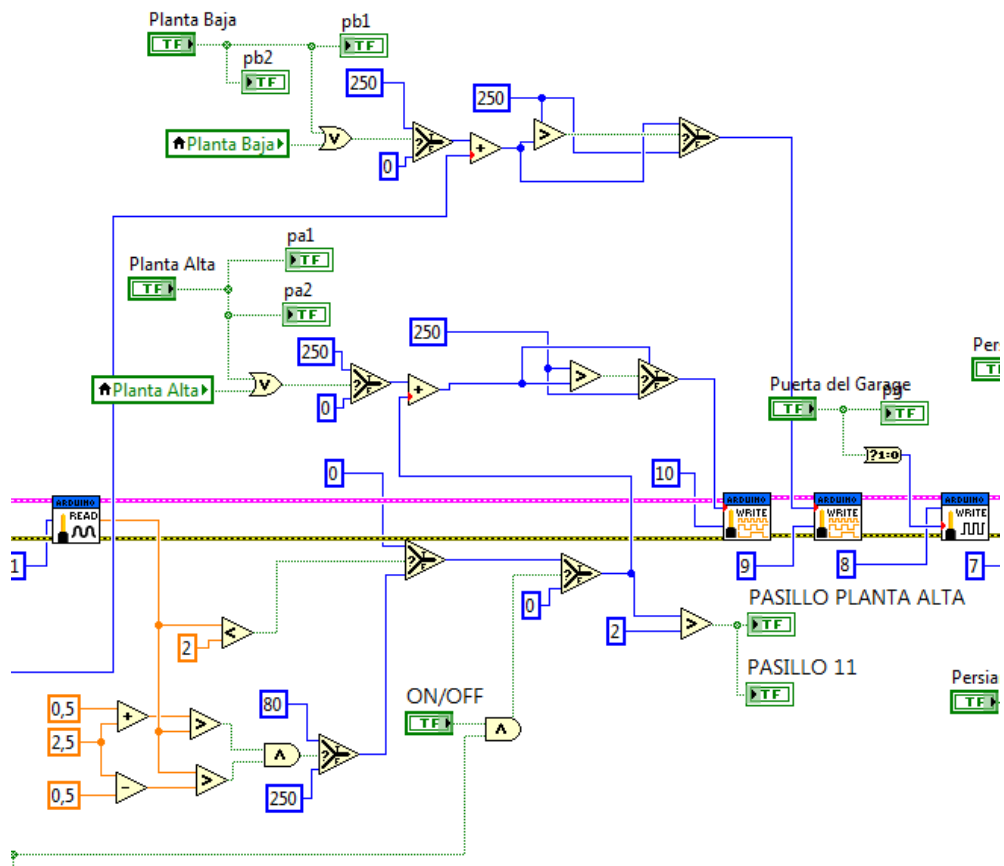


Fig. 5.14 Control del sensor lumínico pasillo 2

Para poder comprender esta programación necesitaremos desglosar el mismo, el primer análisis que se hará es al sensor de luz o LDR y los rangos programados para que se ejecute el control, consecutivamente se estudiará el control de hora y la afectación que ejerce al sistema y finalmente se comprenderá la función de las botonerías o push button.

Los dos sensores resistivos luminicos LDR estan conectados a los pines análogos cero y uno de la tarjeta Arduino mega, los valores receptados del LDR estan entre 0 y 5 ya que cuando hay mayor presencia de luz la resistencia disminuye y el valor de voltaje es cercano a cero, mientras cuando se está en oscuridad la resistencia es alta y el valor de voltaje medido es alto dependiendo del valor del voltaje de alimentación.

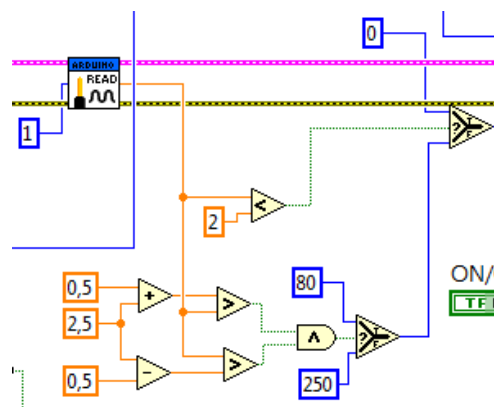


Fig. 5.15 Control del LDR

Cuando el valor sensado del LDR es menor a 2, se manda un cero a la salida de PWM, sucede lo contrario si el valor sensado esta entre 3 y 2 ya que de cumplirse esa condicion se enviará el valor de 80 y para valores mayores a 3 se manda el valor de 250 a la salida del pin digital, como se puede apreciar en la Fig. 5.15.

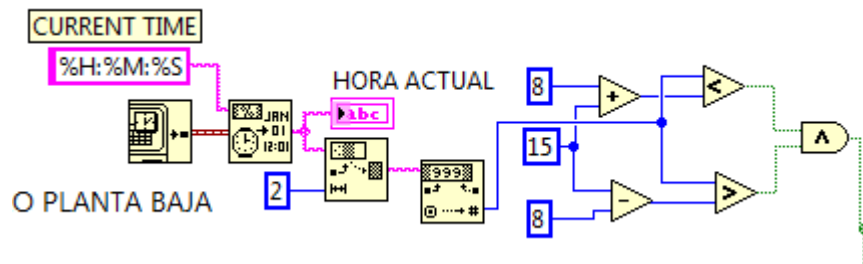


Fig. 5.16 Control de horas para el sensor

Para poder ejercer un correcto control de horas, se ha tomado la hora del sistema de la computadora como se indica en la figura 5.16 utilizando los bloques get date/time in seconds, format date/time string; con el string subset se sitúa la longitud numérica en dos para obtener solo los valores de hora sin considerar minutos ni segundos.

El bloque decimal string to number convierte el dato obtenido en string a número, dicho valor es comparado y solo enviará un true el bloque and cuando tal valor esté dentro del rango de 7 a 23 horas.

Si el control de horas se cumple, es decir está dentro de los límites de tiempo, y si el sensor LDR envía el dato indicando que es mayor a 3, solo se activarán las luminarias si el pulsador o push button ON/OFF ha sido activado como se puede apreciar en la Fig. 5.17.

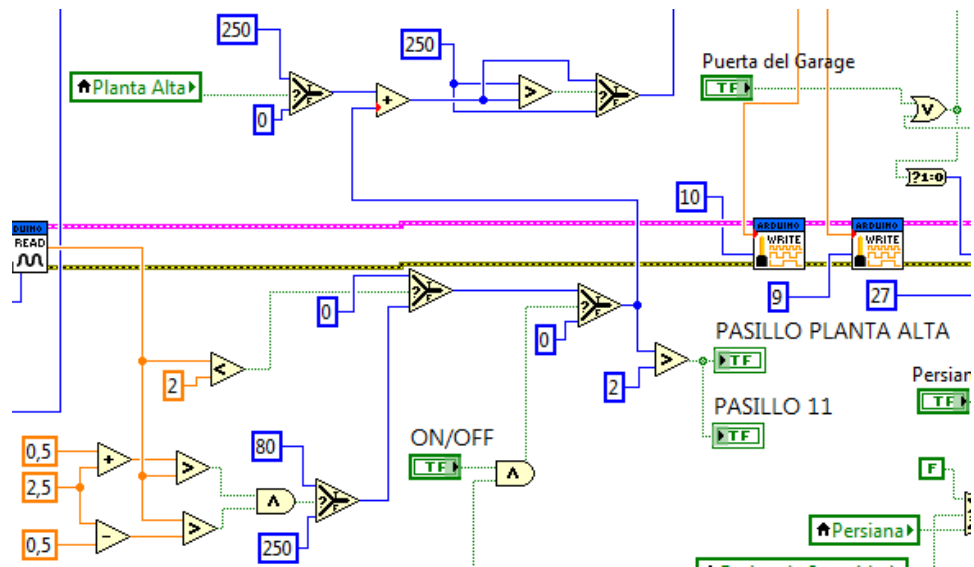


Fig. 5.17 Control del pulsador ON/OFF

Adicional del control lumínico se ha dimerizado la sala, el comedor y el pasillo de la planta baja; cuyos controles están situados en la sección manual del sistema.

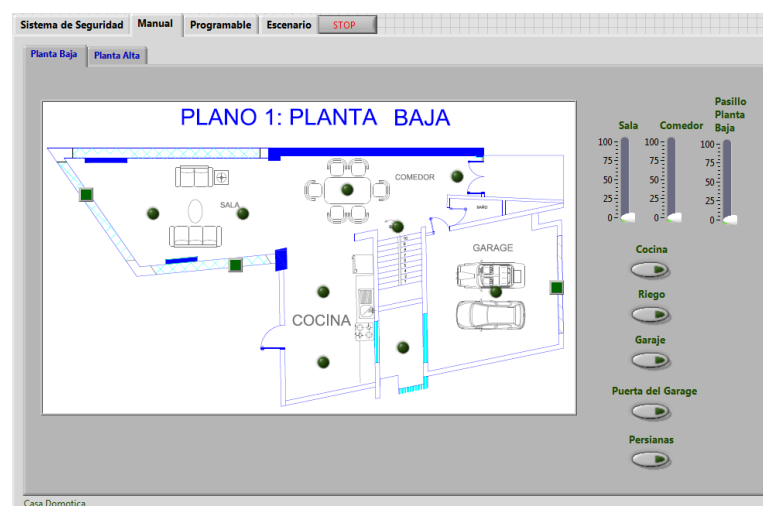


Fig. 5.18 Control manual planta baja del panel frontal

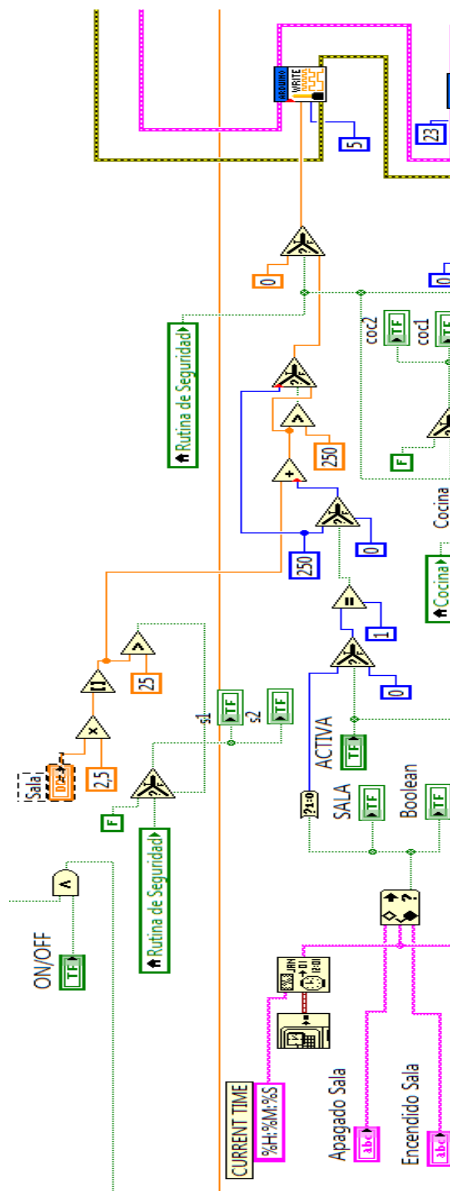


Fig. 5.19 Control programable y dimmer para la sala

En la Fig. 5.19 podemos visualizar la programación para dimerizar la sala que está de color naranja, el slide con nombre de sala es conectado a un bloque multiplicador ya que el valor máximo presente en el slide es 100, se puede apreciar la Fig. 5.20.

Dicho valor es multiplicado por 2.5 ya que el valor máximo de PWM fue establecido a 250, y el resultado del bloque multiplicador pasa por el bloque round to nearest para ser redondeado al entero mas cercano y solo se escribirá ese resultado en el pin asignado si el valor de PWM es mayor a 25.

La misma programación descrita en los párrafos anteriores se aplica para ejercer el control de dimmerización para el comedor, pasillo de planta baja y alta, dormitorio principal y secundario; hay que tener en consideración que el código adicional corresponde a otro control que será explicado más adelante.

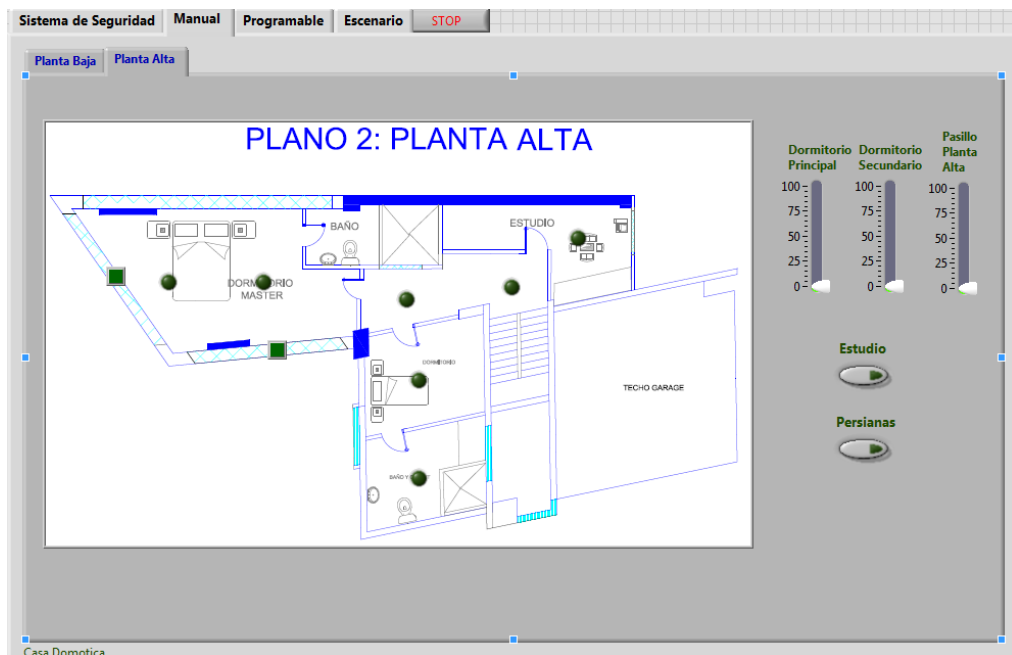


Fig. 5.20 Control manual planta alta del panel frontal

5.1.3 CONTROL POR TIEMPO

En la Fig. 5.12 se puede apreciar que existe un segmento en el panel que dice control de escenario programable, el objetivo de este control es que sea manipulado por el usuario a conveniencia; es decir, el usuario podrá disponer del tiempo de duración de encendido de cierta luminaria o de todas las luminarias de la vivienda.

La ventaja de esta programación es que; mientras se encuentre la botonera *activa* en estado On, funcionará el control todos los días sin tener que realizar comparaciones de meses o días.



Fig. 5.21 Control programable

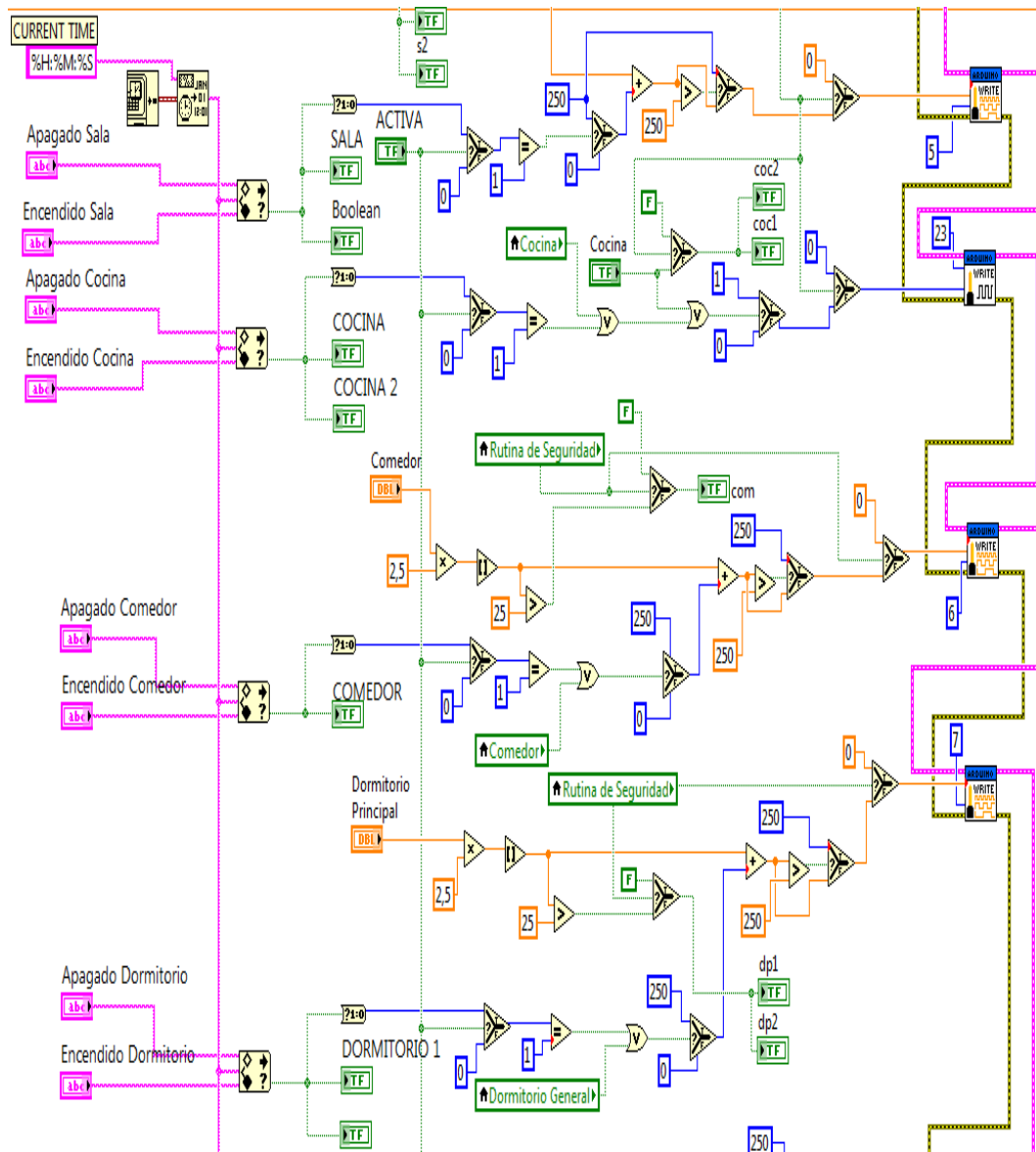


Fig. 5. 22 Control de encendido de luminarias por tiempo

Las variables locales *Cocina*, *Comedor*, *Dormitorio General* que están conectadas a las puertas lógicas or corresponden a los controles de escenario y manual del sistema; los mismos que serán detallados en las secciones correspondientes a cada control.

Para una mejor comprensión del código se tomará una muestra de la programación, es decir se analizará el control de la sala como se aprecia en la Fig. 5.22 ya que también está asociado al control por dimerización a excepción del control para la cocina.

Si por un instante asumimos que el usuario programe el encendido a las 18:00 pm y el apagado esté en las 05:00 am, siendo la hora del sistema las 17:59 pm; esto significa que dentro de un minuto se encenderá el led asignado a la sala ya que el bloque in range and coerse enviará una señal true si la hora del sistema esta dentro del rango asignado.

En el caso de que el bloque envíe una señal true, además de encenderse los LEDS, se convertirá el dato de boolean a numérico siendo éste el valor de uno.

Si el push button *activa* es presionado es decir es activado, permitirá que se ejecute la comparación entre el dato convertido y el valor numérico uno; y, al ser verdadera la condición se escribirá el valor de 250 a la salida del pin PWM correspondiente.

La programación adicional no la analizaremos puesto que corresponde al control por dimerización, tal programación ya fue explicada anteriormente.

Si analizamos la programación realizada para el control de la cocina, determinaremos que no está asociado a algún control dimerizable, por tal razón solo se envía el valor numérico de uno a la salida del pin 23; como se ve en la figura 5.23.

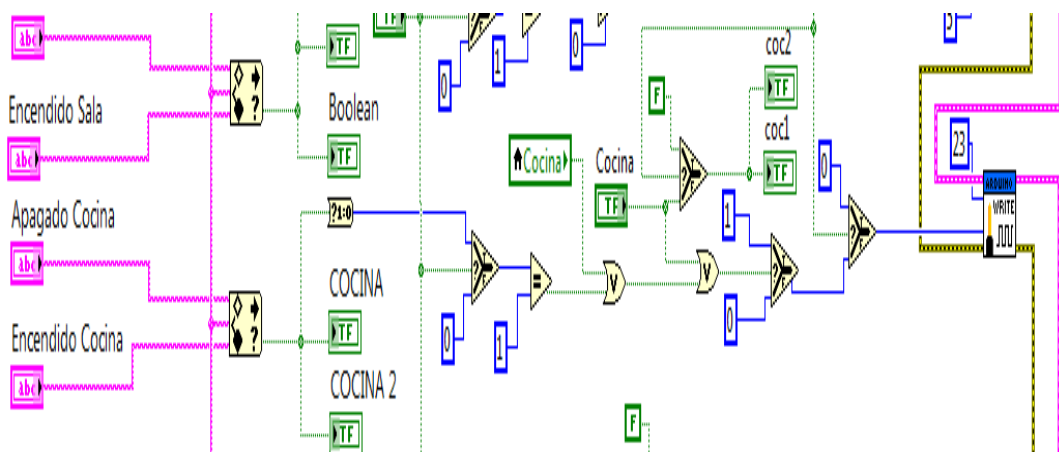


Fig. 5.23 Control programable para la cocina

Hay que considerar que la rutina de seguridad afecta al funcionamiento de todo el sistema; por tal motivo si éste estuviese activado no se ejecutará ningún control programable para luminarias.

5.1.4 CONTROL MANUAL

Esta sección fue creada bajo la consideración de que si el usuario por algún motivo no desee los controles anteriormente expuestos como el programable o los escenarios, pueda encender las luminarias sin complicación alguna manipulando botoneras y los controles para la dimerización.

En la Fig. 5.20 podemos observar que hay sliders para la dimerización y push buttons para una sencilla activación; y en la Fig. 5.24 se aprecia que no se ejerce ningún tipo de control para los push button y solo se enviará un uno digital a la salida del pin correspondiente.

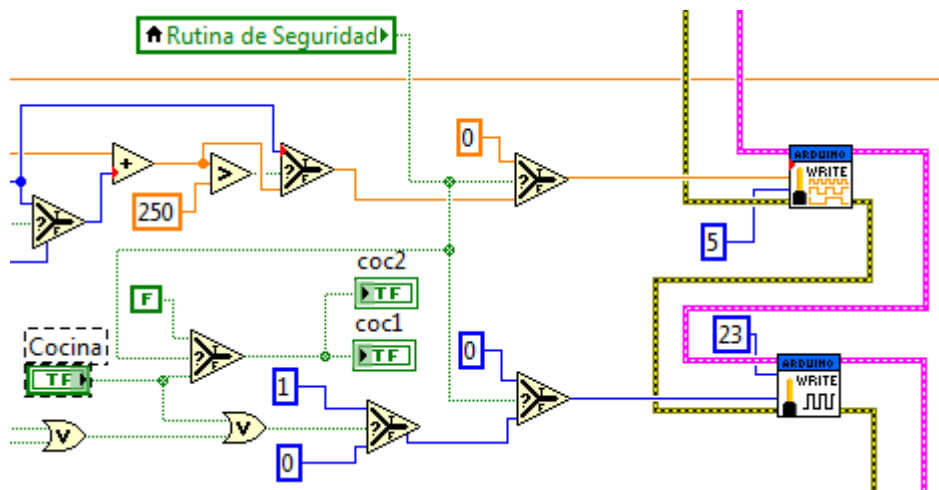


Fig. 5.24 Control manual diagrama de bloques.

De manera similar se realizó la programación para la planta alta, teniendo las mismas consideraciones para los sliders y push button como anteriormente se explicó.

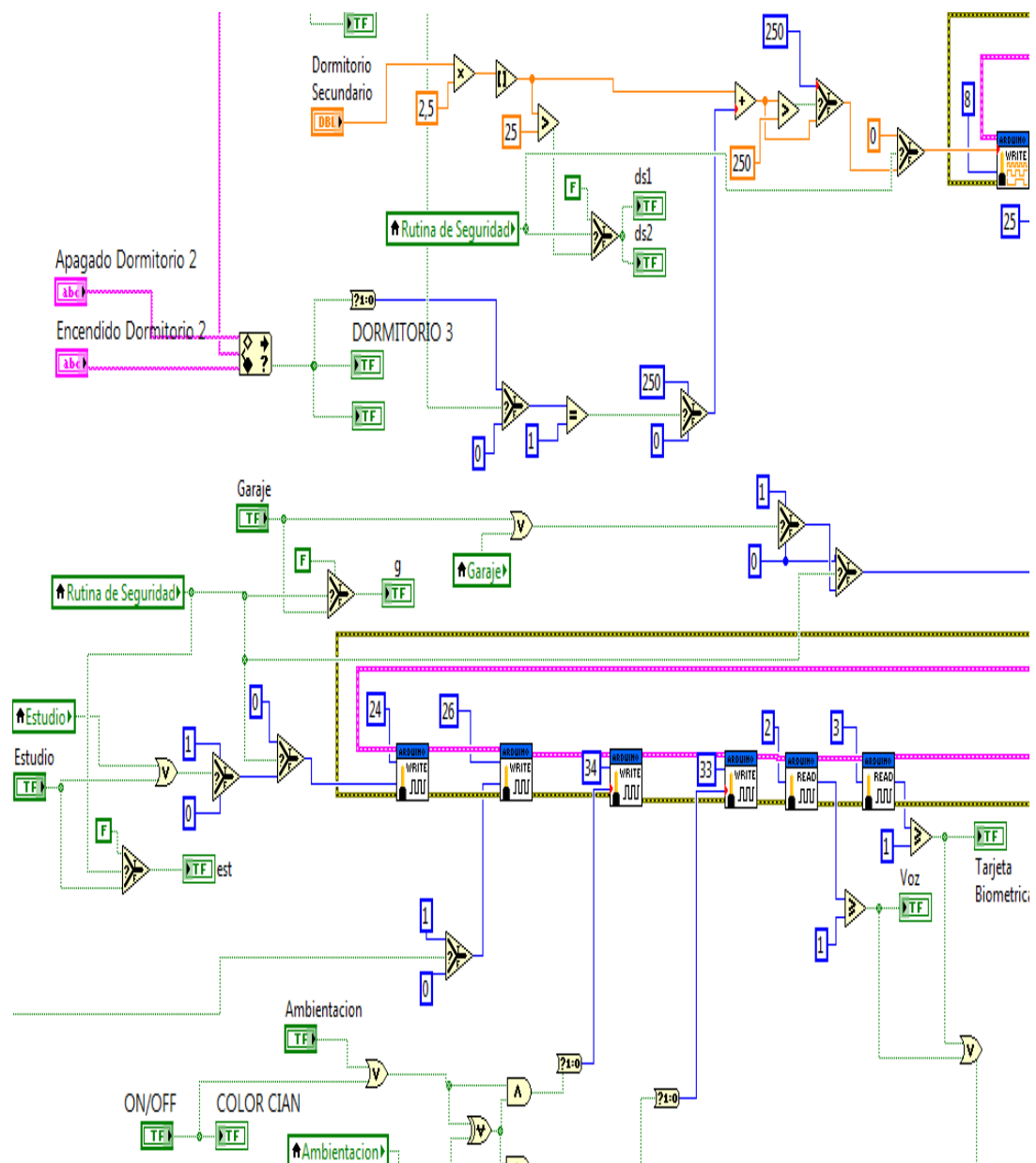


Fig. 5.25 Programación para control manual planta alta

Los slider de dormitorio principal y de dormitorio secundario al ser manipulados activarán los indicadores dp1, dp2, ds1 y ds2 dependiendo de los valores que se manipulen; aumentará o disminuirá la intensidad luminosa del led; y se escribirán a la salida PWM digital de los pines 7 y 8 del Arduino mega, se puede apreciar la programación realizada en la Fig. 5.25.

5.1.5 AMBIENTACIÓN

Previamente para que la tira de led funcione de acuerdo a las condiciones establecidas; se programó en el microcontrolador atmega 328 del Arduino uno para que exista 2 opciones de ambientación.

La opción uno quedo establecida para que el color que el rgb presentará sea el cian y sea aplicado a los 60 LEDS presentes en la tira.

La opción dos fue establecida para que exista un juego de colores, es decir la tira de LEDS presentará el color rojo, morado y azul; y, cada color tendra un determinado tiempo de duración; además de ejecutarse la secuencia hasta que el usuario desactive el push button programado en Labview.

Se presenta a continuación la programación realizada en el ide de Arduino para una mejor comprensión.

```

#include <PololuLedStrip.h>
#include <LabVIEWInterface.h>

PololuLedStrip <12> ledStrip;
#define LED_COUNT 60

rgb_color color[LED_COUNT];

int pin_ah = 3;
int pin_h = 2;
int On_Play= 0;
int On_Esc= 0;
int cent = 0 ;

void setup (){
  Serial.begin(9600);
  pinMode (pin_ah, INPUT);
  pinMode (pin_h, INPUT);
}

void loop (){
  On_Play = digitalRead (pin_h);
  On_Esc = digitalRead (pin_ah);

  do {
    On_Play = digitalRead (pin_h);
    if (On_Play ==1 ){
      aleat_R();
      delay (1000);
    }else
  }

apaga ();

On_Play = digitalRead (pin_h);
if (On_Play ==1){
  aleat_M();
  delay (1000);
}

```

```

    } else
    apaga ();

    On_Play = digitalRead (pin_h);
    if (On_Play ==1){
    aleat_A();
    delay (1000);
    }else
    apaga ();
} while ( On_Play ==1);

if (On_Esc ==1)
do{
    On_Esc = digitalRead (pin_ah);
    aleat_C();
} while (On_Esc ==1);
else
apaga ();
}

void aleat_R(){
    for(uint16_t i = 0; i< 60; i++)
    {
        color[i] = (rgb_color){255, 0, 0}; // Asigna al led strip el color rojo
    }
    ledStrip.write(color, LED_COUNT);
    delay ( 900);
}
void aleat_M(){
    for(uint16_t i = 0; i< 60; i++)
    {
        color[i] = (rgb_color){128, 0, 128}; // Asigna al led strip el color morado
    }
    ledStrip.write(color, LED_COUNT);
    delay ( 900);
}

void aleat_A(){
    for(uint16_t i = 0; i< 60; i++)
    {
        color[i] = (rgb_color){0, 0, 255}; // Asigna al led strip el color azul
    }
    ledStrip.write(color, LED_COUNT);
    delay (900);
}

```

```

void aleat_C(){
  for(uint16_t i = 0; i< 30; i++)
  {
    color[i] = (rgb_color){0, 255, 255}; // Asigna al led strip el color cian
  }
  ledStrip.write(color, LED_COUNT);
  delay ( 900);
}

void apaga (){
  for(uint16_t i = 0; i< 60; i++)
  {
    color[i] = (rgb_color){0, 0, 0}; //
  }
  ledStrip.write(color, LED_COUNT);
  delay (900);
}

```

Fig. 5.26 Código ambientación en IDE Arduino



Fig. 5.27 Controles para ambientación

En la Fig. 5.27 se puede apreciar los controles establecidos para la ambientación, cada push button activará cierta secuencia de luces las cuales ya fueron descritas anteriormente.

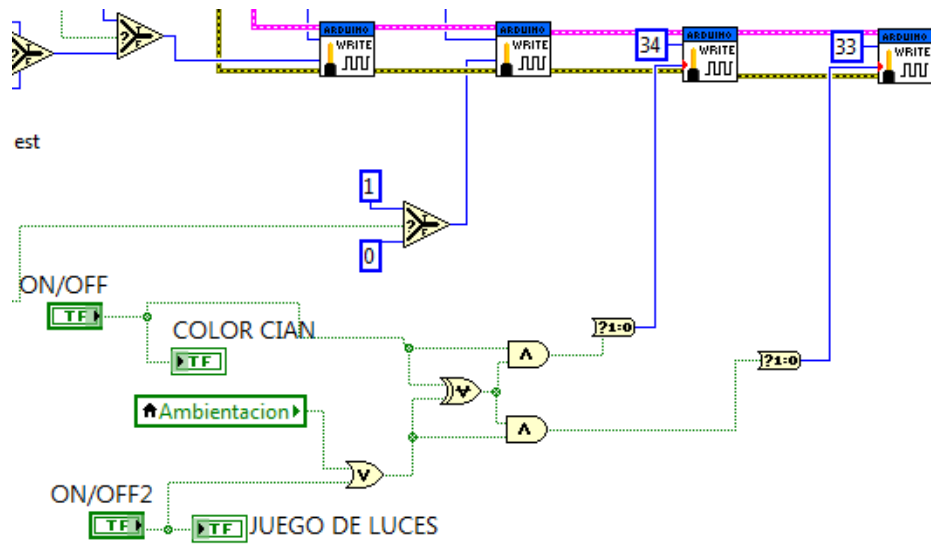


Fig. 5.28 Control programable de ambientación

5.2 SISTEMAS DE PERSIANAS

Toda vivienda cuenta con al menos tres ventanales y para demostración en la maqueta se ha optado por controlar las persianas de los ventanales de la planta baja y alta siendo en total cuatro, los mismos serán visualizados en el panel frontal como se indica en la Fig. 5.18 y 5.20.

Cabe indicar que la programación para el control de los motores para las persianas fue realizado en el PIC16F887 y solo se programó la activación en Labview.

Para comprender a totalidad este sistema es necesario incluir la programación realizada en el lenguaje c del mikroBASIC PRO for pic tal como se presenta a continuación:

```

const valor as byte[8] = (0x08,0x0C,0x04,0x06,0x02,0x03,0x01,0x09)
'arreglo para la secuencia de giro horario
'numero de vueltas completas que giran los motores de paso para el control de
persianas
const giro1 as word = 2
const giro2 as word = 2
const giro3 as word = 2
const pasos as word = 8      'numero de pasos de motor 28byg
const vuelta as word = 512  'cálculo de un vuelta completo

'variable para el control de pasos completos a realizar para las persianas
DIM mover1 AS INTEGER
DIM mover2 AS INTEGER
DIM mover3 AS INTEGER

'variable para el control de pasos del motor 1
DIM mod1 AS INTEGER
DIM mod2 AS INTEGER
DIM mod3 AS INTEGER

'creación de variables a operación numéricos
DIM STEPA AS INTEGER
DIM TURNA AS INTEGER
DIM STEPB AS INTEGER
DIM TURNB AS INTEGER
DIM STEPC AS INTEGER
DIM TURNC AS INTEGER
DIM STEPD AS INTEGER
DIM TURND AS INTEGER

'Declaración de los bits para las banderas de los motores de paso
DIM servo1 as bit
DIM servo2 as bit
DIM servo3 as bit

'Seteo de los puertos I/O (entradas y salidas del PIC)

```

```

TRISA=%11111111 ' 1 entrada
TRISB=%00000000 ' 0 salida
TRISC=%00000000 ' 0 salida
TRISD=%00000000 ' 0 salida
TRISE=%00000000 ' 0 salida

```

'Detalles de los puertos de Entrada sean analógicos

```

ANSEL=%00000000
ANSELH=%00000000
ADCON1 = 0x0F
OSCCON = 0x65
OPTION_REG = %10000110
Delay_ms(50)

```

main:

' Main program

```

STEPB = 0
STEPD = 0
STEPD = 0

```

'Contador de pasos y vueltas de los motores de paso

```

mover1 = vuelta * pasos * giro1
mover2 = vuelta * pasos * giro2
mover3 = vuelta * pasos * giro3

```

while(1) ' programa principal de control de motores de pasos

```

if(PORTA.B0)then

```

'sistema de reinicio; cuando pin es 1 se resetea STEPS a 0 y paraliza la acción de los siguientes puertos de control

```

    if(STEPB > 0)then
        STEPB = STEPB - 1
    end if
    if(STEPC > 0)then
        STEPC = STEPC -1
    end if
    if(STEPD > 0)then
        STEPD = STEPD -1
    end if
    servo1 = 0
    servo2 = 0
    servo3 = 0

```

```

else

```

```

    if(PORTA.B1)then

```

'puerto de activación del 1 motor de paso llamado servo cuando llegan al máximo se auto detiene

```

        servo1 = 1

```

```

    else

```



```

        if(STEPB >= mover1)then
            servo1 = 0
        end if
    end if
    if(PORTA.B2)then
        servo2 = 1
    else
        if(STEPC >= mover2)then
            servo2 = 0
        end if
    end if

    if(PORTA.B3)then
        servo3 = 1
    else
        if(STEPD >= mover3)then
            servo3 = 0
        end if
    end if

```

'sistema sumador y restador para escribir los movimientos de los motores de paso

```

    if(servo1)then
        if (STEPB >= mover1 )then
            STEPB = mover1
        else
            STEPB = STEPB + 1
        end if
    else
        if(STEPB > 0)then
            STEPB = STEPB - 1
        end if
    end if
    if(servo2)then
        if (STEPC >= mover2 )then
            STEPC = mover2
        else
            STEPC = STEPC + 1
        end if
    else
        if(STEPC > 0)then
            STEPC = STEPC - 1
        end if
    end if
    if(servo3)then
        if (STEPD >= mover3 )then
            STEPD = mover3
        end if
    end if

```

```

        else
            STEPD = STEPD + 1
        end if
    else
        if(STEPD > 0)then
            STEPD = STEPD - 1
        end if
    end if
end if

```

'Sistema de colocador del puerto B, C y D el control de motor de pasos
'Con su sistema de terminación en los extremos

```

if(STEPB >= mover1)then
    mod1 = 0
    PORTB = 0x00
else
    if(STEPB <= 0)then
        mod1 = 0
        PORTB = 0x00
    else
        mod1 = ( STEPB )mod( pasos )
        PORTB = valor[mod1]
    end if
end if
if(STEPC >= mover2)then
    mod2 = 0
    PORTC = 0x00
else
    if(STEPC <= 0)then
        mod2 = 0
        PORTC = 0x00
    else
        mod2 = ( STEPC )mod( pasos )
        PORTC = valor[mod2]
    end if
end if
if(STEPD >= mover3)then
    mod3 = 0
    PORTD = 0x00
else
    if(STEPD <= 0)then
        mod3 = 0
        PORTD = 0x00
    else
        mod3 = ( STEPD )mod( pasos )
        PORTD = valor[mod3]
    end if
end if

```

```
        end if
    end if
    'Retardo de 1 ms de cada paso
    delay_ms(1)
end
end
```

Cada sección del código c fue documentado para un mejor análisis; y, los cuatro pines del puerto B, C y D van conectados al ULN 2003APG el cual actúa como un módulo de fuerza para emitir una secuencia y girar los motores¹³.

Los pines digitales del Arduino mega van conectados a los del puerto A del PIC16F887 y la programación en Labview controla el accionamiento del sistema de persianas ya que está conectado al Arduino mega.

La botonera o push button programada en Labview enviará un alto o bajo a la salida de los pines digitales, tal pin estará conectado al microcontrolador correspondiente permitiendo la activación o no del sistema. Si observamos el panel frontal en la sección manual como indica la Fig. 5.29, podremos notar que cada piso de la vivienda cuenta con su push button para controlar las persianas.

¹³ Se usa el stepper motor 28BYJ-48, para comprender mejor el funcionamiento del mismo véase el datasheet que se ha adjuntado en los anexos.

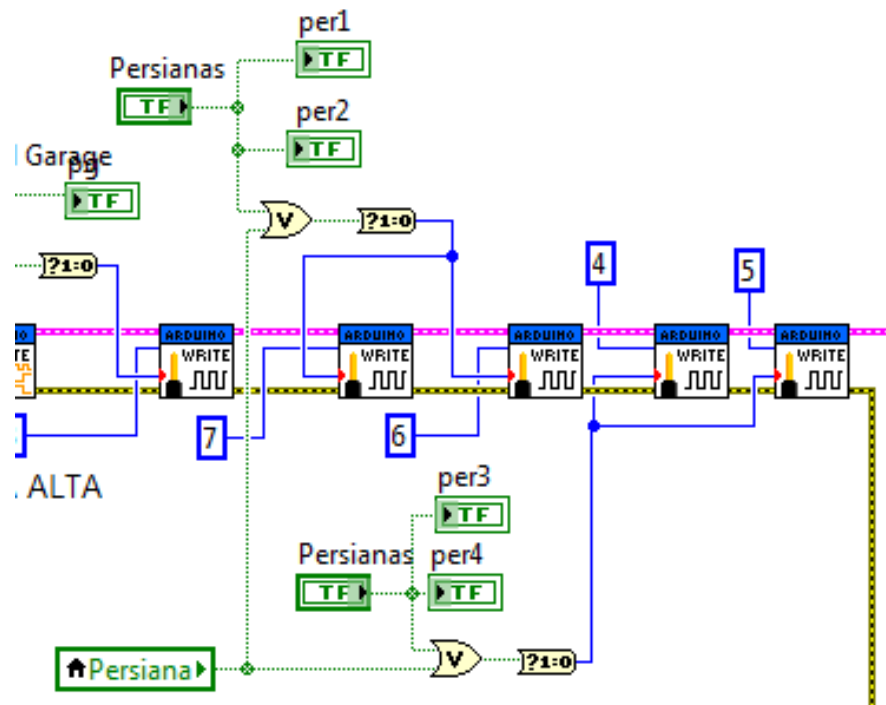


Fig. 5.29 Control del sistema de persiana

Si se presiona el push button *Persianas* correspondiente a la planta baja, se encenderán los LEDs *per1* y *per2*; y, se enviará un uno digital a la salida de los pines 6 y 7. El comportamiento es similar si presionamos el push button correspondiente a la planta alta, ya que se encienden los LEDs *per3* y *per4*; y, adicionalmente se envía un uno digital a la salida de los pines 4 y 5.

5.3 SISTEMA DE SEGURIDAD

5.3.1 MEMORIAS DE RUTINA PARA SIMULAR PRESENCIA DE UNA PERSONA EN LA CASA

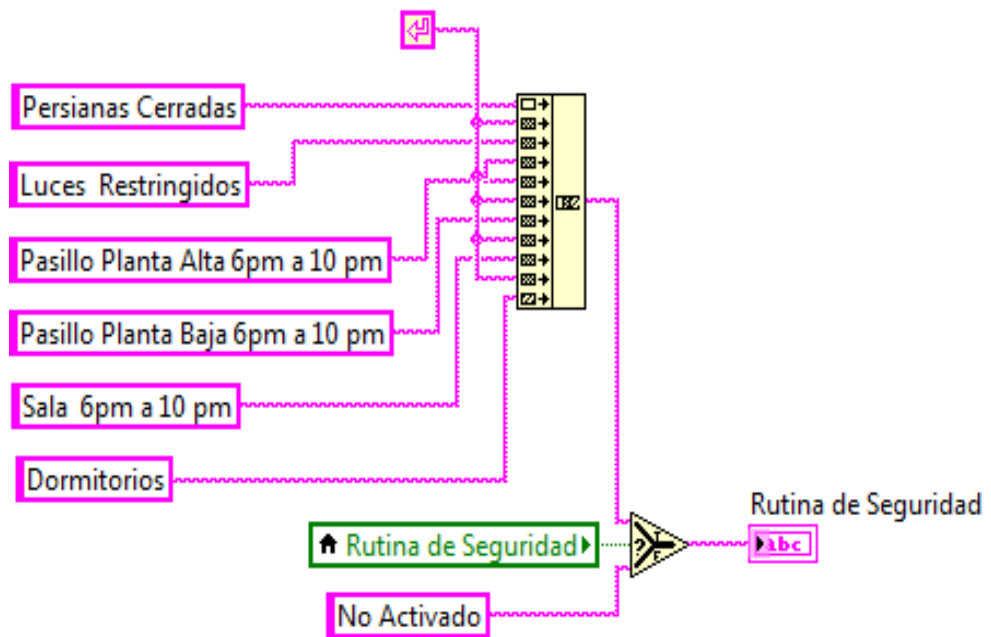


Fig. 5.30 Indicadores string–rutina de seguridad

En la Fig. 5.1 de la sección correspondiente a los escenarios del panel frontal, se encuentra los indicadores string y led para la rutina de seguridad, tal indicador string mostrará *No Activado* si la variable local *Rutina de Seguridad* no ha sido activada, caso contrario se presentará un detalle de lo que estará activo mientras se encuentre operando la rutina, véase la Fig. 5.30.

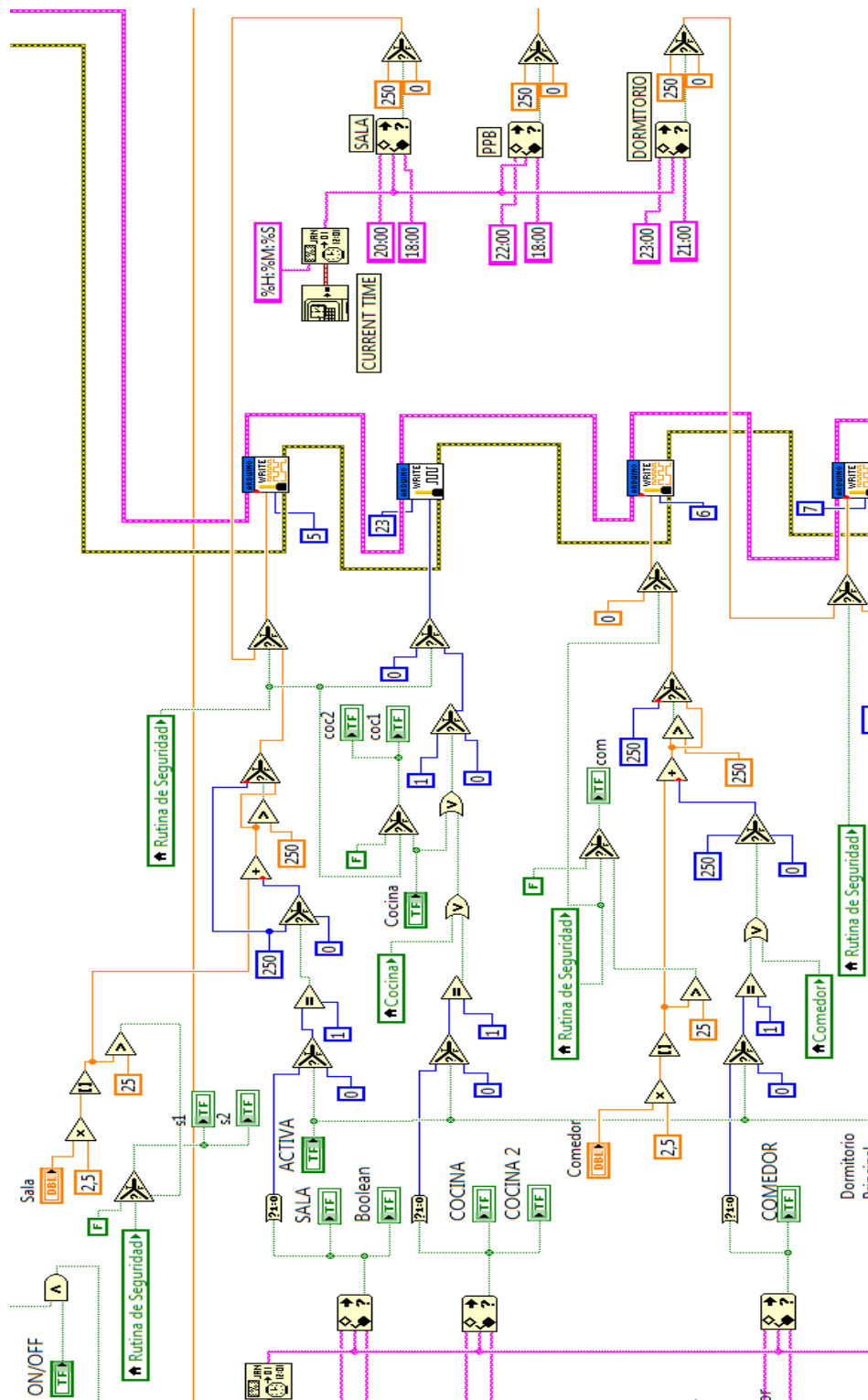


Fig. 5.31 Programación rutina de seguridad

La Fig. 5.31 detalla la programación para el funcionamiento de la rutina, y se puede apreciar cómo afecta al funcionamiento del control programable y dimerizable ya que solo se ejecutarán si la seguridad no es activada.

Las únicas programaciones que no son afectadas por la rutina es el reconocimiento de voz, control con tarjeta magnética, cámara de seguridad, sistema de control para garaje y riego; por evidentes razones no se puede permitir que la rutina ejerza control sobre lo anteriormente mencionado ya que es necesario que estén funcionando al momento de que el usuario decida ingresar a la vivienda, guardar el vehículo y asegurarse que el jardín se encuentre en buen estado.

La rutina de seguridad permite que se encienda la luminaria de la sala por un lapso de tiempo de 2 horas es decir desde las 18:00 pm hasta las 20:00 pm, la luminaria del pasillo de la planta baja se encenderá desde las 18:00 pm hasta las 22:00 pm y la del dormitorio principal se encenderá desde las 21:00 pm hasta las 23:00 pm inhabilitando todos los controles anteriormente programados; y, se ha configurado de esa manera porque es el patrón de evento más común en toda vivienda.

5.3.2 SISTEMA POR VOZ Y TARJETA PARA PUERTA

Antes de analizar la programación en Labview hay que considerar que para manipular la tarjeta NFC shield v1.6 se la programó en el entorno de Arduino, y de la misma manera para poder realizar el reconocimiento de voz la programación principal tuvo que realizarse en el ide de Arduino.

En Labview se ejecuta el control para los dos modulos permitiendo al usuario manipular la activación de los mismos, y a la vez ambos controles estan vinculados con la captura de imágenes usando la cámara digital. Ya que ambos módulos estan asociados con el funcionamiento de la cámara, analizaremos la programación de la captura de imágenes.

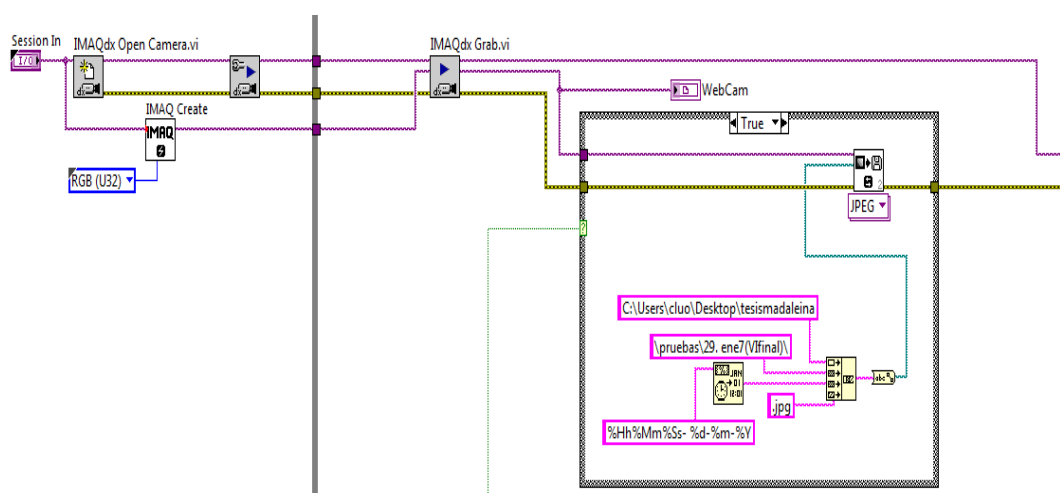


Fig. 5.32 Programación de captura de imagen

En la Fig. 5.32 se puede ver que se selecciona la cámara instalada con el bloque *sesion in*; mientras que con los bloques *IMAQdx open camera* e *IMAXdx configure grab* se inicia sesión con la cámara seleccionada y comienza la configuración para la grabación.

Con el *IMAQ create* se crea una localización temporal de memoria para las imágenes y con el bloque *IMAQdx grab* se puede visualizar la imagen cuando se haya conectado el bloque *image display* a *image out* del *imaqdx grab* cuyo tipo de dato será *rgb unsigned long* de 32 bytes.

Se realizará la captura y se guardará en una dirección siempre y cuando se tenga un alto o *high* al identificar la tarjeta magnética o cuando se haya reconocido la voz.

Se guarda un máximo de dos capturas por segundo en la dirección establecida en las constantes *string* y para evitar que se sobrescriba la captura se ha añadido el *format date/time string* con la finalidad de que se guarde la imagen bajo el nombre de la hora, minutos, segundos, día, mes y año del sistema con formato (.jpg).

Al final se cierra la grabación y se da el tratamiento de errores para evitar afectaciones al sistema domótico como se indica en la Fig. 5.33.

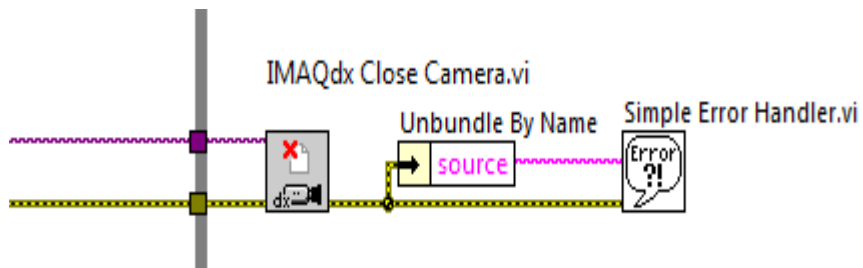


Fig. 5.33 Programación de cierre de captura de imagen

Antes de continuar con la programación en Labview para los controles de los módulos, es necesario incluir la programación realizada en el ide de Arduino, tal programación hace posible el funcionamiento de todo el sistema de seguridad.

El código que se presentará a continuación corresponde al easyvr 2.0 siendo el siguiente:

```
// Declaración de las librerías del Arduino, comunicación serial, control de puertos y pines

#if defined(ARDUINO) && ARDUINO >= 100
#include "Arduino.h"
#include "SoftwareSerial.h"
SoftwareSerial port(12,13);

#else // Arduino 0022 - use modified NewSoftSerial
```

```

#include "WProgram.h"
#include "NewSoftSerial.h"
NewSoftSerial port(12,13);
#endif

// Añadimos las librerías descargables de EasyVR para acoplar al ARDUINO y logre
una comunicación con el MEGA 2560
#include "EasyVR.h"
#define SND_Access_denied      1
#define SND_Access_granted    2
#define SND_Hello              3
#define SND_Please_repeat     4
#define SND_Please_say_your_password 5
#define SND_Please_talk_louder 6

// Declaración de los puertos a easyVR y logre una comunicación UART

EasyVR easyvr(port);
EasyVRBridge bridge;

// Declaración de las variables para la comunicación de datos del easyVR y Arduino

uint32_t mask = 0;
int8_t group = 0;
uint8_t train = 0;
char name[32];
// Declaración de pin 8 como salida digital para la integración con el Arduino mega

int led = 8;
void setup()
{
  // bridge mode?

  pinMode(led,OUTPUT); // el pin 8 ( led ) una salida digital

// Preparación del puente de comunicación de ARDUINO con el easyVR usando los
pines 0 1 12 y 13

  if (bridge.check())
  {
    cli();
    bridge.loop(0, 1, 12, 13);
  }
// Declaración del tiempo serial de comunicación de 9600 bps

Serial.begin(9600);
port.begin(9600);

```

```

digitalWrite(led,LOW);

// Programación para detector el easyVR y poder realizar la programación deseada
para el reconocimiento de VOZ

if (!easyvr.detect())
{
  Serial.println("EasyVR not detected!");
  for (;;)
  }
easyvr.setPinOutput(EasyVR::IO1, LOW);

//Una vez el easyVR se ha detectado se programa el lenguaje a ESPAÑOL

Serial.println("EasyVR detected!");
easyvr.setTimeout(5);
easyvr.setLanguage(EasyVR::SPANISH);
int16_t count = 0;

//Se realiza en enlace de la memoria del EASYVR para grupo de palabras grabadas
con sus respectivas voces para utilizar como la palabra ABRIR y con su clave SOL

if (easyvr.getGroupMask(mask)) // get trained user names and passwords
{
  uint32_t msk = mask;
  for (group = 0; group <= EasyVR::PASSWORD; ++group, msk >>= 1)
  {
    if (!(msk & 1)) continue;
    if (group == EasyVR::TRIGGER) // Busca las palabras del Grupo 1 como ABRIR
      Serial.print("Trigger: ");
    else if (group == EasyVR::PASSWORD) // Busca la palabra del PASSWORD o
      Contraseña NOW
      Serial.print("Password: ");
    else
    {
      Serial.print("Group ");
      Serial.print(group);
      Serial.print(": ");
    }
    count = easyvr.getCommandCount(group);
    Serial.println(count);
    for (int8_t idx = 0; idx < count; ++idx)
    {
      if (easyvr.dumpCommand(group, idx, name, train))
      {
        Serial.print(idx);

```

```

Serial.print(" = ");
Serial.print(name);
Serial.print(", Trained ");
Serial.print(train, DEC);
if (!easyvr.isConflict())
  Serial.println(" times, OK");
else
{
  // Añade en código de 8 bits la palabra obtenida

  int8_t confl = easyvr.getWord();
  if (confl >= 0)
    Serial.print(" times, Similar to Word ");
  else
  {
    confl = easyvr.getCommand();
    Serial.print(" times, Similar to Command ");
  }
  Serial.println(confl);
}
}
}
}
}
easyvr.setLevel(EasyVR::HARDER);
easyvr.playSound(SND_Hello, EasyVR::VOL_FULL);
}

// Programación sin fin del control de VOZ

void loop()
{
  // Declaración de las variables principales

  int idx_cmd;
  int idx_pwd;

  // easy empieza a escuchar con su led integrado en ON

  easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)

  // easy VR esperando la palabra del grupo 1 ABRIR

  Serial.println("Say a name in Group 1");
  easyvr.recognizeCommand(1); // recognise command in group 1
  // Se ha reconocido la palabra del Grupo 1

```

```

while (!easyvr.hasFinished()); // wait for user name
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
// easyVR terminado de escuchar con su led integrado en OFF

idx_cmd = easyvr.getCommand(); // get recognised user name
if (idx_cmd >= 0)
{
  Serial.print("Name: ");
  if (easyvr.dumpCommand(1, idx_cmd, name, train))
    Serial.println(name);
  else
    Serial.println();

// Una vez realizado el paso anterior se emitirá un sonido y se prepara el password o
contraseña

  easyvr.playSound(SND_Please_say_your_password , EasyVR::VOL_FULL); // ask
for password

  easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)
  Serial.println("Say the password");
// Donde se repita la funcion de esperar, oir y comprobar si la contraseña es
correcta, donde si no lo es se reanuda el procedimiento desde el principio y si lo es
se energiza el led de salida 8

  easyvr.recognizeCommand(EasyVR::PASSWORD); // set group 16
  while (!easyvr.hasFinished()); // wait for password

  easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off

  idx_pwd = easyvr.getCommand(); // get recognised password

  if (idx_pwd >= 0)
  {
    Serial.print("Password: ");

    if (easyvr.dumpCommand(EasyVR::PASSWORD, idx_pwd, name, train))
    {
      Serial.print(" = ");
      Serial.println(name);
    }
    else

      Serial.println();

    if ( idx_pwd == idx_cmd) // index of username and password are the same,
access granted

```

```

    {
// Se logra escuchar las 2 palabras y se confirma que son las palabras guardadas en
la memoria del easyVR donde se ejecuta la acción del easyVR en energizar un led

        Serial.println("Access granted");

        easyvr.playSound(SND_Access_granted , EasyVR::VOL_FULL);

// Se energiza el led por 2500 milisegundos y luego se apaga

        digitalWrite(led,HIGH);
        delay(2500);           // wait a while
        digitalWrite(led,LOW);
    }
    else // index of username and password differ, access is denied

    {

// Paso de fallo la confirmacion del LED

        Serial.println("Access denied");
        easyvr.playSound(SND_Access_denied , EasyVR::VOL_FULL);
    }

}

int16_t err = easyvr.getError();

if (easyvr.isTimeout() || (err >= 0)) // password timeout, access is denied
{

    Serial.println("Error, try again...");
    easyvr.playSound(SND_Access_denied , EasyVR::VOL_FULL);
// Se reanuda la operacion de nuevo

}
}

else
{

//También el easyVR posee un contador de tiempo para escuchar la voz deseada,
donde easyVR da su tiempo preestablecido por 1 seg, si sobrepasa el tiempo es
como anunciar error y se reanuda el procedimiento sin ejecutar la acción deseada
para la comunicación del mega Arduino

        if (easyvr.isTimeout())

```

```

        Serial.println("Timed out, try again...");
    int16_t err = easyvr.getError();
    if (err >= 0)
    {
        Serial.print("Error ");
        Serial.println(err, HEX);
    }
}
}

```

Se ha agregado comentarios a la programación que ha sido presentada para detallar las funciones de las variables, comandos y la forma de operación del módulo del reconocimiento de voz.

Se especifican las palabras (llaves por voz) que activarán la captura de imágenes y se detalla la programación de los mismos, haciendo que el código de programación sea mas manipulable.

Terminado el análisis anterior se añadira el código que permite el funcionamiento de la tarjeta magnética siendo este el siguiente:

```

//incluir liberaría PN532 para la comunicación de la tarjeta BIOMÉTRICA
#include <PN532.h>

// Declaración de los pines 10 11 12 y 13 para la comunicación con el módulo de la
tarjeta biométrica
#define SCK 13
#define MOSI 11
#define SS 10

```



```

#define MISO 12

// Declaración de pin 8 como constante door como puerto de salida
#define door 8

// Declaración de la tarjeta BIOMETRICA con su respectiva ID de la tarjeta a utilizar
#define ID_TARJETA 2868713947

// Usando el comando de PN532 para declarar la comunicación dada
PN532 nfc(SCK, MISO, MOSI, SS);
void setup()
{
    Serial.begin(9600); // Declaración de Puerto de comunicación serial a 9600 bps
    nfc.begin();
    nfc.SAMConfig();
    // Declaración de la constante door como Salida
    pinMode(door, OUTPUT);
    Serial.println("Placa preparada para leer !");
}
// Programación Principal
void loop()
{
    //uint32 como variable para leer el código dada de la tarjeta BIOMETRICA
    uint32_t id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);

    // Si el id es igual a código de la tarjeta BIOMETRICA realiza la acción de encender o
    apagar el LED
    if ( id == ID_TARJETA)
    {
        digitalWrite( door, !digitalRead (door)); // Escritura en el pin door
        delay (1000); // Tiempo de retardo un segundo
    }
}

```

Para determinar cuál es el id de la tarjeta es necesario realizar otra programación en la cual los códigos para el void setup y loop serán los siguientes:

```
void setup()
{
  Serial.begin(9600);
  nfc.begin();
  // Configura la Shield para leer etiquetas RFID:
  nfc.SAMConfig();
  Serial.println("Placa preparada para leer!");
}

void loop()
{
  uint32_t id = nfc.readPassiveTargetID(PN532_MIFARE_ISO14443A);
  if( id != 0 )
  {
    Serial.print("ID= "); Serial.println(id);
    delay(1000);
  }
}
```

Obtenida la id de cualquier tarjeta se puede programar de acuerdo a las necesidades presentadas; en nuestro caso permitirá la apertura de la puerta y el cerrado de la misma.

Comprendido el código anterior se retoma la programación en Labview para incluir los controles para la tarjeta magnética y el módulo de voz.

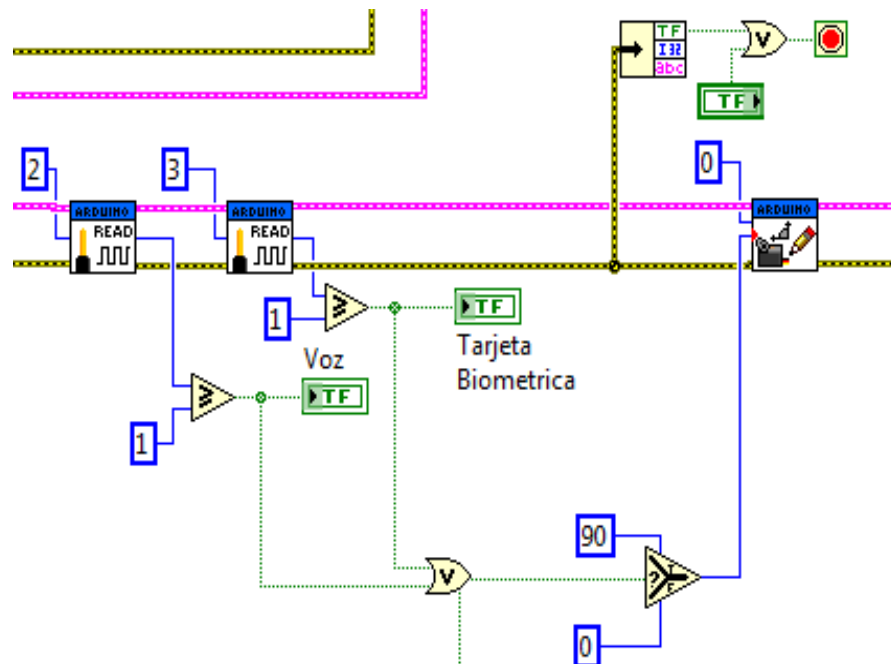


Fig. 5.34 Programación de control para los módulos

Cuando el módulo nfc identifica el id de la tarjeta, enviará un alto a la salida del pin 8, tal pin es leído por la tarjeta Arduino mega en el pin 2 y procesado por Labview; permitiendo que se encienda el led indicador y a la vez la captura de imágenes; se puede apreciar la programación en la Fig. 5.34.

De una manera muy similar se manipula el modulo easyvr, ya que es configurado para que cuando haga el reconocimiento de voz envíe un alto a la salida del pin 8, tal señal es leída en el pin 3 de la tarjeta

Arduino mega y procesada por Labview para que se active el led indicador y se proceda a la captura de imágenes.

En la Fig. 5.34 podemos apreciar que cuando esté activado el led indicador de voz o tarjeta magnética; el bloque selector if enviará el valor de 90 a la entrada *angle/degrees* del bloque *servo write angle*.

El bloque de escritura de servos permite manipular los servos conectados, se puede apreciar en la Fig. 5.35, para ello es necesario escribir el ángulo a mover y especificar el número del servo que se ha conectado.

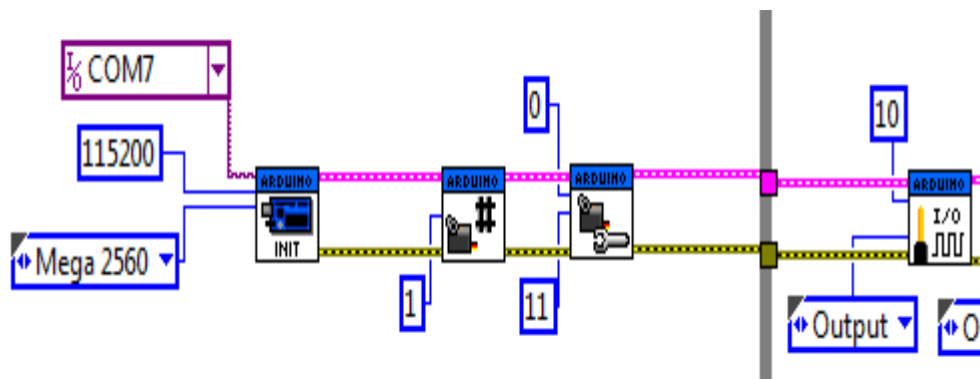


Fig. 5.35 Bloques para configuración de servo

Se presenta a continuación la imagen del panel frontal del sistema de seguridad , y hay que considerar que se encuentra en estado de sin ejecución; mas adelante se verifica el funcionamiento del mismo.

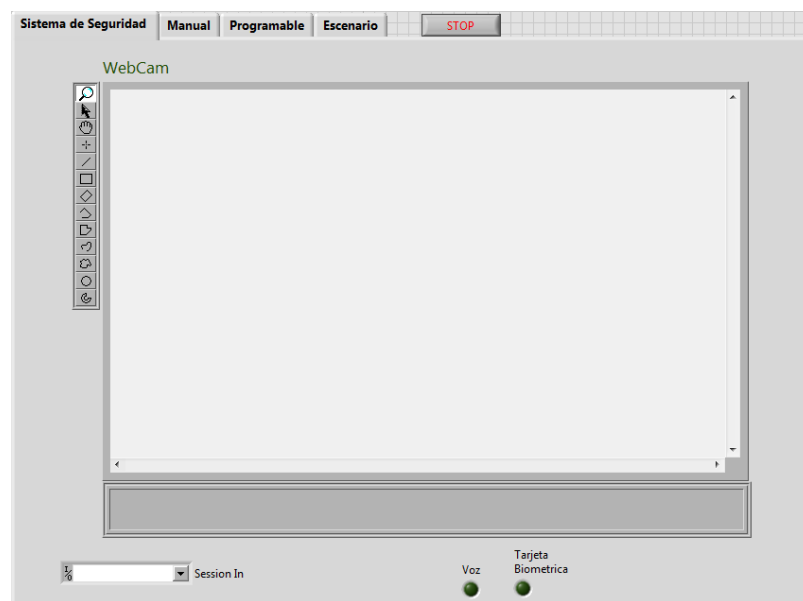


Fig. 5.36 Panel frontal del sistema de seguridad

5.4 SISTEMA DE RIEGO

Una electroválvula esta conectada con un switch y la fuente de alimentación, por demostración se ha incorporado un led asumiendo que ejecutará la función del switch en la vida real; y cada vez que se active la botonera en el panel frontal de Labview se encenderá el led, vease la Fig. 5.18; en la imagen que se presenta en la pagina siguiente se detalla la programación realizada.

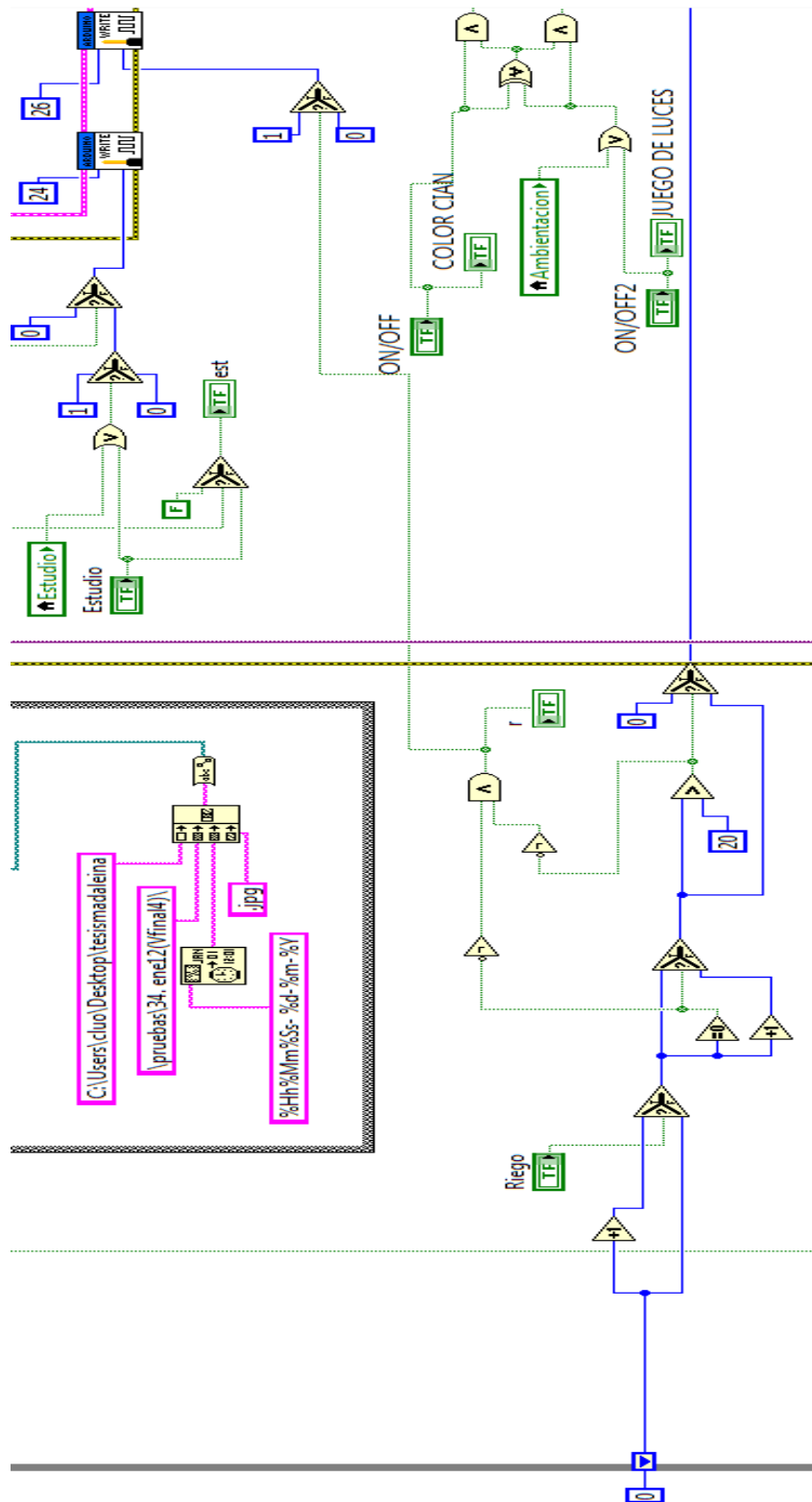


Fig. 5.37 Control del sistema de riego

El control del sistema de riego no presenta mayor complejidad puesto que solo se manipula la salida digital del pin 26 de la Arduino mega y se ejerce un mínimo control de tiempo para duración del regado.

5.5 SISTEMA DE GARAGE

La programación principal para el funcionamiento del motor se realizó en el PIC16F887 dejando unicamente el control para programar en Labview.

El pin asignado para ejecutar el control es el 27, tal pin va conectado al puerto A del pic y cuyas salidas van conectados al puente h permitiendo la secuencia de giro del motor.

La programación de secuencia del motor fue realizada en lenguaje c usando el microPRO for pic como se visualiza a continuación:

```
const valor as byte[8] = (0x08,0x0C,0x04,0x06,0x02,0x03,0x01,0x09)
Arreglo para la secuencia de giro horario
Número de vueltas completas que gira el motor de paso para el control del garaje
const giro1 as word = 2
const pasos as word = 8      'numero de pasos de motor 28byg
const vuelta as word = 512  'cálculo de un vuelta completo

'variable para el control de pasos completos a realizar para el garaje
DIM mover1 AS INTEGER
```

Variable para el control de pasos del motor 1

DIM mod1 AS INTEGER

Creación de variables a operación numéricos

DIM STEPA AS INTEGER

DIM TURNA AS INTEGER

DIM STEPB AS INTEGER

DIM TURNB AS INTEGER

DIM STEPC AS INTEGER

DIM TURNC AS INTEGER

DIM STEPD AS INTEGER

DIM TURND AS INTEGER

'Declaración de los bits para las banderas de los motores de paso

DIM servo1 as bit

'Seteo de los puertos I/O (entradas y salidas del PIC)

TRISA=%11111111 ' 1 entrada

TRISB=%00000000 ' 0 salida

TRISC=%00000000 ' 0 salida

TRISD=%00000000 ' 0 salida

TRISE=%00000000 ' 0 salida

'Detalles de los puertos de Entrada sean analógicos

ANSEL=%00000000

ANSELH=%00000000

ADCON1 = 0x0F

OSCCON = 0x65

OPTION_REG = %10000110

Delay_ms(50)

main:

' Main program

STEPB = 0

STEPC = 0

STEPD = 0

'Contador de pasos y vueltas del motor de paso

mover1 = vuelta * pasos * giro1

mover2 = vuelta * pasos * giro2

mover3 = vuelta * pasos * giro3

while(1) ' programa principal de control de motores de pasos

if(PORTA.B0)then

Sistema de reinicio; cuando pin es 1 se resetea STEPS a 0 y paraliza la acción de los siguientes puertos de control

if(STEPB > 0)then


```

        STEPB = STEPB - 1
    end if
    servo1 = 0
else
    if(PORTA.B1)then
        Puerto de activación del 1 motor de paso llamado servo cuando llegan al máximo se
        auto detiene
        servo1 = 1
    else
        if(STEPB >= mover1)then
            servo1 = 0
        end if
    end if
end if
Sistema sumador y restador para escribir los movimientos de los motores de paso
if(servo1)then
    if (STEPB >= mover1 )then
        STEPB = mover1
    else
        STEPB = STEPB + 1
    end if
    else
        if(STEPB > 0)then
            STEPB = STEPB - 1
        end if
    end if
end if
Sistema de colocador del puerto B, C y D el control de motor de pasos con su
sistema de terminación en los extremos
if(STEPB >= mover1)then
    mod1 = 0
    PORTB = 0x00
else
    if(STEPB <= 0)then
        mod1 = 0
        PORTB = 0x00
    else
        mod1 = ( STEPB )mod( pasos )
        PORTB = valor[mod1]
    end if
end if

end if
'Retardo de 1 ms de cada paso
delay_ms(1)
end
end

```

El sistema de garaje además de podérsele controlar con un push button virtual también se lo puede activar haciendo uso de un control remoto y la programación que permite el funcionamiento del mismo se muestra en la figura 59.

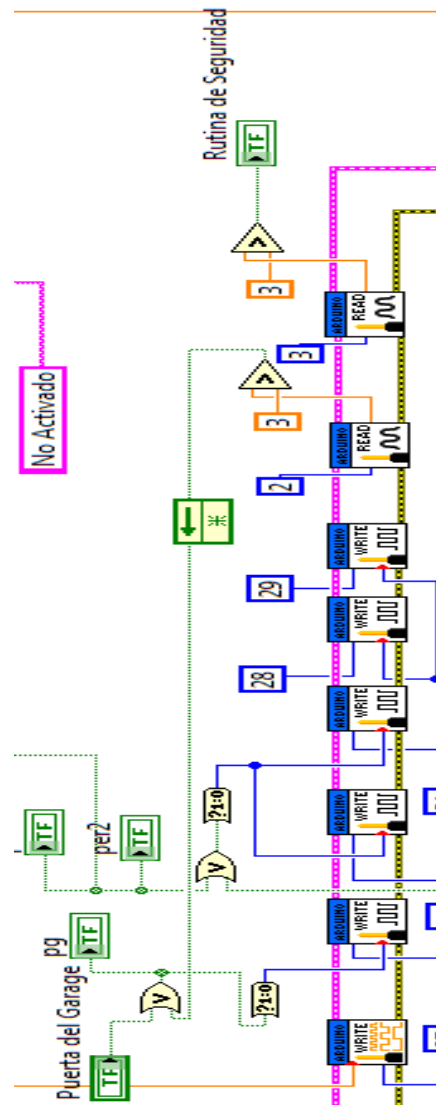


Fig. 5.38 Control remoto para el garaje

La frecuencia obtenida del control remoto es leída por el receptor y éste envía un valor de 0 o 5 voltios al pin 2; en caso de ser 5 voltios se envía un uno digital al pin 27 permitiendo el funcionamiento del motor.

5.6 DISEÑO DE PRUEBAS

Se ha optado realizar 3 secciones de verificación del sistema siendo éstos los siguientes:

La primera sección engloba todos los subsistemas en funcionamiento independiente; es decir, no se realizó asociaciones, integraciones o uniones entre sistemas.

Las pruebas correspondientes a la segunda sección fueron realizadas en base a las gestiones de la domótica propuestas para el desarrollo del proyecto.

La tercera sección corresponde a la integración de todos los subsistemas y gestiones para la validación del sistema domótico total.

A continuación validaremos cada subsistema acorde a lo especificado anteriormente:

SECCION I

PROGRAMACIÓN DE ESCENARIOS:

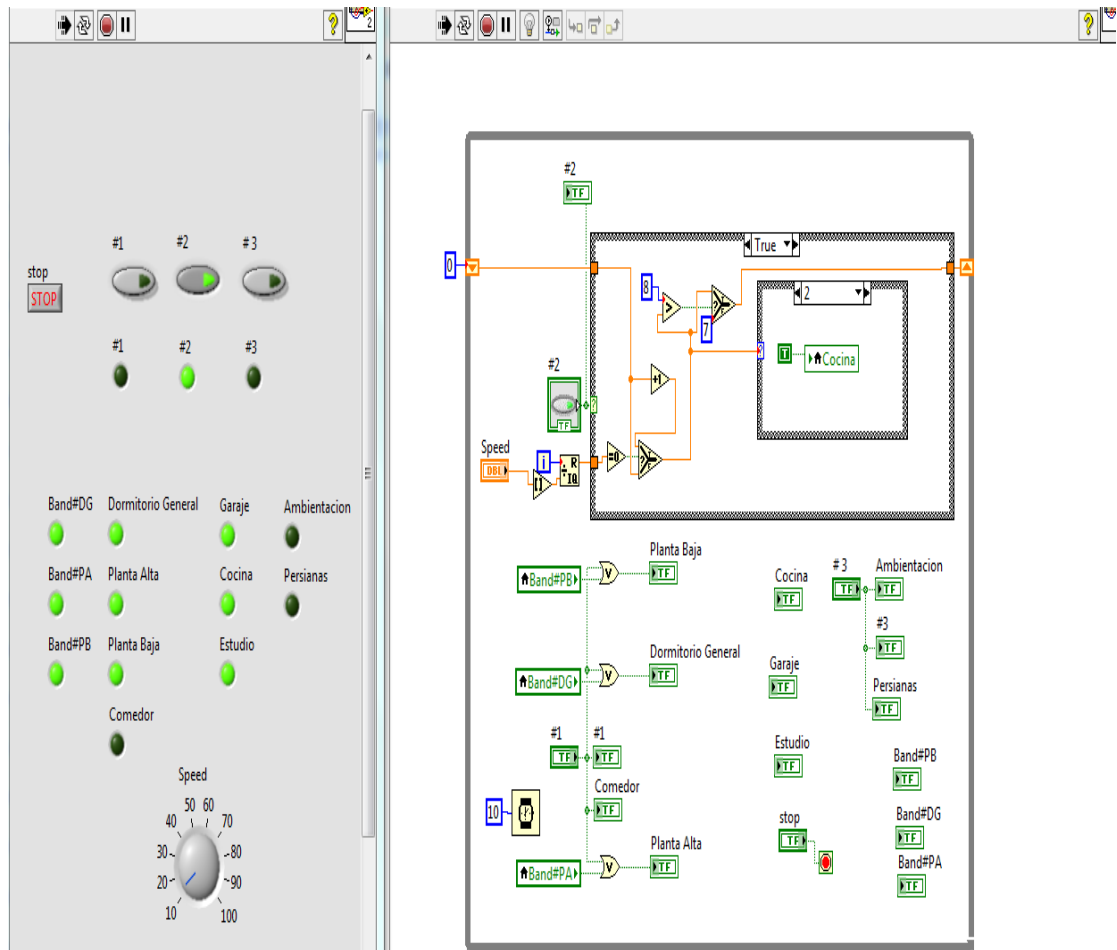


Fig. 5.39 Prueba uno para escenarios

Se realizó los controles para controlar el encendido de las luminarias permitiendo una secuencia para encender y apagar los LEDS, y se agregó un speed para posteriormente setear a una velocidad determinada de encendido y apagado.

En esta etapa no se realiza la distribución de pines ni tampoco se agrega los bloques de escritura para pines digitales de la Arduino ya que se está desarrollando lo que va a ser el sistema de iluminación.

DIMERIZACIÓN POR AMBIENTE:

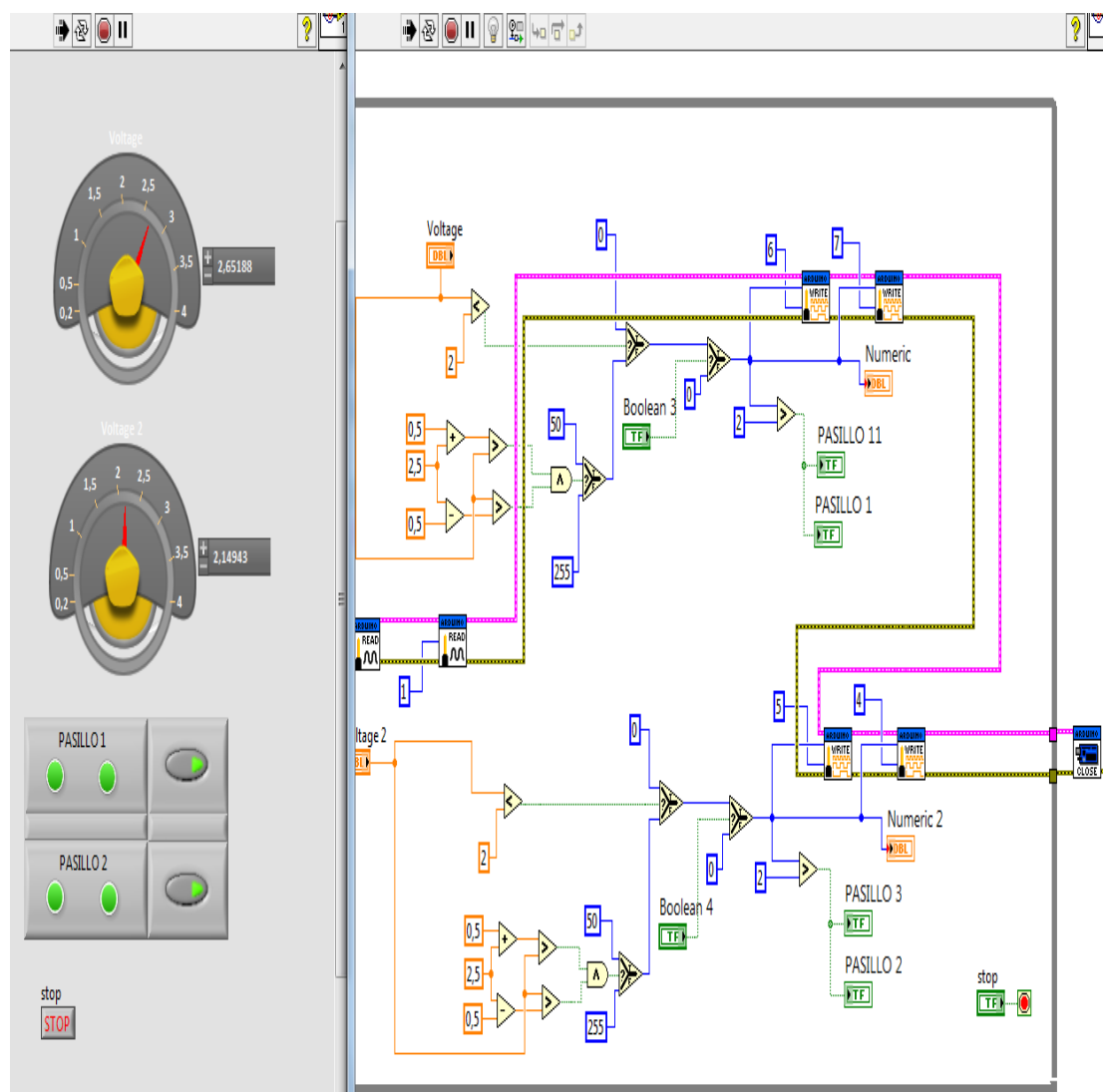


Fig. 5.40 Prueba dos para dimerización

Para poder dimerizar cierto led es necesario conocer la cantidad lumínica presente en el ambiente, para ello las primeras pruebas fueron de lectura del LDR y establecer rangos de voltaje para determinar los valores de PWM a escribir en los pines de la Arduino.

Los pines seleccionados son solo para la verificación de la programación y una vez concluido ese segmento de código se añadió el control de hora para completar el funcionamiento del control lumínico; se puede apreciar la programación en la Fig. 5.41.

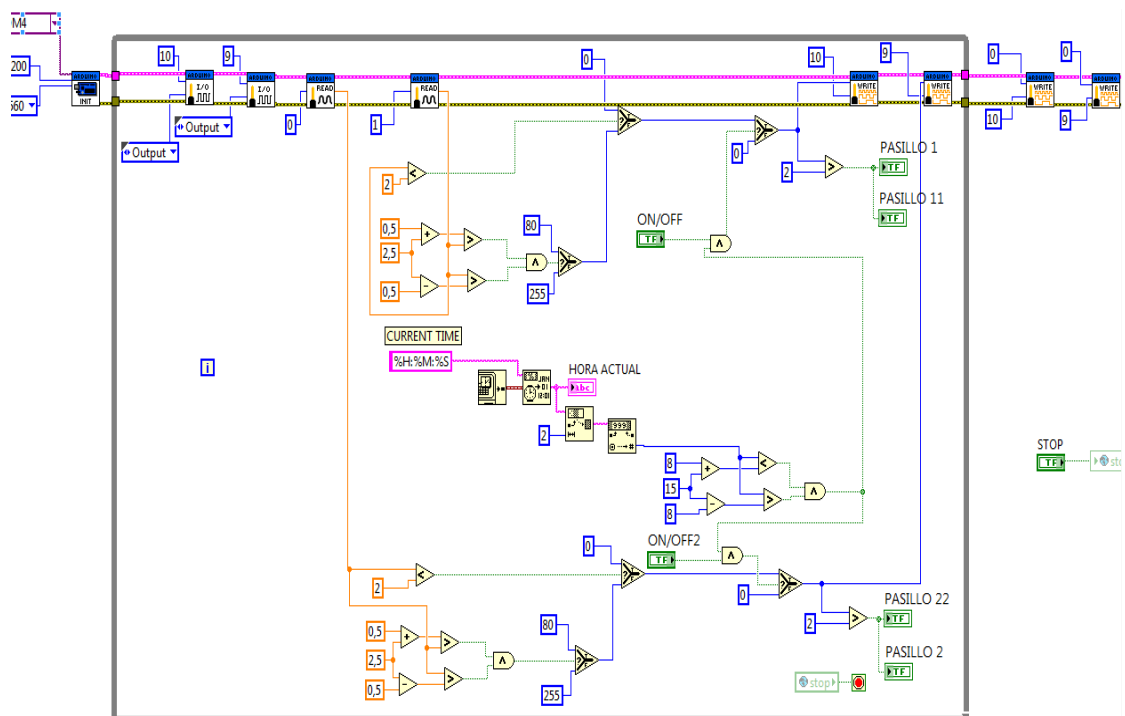


Fig. 5.41 Prueba tres dimerización por hora

CONTROL POR TIEMPO:

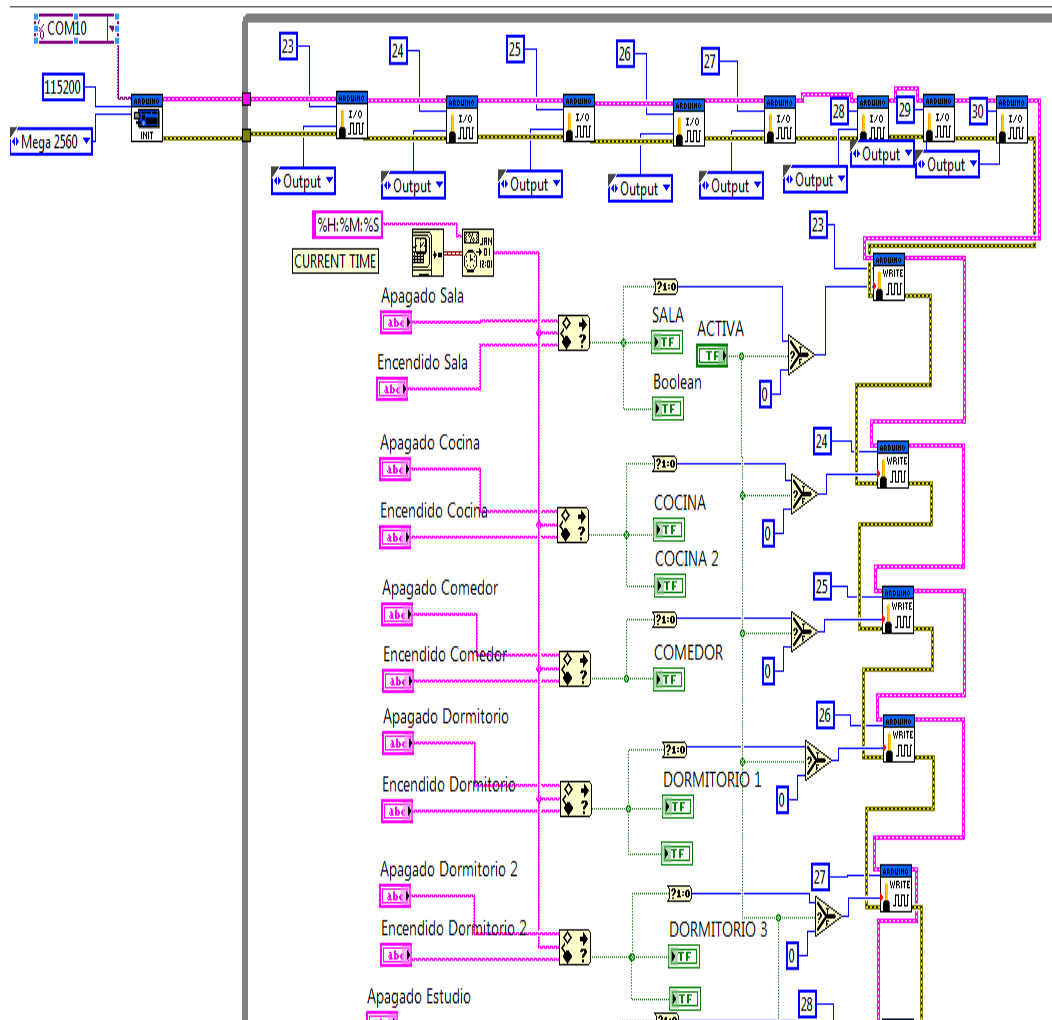


Fig. 5.42 Prueba cuatro para el control por tiempo

Se realiza las comparaciones de hora y minutos entre la hora del sistema y la ingresada por el usuario; en el caso de que se cumpla la condición además de encenderse el led indicador virtual se encenderá los LEDS que se han conectado a la Arduino.

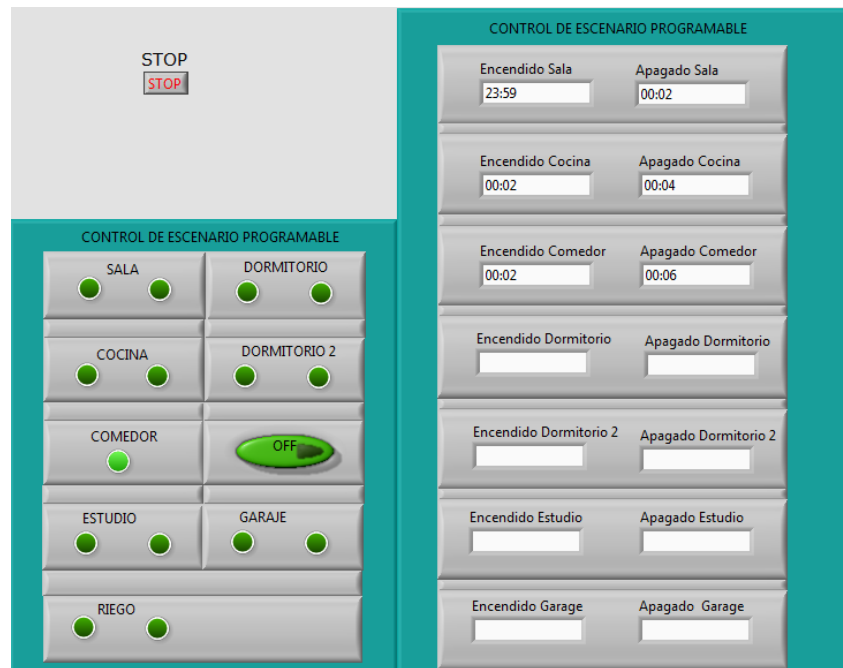


Fig. 5.43 Prueba cinco-panel frontal

Para este ejemplo la hora del sistema es 00:04 am y se verificó el funcionamiento del control de luces por tiempo, véase la Fig. 5.43.

AMBIENTACIÓN:

La programación de la tira fue realizada en código c haciendo uso del ide de Arduino y para constatar que funciona conforme a lo programado se diseñó dos controles que activarán cada secuencia; ésta primera programación fue para prueba y se minimizó bloques para la codificación final, como se indica en la Fig. 5.44.

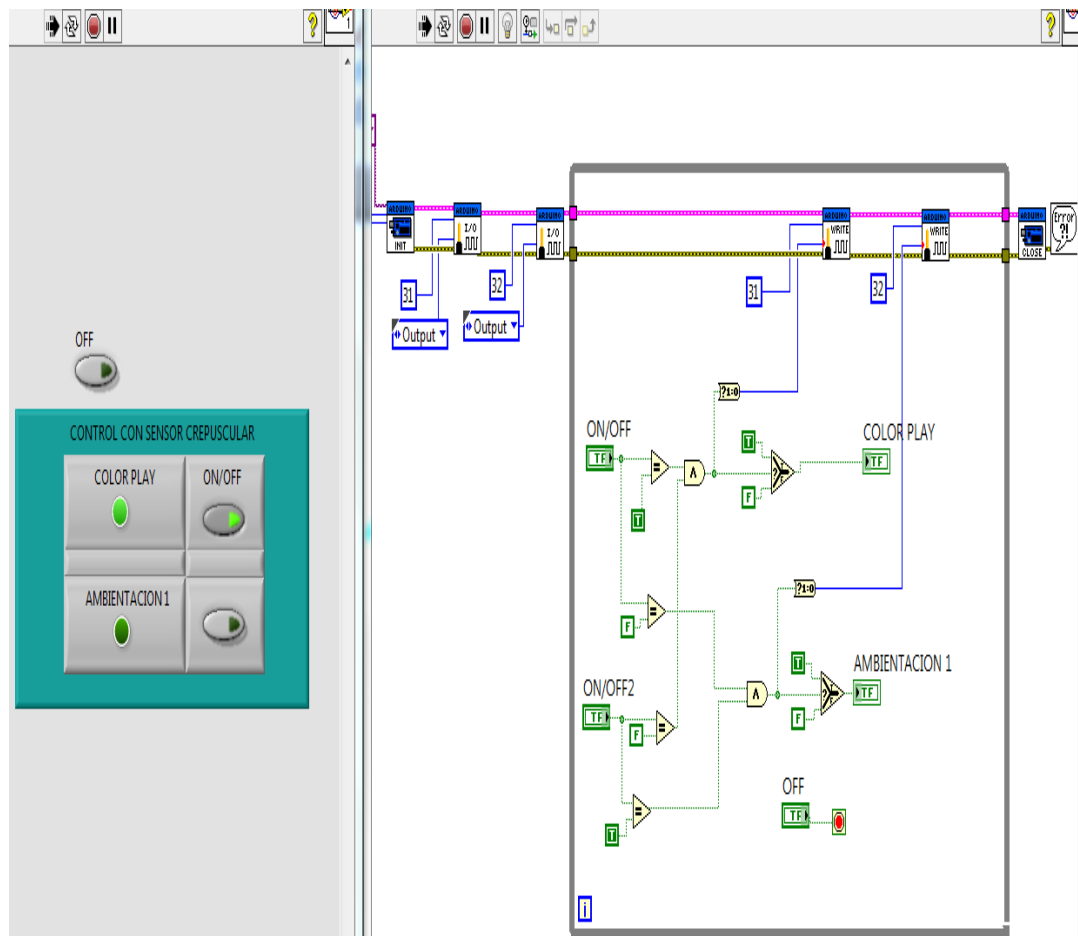


Fig. 5.44 Prueba seis del control para ambientación

CAPTURA DE IMÁGENES:

Como el sistema de seguridad funciona con la tarjeta y el módulo reconocimiento de voz, por simulación se creó botoneras o push button en reemplazo del funcionamiento de los módulos tal como se indica en la Fig. 5.45.

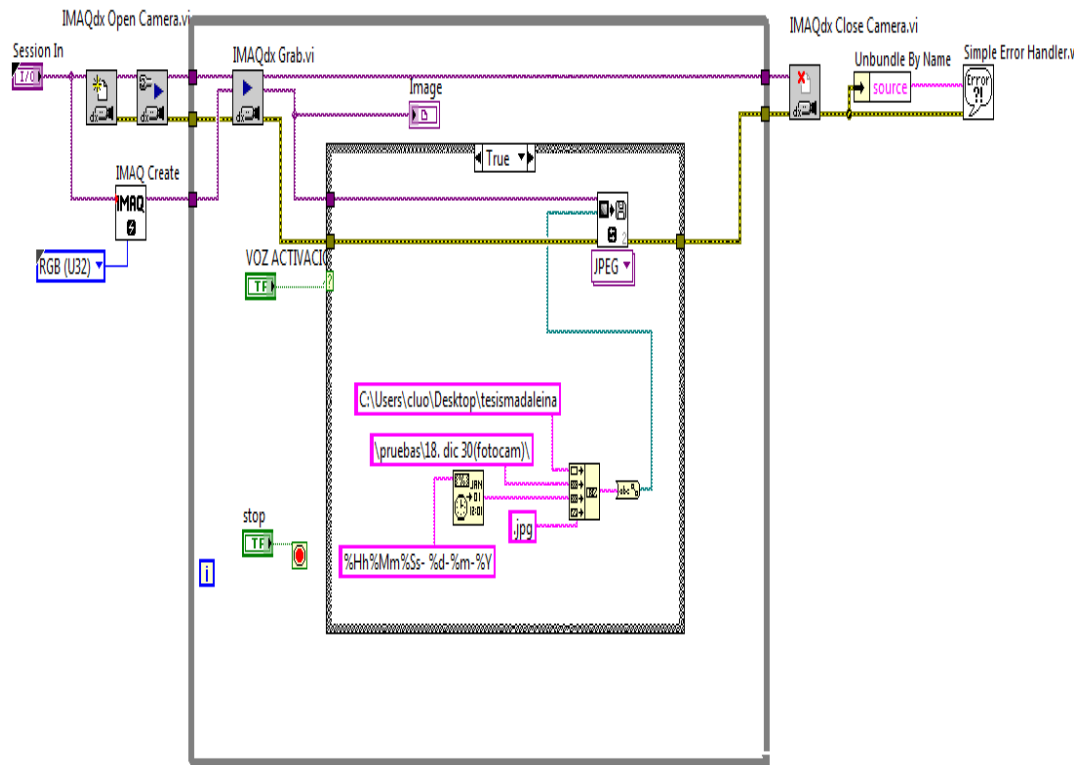


Fig. 5.45 Prueba siete para la captura de imágenes

Los módulos son programados en Arduino uno y para verificar el funcionamiento de los mismos se usó el monitor serial, el dato que enviarán a la Arduino mega es digital por tal motivo se usó un push button con el nombre de VOZ ACTIVACIÓN en la simulación de la figura presentada.

De la misma manera el funcionamiento de los motores fue programado en otro software y a diferencia de lo anterior descrito el microcontrolador PIC16F887 recibirá el dato digital que la Arduino

enviará; para que la Arduino envíe un dato basta con programar un push button a la salida del pin digital y de esa manera se ejerce un control en esta primera etapa de pruebas.

CONTROL DEL RIEGO:

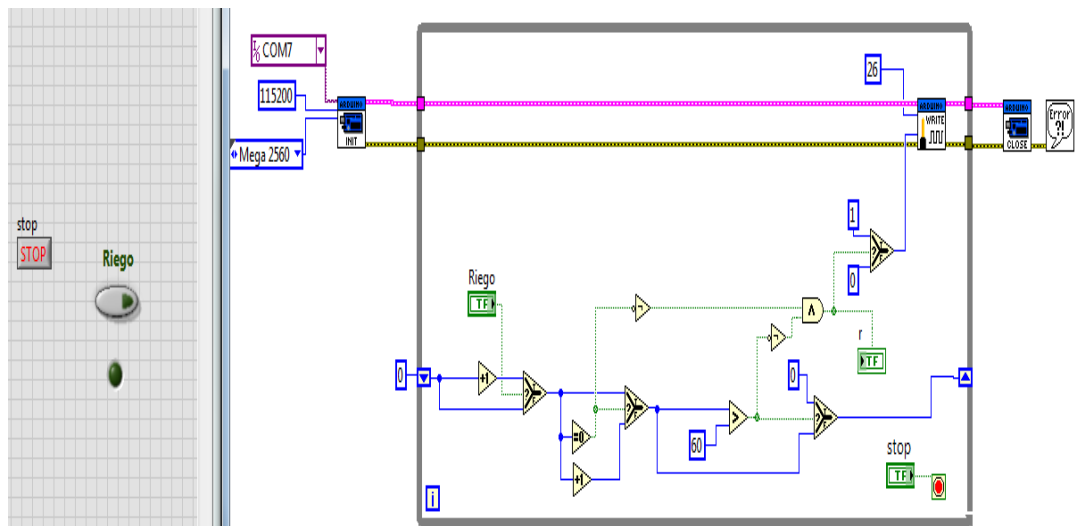


Fig. 5.46 Prueba ocho para el sistema de riego

Se controla el tiempo de duración para el encendido del sistema de riego, programándose para que sea por segundos y con la ayuda del indicador led se verificó el funcionamiento del mismo; ver la Fig. 5.46.

Este control es el más sencillo ya que no presenta integración de otros módulos para el funcionamiento del mismo.

SECCION II

La prueba realizada fue la integración del sistema de luces; es decir, unión de los controles lumínico, dimerización, ambientación y luminarias controladas por push buttons, como se presenta en la Fig.5.47.

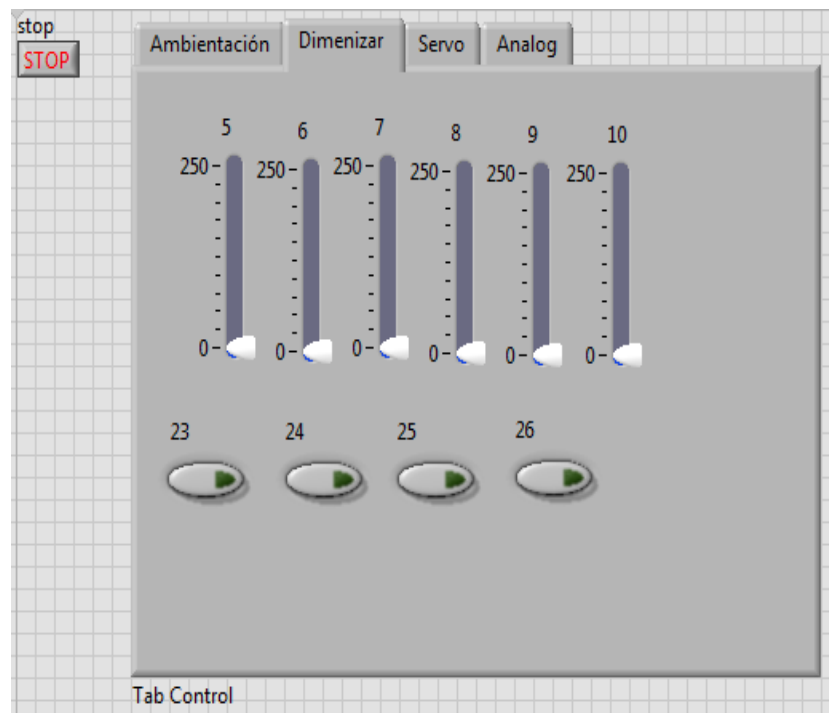


Fig. 5.47 Prueba nueva para la sección dos

Toda la programación realizada para probar los controles para la iluminación hace posible que la gestión del confort esté lista para ser

evaluada en la programación final es decir cuando todas las gestiones hayan sido incluidas, a pesar que también se ha incorporado la gestión de programación y ahorro energético en la cual se controla persianas y garaje como se puede observar en la Fig. 5.48.

No se realizó programaciones adicionales para verificar el funcionamiento de la gestión de la seguridad ya que con las pruebas iniciales realizadas se estima que los módulos funcionarán acorde a lo programado.

SECCION III

Anteriormente se ha verificado el funcionamiento de las programaciones realizadas, pero ahora se englobará todo el sistema dentro de un solo lazo while para determinar si entre las gestiones no hay error de comunicación o tiempo de respuesta agotados, situaciones en la que el programa se volvería ineficiente y adicionalmente hay que considerar la correcta comunicación entre la DAQ y el software.

Ésta sección fue ejecutada una vez que la maqueta ha sido terminada; así como la instalación de todas las conexiones eléctricas,

ubicación de dispositivos y ajustes considerables para poner a prueba la validez del programa.

Los resultados obtenidos de la ejecución del programa se describirán más adelante en los capítulos correspondientes.

En la Fig. 5.49 se presenta la programación completa; cabe recalcar que se lo puede visualizar en los anexos ya que se lo ha imprimido en formato A1.

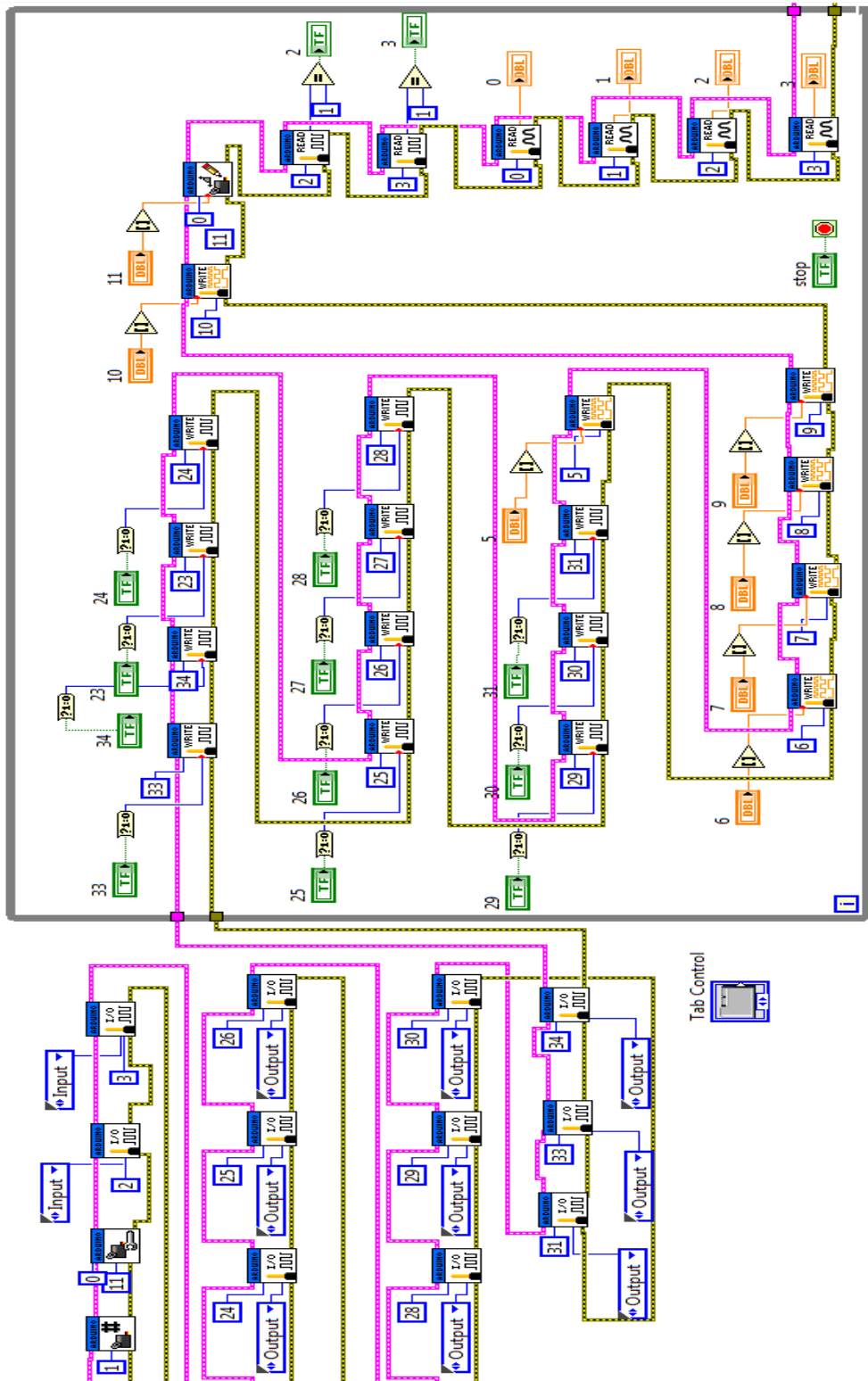


Fig. 5.48 Código para la prueba nueva

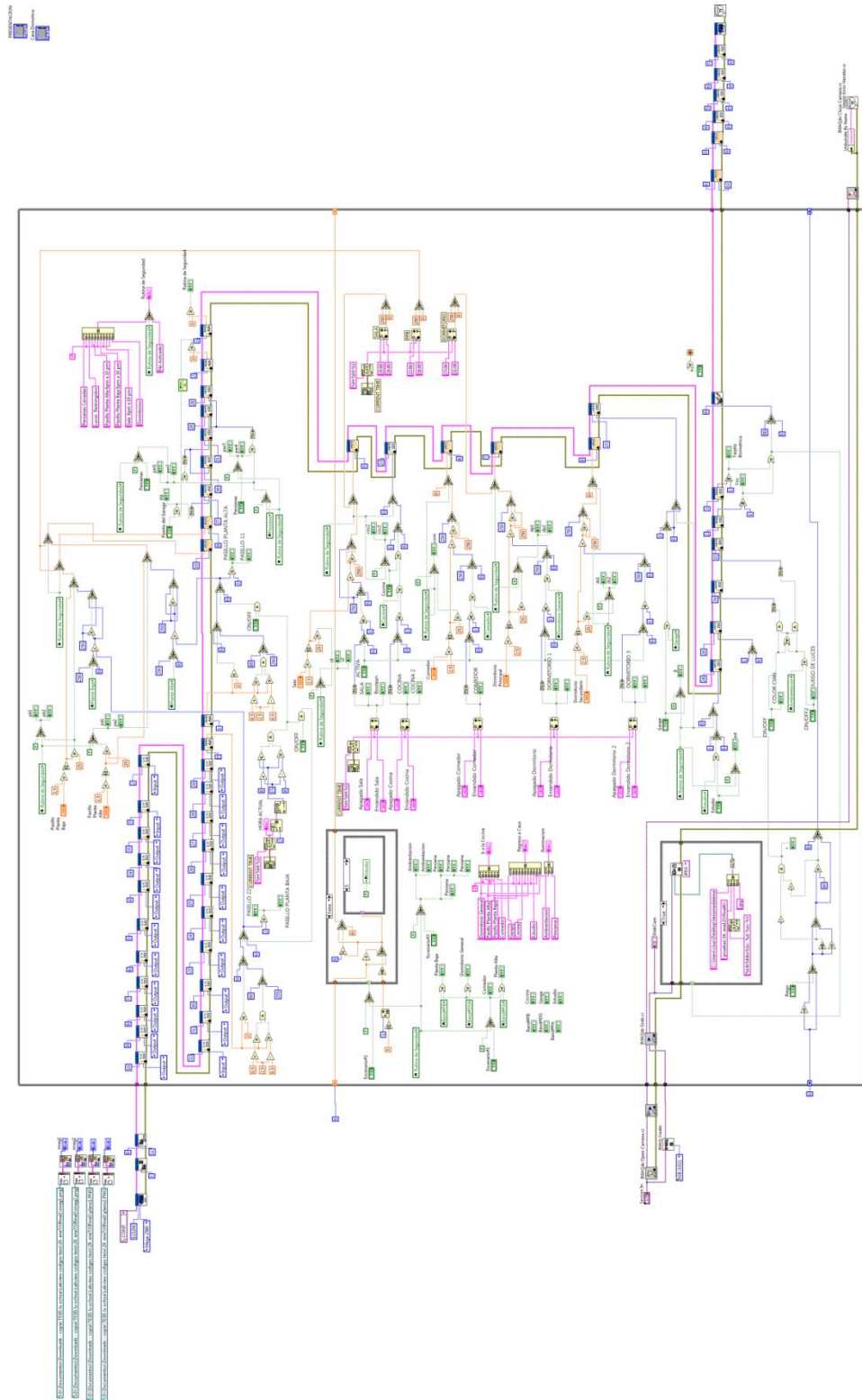


Fig. 5.49 Código de la prueba final

CAPÍTULO 6

MONTAJE Y ACOPLAMIENTO DE LOS DISPOSITIVOS ELECTRÓNICOS

6.1 SOFTWARE Y DISPOSITIVOS ELECTRÓNICOS

Se incorporó en la construcción de la maqueta diversas funciones para cumplir un fin domótico, para ello se utilizó equipos con fácil accesibilidad y amplia gama de información proporcionada por los fabricantes y terceros. A continuación se describirá los componentes que se utilizaron en el proyecto:

Tarjeta de Adquisición de Datos:	Arduino Mega 2560
Sistema de Iluminación y Riego:	Placa Electrónica con Relé
	Placa Electrónica de DIMMER
Sistema de Control Persianas y Garaje:	Microcontroladores - PIC16F887
	ULN2003APG Drivers
	Motor de Paso – 28BYJ-48
Sistema de Seguridad:	Webcam
	Arduino UNO
	EasyVR
	PN532 NFC/RFID
	Servomotor – HS-311
Sistema de Control Remoto por Radio Frecuencia	Módulo para transmisión
	Módulo para recepción
Sistema de Ambientación:	Tira de LEDS – Pololu

Tabla # 7 Módulos implementados en la maqueta

En la página siguiente se presenta un organizador gráfico en el cual se describe los componentes usados por sistema, ver la Fig. 6.1.

Tarjeta de Adquisición de Datos

La tarjeta de adquisición de datos es un procesador de señales eléctricas a tiempo real que permite la interpretación y manipulación de datos con el ordenador.

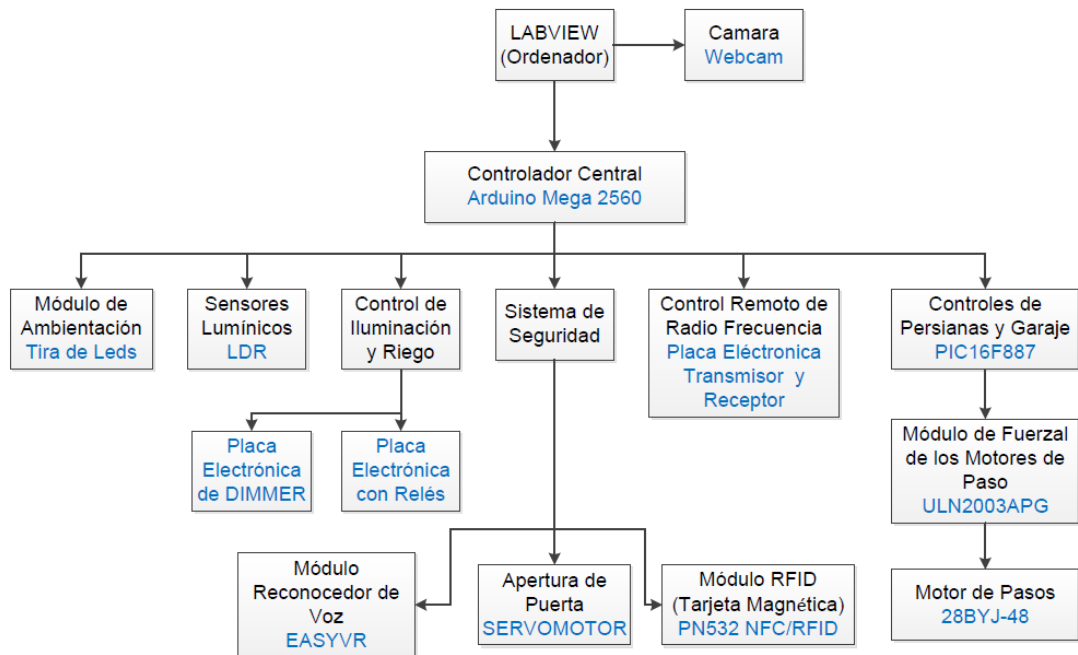


Fig. 6. 1 Organizador gráfico de los controles implementados

Arduino Mega 2560

El Arduino Mega 2560¹⁴ es una placa electrónica basada en el Atmega2560 que permite trabajarla como un módulo de control o como tarjeta de adquisición de datos para el ordenador y el resto de los dispositivos. Posee 54 pines digitales que pueden usarse como entradas o salidas digitales, de los cuales 15 pines son para salidas PWM, adicionalmente presenta 16 pines para entrada analógica, 4 puertas seriales UART, un oscilador o reloj interno de 16 MHz (Mega

¹⁴ Véase Anexo 6 para características y esquemático del Arduino Mega 2560

Hertz), cabecera ICSP, un puerto USB y alimentación externa, incluyendo el integrado ATmega8U2 para la programación USB-Serial. [17]

Comunicación entre Labview y Arduino Mega 2560

El Arduino se conecta al computador usando el protocolo de comunicación serial UART (Transmisor y Receptor Asincrónico Universal), y se logra establecer una comunicación entre el ordenador y el módulo de manera básica, efectiva y segura.

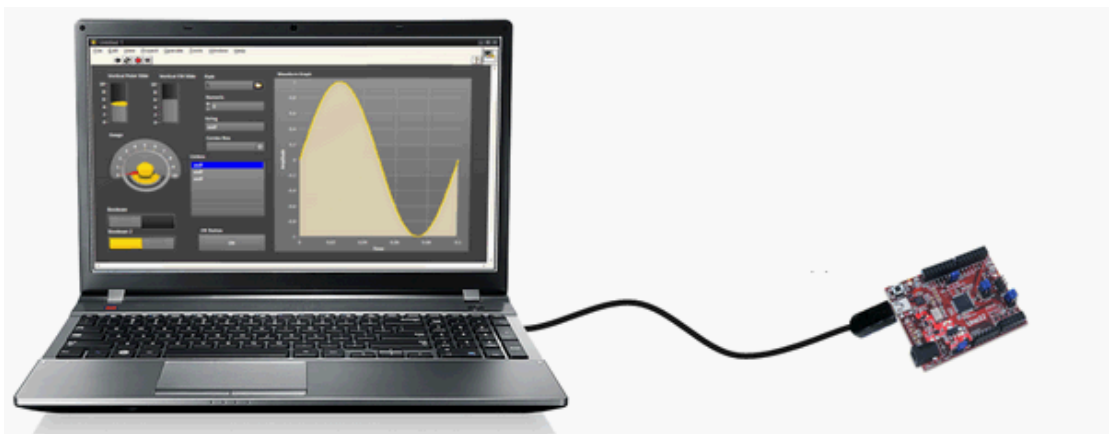


Fig. 6.2 Comunicación entre pc y tarjeta

El UART es un protocolo de comunicación Serial que permite la transmisión de datos a través de cables físicos TX y RX que significan

transmisor y receptor respectivamente, bajo una trama de bits. La trama contiene 10 bits que operan de manera lógica, entiéndase que los bits lógicos son 0 y 1. La información se mantiene en 1 hasta recibir el bit de inicio y permitir el envío de información; además, el protocolo contiene un bit de inicio, que es el 0; luego envía 8 bits para la trama de información y se finaliza con un bit de parada, el cual es el 1, cerrando la información. [8]

Los datos son procesados por el Integrado ATmega8U2 que se encuentra en el Arduino Mega 2560 para que las señales sean convertidas de TX y RX a Data+ y Data- respectivamente; permitiendo establecer comunicación usando la interface USB. [9]

Los pines Data+ y Data- pertenecen a la interface USB.

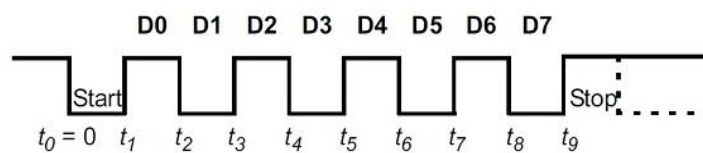


Fig. 6.3 Modelo de la trama de bits

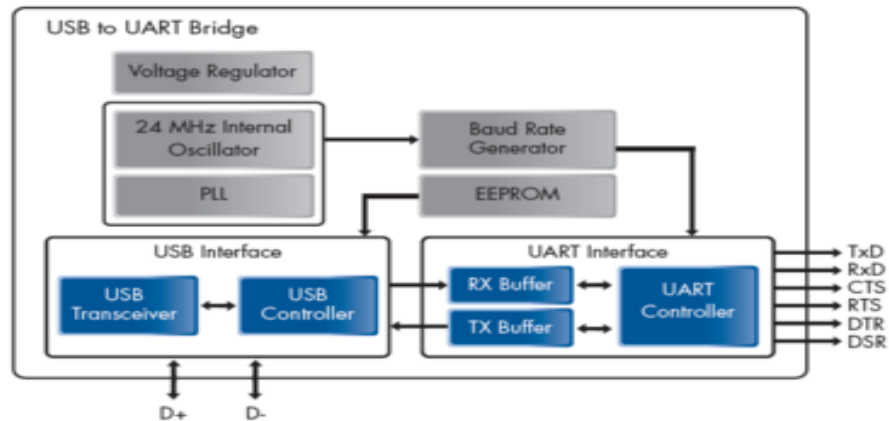


Fig. 6.4 Esquema de puente UART a USB¹⁵

Se presenta a continuación la programación del Arduino Mega 2560¹⁶ para habilitar la comunicación Serial con el Labview.

```
#define FIRMWARE_MAJOR 02
#define FIRMWARE_MINOR 00 // Preparación de los registros del Arduino para
// permitir la comunicación serial
#if defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__)
//Definición del ATmega2560 para el LIFA BASE (Interface de Labview )
#define DEFAULTBAUDRATE 9600 //Velocidad de bits por segundo natural
#else
#define DEFAULTBAUDRATE 115200 // Velocidad de bits por Segundo
#endif
#define MODE_DEFAULT 0
// Modulo Normal de los comandos de Labview si no hay comandos se coloca el
comando nulo
#define COMMANDLENGTH 15 // Paquete de 15 bytes de datos
```

¹⁵ Véase la bibliografía en el apartado [10]

¹⁶ Véase la bibliografía en el Anexo 7

Para que se establezca la sincronización entre Labview y Arduino es necesario definir el tipo de Arduino que se conectará, la velocidad de bits por segundo que es 9600 bps y la cantidad de 15 bytes de información para la correcta comunicación entre ambos.

En primera instancia se realizaron pruebas de lectura en el puerto de transmisión y de recepción del Arduino, en primera instancia solo se colocó la punta del osciloscopio en el pin 1 TX sin enviar dato alguno y se obtuvo el resultado que se presenta en la Fig. 6.5.

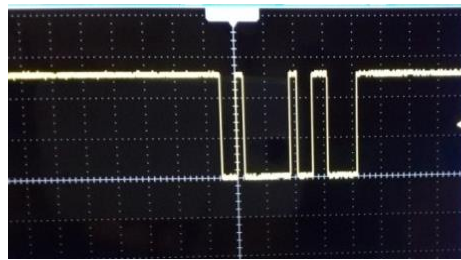


Fig. 6.5 Pin1 Tx

Se realizó el mismo procedimiento para la recepción, es decir se situó la punta del osciloscopio en el pin 0 RX de la Arduino, obteniendo el resultado como se observa en la Fig. 6.6.

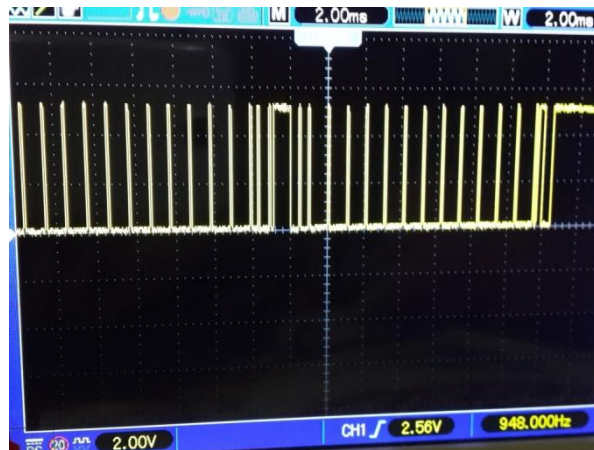


Fig. 6.6 Pin 0 RX

También se realizó muestreos en los pines Data- y Data+, como se muestran en las Fig. 6.7 y 6.8 respectivamente.

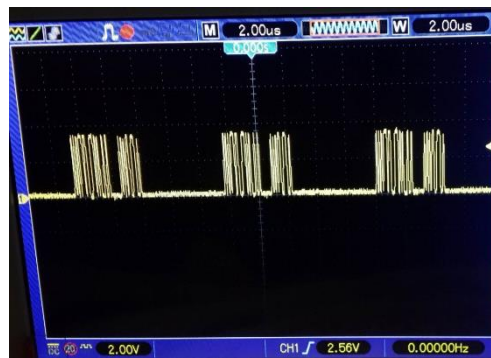


Fig. 6.7 Data -

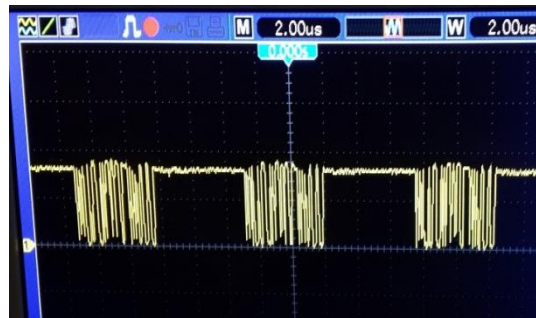


Fig. 6.8 Data +

Diagrama de Conexiones de la Comunicación Serial

En la Fig. 6.9 se puede apreciar los componentes del Arduino Mega 2560 que son utilizados para realizar la comunicación con la interfaz USB y permitir la comunicación serial.

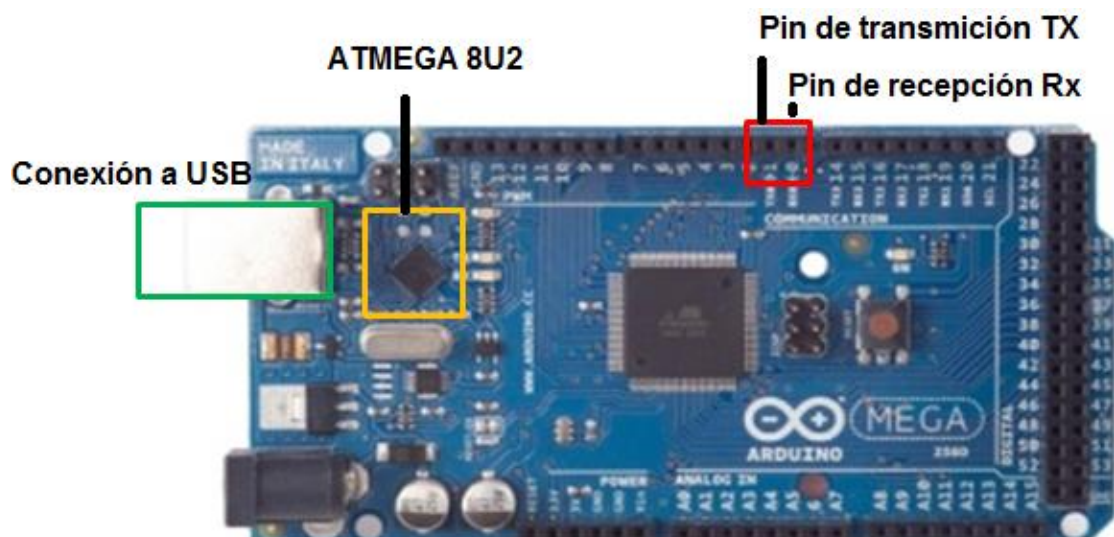


Fig. 6.9 Arduino Mega 2560

Puerto de comunicación:

	Pin	Puerto	Detalle
TX	1	Comunicación	Pin de transmisión de Datos
RX	0	Comunicación	Pin de recepción de Datos

Tabla # 8 Descripción de puertos

Integrado ATMEGA8U2: Integrado que permite la conversión Lógico de los pines TX y RX a Data+ y Data-. [11]

Puerto USB JACK: Conecta el cable que permite las conexiones de ambas partes.

Nombre	Color	Detalle
5V	Rojo	Cable de Alimentación con 5 V
Data +	Blanco	Cable de dato con diferencial positivo
Data -	Verde	Cable de dato con diferencial negativo
GND	Negro	Cable de punto común

Tabla # 9 Descripción del cable

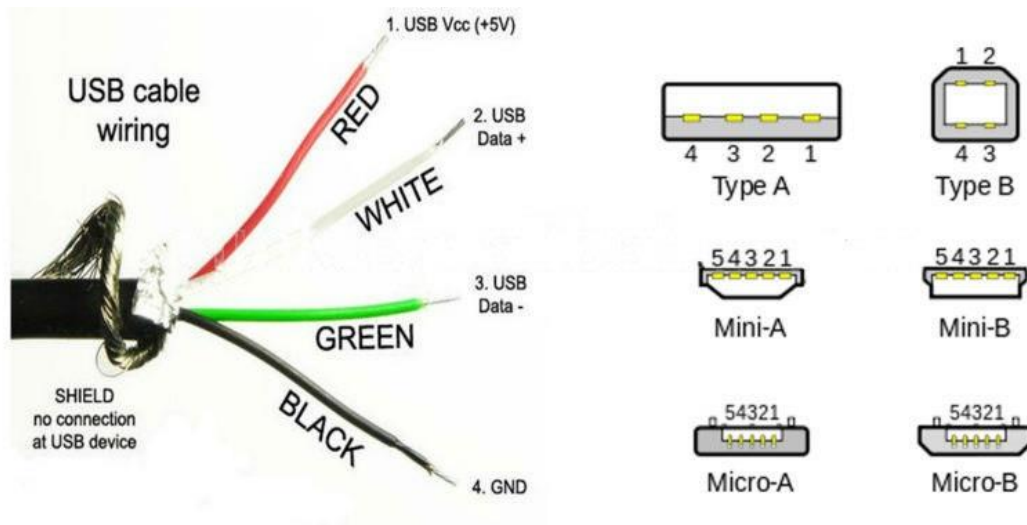


Fig. 6.10 Descripción de los pines de la interfaz USB

Una vez realizada la conexión entre los puertos de la tarjeta Arduino y del computador, es necesario revisar el administrador de dispositivos; véase la Fig. 6.11 ya que se instala y se prepara un puerto COM para la comunicación con el Arduino, ésta información es necesaria para realizar las programaciones en el bloque de inicialización en LABVIEW. Labview puede reconocer y escribir la información de 8 bits y decodificarlas a código de letras ASCII, usando el VISA (Arquitectura de Instrumento Visual) que enlaza el puerto COM con el Labview.

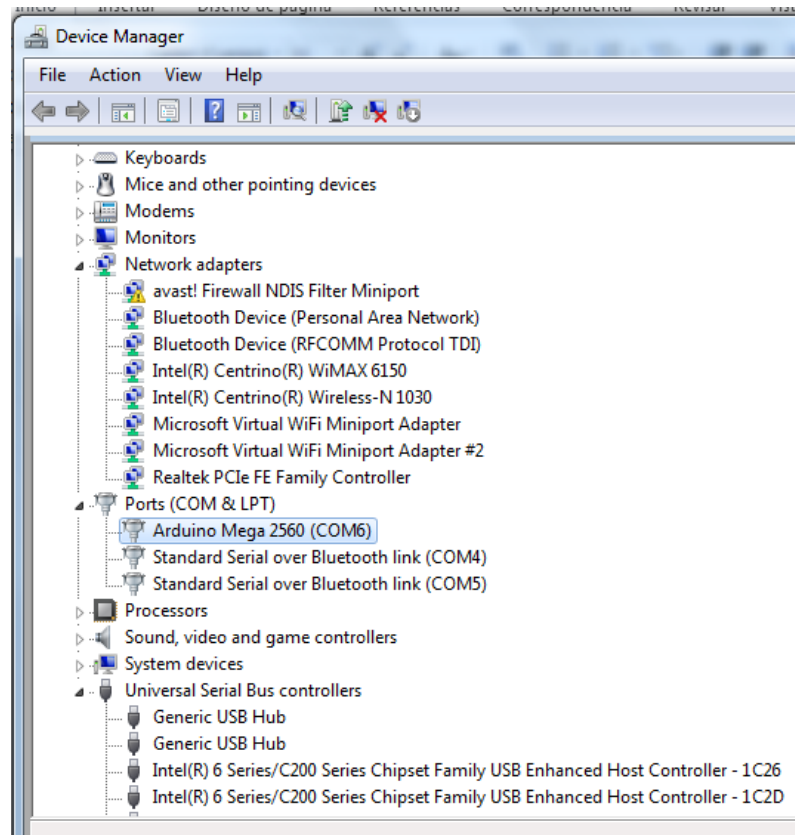


Fig. 6.11 Administrador de dispositivos

Programación en Labview para la comunicación con el Arduino:

Para poder realizar la programación en Labview, es necesario haber instalado los toolkit para *Arduino-Interface*; ya que contiene el sketch para que la tarjeta Arduino se comuniquen con el software LABVIEW y poder ejercer control sobre la tarjeta.

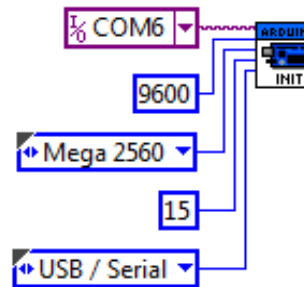


Fig. 6.12 Bloque Init- Labview

Dentro de la programación en LABVIEW se utiliza el toolkit para Arduino; en la Fig. 6.12 se muestra el Bloque Init, que inicia y procede a establecer la transmisión y recepción de datos con la tarjeta; para ello se necesita detallar el modo de comunicación; es decir, el puerto COM que fue reconocido por el ordenador como se visualiza en la Fig. 6.11, la velocidad de transmisión que fue establecido a 9600 bps, el tipo de Arduino que se va a utilizar en este caso es el Mega 2560 y la cantidad de paquetes que son 15 datos con 8 bits integrados para la transmisión de información dada.

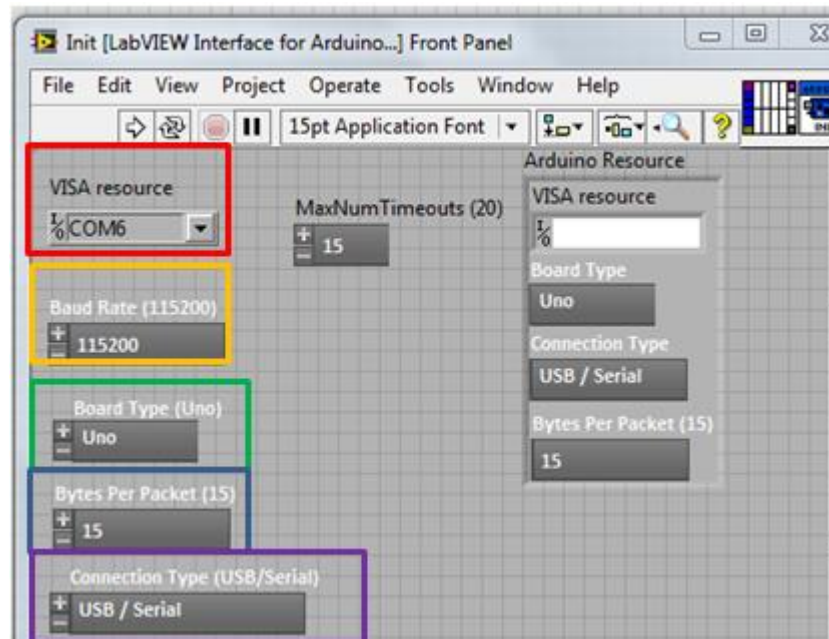


Fig. 6.13 Configuración del bloque Init

Visa Resource	Selección del puerto
Baud Rate	Velocidad de Transmisión
Board Type	Selección del Arduino que se está usando
Bytes Per Packet	Transmisión de 15 bytes para cada paquete
Connection Type	Tipo de Cable para la conexión

Tabla # 10 Descripción de los bloques de configuración

En la Fig. 6.14 se muestra la programación del bloque Arduino Init, el programa lee los valores iniciales configurados como se visualiza en la Fig. 6.13 y se prepara para iniciar la comunicación, verifica si el Arduino está conectado; en el caso de haber respuesta de

comunicación se procede a mantenerla; caso contrario se cancela y se envía un anuncio de error.

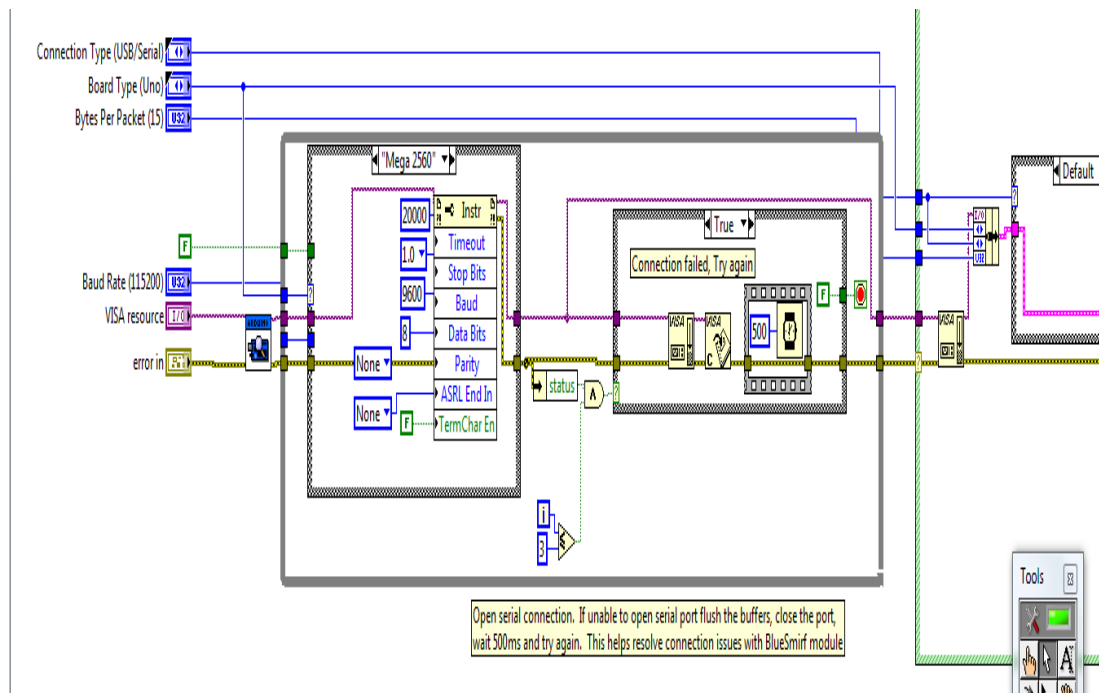


Fig. 6.14 Programación interna del bloque Init

Sistema de Iluminación y Riego

Para nuestro sistema de iluminación y riego se utilizó una placa electrónica con Relé con la cual es posible realizar las demostraciones para el control del encendido y apagado de luminarias y al mismo tiempo se simula el accionamiento de una electroválvula para el sistema de riego.

La placa electrónica DIMMER controla la intensidad lumínica que se ha aplicado a ciertas luminarias. En la siguiente tabla se describe la ubicación en la maqueta y la cantidad de LEDS que han sido dimerizados y aquellos que tienen un control manual.

Lugar	Cantidad	Tipo
Dormitorio Principal	2	Es dimerizable
Dormitorio Secundario	2	Es dimerizable
Pasillo Planta Alta	2	Es dimerizable
Pasillo Planta Baja	2	Es dimerizable
Sala	2	Es dimerizable
Comedor	1	Es dimerizable
Estudio	2	ON/OFF
Garaje	1	ON/OFF
Cocina	2	ON/OFF
Riego	1	ON/OFF

Tabla # 11 Espacios en la maqueta con el sistema de iluminación

En la Fig. 6.15 se visualiza la placa electrónica con Relés; la misma que ha sido energizada por transistores ya que operan como

interruptores y a la vez la placa es controlada por la Arduino Mega 2560.

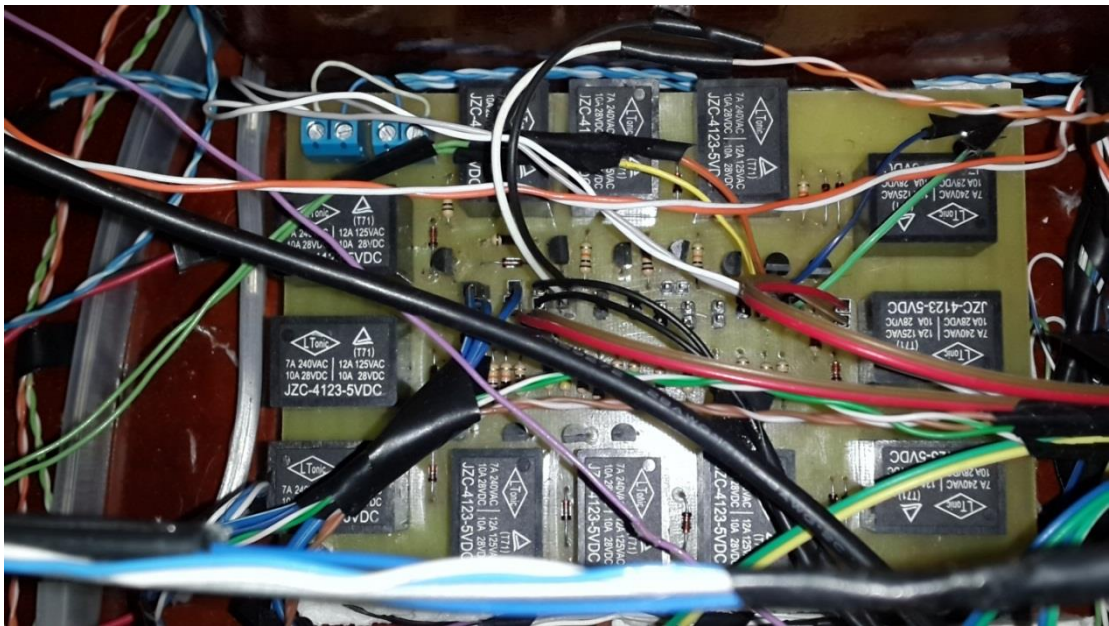


Fig. 6.15 Placa con Relays

La placa electrónica DIMMER permite controlar la intensidad de los LEDS usando transistores bipolares (Abierto y Cerrado) 2N2222¹⁷ que permite el control de voltaje con la técnica PWM (Modulación de Ancho de Pulso) , y permite el control de accionamientos por pulsos a los transistores. La tarjeta Arduino Mega 2560 posee, se ha programado a operar a 1000 Hertz y LABVIEW controla al Arduino

¹⁷ Visualícese al Anexo 8 Datasheet 2N2222

ya que los datos son convertidos de 0 - 255 valores enteros a 0 - 100% de PWM, se la puede apreciar en la Fig. 6.16.

En la Fig. 5.18 y 5.20 se pueden observar los indicadores LEDS y las placas que han sido mencionadas ejecutarán el control programado.

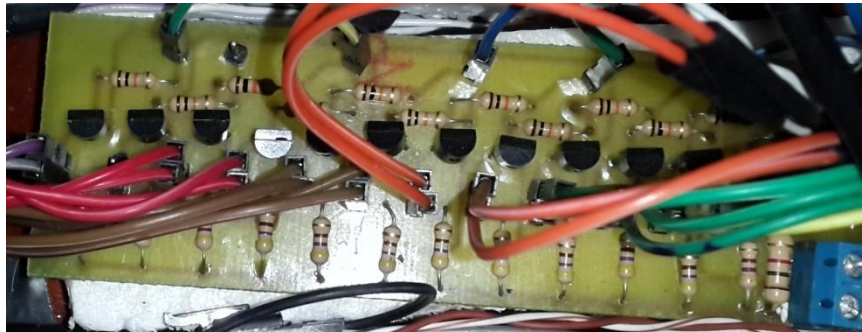


Fig. 6.16 Placa electrónica Dimmer

Sistema de Control de Persianas y Garaje

Para el control de persianas y garaje se implementó motores de paso que permitirán girar y enrollar la tela con bambú, permitiendo visualizar el control para abrir y cerrar persianas y la puerta del garaje. El Arduino Mega 2560 se comunica con los micro controladores PIC16F887 ya que son usados como módulo de control

de giro para los motores y el ULN2003APG se usa como integrado de fuerza.

Control de los motores de paso

El microcontrolador PIC16F887¹⁸ (Controlador de Interfaz Periférico), es un circuito integrado con memoria, procesador y pines programables con 35 pines de entrada y salida, de los cuales algunos pines fueron programados¹⁹ para controlar los motores de paso, y generar pulsos de manera secuencial que permitirán el giro para cada motor. Los motores de paso 28BYJ-48²⁰ de 5 voltios poseen ocho pasos para realizar el giro adecuado y se debe repetir 512 veces los 8 pasos para dar una vuelta completa. Los motores fueron programados para que realicen 2 giros cuando las persianas y la puerta del garaje se estén abriendo.

¹⁸ Obsérvese en el Anexo 9 Características del PIC16F887

¹⁹ Visualice página 75 Programación en C de los microcontroladores

²⁰ Revise en el Anexo 10 Características del Motor de Paso

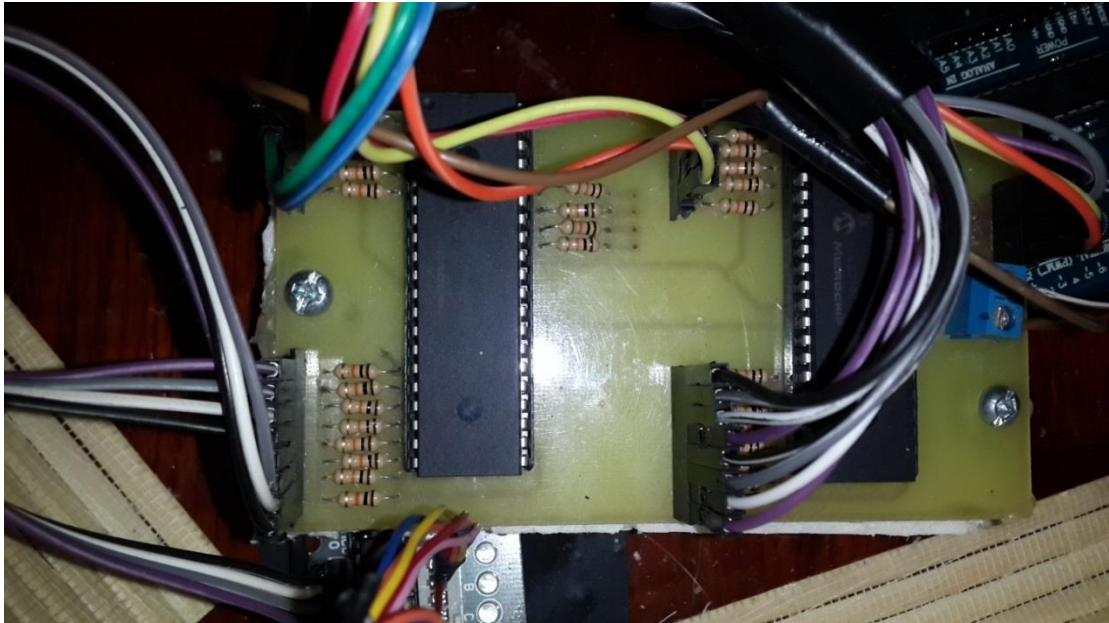


Fig. 6.17 Placa con los pics PIC16F887

Los microcontroladores PIC16F887 se comunican de manera lógico al Mega Arduino 2560 para iniciar el movimiento de los motores de paso y generar la secuencia que se requiere para girar a los mismos. Se utilizan cuatro puertos del microcontrolador de ocho pines; es decir, se usan los puertos A,B,C, y D que fueron programados como puertos digitales de entrada, el puerto A se comunica a la tarjeta Mega Arduino 2560, mientras que el resto de los puertos se conectan con los integrados de fuerza para cada motor.

El ULN2003APG²¹ es un conductor Darlington que permite la alimentación de cargas, se comunica con los 5 pines del motor de paso y opera de manera lógica. El motor de paso presenta 5 cables cuyos colores son azul, rosado, amarillo, naranja y rojo; el cable Rojo es para la alimentación y se conectan a la salida del conductor Darlington; los demás cables a excepción de la alimentación puede estar conectados a tierra o permanecer sin conexión; se puede apreciar la Fig. 6.18 y 6.19.



Fig. 6.18 ULN2003APG Driver

En la siguiente imagen se puede visualizar que el cable rojo es el cable de poder y el resto son para control.

²¹ Véase en el Anexo 11 Características del ULN2003APG



Fig. 6.19 Motor de pasos 28BYJ-48

Sistema de Seguridad

Se implementó estos dispositivos simulado un video portero, se realiza una breve descripción en la tabla # 12 que se presenta en la página siguiente:

Dispositivos	Nombre	Descripción
Cámara	WebCam ZEN	Cámara que permite monitorear la entrada de la puerta de la maqueta y tomar fotografías
Módulo RFID	PN532	Módulo que permite el censado de la tarjeta magnética
Módulo Reconocedor de Voz	EasyVR	Módulo que permite la identificación de las voces grabadas
Servo Motor	HS-311	Motor que permite el control de la apertura de la puerta

Tabla # 12 Elementos que simulan un videoportero

El Arduino UNO²² es una placa con un microcontrolador ATmega 328 desarrollado por Atmel AVR, y un entorno desarrollado para múltiples usos electrónicos. La placa Arduino posee 14 pines de entrada y salida, 6 de esos pines son salidas PWM y 4 pines la comunicación SPI, 6 pines de entrada analógica y 2 pines para la comunicación serial UART con el integrado ATmega8U2 para el interfaz USB. La placa Arduino UNO se implementó con un módulo de control con el módulo reconocedor EasyVR, PN532 NFC/RFID y la tira de LEDs para el control de ambientación de la maqueta.

²² Observe el Anexo 12 Características del Arduino UNO

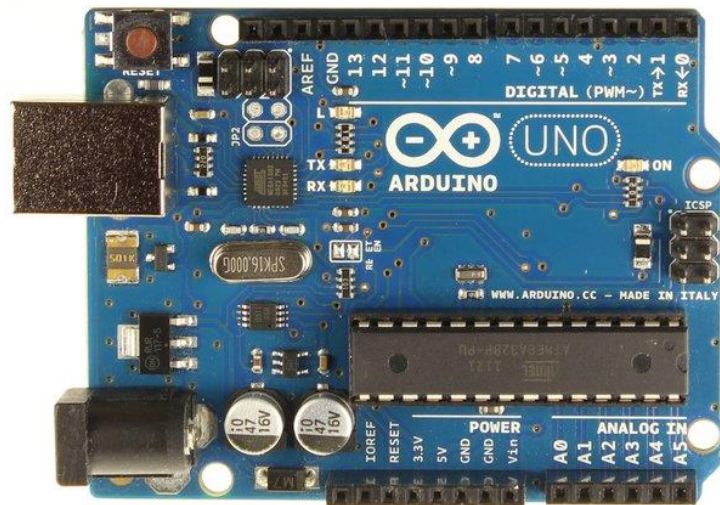


Fig. 6. 20 Placa Arduino UNO

Se incorporó una cámara webcam la cual se comunica con el Ordenador y usando el programa LABVIEW se monitoreará y se capturará fotos que serán almacenadas en la memoria del ordenador.

El Módulo PN532 NFC / RFID Shield²³ (Comunicación de Campo Cercano / Identificador de Radio Frecuencia) puede implementarse en el Arduino, desarrollado por ELECFREAKS que permite la detección de varias tarjetas magnéticas e identificarlas por sus propios ID (Identidad de la tarjeta), el Arduino se comunica con el módulo para identificar el censado de la tarjeta y realizar la apertura de la misma.

²³ Visualice Anexo 13 DataSheet PN532 NFC/RFID

Comunicación de la Tarjeta magnética

El NFC PN532 es un módulo RFID que permite censar la proximidad de la tarjeta magnética y reconocerla mediante su ID. El controlador que permite programar a la tarjeta es el Arduino UNO y se puede conectar los pines digitales de salida del módulo a la Arduino Mega 2560; al realizar esta acción se crea un puente con la finalidad de establecer comunicación y control. [12]

El SPI (Bus de interface serial periférico) permite la comunicación del Arduino UNO como Maestro y NFC PN532 como Esclavo. [13]

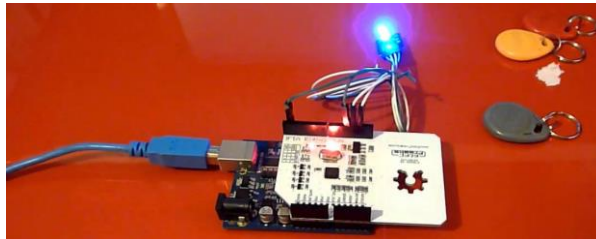


Fig. 6.21 Arduino UNO y NFC PN532

El SPI es un bus de comunicación serial Síncrono para distancias cortas, cuyo modo de comunicación es full dúplex usando la

arquitectura maestro – esclavo con un único Maestro y varios Esclavos.

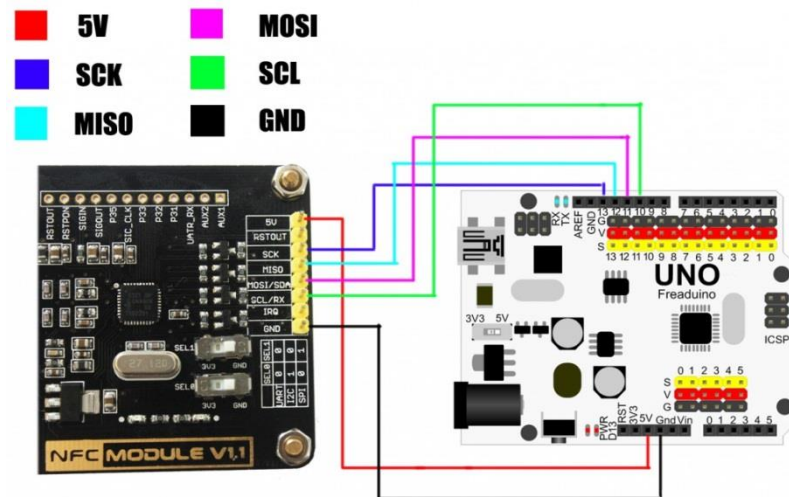


Fig. 6. 22 Conexión entre la Arduino y la tarjeta

Entiéndase que la comunicación full dúplex permite la transmisión de datos en ambas direcciones usando el mismo canal. Algunas veces se lo llama al SPI bus serial de 4 cables, y las funciones de los cables para la comunicación son:

Nombre	Simbología	Detalle
SCK	Clock	Reloj Sincronizador
MISO	Master In Slave Out	Información del Esclavo al Maestro
MOSI	Master Out Slave In	Información Maestro a Esclavo
SS	Slave Select	Activador del Esclavo Asignado

Tabla # 13 Funciones del SPI

Para realizar la comunicación SPI los pines de la Arduino están en 1 es decir en alto, el SS o selector de esclavo envía un 0 para entablar una conexión Maestro - Esclavo y se sincroniza el reloj SCK para enviar la trama de datos. El MOSI envía la trama para que el Maestro le pregunte al Esclavo y luego el MISO responde cuando finaliza la pregunta, en la Fig. 6.23 que se presenta a continuación se puede apreciar con mayor claridad la comunicación entre dispositivos. [13]

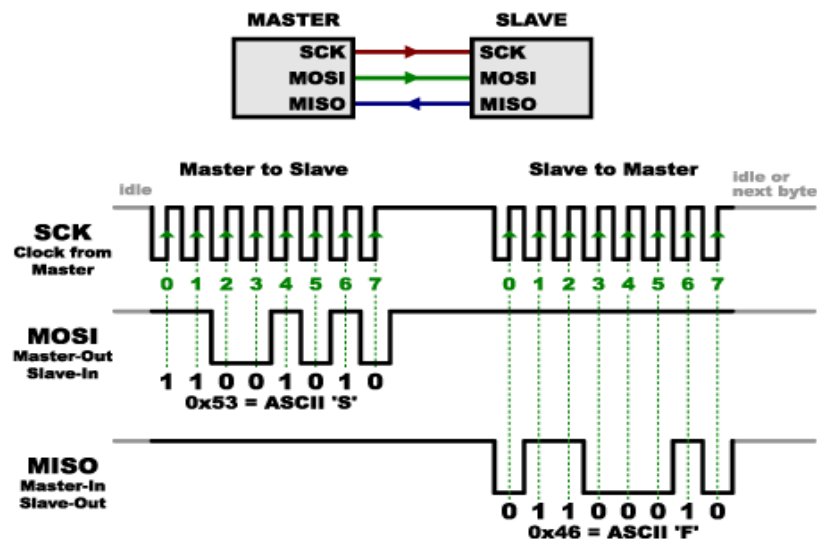


Fig. 6.23 Comunicación serial SPI

Introducción de la programación para el PN532 en el archivo .cpp²⁴

```
#include <Arduino.h>
```

²⁴ Obsérvese la página 92 Programación Completa del Arduino UNO - PN532

```

#include "PN532.h"
byte pn532ack[] = {0x00, 0x00, 0xFF, 0x00, 0xFF, 0x00};
byte pn532response_firmwarevers[] = {0x00, 0xFF, 0x06, 0xFA, 0xD5, 0x03};
#define PN532_PACKBUFFSIZ 64
byte pn532_packetbuffer[PN532_PACKBUFFSIZ];
PN532::PN532(uint8_t clk, uint8_t miso, uint8_t mosi, uint8_t ss) {
    _clk = clk;
    _miso = miso;
    _mosi = mosi;
    _ss = ss;
    pinMode(_ss, OUTPUT);
    pinMode(_clk, OUTPUT);
    pinMode(_mosi, OUTPUT);
    pinMode(_miso, INPUT);
}
...

/***** low level SPI */
void PN532::spiwrite(uint8_t c) {
    int8_t i;
    digitalWrite(_clk, HIGH);
    for (i=0; i<8; i++) {
        digitalWrite(_clk, LOW);
        if (c & _BV(i)) {
            digitalWrite(_mosi, HIGH);
        } else {
            digitalWrite(_mosi, LOW);
        }
        digitalWrite(_clk, HIGH);
    }
}

```

```

uint8_t PN532::spiread(void) {
    int8_t i, x;
    x = 0;
    digitalWrite(_clk, HIGH);
    for (i=0; i<8; i++) {
        if (digitalRead(_miso)) {
            x |= _BV(i);
        }
        digitalWrite(_clk, LOW);
        digitalWrite(_clk, HIGH);
    }
    return x;
}

```

En la programación que se le realiza al Arduino se incluye las librerías *Arduino.h* y *PN532.h* que son dadas por el fabricante, los paquetes de datos *pn532ack*, *pn532response_firmwarevers* y *pn532_packetbuffer*, son establecidos para que se efectúe la comunicación SPI al *PN532* definiendo el (*clk*, *miso*, *mosi*, *ss*) que ha sido mencionado en los párrafos anteriores.

El MISO y MOSI van en una trama de 8 bits, usando el CLK como reloj sincronizador y los datos son recibidos cuando hay un flanco positivo del CLK como se visualiza en la Fig. 6.23.

El Módulo EasyVR²⁵ pertenece a la marca TIGAL VEEAR, la cual elabora varios productos que son implementados para efectuar el reconocimiento de voz y comandos por voz. Posee una adaptación con la plataforma Arduino que es EasyVR Arduino Shield 2.0 TIGAL y permite al módulo operar con un sensor para recibir e identificar la palabra.

Comunicación entre el Arduino y el EASY VR

La comunicación con EASYVR y el Arduino se realiza bajo una comunicación Serial –UART, donde interactúan los pines TX y RX para la comunicación entre ellos. Para manipular el EASYVR se usa librerías y ejemplos dados por los programadores de la tarjeta. [14]

Es posible que el ARDUINO y el EASYVR interactúen entre sí estableciendo comunicación con el ordenador; cuya finalidad es el monitoreo del módulo y realizar cambios en la programación.

²⁵ Visualice Anexo 14 Propiedades del EasyVR

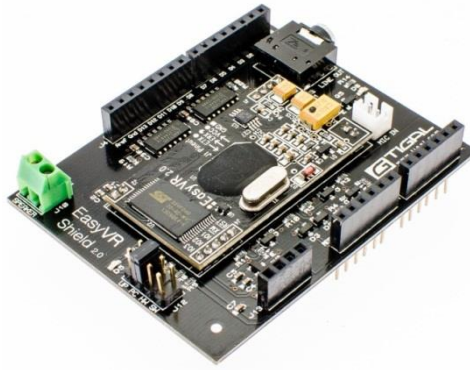


Fig. 6.24 Módulo EASYVR

La Fig. 6.25 describe la trama de los Bits de la comunicación Serial UART. La comunicación inicia con 1 y se espera el bit 0 para iniciar la transmisión en la cual se reciben los 8 bits de código ASCII; se finaliza con el bit de Parada en 1 y se regresa al estado inicial 1.

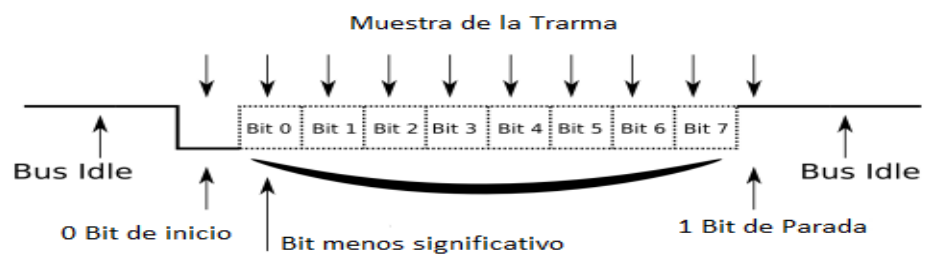


Fig. 6. 25 Trama de bits

Declaración de librerías e introducción a la programación del Arduino UNO²⁶ con el EasyVR.

```

#if defined(ARDUINO) && ARDUINO >= 100 //Detección de la versión del Arduino
#include "Arduino.h" //Uso de las librerías del Arduino
#include "SoftwareSerial.h" //Uso conexión con pines seriales
SoftwareSerial port(12,13); // Declaración de los pines para la comunicación serial

#else // Arduino 0022 - use modified NewSoftSerial
// si no se detecta el Arduino apropiado se busca otra librería de Arduino
#include "WProgram.h"
#include "NewSoftSerial.h"
NewSoftSerial port(12,13);
#endif
    #include "EasyVR.h" //Uso de las librerías EasyVR
if (bridge.check())
{
    cli();
    bridge.loop(0, 1, 12, 13); Selección de los pines para la comunicación de los
módulos
}
Serial.begin(9600); //Setea de la velocidad de transmisión
port.begin(9600);
digitalWrite(led,LOW);

```

Esquema de comunicación entre módulos.

Dada la declaración de comunicación, el Arduino utiliza los pines 12 y 13 para la transmisión y recepción de datos con el EasyVR. Los pines

²⁶ Véase página 86 Programación Completa del Arduino UNO – EasyVR

0 y 1 permiten realizar una comunicación serial con el ordenador y permite un monitoreo de ello.

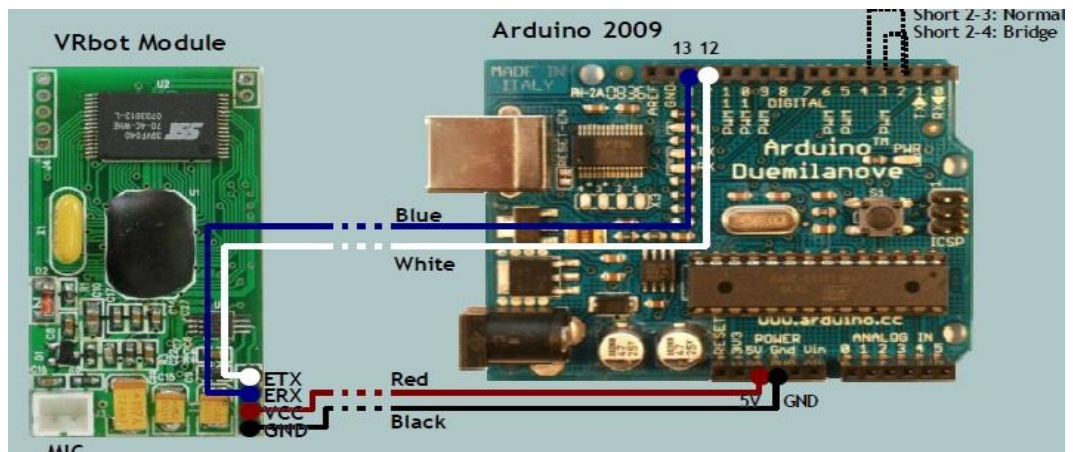


Fig. 6. 26 Diagrama de conexión

El Control de Apertura de la Puerta

El servomotor es un actuador que permiten controlar de manera precisa para mantener la posición del eje en un ángulo determinado.

Se implementó un servomotor para simular un sistema de apertura en la puerta principal de la casa, se incorpora un brazo en el eje del motor para proporcionar el movimiento de la puerta. El servomotor Hitec HS-

311²⁷, posee 3 pines de color negro que se conectan a un punto en común, el rojo se conecta a la fuente de alimentación de 5 voltios y el cable amarillo es para el control, el Arduino Mega 2560 controla el servomotor bajo señal PWM de 50 Hertz o 20 milisegundos, y el servomotor opera de 1 a 2 milisegundos de ancho de pulso posicionando el eje en un ángulo de 0 a 180 grados respectivamente.

La Fig. 6.27 describe el ancho de pulso con la posición del ángulo.

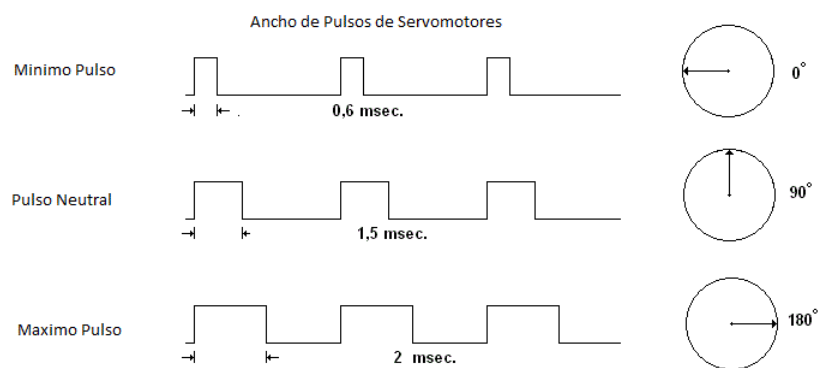


Fig. 6. 27 Ancho de pulso y ángulo de giro del motor

Sistema de Control Remoto de Radio Frecuencia

El control remoto se diseñó para controlar de forma manual la activación de la apertura de la puerta del Garaje, y la activación de la

²⁷ Véase Anexo 15 Descripción de Servomotor

Rutina de Seguridad, simulando un circuito electrónico de control externo que permita ejecutar las acciones como: estacionar el auto del dueño del hogar o salidas de viajes familiares.

Se implementó 2 placas para el sistema de control remoto, una es para el transmisor y otra para el receptor²⁸.

La placa electrónica de transmisión posee un módulo de radio frecuencia transmisor que opera a modulación ASK (Modulación por Desplazamiento de Amplitud) enviando una frecuencia de 433 Mega Hertz si recibe un 1 lógico, caso contrario no generará ninguna señal y es tratado como un 0 lógico. El módulo posee 3 pines los cuales están conectados a tierra o GND, punto para la alimentación y el Data que comunica con el codificador HT12E, que es controlado de manera lógico.

El integrado HT12E²⁹ es un codificador de 8 bits para dirección y 4 bits para datos formando la trama que será transmitida hacia el receptor, los 4 bits son operados de manera manual por el usuario a través de pulsadores.

²⁸ Revisar Anexo 16 Módulos de Radio Frecuencia

²⁹ Observe Anexo 17 Datasheet del Integrado HT12E

El módulo receptor recibe la señal de frecuencia y lo transforma en una trama de datos usando el integrado decodificador HT12D³⁰ obteniendo la señal de los 12 bits, 8 bits de la dirección del decodificador asignado y 4 bits de datos mencionado anteriormente y enviadas al Arduino Mega 2560.

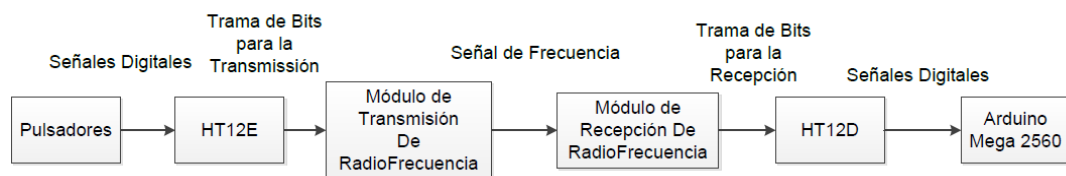


Fig. 6.28 Modo de envío de la trama para la comunicación



Fig. 6.29 Módulo de Transmisión

³⁰ Revise Anexo 18 Datasheet del Integrado HT12D

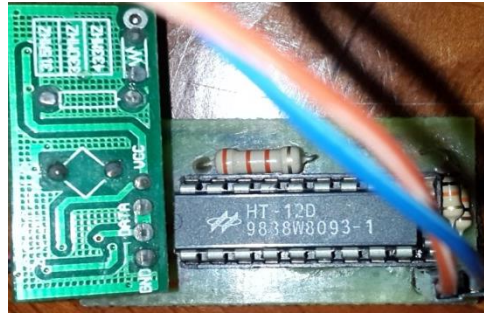


Fig. 6.30 Módulo de recepción

Sistema de Sensor Lumínico

El LDR es un fotoresistor es decir varia su resistencia en función de la intensidad lumínica, cuando hay incidencia de luz el valor de resistencia puede oscilar entre los 50 a 1000 ohmios y de 50k a varios mega ohmios cuando está a oscuras.

Sistema de Ambientación

La tira de LEDS de la marca POLOLU posee el integrado WS2812B³¹ para el control de cada LED, se utilizó el módulo Arduino UNO para controlar la tira y generar los colores deseados.

³¹ Véase Anexo 19 Descripción del WS2812B

Comunicación con la tira de LEDS

Se establece comunicación serial ya que se envía datos de manera lógica a cada módulo LED independiente cuyo tamaño de trama es de 24 bits; se repite varias veces según la cantidad de LEDS. [15]

Cada LED posee un registro de 24 bits en la cual es posible programar los 3 colores en código RGB teniendo 8 bits de datos por color (Rojo, Verde y Azul).

La trama de datos opera de manera lógica y son codificadas por un control de tiempo en alto (5 voltios) y tiempo en bajo (0 voltios) a 1,25 microsegundos. En la figura 6.31 y 6.32 se visualiza el código en 0 y 1 para permitir el control del color que deseamos. El RESET es para eliminar el color asignado.

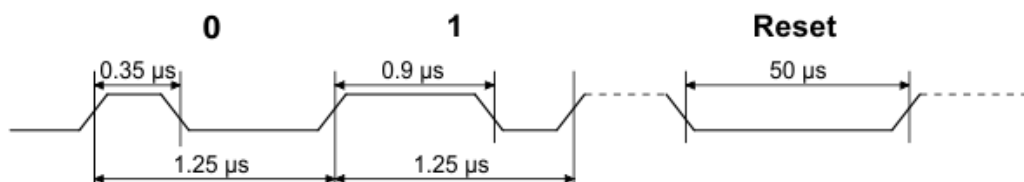


Fig. 6.31 Obtención de los datos

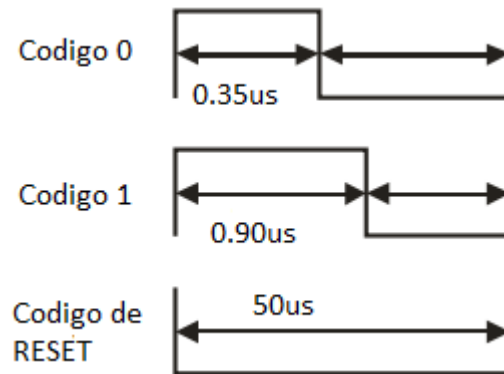


Fig. 6.32 Tiempos de los códigos

Se dice que está en código cero; cuando la tira de LEDS detecta que la duración del High es de $0.35 \mu\text{s}$. y la duración del Low es $0.9 \mu\text{s}$, se puede observar la Fig. 6.31. Se dice que el código está en uno; cuando la tira de LEDS detecta que la duración del High es de $0.90 \mu\text{s}$ y la duración del Low es $0.35 \mu\text{s}$.

Si no hay datos por $50 \mu\text{s}$ se resetea la secuencia de LEDS y se aplica la trama de datos a todos los LEDS.

Armado de la trama de datos:

Cada módulo led WS2812B posee 1 registro con 24 bits y para controlar mediante código de color RGB se debe programar la intensidad del color a mostrar en los LEDS; para ello la trama está

dividida en 3 arreglos con 8 bits cada uno, los primeros 8 bits es para la intensidad del color verde, el segundo arreglo es para el color rojo y la última secuencia es para el color azul.



Fig. 6.33 Trama de datos de los LEDs

Los LEDs se encuentran interconectados entre ellos mediante el pin Dout y el pin Din permitiendo la comunicación entre ellos.

El Arduino enviará la trama de datos que ha sido codificada (véase la figura 90 y 91), donde destinará los 24 bits para cada LED, es decir si la tira posee 3 LEDs, el Arduino mandará trama de 96 bits y cada módulo adquirirá los primeros 24 bits para almacenarlos y producir la luz que ha sido programada. Finalmente se elimina la trama enviada.



Fig. 6.34 Imagen de la Tira de LEDS WS2812B

Se ha utilizado el pin 12 de la Arduino UNO para ejercer el control sobre la tira; tal pin está conectado al DIN del LEDSTRIP (cable VERDE), y le envía la trama de datos con sus respectivos tiempos para que cada led presente el color que ha sido programado. Se usa el cable Rojo para la alimentación a 5 V y el negro para el punto común (GND).



Fig. 6.35 Imagen de los conectores de la tira de LEDS

La Figura 6.35 representa una tira de 3 módulos de LEDS, y se muestra como la trama de datos es desglosado por cada estado D1, D2, D3 y D4.

Se toma como referencia D1, D2 y D3 como módulos ya que D4 es un dispositivo con pin Dout sin conexión. Se comienza con el primer ciclo, donde el D1 recibe trama de 96 bits y toma sus 24 bits que le corresponden aminorando la secuencia a 48, D2 recibe los 48 bits y repetirá el proceso que realizó D1, el D3 recibe y repite la secuencia y en el caso de que exista más módulos se deberá enviar una trama cuya cantidad de bits resultará de la multiplicación del número total de LEDS existentes en la tira por los 24 bits.

En D4 no hay bits por leer y escribir por lo tanto se genera el RESET para resetear los módulos anteriores y generar el segundo ciclo; de esta manera se repetirá infinitas veces.

La Fig. 6.36 permite visualizar la forma de escritura de la trama de datos por led; es decir, de la trama de datos general el D1 (primer led) obtiene los bits que le corresponden y tales bits son eliminados de la trama general, el D2 repite la acción y obtendrá los bits que le fueron asignados; sucederá lo mismo hasta que todos los bits hayan sido tomados y se vuelve a enviar el código para los siguientes LEDS repitan la secuencia.

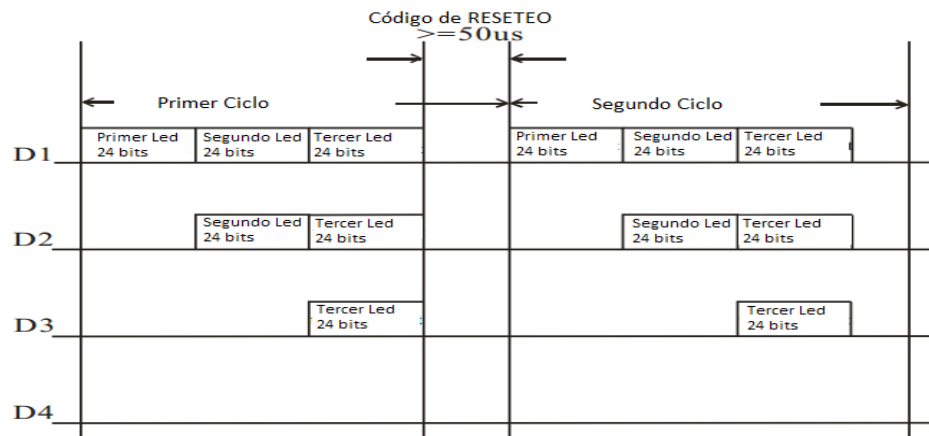


Fig. 6.36 Comunicación al Conjunto de LEDS

6.2 ACOPLAMIENTO GENERAL

En la página siguiente se presenta el diagrama de conexiones que se realizó en la maqueta:

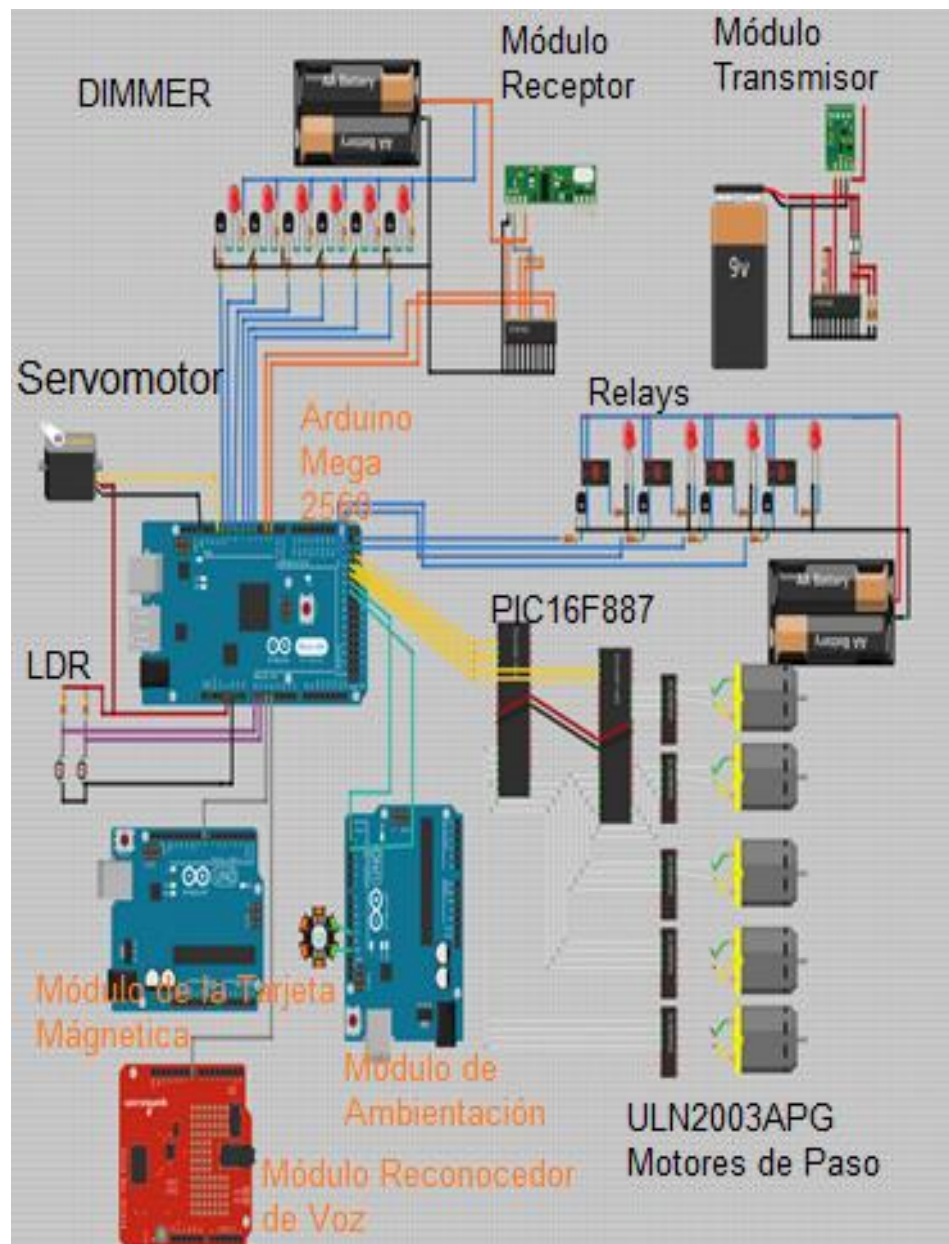


Fig. 6.37 Acoplamiento del sistema

Se presenta a continuación las imágenes de las baquelitas:

Dimmer

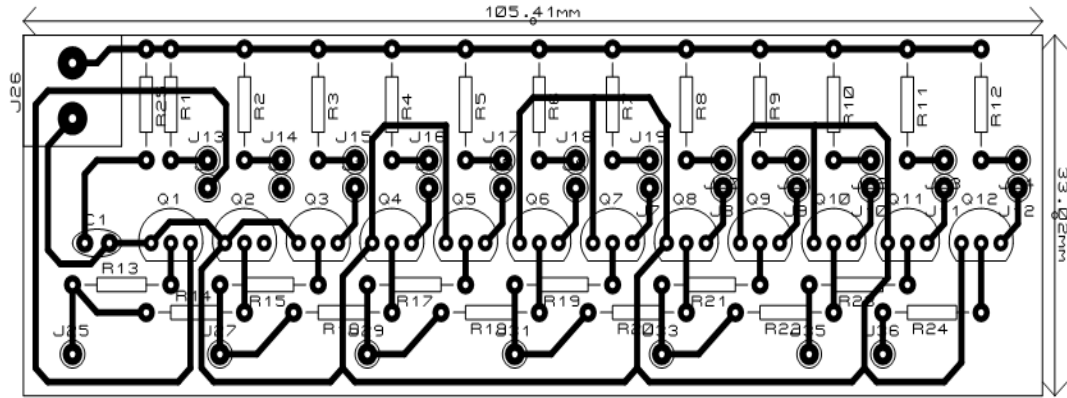


Fig. 6.38 Dimmer baquelita

Control Remoto y Receptor

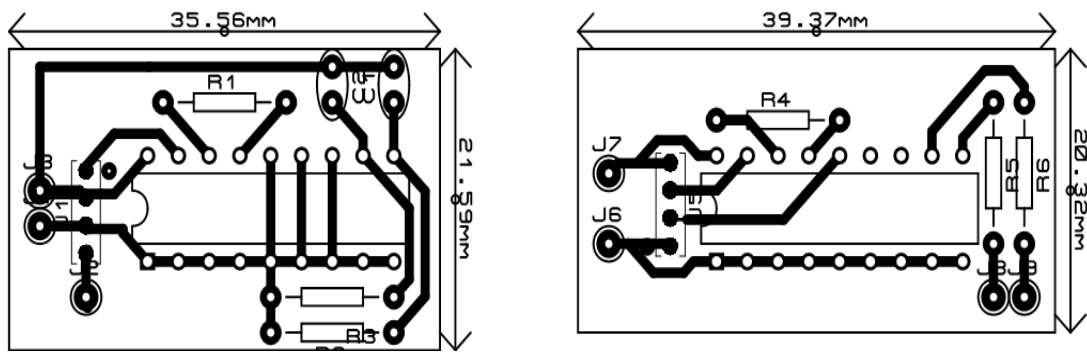


Fig. 6.39 Control Remoto y Receptor

PIC Motor de paso

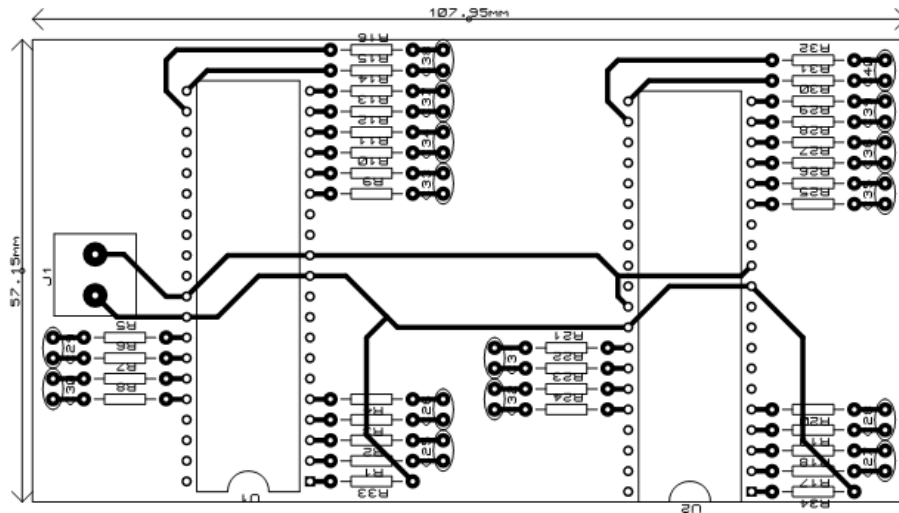


Fig. 6.40 PIC para los motores de paso

Relay

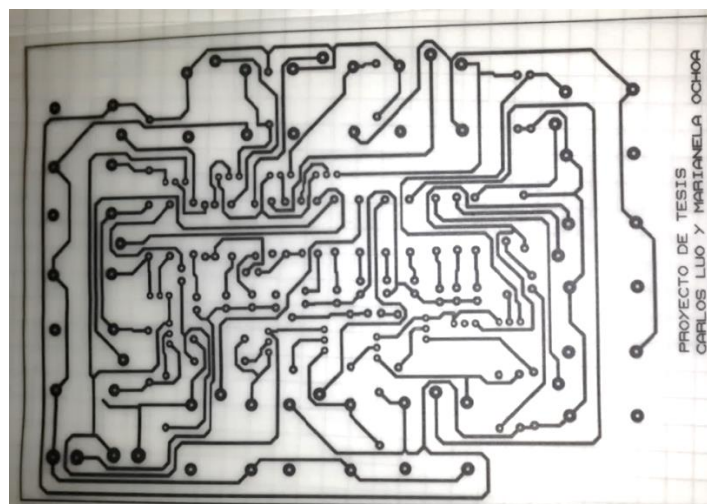


Fig. 6.41 Relays

CAPÍTULO 7

ANÁLISIS DE RESULTADOS

7.1 ELEMENTOS UTILIZADOS Y COSTOS

La tabla que se presenta a continuación corresponde a los componentes electrónicos así como se incluye también costos de cable, conectores y módulos.

ITEM	DESCRIPCION	CANTIDAD	COSTO UNITARIO	TOTAL
1	Relé 5V	20	1	20
2	Transistores 2N3904	10	0,1	1
3	Diodos 1N4148	10	0,05	0,5
4	Borneras	3	0,2	0,6
5	Resistencia 1/4 Watt	40	0,05	2

6	Zócalos 40 pines	3	0,2	0,6
7	Espadines	4	0,325	1,3
8	2N2222	5	0,2	1
9	IRF640	2	1,5	3
10	PLUG 9V	4	0,25	1
11	105 250	2	0,8	1,6
12	ULN2003APG	2	0,8	1,6
13	Cables	20	0,4	8
14	Placa DIMMER	1	4,78	4,78
15	Placa Control Remoto	1	3,25	3,25
16	Placa Fuente	1	7,96	7,96
17	Películas	3	1	3
18	2N2222	14	0,1	1,4
19	Resistencias 10K	24	0,05	1,2
20	Resistencias 470	12	0,05	0,6
21	Zócalos 18 pines	3	0,15	0,45
22	9V Conector	2	0,25	0,5
23	Resistencia 43K	1	0,05	0,05
24	Tira de led	1	54,21	54,21
25	Mega 2560	3	32,51	97,53
26	Motor de pasos	3	10,43	31,29
27	Matrimonio 1/4" a 5 mm y eje de 1/4 x7" acero inoxidable	7	16,68	116,76
28	Modulo lector de memoria SD	1	13,9	13,9
29	Cable plano de 40 pines	2	11,26	22,52
30	Conector + Terminal macho/ hembra para espadines	45	0,34	15,2
31	Motor de pasos	2	10,43	20,86
32	Llave hexagonal 3/32" para ajustar polea y matrimonio	2	1,24	2,48
33	Servomotor	1	12	12
34	Arduino uno	3	30	90

35	Easyvr 2.0 shield	1	90	90
36	Cámara	1	20	20

Total	\$ 652,14
-------	-----------

Tabla # 14 Precios de elementos adquiridos.

La tabla que se presenta a continuación corresponde a los materiales obtenidos para la construcción de la maqueta, y herramientas que ayudarán para que la implementación del sistema funcione.

ITEM	DESCRIPCION	CANTIDAD	COSTO UNITARIO	COSTO TOTAL
1	Blancola 1 lt	1	2,95	2,95
2	Amoladora	1	70	70
3	Arco sierra	1	5,35	5,35
4	Dcorlac	2	7,78	15,56
5	Plancha de plywood de 4 mm	1	26,4	26,4
6	Plancha de plywood de 9 mm	1	15,54	15,54
7	Lija agua	3	0,54	1,62
8	Lija madera	3	0,36	1,08
9	Funda de tornillo para madera	2	0,2	0,4
10	Funda de tornillo pequeño	2	0,44	0,88
11	Pistola para silicón	1	1,78	1,78
12	Silicona barras	1	1,78	1,78
13	Brochas 2"	2	0,49	0,98
14	Brocha prensada	1	1,78	1,78
15	Madera Balsa	3	1	3
16	Plumafon	1	0,7	0,7
17	Cinta aislante	2	6,24	12,48

18	Rollo de alambre de 1.6	1	7,84	7,84
19	Baterías 9 V	2	3	6
20	Martillo Stanley	1	7,11	7,11
21	Clavo fino paquete	1	0,5	0,5
22	Destornillador	2	1,27	2,54
23	Estaño	1	21	21
24	Brujitas	10	0,8	8
25	Plancha de acrílico	1	80	80
			Subtotal	\$ 264,85
			Total	\$ 295,27

Tabla # 15 Precios de materiales adquiridos.

El costo total por realización de baquelitas es 38.03 dólares.

7.2 PRESUPUESTOS

Para la realización del presente proyecto es necesario tener a disposición la cantidad de mil dólares americanos considerando que no se realiza montajes reales sino en maqueta y se está demostrando la eficacia del sistema. A continuación se detalla el costo total:

DESCRIPCION	PRESUPUESTO
Componentes electrónicos	652,14
Herramientas para carpintería	295,27
Baquelitas	38,03
Inversión por pruebas y fallos	10
Total	\$ 995,44

Tabla # 16 Inversión total.

7.3 PRUEBAS Y RESULTADOS DE PRUEBAS

Se presenta los resultados de los diseños de pruebas que anteriormente se expusieron correspondientes a cada sección.

La primera prueba realizada fue la verificación del sistema de escenarios (escenario #1, #2 y #3) teniendo los siguientes resultados:

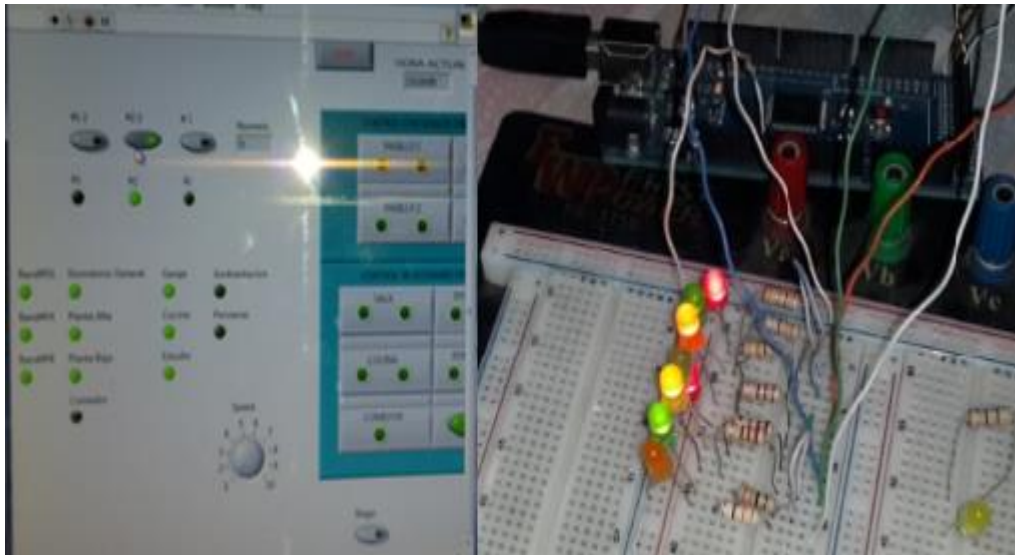


Fig. 7.1 Resultado uno en el cual se visualizan los LEDS encendidos

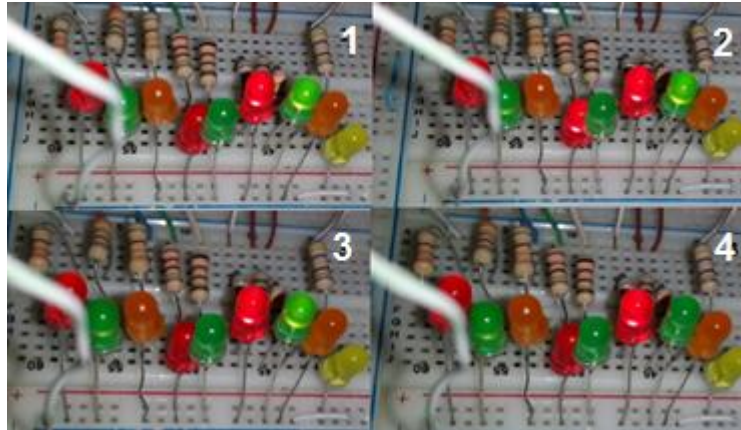


Fig. 7.2 Resultado de escenario

En la Fig. 7.2 se puede apreciar que en las 2 primeras imágenes se está iniciando la secuencia de encendido de luces mientras que en las imágenes 3 y 4 se activa la secuencia de apagado de LEDs. La segunda prueba efectuada fue el control lumínico como se visualiza en la Fig. 7.3.

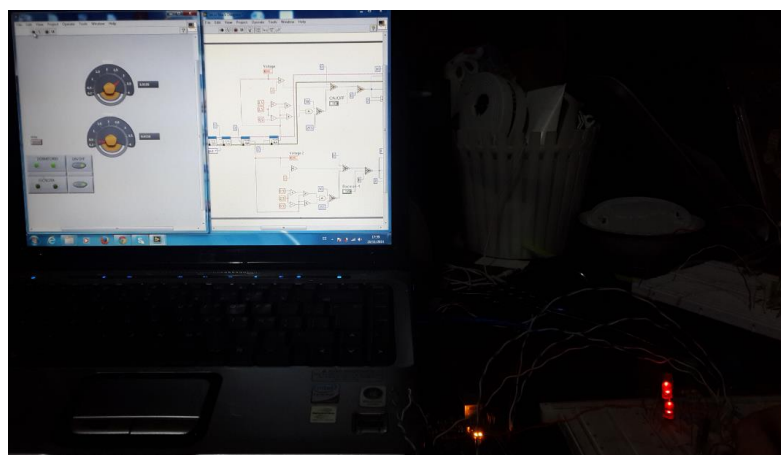


Fig. 7.3 Resultado del control lumínico.

Para que la prueba tenga los resultados esperados; fue realizada en las horas programadas; es decir, se esperó a que exista presencia de oscuridad y si observamos con detalle la imagen, notaremos lo siguiente:

- La lectura de los LDR son diferentes debido a que uno fue manipulado con la mano para variar la intensidad de luminosidad.
- Los indicadores LEDS virtuales están activados y por consiguiente los LEDS físicos están encendidos al cumplirse el control.

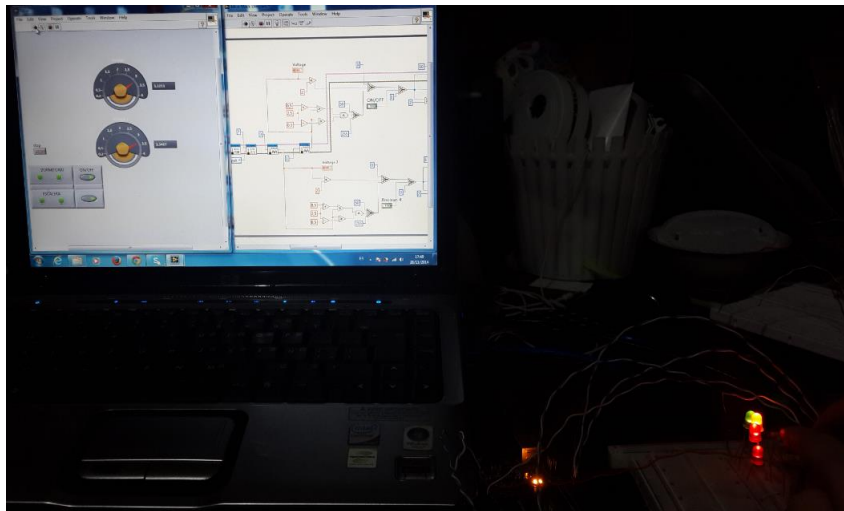


Fig. 7.4 Resultado del control

En la Fig. 7.4 se aprecia que el control programado correspondiente al lumínico funciona acorde a lo esperado.

La siguiente prueba realizada fue la integración de los controles de iluminación como se comentó en la sección 2 del diseño de pruebas.

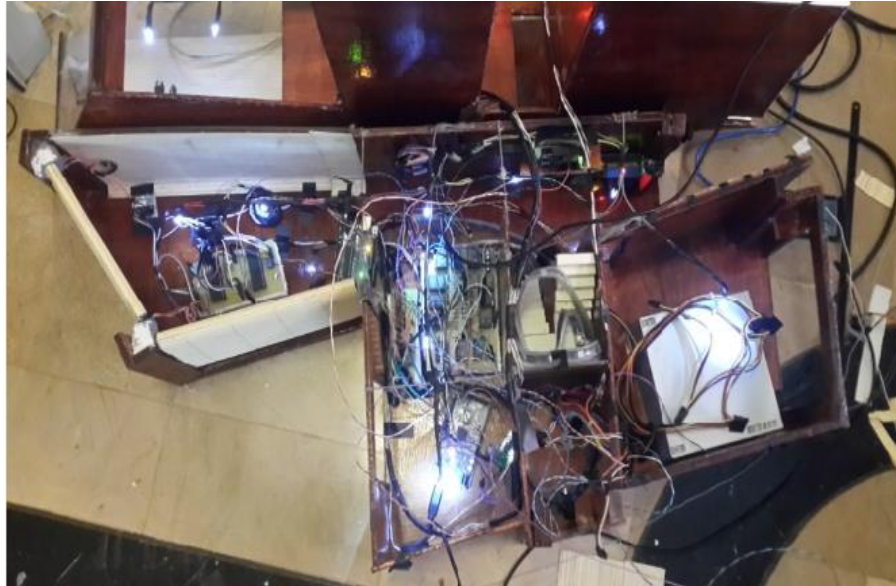


Fig. 7. 5 Resultado de la maqueta

En la Fig. 7.5 se puede apreciar que todas las luminarias están energizadas; esto nos indica que funciona todo el control para iluminación, es decir; los controles manuales, dimerización, control de escenarios, están activos completando una gestión; la cual es programación y ahorro energético.

Para la sección tres se realizaron varias pruebas en diferentes días para determinar si el sistema funciona correctamente, y se concluyó que los controles para el sistema de iluminación no presenta complicación alguna y de manera similar podemos afirmar el funcionamiento de la tarjeta magnética junto con la captura de imágenes usando la cámara que forman parte del sistema de seguridad.

En el caso del reconocimiento de voz se observó que si el usuario modifica la pronunciación así como la tonalidad; el módulo no reconoce la voz que fue programada y por lo tanto no se ejecutará ninguna acción.

A continuación se presenta imágenes en los cuales se puede observar el funcionamiento de las persianas y ambientación.



Fig. 7. 6 Resultado del control de persianas

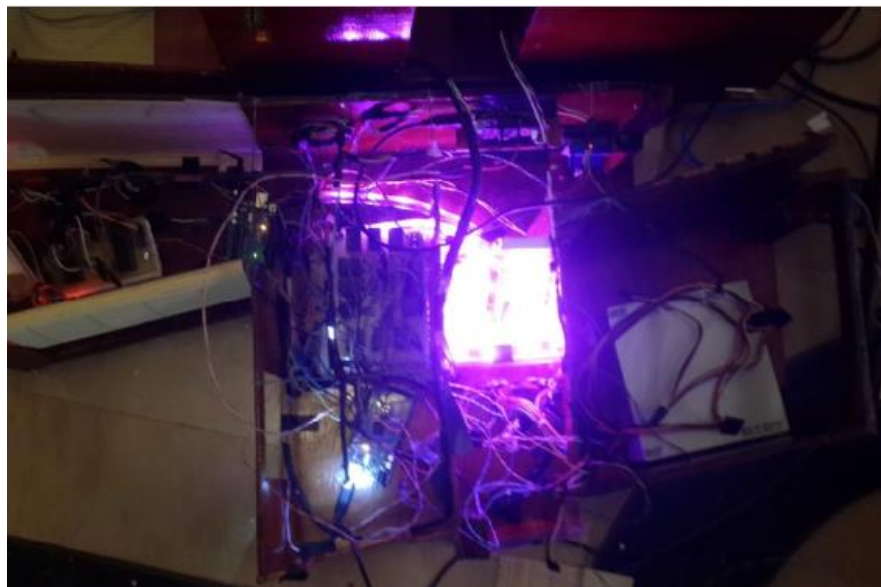


Fig. 7. 7 Resultado del control de tira de LEDS

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES:

1. El resultado del funcionamiento del sistema domótico implementado en la maqueta utilizando los microcontroladores Arduino y la interfaz de usuario Labview funcionó conforme a lo programado.
2. Se podrá realizar demostraciones del sistema domótico en la clase de domótica y usarla a la maqueta para futuros proyectos.
3. Se obtienen buenos resultados usando los módulos Arduino cuando se trata de automatización ya que además de ser didácticos, fácil para implementación y de bajo costo; son también amigables para la programación.

RECOMENDACIONES:

1. Cotizar módulos considerando la funcionalidad, eficacia y durabilidad para comparar dispositivos en varias marcas como Bticino, simonvis, e insteon.
2. Como el sistema domótico se presentó en una maqueta, es necesario darle el mantenimiento correcto del mismo para las demostraciones académicas y realizaciones de experimentos.
3. Para que el módulo de reconocimiento de voz funcione establemente se necesita un ambiente silencioso; caso contrario no funcionará y no se ejecutará el reconocimiento de voz.
4. En el caso de ejecutar el control lumínico en un lugar cerrado donde exista poca intensidad de luz; es preferible usar una linterna para maniobrar el control.
5. Permanecer a una distancia menor a 5 metros para un adecuado funcionamiento del control remoto siempre y cuando el nivel de la batería sea óptimo, caso contrario la distancia debe acortarse. Si se está usando

el control remoto permanecer a una distancia menor a 5 metros para la funcionalidad del receptor, ignorando el déficit de energía de la batería.

6. En el caso de usar las tarjetas Arduino para la realización de proyectos; añadir el código ya programado al sketch modificado considerando las librerías incorporadas, caso contrario no funcionará el sistema.

7. Considerar el uso de reguladores de voltaje para la seguridad de los módulos, los picos de corrientes dañan los integrados ocasionando pérdidas de control, el Jack de la Arduino no soporta voltajes mayores a 10 voltios ya que se sobrecalienta hasta ocasionar el daño total de la tarjeta.

BIBLIOGRAFÍA

- [1] Live Modern, DOMÓTICA, <http://www.livemodern.org/domotica-definicion/>
- [2] Huidobro José Manuel, <http://www.monografias.com/trabajos14/domotica/domotica.shtml>
- [3] Pérez Guzmán Julio César, EDIFICIOS INTELIGENTES, <http://www.monografias.com/trabajos15/edific-inteligentes/edific-inteligentes.shtml>
- [4] Reyes Padilla Karenh Gyssel, DOMÓTICA, <http://www.monografias.com/trabajos35/domotica/domotica.shtml>
- [5] Escobar Alex, <http://www.monografias.com/trabajos5/edin/edin.shtml>
- [6] Quiñones Ponte Bhily, <https://es.scribd.com/doc/218307407/Monografia-de-Domotica>
- [7] Pololu Robotics & Electronics, <https://www.pololu.com/product/2546>
- [8] MyHDL, Python silicon, http://old.myhdl.org/doku.php/projects:uart_rs232_receiver_transmitter
- [9] SILICA Avnet Company, <http://www.silica.com/product/cy7c64225-28pvxc-plug-and-play-usb-to-uart-brid.html>

- [10] Maxim Integrated, <http://www.maximintegrated.com/en/app-notes/index.mvp/id/214>
- [11] Atmel Corporation, <http://www.atmel.com/Images/doc7799.pdf>
- [12] Elec Freaks, <http://www.elec Freaks.com/5418.html>
- [13] Sparkfun, <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>
- [14] TIGAL VEEAR, <http://www.veear.eu/products/easyvr-Arduino-shield/>
- [15] Pololu Robotics & Electronics, <https://www.pololu.com/product/2551>
- [16] Wikipedia, <http://en.wikipedia.org/wiki/USB>
- [17] Arduino Software, <http://Arduino.cc/en/Main/ArduinoBoardMega2560>
- [18] José Rivas A. , MANUAL ILUSTRADO PARA LA INSTALACIÓN DOMÓTICA, http://books.google.com.ec/books?id=eB5Mp-b0asAC&pg=PA24&dq=instalacion+alarma&pg=PP1&redir_esc=y#v=onepage&q&f=true
- [19] Stefan Junstrand, Xavier P., DOMÓTICA Y HOGAR DIGITAL, 2004 http://books.google.com.ec/books?id=8ERFqWcdHAEC&pg=PA51&hl=es&source=gbs_toc_r&cad=3#v=onepage&q&f=false
- [20] Luciano Redolfi, DOMÓTICA, Pocket Users, 2013, <http://www.mediafire.com/view/Intn44jni0d1juw/DOMOTICA.pdf>

[21] Simon Monk, 30 ARDUINOS PROJECTS for the EVIL GENIUS, McGraw Hill,
http://www.nfiautomation.org/FREE_Download/Technical%20Documents/Arduino/30%20Arduino%20projects%20for%20the%20evil%20genius.pdf

ANEXOS

ANEXO 1. PROGRAMACIÓN TOTAL DEL SISTEMA DOMÓTICO



- 1. Documento de especificaciones técnicas de los materiales eléctricos.
- 2. Documento de especificaciones técnicas de los materiales de cableado eléctrico.
- 3. Documento de especificaciones técnicas de los materiales de cableado eléctrico.
- 4. Documento de especificaciones técnicas de los materiales de cableado eléctrico.

ANEXO 2. PRUEBAS DE VOLTAJE Y CORRIENTE DE LA MAQUETA

Pruebas de Voltaje y Corriente

Nombre	Voltaje [V]	Corriente Inicial [A]	Corriente Nominal con Carga [A]	Función
Servomotor	5	0,13	0,075	Movimiento de apertura de la puerta del hogar
Arduino -Pololu	9		0,05	Módulo de control de la tira de LEDS
Arduino-NFC	9		0,13	Lector de la tarjeta magnética
Tira de LEDS	5	0,05	0,7	Tira de LEDS para a ambientación
7805	12		0,18	Reductor de voltaje para los Arduinos
Arduino Mega	5		0,3	Procesador central
Placa Dimmer	5		0,055	Módulo que puede dimerizar 11 LEDS
Placa Relay	5		0,38	Módulo de control de apertura de 5 LEDS y el riego
Placa PIC	5		0,003	Módulo de control de persianas
Motor de Paso	5		0,16	Motor que permite el movimiento de persiana y garaje
Total		0,18	2,033	

Pruebas de Potencia

Nombre	Potencia Inicial [W]	Potencia Nominal con Carga [W]	Función
Servomotor	0,65	0,375	Movimiento de apertura de la puerta del hogar
Arduino-Pollolu	0	0,45	Módulo de control de la tira de LEDS
Arduino-NFC	0	1,17	Lector de la tarjeta magnética
Tira de LEDS	0,25	3,5	Tira de LEDS para a ambientación
7805	0	2,16	Reductor de voltaje para los Arduinos
Arduino Mega	0	1,5	Procesador central
Placa Dimmer	0	0,275	Módulo que puede dimerizar 11 LEDS
Placa Relay	0	1,9	Módulo de control de apertura de 5 LEDS y el riego
Placa PIC	0	0,015	Módulo de control de persianas
Motor de Paso	0	0,8	Motor que permite el movimiento de persiana y garaje
Total	0,9	12,145	

ANEXO 3. COTIZACIÓN PRESENTADA POR BTCINO



PROHOME
HOGARES INTELIGENTES

Av. Juan Tanco Marengo Km. 1.5
C.C. Dientro Local 11 y 12. Guayaquil - Ecuador
PBX: (593-4) 292 2346 _ Fax: (593-4) 292 2281

www.prohome.ec

PROFORMA

ATENCIÓN: Sr. Carlos Luo Xu

FECHA: 13 de enero de 2015

OBRA:

SISTEMAS

MY HOME

COSTO



ILUMINACIÓN

JARDIN

- Control riego
- Control puerta garaje
- Control 4 golpes de luces ON / OFF

\$ 396,15

HALL

- Control 1 golpe de luz ON / OFF
- Control 2 golpes de luces dimerizables
- Control crepuscular de luces dimables
- Comando Nighter de 6 funciones
- Central de escenas (creación de 4 escenas)

\$ 1.163,83

SALA

- Control 1 golpe de luz ON / OFF
- Control 2 golpes de luces dimerizables
- Control 2 persianas eléctricas
- Control crepuscular de luces dimables
- Pantalla Touch Multimedia

\$ 4.817,81

COMEDOR

- Control 1 golpe de luz ON / OFF
- Control 2 golpes de luces dimerizables
- Control crepuscular de luces dimables
- Comando triple

\$ 768,00

ESCALERA / CORREDOR PA

- Control 2 golpes de luces ON / OFF
- Comandos triples

\$ 293,48

DORMTORIO MASTER

- Control 1 golpe de luz ON / OFF
- Control 2 golpes de luces dimerizables
- Control 2 persianas eléctricas

\$ 1.755,62

Sensores de movimiento PIR en accesos

Comunicación mediante línea telefónica urbana en caso de intrusión

Sirena de exterior

Ingreso mediante llavero transponder

\$ 4.489,39

SUBTOTAL	\$ 14.884,03
DESCUENTO 10%	\$ 1.488,40
SUBTOTAL	\$ 13.395,63
INSTALACIÓN, CONFIGURACIÓN Y PUESTA EN MARCHA	\$ 2.332,60
SUBTOTAL	\$ 15.728,23
IVA	\$ 1.887,39
TOTAL	\$ 17.615,62

NO INCLUYE: INSTALACION ELECTRICA O CIVIL

CONDICIONES DE VENTA:

Tiempo de entrega: Confirmando stock con fábrica

Promedio: 16 semanas

Forma de Pago: 50 % de anticipo

40% a la entrega del material

10% al termino del proyecto

Validez: 15 días

Elaborado por: Andrés Andrade

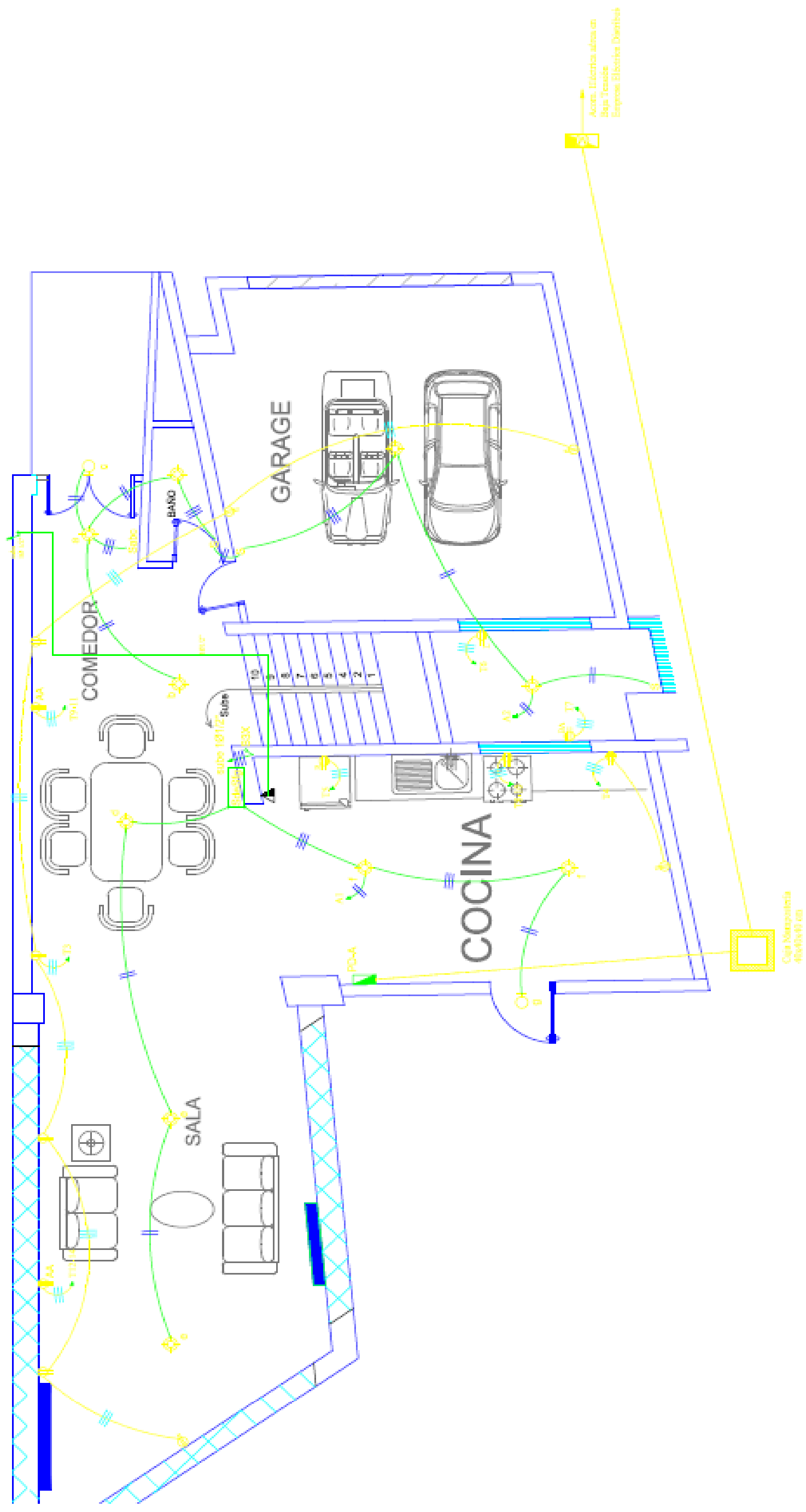
Cotizado por: Angélica León

Distribuidor Autorizado

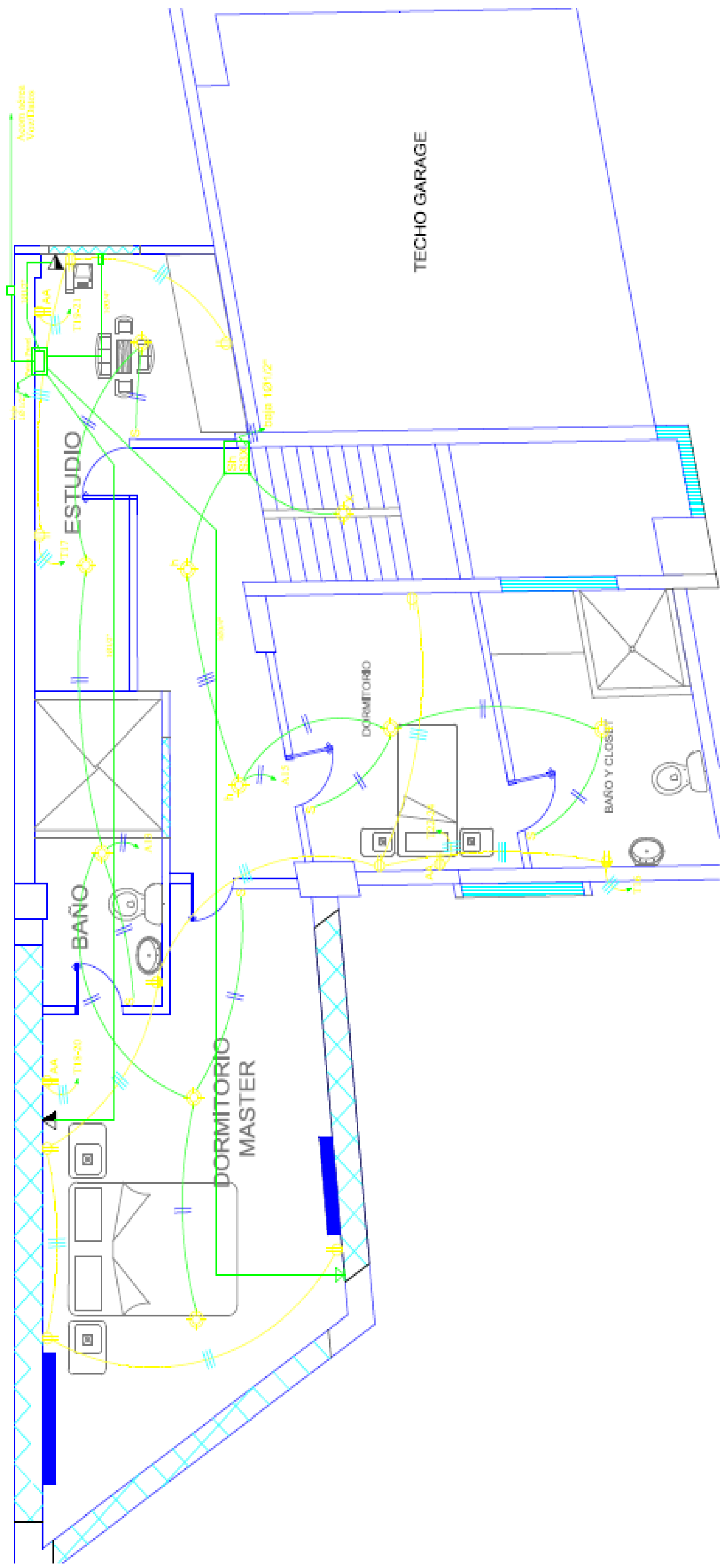


ANEXO 4. PLANOS ELÉCTRICOS DE LA VIVIENDA

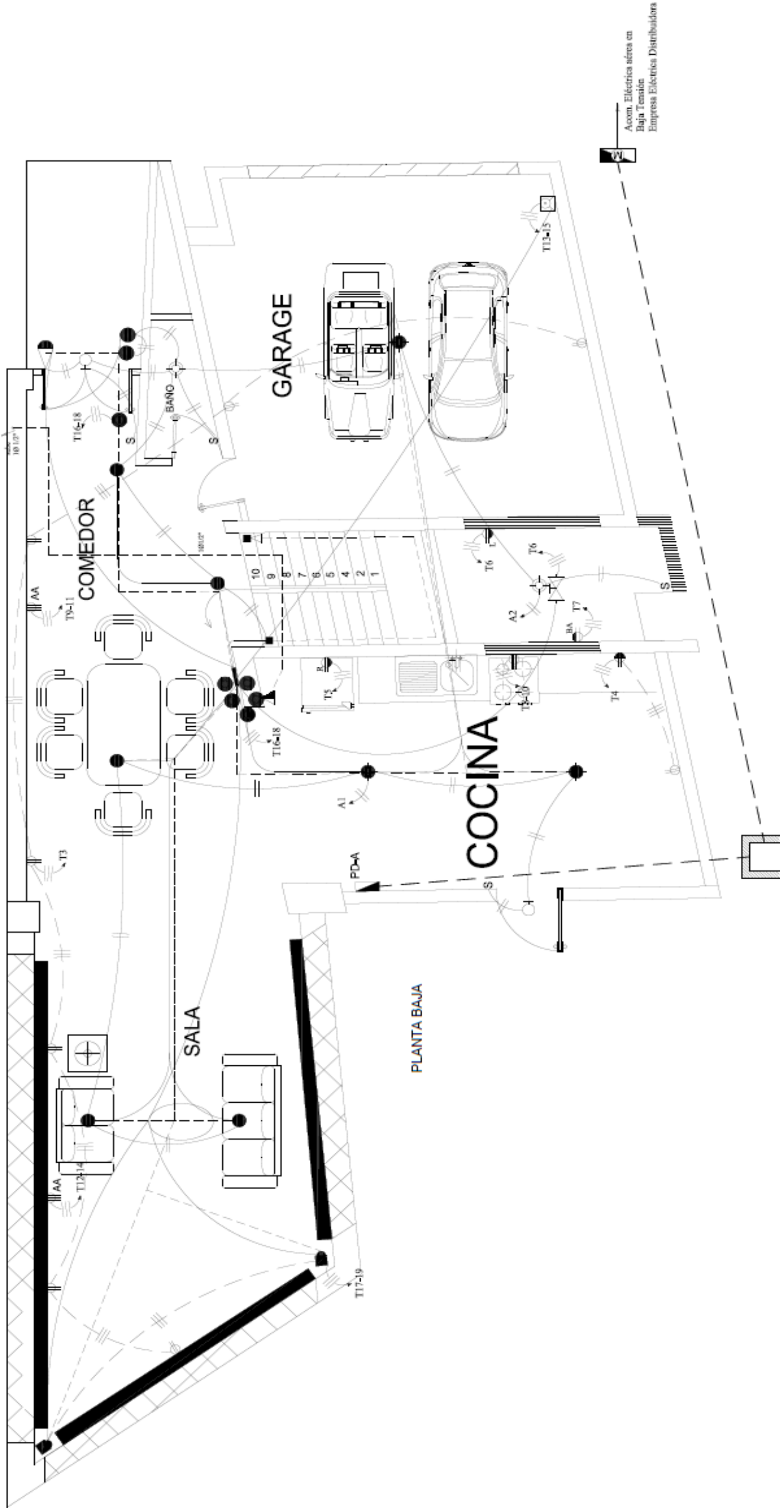
PLANO 1: PLANTA BAJA



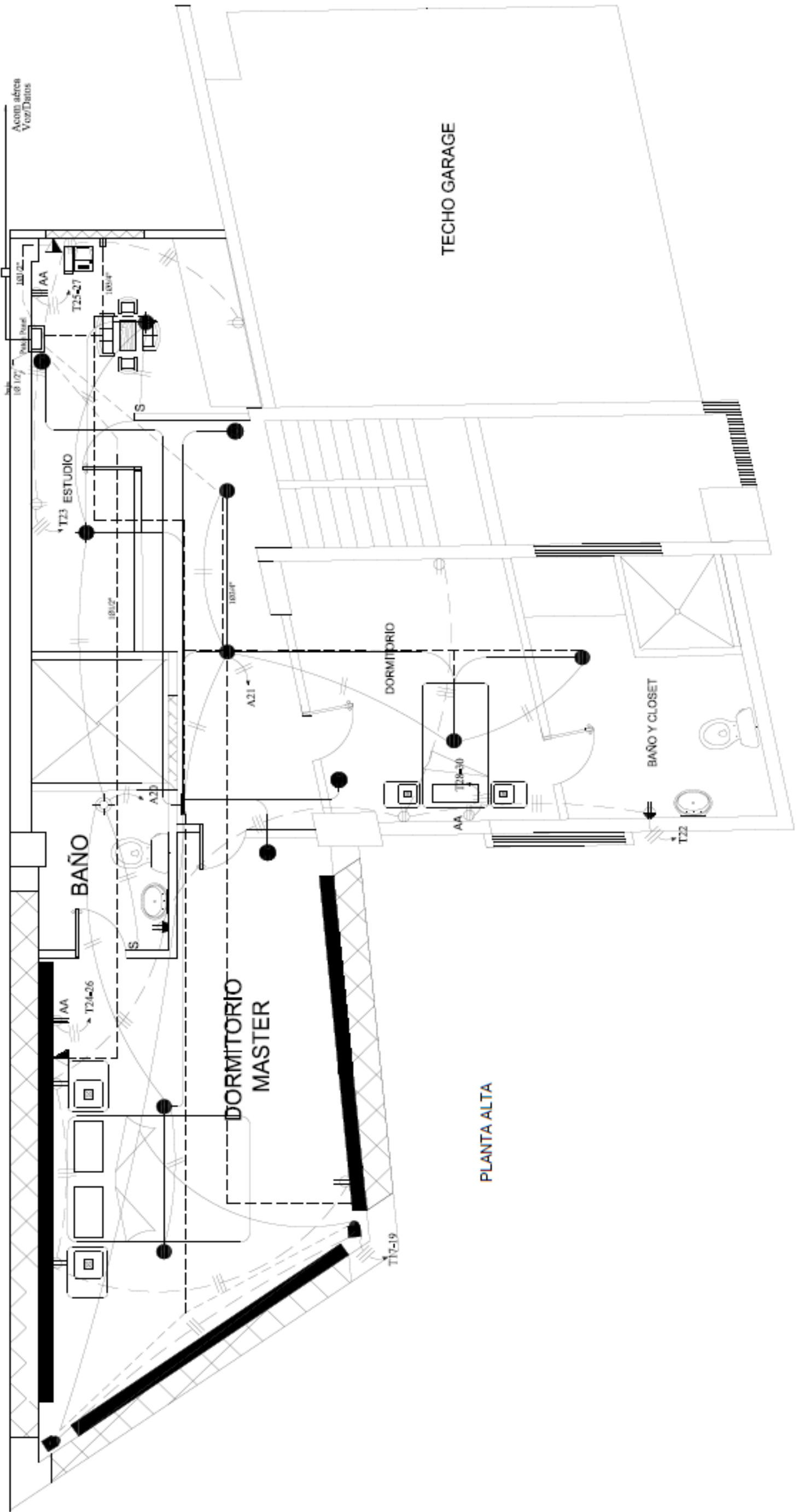
PLANO 2: PLANTA ALTA



ANEXO 5. PLANOS DOMÓTICOS DE LA MAQUETA



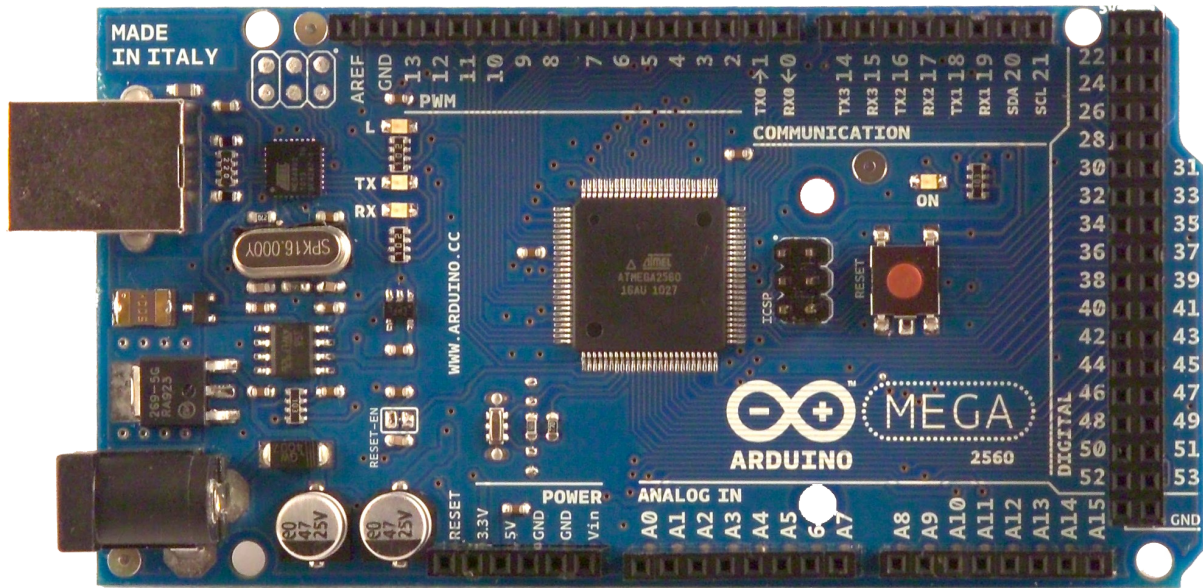
Acorn. Eléctrica aérea en
Baja Tensión
Empresa Eléctrica Distribuidora



PLANTA ALTA

ANEXO 6. ESQUEMÁTICO DEL ARDUINO MEGA 2560

Arduino MEGA 2560



Product Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

Index

Technical Specifications

Page 2

How to use Arduino
Programming Enviroment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Enviromental Policies
half sqm of green via Impatto Zero®

Page 7



RADIOSPARES

RADIONICS



Technical Specification

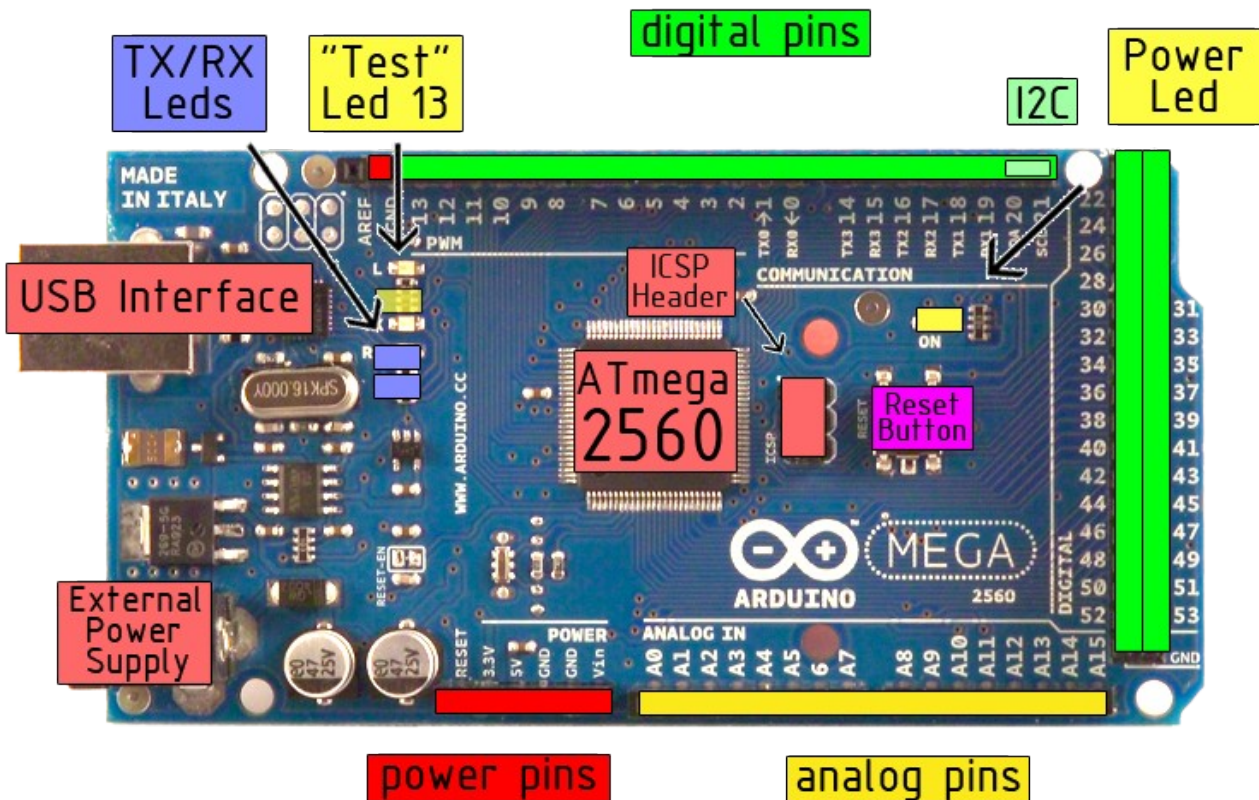


EAGLE files: [arduino-mega2560-reference-design.zip](#) Schematic: [arduino-mega2560-schematic.pdf](#)

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Mega2560 can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and [analogReference\(\)](#) function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.



radiospares

RADIONICS



Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega's digital pins.

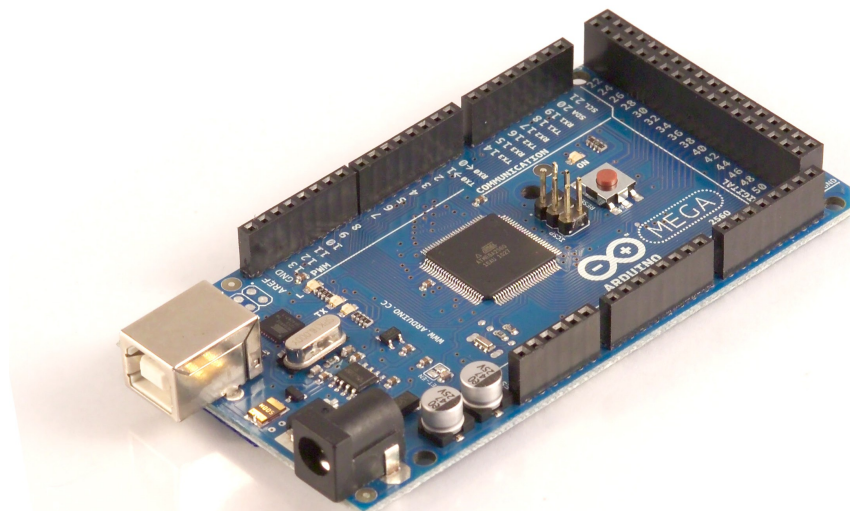
The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. To use the SPI communication, please see the ATmega2560 datasheet.

Programming

The Arduino Mega2560 can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.



radiospares

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega is designed to be compatible with most shields designed for the Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega and Duemilanove / Diecimila. **Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5).**



radiospares

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select MEGA

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!



radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.

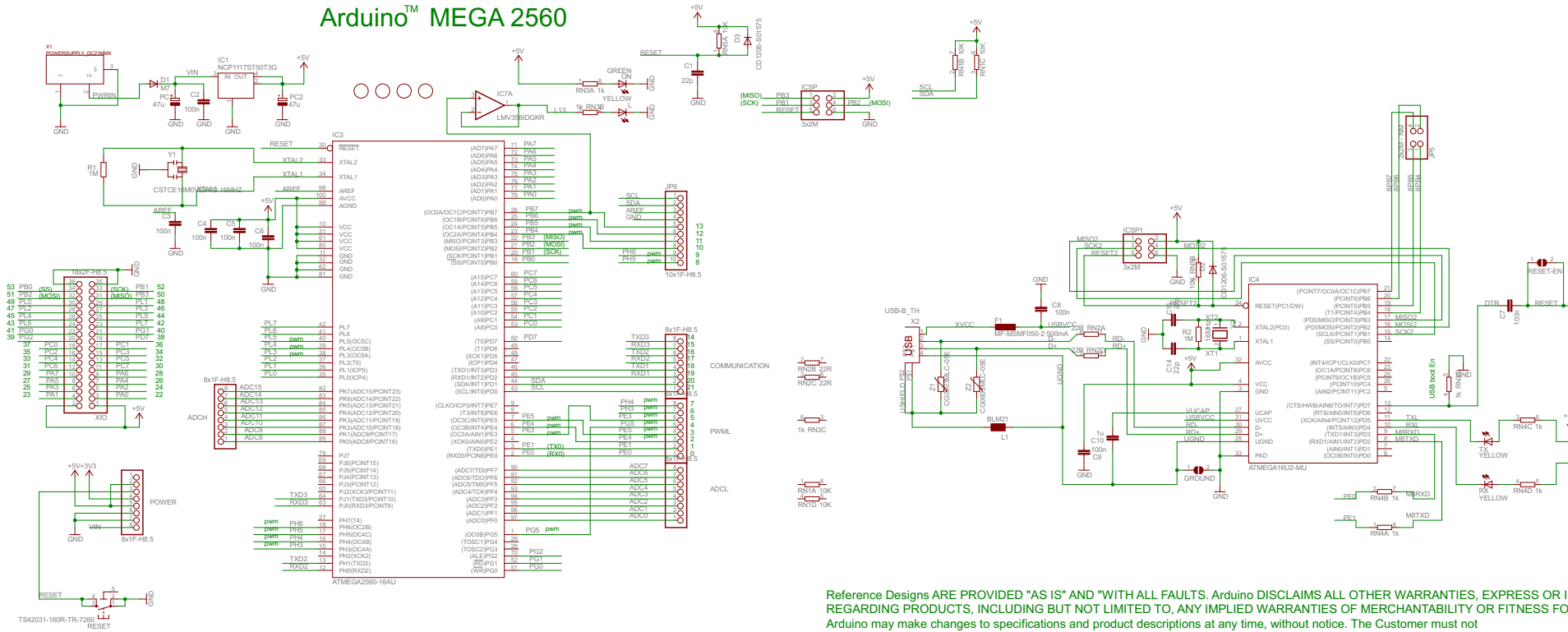


radiospares

RADIONICS



Arduino™ MEGA 2560



Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

ANEXO 7. PROGRAMACIÓN DEL ARDUINO MEGA 2560

LVIFA_BASE

```
/**
**
** LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With LabVIEW.
**
** Written By: Sam Kristoff - National Instruments
** Written On: November 2010
** Last Updated: Dec 2011 - Kevin Fort - National Instruments
**
** This File May Be Modified And Re-Distributed Freely. Original File Content
** Written By Sam Kristoff And Available At www.ni.com/arduino.
**
***/

/**
**
** Includes.
**
***/

// Standard includes. These should always be included.
#include <Wire.h>
#include <SPI.h>
#include <Servo.h>
#include "LabVIEWInterface.h"

/**
** setup()
**
** Initialize the Arduino and setup serial communication.
**
** Input: None
** Output: None
***/

void setup()
{
// Initialize Serial Port With The Default Baud Rate
syncLV();

// Place your custom setup code here
```



```
}
```

```
/******
```

```
** loop()
```

```
**
```

```
** The main loop. This loop runs continuously on the Arduino. It
```

```
** receives and processes serial commands from LabVIEW.
```

```
**
```

```
** Input: None
```

```
** Output: None
```

```
*****/
```

```
void loop()
```

```
{
```

```
  // Check for commands from LabVIEW and process them.
```

```
  checkForCommand();
```

```
  // Place your custom loop code here (this may slow down communication with LabVIEW)
```

```
  if(acqMode==1)
```

```
  {
```

```
    sampleContinuously();
```

```
  }
```

```
}
```



```

    motor2.onestep(FORWARD, SINGLE);
}
void backwardstep2() {
    motor2.onestep(BACKWARD, SINGLE);
}
AccelStepper steppers[8]; //Create array of 8 stepper objects
#endif
// Variables
unsigned int retVal;
int sevenSegmentPins[8];
int currentMode;
unsigned int freq;
unsigned long duration;
int i2cReadTimeouts = 0;
char spiBytesToSend = 0;
char spiBytesSent = 0;
char spiCSPin = 0;
char spiWordSize = 0;
Servo *servos;
byte customChar[8];
LiquidCrystal lcd(0,0,0,0,0,0,0);
// Sets the mode of the Arduino (Reserved For Future Use)
void setMode(int mode)
{
    currentMode = mode;
}
// Checks for new commands from LabVIEW and processes them if any exists.
int checkForCommand(void)
{
#ifdef STEPPER_SUPPORT
    // Call run function as fast as possible to keep motors turning
    for (int i=0; i<8; i++){
        steppers[i].run();
    }
#endif
    int bufferBytes = Serial.available();
    if(bufferBytes >= COMMANDLENGTH)
    {
        // New Command Ready, Process It
        // Build Command From Serial Buffer
        for(int i=0; i<COMMANDLENGTH; i++)
        {

```

```

    currentCommand[j] = Serial.read();
}
processCommand(currentCommand);
return 1;
}
else
{
    return 0;
}
}

```

// Processes a given command

```
void processCommand(unsigned char command[])
```

```

{
    // Determine Command
    if(command[0] == 0xFF && checksum_Test(command) == 0)
    {
        switch(command[1])
        {
            /*****
            ** LIFA Maintenance Commands
            *****/
            case 0x00: // Sync Packet
                Serial.print("sync");
                Serial.flush();
                break;
            case 0x01: // Flush Serial Buffer
                Serial.flush();
                break;
            /*****
            ** Low Level - Digital I/O Commands
            *****/
            case 0x02: // Set Pin As Input Or Output
                pinMode(command[2], command[3]);
                Serial.write('0');
                break;
            case 0x03: // Write Digital Pin
                digitalWrite(command[2], command[3]);
                Serial.write('0');
                break;
            case 0x04: // Write Digital Port 0
                writeDigitalPort(command);

```

```

    Serial.write('0');
    break;
case 0x05: //Tone
    freq = ( command[3]<<8) + command[4];
    duration=(command[8]+ (command[7]<<8)+(command[6]<<16)+(command[5]<<24));
    if(freq > 0)
    {
        tone(command[2], freq, duration);
    }
    else
    {
        noTone(command[2]);
    }
    Serial.write('0');
    break;
case 0x06: // Read Digital Pin
    retVal = digitalRead(command[2]);
    Serial.write(retVal);
    break;
case 0x07: // Digital Read Port
    retVal = 0x0000;
    for(int i=0; i <=13; i++)
    {
        if(digitalRead(i))
        {
            retVal += (1<<i);
        }
    }
    Serial.write( (retVal & 0xFF));
    Serial.write( (retVal >> 8));
    break;
/*****
** Low Level - Analog Commands
*****/
case 0x08: // Read Analog Pin
    retVal = analogRead(command[2]);
    Serial.write( (retVal >> 8));
    Serial.write( (retVal & 0xFF));
    break;
case 0x09: // Analog Read Port
    analogReadPort();
    break;

```

```

/*****
** Low Level - PWM Commands
*****/

case 0x0A: // PWM Write Pin
  analogWrite(command[2], command[3]);
  Serial.write('0');
  break;
case 0x0B: // PWM Write 3 Pins
  analogWrite(command[2], command[5]);
  analogWrite(command[3], command[6]);
  analogWrite(command[4], command[7]);
  Serial.write('0');
  break;
/*****
** Sensor Specific Commands
*****/

case 0x0C: // Configure Seven Segment Display
  sevenSegment_Config(command);
  Serial.write('0');
  break;
case 0x0D: // Write To Seven Segment Display
  sevenSegment_Write(command);
  Serial.write('0');
  break;
/*****
** I2C
*****/

case 0x0E: // Initialize I2C
  Wire.begin();
  Serial.write('0');
  break;
case 0x0F: // Send I2C Data
  Wire.beginTransmission(command[3]);
  for(int i=0; i<command[2]; i++)
  {
    #if defined(ARDUINO) && ARDUINO >= 100
      Wire.write(command[i+4]);
    #else
      Wire.send(command[i+4]);
    #endif
  }
}

```

```

Wire.endTransmission();
Serial.write('0');
break;
case 0x10: // I2C Read
i2cReadTimeouts = 0;
Wire.requestFrom(command[3], command[2]);
while(Wire.available() < command[2])
{
i2cReadTimeouts++;
if(i2cReadTimeouts > 100)
{
return;
}
else
{
delay(1);
}
}
for(int i=0; i<command[2]; i++)
{
#ifdef ARDUINO && ARDUINO >= 100
Serial.write(Wire.read());
#else
Serial.write(Wire.receive());
#endif
}
break;
/*****
** SPI
*****/
case 0x11: // SPI Init
SPI.begin();
Serial.write('0');
break;
case 0x12: // SPI Set Bit Order (MSB LSB)
if(command[2] == 0)
{
SPI.setBitOrder(LSBFIRST);
}
else
{
SPI.setBitOrder(MSBFIRST);
}

```

```

}
Serial.write('0');
break;
case 0x13: // SPI Set Clock Divider
spi_setClockDivider(command[2]);
Serial.write('0');
break;
case 0x14: // SPI Set Data Mode
switch(command[2])
{
case 0:
SPI.setDataMode(SPI_MODE0);
break;
case 1:
SPI.setDataMode(SPI_MODE1);
break;
case 2:
SPI.setDataMode(SPI_MODE2);
break;
case 3:
SPI.setDataMode(SPI_MODE3);
break;
default:
break;
}
Serial.write('0');
break;
case 0x15: // SPI Send / Receive
spi_sendReceive(command);
break;
case 0x16: // SPI Close
SPI.end();
Serial.write('0');
break;
/*****
** Servos
*****/
case 0x17: // Set Num Servos
free(servos);
servos = (Servo*) malloc(command[2]*sizeof(Servo));
for(int i=0; i<command[2]; i++)
{

```



```

    servos[i] = Servo();
}
if(servos == 0)
{
    Serial.write('1');
}
else
{
    Serial.write('0');
}
break;
case 0x18: // Configure Servo
    servos[command[2]].attach(command[3]);
    Serial.write('0');
    break;
case 0x19: // Servo Write
    servos[command[2]].write(command[3]);
    Serial.write('0');
    break;
case 0x1A: // Servo Read Angle
    Serial.write(servos[command[2]].read());
    break;
case 0x1B: // Servo Write uS Pulse
    servos[command[2]].writeMicroseconds( (command[3] + (command[4]<<8)) );
    Serial.write('0');
    break;
case 0x1C: // Servo Read uS Pulse
    retVal = servos[command[2]].readMicroseconds();
    Serial.write ((retVal & 0xFF));
    Serial.write( (retVal >> 8));
    break;
case 0x1D: // Servo Detach
    servos[command[2]].detach();
    Serial.write('0');
    break;
/*****
**                               LCD
*****/
case 0x1E: // LCD Init
    lcd.init(command[2], command[3], command[4], command[5], command[6], command[7],
command[8], command[9], command[10], command[11], command[12], command[13]);

```

```
Serial.write('0');
break;
case 0x1F: // LCD Set Size
  lcd.begin(command[2], command[3]);
  Serial.write('0');
  break;
case 0x20: // LCD Set Cursor Mode
  if(command[2] == 0)
  {
    lcd.noCursor();
  }
  else
  {
    lcd.cursor();
  }
  if(command[3] == 0)
  {
    lcd.noBlink();
  }
  else
  {
    lcd.blink();
  }
  Serial.write('0');
  break;
case 0x21: // LCD Clear
  lcd.clear();
  Serial.write('0');
  break;
case 0x22: // LCD Set Cursor Position
  lcd.setCursor(command[2], command[3]);
  Serial.write('0');
  break;
case 0x23: // LCD Print
  lcd_print(command);
  break;
case 0x24: // LCD Display Power
  if(command[2] == 0)
  {
    lcd.noDisplay();
  }
  else
```

```
{
  lcd.display();
}
Serial.write('0');
break;
case 0x25: // LCD Scroll
  if(command[2] == 0)
  {
    lcd.scrollDisplayLeft();
  }
  else
  {
    lcd.scrollDisplayRight();
  }
  Serial.write('0');
  break;
case 0x26: // LCD Autoscroll
  if(command[2] == 0)
  {
    lcd.noAutoscroll();
  }
  else
  {
    lcd.autoscroll();
  }
  Serial.write('0');
  break;
case 0x27: // LCD Print Direction
  if(command[2] == 0)
  {
    lcd.rightToLeft();
  }
  else
  {
    lcd.leftToRight();
  }
  Serial.write('0');
  break;
case 0x28: // LCD Create Custom Char
  for(int i=0; i<8; i++)
  {
    customChar[i] = command[i+3];
```

```

}
lcd.createChar(command[2], customChar);

Serial.write('0');
break;
case 0x29: // LCD Print Custom Char
lcd.write(command[2]);
Serial.write('0');
break;
/*****
** Continuous Aquisition
*****/

case 0x2A: // Continuous Aquisition Mode On
acqMode=1;
contAcqPin=command[2];
contAcqSpeed=(command[3])+(command[4]<<8);
acquisitionPeriod=1/contAcqSpeed;
iterationsFlt =.08/acquisitionPeriod;
iterations=(int)iterationsFlt;
if(iterations<1)
{
iterations=1;
}
delayTime= acquisitionPeriod;
if(delayTime<0)
{
delayTime=0;
}
break;
case 0x2B: // Continuous Aquisition Mode Off
acqMode=0;
break;
case 0x2C: // Return Firmware Revision
Serial.write(byte(FIRMWARE_MAJOR));
Serial.write(byte(FIRMWARE_MINOR));
break;
case 0x2D: // Perform Finite Aquisition
Serial.write('0');
finiteAcquisition(command[2],(command[3])+(command[4]<<8),command[5]+(command[6]<<8));
break;
/*****
** Stepper

```

```

*****/
#ifdef STEPPER_SUPPORT
case 0x30: // Configure Stepper
  if (command[2] == 5){ // Support AFMotor Shield
    switch (command[3]){
      case 0:
        steppers[command[3]] = AccelStepper(forwardstep1, backwardstep1);
        break;
      case 1:
        steppers[command[3]] = AccelStepper(forwardstep2, backwardstep2);
        break;
      default:
        break;
    }
  }
  else if(command[2]==6) { // All other stepper configurations
    steppers[command[3]] = AccelStepper(1, command[4],command[5],command[6],command[7]);
  }
  else{
    steppers[command[3]] = AccelStepper(command[2],
command[4],command[5],command[6],command[7]);
  }
  Serial.write('0');
  break;
case 0x31: // Stepper Write
  AccelStepper_Write(command);
  Serial.write('0');
  break;
case 0x32: // Stepper Detach
  steppers[command[2]].disableOutputs();
  Serial.write('0');
  break;
case 0x33: // Stepper steps to go
  retVal = 0;
  for(int i=0; i<8; i++){
    retVal += steppers[i].distanceToGo();
  }
  Serial.write( (retVal & 0xFF) );
  Serial.write( (retVal >> 8) );
  break;
#endif
/*****

```

```

** Unknown Packet
*****/

default: // Default Case
  Serial.flush();
  break;
}
}
else{
  // Checksum Failed, Flush Serial Buffer
  Serial.flush();
}
}
/*****
** Functions
*****/

// Writes Values To Digital Port (DIO 0-13). Pins Must Be Configured As Outputs Before Being Written To
void writeDigitalPort(unsigned char command[])
{
  digitalWrite(13, (( command[2] >> 5) & 0x01) );
  digitalWrite(12, (( command[2] >> 4) & 0x01) );
  digitalWrite(11, (( command[2] >> 3) & 0x01) );
  digitalWrite(10, (( command[2] >> 2) & 0x01) );
  digitalWrite(9, (( command[2] >> 1) & 0x01) );
  digitalWrite(8, (command[2] & 0x01) );
  digitalWrite(7, (( command[3] >> 7) & 0x01) );
  digitalWrite(6, (( command[3] >> 6) & 0x01) );
  digitalWrite(5, (( command[3] >> 5) & 0x01) );
  digitalWrite(4, (( command[3] >> 4) & 0x01) );
  digitalWrite(3, (( command[3] >> 3) & 0x01) );
  digitalWrite(2, (( command[3] >> 2) & 0x01) );
  digitalWrite(1, (( command[3] >> 1) & 0x01) );
  digitalWrite(0, (command[3] & 0x01) );
}

// Reads all 6 analog input ports, builds 8 byte packet, send via RS232.
void analogReadPort()
{
  // Read Each Analog Pin
  int pin0 = analogRead(0);
  int pin1 = analogRead(1);
  int pin2 = analogRead(2);
  int pin3 = analogRead(3);
  int pin4 = analogRead(4);
}

```

```

int pin5 = analogRead(5);
//Build 8-Byte Packet From 60 Bits of Data Read
char output0 = (pin0 & 0xFF);
char output1 = ( ((pin1 << 2) & 0xFC) | ( (pin0 >> 8) & 0x03) );
char output2 = ( ((pin2 << 4) & 0xF0) | ( (pin1 >> 6) & 0x0F) );
char output3 = ( ((pin3 << 6) & 0xC0) | ( (pin2 >> 4) & 0x3F) );
char output4 = ( (pin3 >> 2) & 0xFF);
char output5 = (pin4 & 0xFF);
char output6 = ( ((pin5 << 2) & 0xFC) | ( (pin4 >> 8) & 0x03) );
char output7 = ( (pin5 >> 6) & 0x0F );
// Write Bytes To Serial Port
Serial.print(output0);
Serial.print(output1);
Serial.print(output2);
Serial.print(output3);
Serial.print(output4);
Serial.print(output5);
Serial.print(output6);
Serial.print(output7);
}
// Configure digital I/O pins to use for seven segment display
void sevenSegment_Config(unsigned char command[])
{
    // Configure pins as outputs and store in sevenSegmentPins array for use in sevenSegment_Write
    for(int i=2; i<10; i++)
    {
        pinMode(command[i], OUTPUT);
        sevenSegmentPins[(i-1)] = command[i];
    }
}
// Write values to sevenSegment display. Must first use sevenSegment_Configure
void sevenSegment_Write(unsigned char command[])
{
    for(int i=1; i<9; i++)
    {
        digitalWrite(sevenSegmentPins[(i-1)], command[i]);
    }
}
// Set the SPI Clock Divisor
void spi_setClockDivider(unsigned char divider)
{
    switch(divider)

```

```

{
  case 0:
    SPI.setClockDivider(SPI_CLOCK_DIV2);
    break;
  case 1:
    SPI.setClockDivider(SPI_CLOCK_DIV4);
    break;
  case 2:
    SPI.setClockDivider(SPI_CLOCK_DIV8);
    break;
  case 3:
    SPI.setClockDivider(SPI_CLOCK_DIV16);
    break;
  case 4:
    SPI.setClockDivider(SPI_CLOCK_DIV32);
    break;
  case 5:
    SPI.setClockDivider(SPI_CLOCK_DIV64);
    break;
  case 6:
    SPI.setClockDivider(SPI_CLOCK_DIV128);
    break;
  default:
    SPI.setClockDivider(SPI_CLOCK_DIV4);
    break;
}
}
void spi_sendReceive(unsigned char command[])
{
  if(command[2] == 1)    //Check to see if this is the first of a series of SPI packets
  {
    spiBytesSent = 0;
    spiCSPin = command[3];
    spiWordSize = command[4];
    // Send First Packet's 8 Data Bytes
    for(int i=0; i<command[5]; i++)
    {
      // If this is the start of a new word toggle CS LOW
      if( (spiBytesSent == 0) || (spiBytesSent % spiWordSize == 0) )
      {
        digitalWrite(spiCSPin, LOW);
      }
    }
  }
}

```



```

// Send SPI Byte
Serial.print(SPI.transfer(command[i+6]));
spiBytesSent++;
// If word is complete set CS High
if(spiBytesSent % spiWordSize == 0)
{
    digitalWrite(spiCSPin, HIGH);
}
}
else
{
    // SPI Data Packet - Send SPI Bytes
    for(int i=0; i<command[3]; i++)
    {
        // If this is the start of a new word toggle CS LOW
        if( (spiBytesSent == 0) || (spiBytesSent % spiWordSize == 0) )
        {
            digitalWrite(spiCSPin, LOW);
        }
        // Send SPI Byte
        Serial.write(SPI.transfer(command[i+4]));
        spiBytesSent++;
        // If word is complete set CS High
        if(spiBytesSent % spiWordSize == 0)
        {
            digitalWrite(spiCSPin, HIGH);
        }
    }
}
// Synchronizes with LabVIEW and sends info about the board and firmware (Unimplemented)
void syncLV()
{
    Serial.begin(DEFAULTBAUDRATE);
    i2cReadTimeouts = 0;
    spiBytesSent = 0;
    spiBytesToSend = 0;
    Serial.flush();
}

// Compute Packet Checksum

```

```

unsigned char checksum_Compute(unsigned char command[])
{
    unsigned char checksum;
    for (int i=0; i<(COMMANDLENGTH-1); i++)
    {
        checksum += command[i];
    }
    return checksum;
}
// Compute Packet Checksum And Test Against Included Checksum
int checksum_Test(unsigned char command[])
{
    unsigned char checksum = checksum_Compute(command);
    if(checksum == command[COMMANDLENGTH-1])
    {
        return 0;
    }
    else
    {
        return 1;
    }
}

// Stepper Functions
#ifdef STEPPER_SUPPORT
void AccelStepper_Write(unsigned char command[]){
    int steps = 0;
    int step_speed = 0;
    int acceleration = 0;
    //Number of steps & speed are a 16 bit values, split for data transfer. Reassemble 2 bytes to an int 16
    steps = (int)(command[5] << 8) + command[6];
    step_speed = (int)(command[2] << 8) + command[3];
    acceleration = (int)(command[7] << 8) + command[8];
    steppers[command[4]].setMaxSpeed(step_speed);
    if (acceleration == 0){
        //Workaround AccelStepper bug that requires negative speed for negative step direction
        if (steps < 0) step_speed = -step_speed;
        steppers[command[4]].setSpeed(step_speed);
        steppers[command[4]].move(steps);
    }
    else {
        steppers[command[4]].setAcceleration(acceleration);
    }
}

```

```

    steppers[command[4]].move(steps);
}
}
#endif
void sampleContinuously()
{
for(int i=0; i<iterations; i++)
{
    retVal = analogRead(contAcqPin);
    if(contAcqSpeed>1000) //delay Microseconds is only accurate for values less than 16383
    {
        Serial.write( (retVal >> 2));
        delayMicroseconds(delayTime*1000000); //Delay for necessary amount of time to achieve desired
sample rate
    }
    else
    {
        Serial.write( (retVal & 0xFF) );
        Serial.write( (retVal >> 8));
        delay(delayTime*1000);
    }
}
}
void finiteAcquisition(int analogPin, float acquisitionSpeed, int numberOfSamples)
{
//want to exit this loop every 8ms
acquisitionPeriod=1/acquisitionSpeed;

for(int i=0; i<numberOfSamples; i++)
{
    retVal = analogRead(analogPin);
    if(acquisitionSpeed>1000)
    {
        Serial.write( (retVal >> 2));
        delayMicroseconds(acquisitionPeriod*1000000);
    }
    else
    {
        Serial.write( (retVal & 0xFF) );
        Serial.write( (retVal >> 8));
        delay(acquisitionPeriod*1000);
    }
}
}

```

```

}
}
void lcd_print(unsigned char command[])
{
  if(command[2] != 0)
  {
    // Base Specified By User
    int base = 0;
    switch(command[2])
    {
      case 0x01: // BIN
        base = BIN;
        break;
      case 0x02: // DEC
        base = DEC;
        break;
      case 0x03: // OCT
        base = OCT;
        break;
      case 0x04: // HEX
        base = HEX;
        break;
      default:
        break;
    }
    for(int i=0; i<command[3]; i++)
    {
      lcd.print(command[i+4], base);
    }
  }
  else
  {
    for(int i=0; i<command[3]; i++)
    {
      lcd.print((char)command[i+4]);
    }
  }
  Serial.write('0');
}

```

ANEXO 8. DATASHEET DEL 2N2222



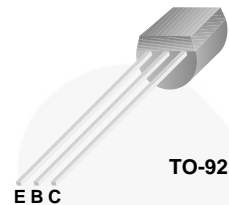
July 2014

PN222A

NPN General-Purpose Amplifier

Features

- This device is for use as a medium power amplifier and switch requiring collector currents up to 500mA.



Ordering Information

Part Number	Top Mark	Package	Packing Method
PN2222ABU	PN2222A	TO-92 3L	Bulk
PN2222ATA	PN2222A	TO-92 3L	Ammo
PN2222ATF	PN2222A	TO-92 3L	Tape and Reel
PN2222ATFR	PN2222A	TO-92 3L	Tape and Reel

Absolute Maximum Ratings^{(1), (2)}

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only. Values are at $T_A = 25^\circ\text{C}$ unless otherwise noted.

Symbol	Parameter	Value	Unit
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	75	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current	1.0	A
T_{STG}	Operating and Storage Junction Temperature Range	-55 to 150	$^\circ\text{C}$

Note:

1. These rating are based on a maximum junction temperature of 150°C .
2. These are steady-state limits. Fairchild Semiconductor should be consulted on applications involving pulsed or low-duty-cycle operation.

Thermal Characteristics⁽³⁾

Values are at $T_A = 25^\circ\text{C}$ unless otherwise noted.

Symbol	Parameter	Max.	Unit
P_D	Total Device Dissipation	625	mW
	Derate Above 25°C	5.0	mW/ $^\circ\text{C}$
$R_{\theta JC}$	Thermal Resistance, Junction to Case	83.3	$^\circ\text{C}/\text{W}$
$R_{\theta JA}$	Thermal Resistance, Junction to Ambient	200	$^\circ\text{C}/\text{W}$

Note:

3. PCB size: FR-4, 76 mm x 114 mm x 1.57 mm (3.0 inch x 4.5 inch x 0.062 inch) with minimum land pattern size.

Electrical Characteristics

Values are at $T_A = 25^\circ\text{C}$ unless otherwise noted.

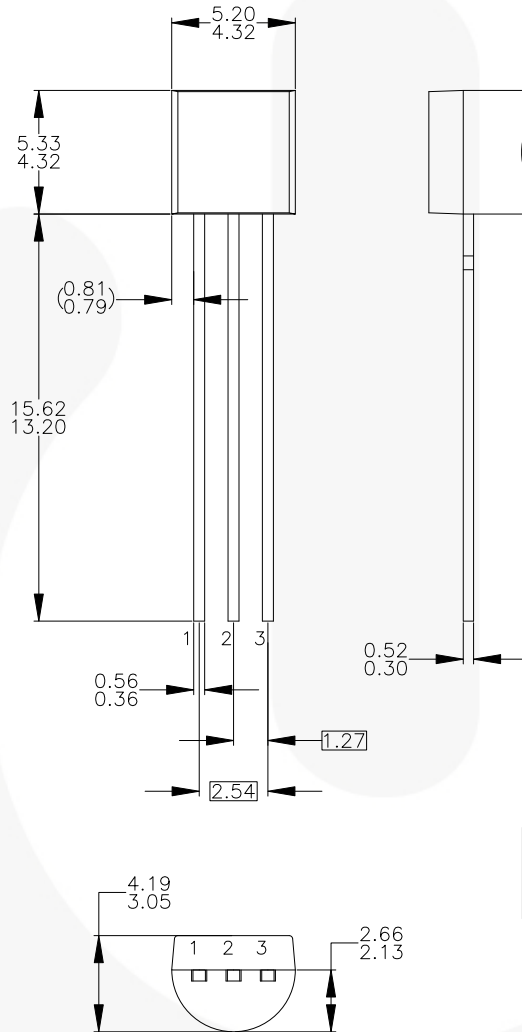
Symbol	Parameter	Conditions	Min.	Max.	Unit
Off Characteristics					
$BV_{(BR)CEO}$	Collector-Emitter Breakdown Voltage ⁽⁴⁾	$I_C = 10\text{ mA}, I_B = 0$	40		V
$BV_{(BR)CBO}$	Collector-Base Breakdown Voltage	$I_C = 10\text{ }\mu\text{A}, I_E = 0$	75		V
$BV_{(BR)EBO}$	Emitter-Base Breakdown Voltage	$I_E = 10\text{ }\mu\text{A}, I_C = 0$	6.0		V
I_{CEX}	Collector Cut-Off Current	$V_{CE} = 60\text{ V}, V_{EB(off)} = 3.0\text{ V}$		10	nA
I_{CBO}	Collector Cut-Off Current	$V_{CB} = 60\text{ V}, I_E = 0$		0.01	μA
		$V_{CB} = 60\text{ V}, I_E = 0, T_A = 125^\circ\text{C}$		10	
I_{EBO}	Emitter Cut-Off Current	$V_{EB} = 3.0\text{ V}, I_C = 0$		10	nA
I_{BL}	Base Cut-Off Current	$V_{CE} = 60\text{ V}, V_{EB(off)} = 3.0\text{ V}$		20	nA
On Characteristics					
h_{FE}	DC Current Gain	$I_C = 0.1\text{ mA}, V_{CE} = 10\text{ V}$	35		
		$I_C = 1.0\text{ mA}, V_{CE} = 10\text{ V}$	50		
		$I_C = 10\text{ mA}, V_{CE} = 10\text{ V}$	75		
		$I_C = 10\text{ mA}, V_{CE} = 10\text{ V}, T_A = -55^\circ\text{C}$	35		
		$I_C = 150\text{ mA}, V_{CE} = 10\text{ V}^{(4)}$	100	300	
		$I_C = 150\text{ mA}, V_{CE} = 1\text{ V}^{(4)}$	50		
		$I_C = 500\text{ mA}, V_{CE} = 10\text{ V}^{(4)}$	40		
$V_{CE(sat)}$	Collector-Emitter Saturation Voltage ⁽⁴⁾	$I_C = 150\text{ mA}, I_B = 15\text{ mA}$		0.3	V
		$I_C = 500\text{ mA}, I_B = 50\text{ mA}$		1.0	
$V_{BE(sat)}$	Base-Emitter Saturation Voltage ⁽⁴⁾	$I_C = 150\text{ mA}, I_B = 15\text{ mA}$	0.6	1.2	V
		$I_C = 500\text{ mA}, I_B = 50\text{ mA}$		2.0	
Small Signal Characteristics					
f_T	Current Gain Bandwidth Product	$I_C = 20\text{ mA}, V_{CE} = 20\text{ V}, f = 100\text{ MHz}$	300		MHz
C_{obo}	Output Capacitance	$V_{CB} = 10\text{ V}, I_E = 0, f = 1\text{ MHz}$		8.0	pF
C_{ibo}	Input Capacitance	$V_{EB} = 0.5\text{ V}, I_C = 0, f = 1\text{ MHz}$		25	pF
$rb'C_c$	Collector Base Time Constant	$I_C = 20\text{ mA}, V_{CB} = 20\text{ V}, f = 31.8\text{ MHz}$		150	pS
NF	Noise Figure	$I_C = 100\text{ }\mu\text{A}, V_{CE} = 10\text{ V}, R_S = 1.0\text{ k}\Omega, f = 1.0\text{ kHz}$		4.0	dB
$Re(h_{ie})$	Real Part of Common-Emitter High Frequency Input Impedance	$I_C = 20\text{ mA}, V_{CE} = 20\text{ V}, f = 300\text{ MHz}$		60	Ω
Switching Characteristics					
t_d	Delay Time	$V_{CC} = 30\text{ V}, V_{EB(off)} = 0.5\text{ V}, I_C = 150\text{ mA}, I_{B1} = 15\text{ mA}$		10	ns
t_r	Rise Time			25	ns
t_s	Storage Time	$V_{CC} = 30\text{ V}, I_C = 150\text{ mA}, I_{B1} = I_{B2} = 15\text{ mA}$		225	ns
t_f	Fall Time			60	ns

Note:

4. Pulse test: pulse width $\leq 300\text{ }\mu\text{s}$, duty cycle $\leq 2.0\%$.

Physical Dimensions

TO-92 (Bulk)



NOTES: UNLESS OTHERWISE SPECIFIED

- A) DRAWING WITH REFERENCE TO JEDEC TO-92 RECOMMENDATIONS.
- B) ALL DIMENSIONS ARE IN MILLIMETERS.
- C) DRAWING CONFORMS TO ASME Y14.5M-1994.
- D) TO-92 (92,94,96,97,98) PIN CONFIGURATION:

PIN	92			94			96			97			98		
	P	F	M	P	F	M	B	F	M	P	F	M	P	F	M
1	E	S	S	E	S	S	B	D	G	C	G	D	C	G	D
2	B	D	G	C	G	D	E	S	S	B	D	G	E	S	S
3	C	G	D	B	D	G	C	G	D	E	S	S	B	D	G

LEGEND:

- P - BIPOLAR
- F - JFET
- M - DMOS
- E - EMITTER
- B - BASE
- C - COLLECTOR
- D - DRAIN
- S - SOURCE
- G - GATE

- E) FOR PACKAGE 92, 94, 96, 97 AND 98: PIN CONFIGURATION DRAIN "D" AND SOURCE "S" ARE INTERCHANGEABLE AT JFET "F" OPTION.
- F) DRAWING FILENAME: MKT-ZA03DREV3.

Figure 1. 3-LEAD, TO92, JEDEC TO-92 COMPLIANT STRAIGHT LEAD CONFIGURATION (OLD TO92AM3)

Package drawings are provided as a service to customers considering Fairchild components. Drawings may change in any manner without notice. Please note the revision and/or date on the drawing and contact a Fairchild Semiconductor representative to verify or obtain the most recent revision. Package specifications do not expand the terms of Fairchild's worldwide terms and conditions, specifically the warranty therein, which covers Fairchild products.

Always visit Fairchild Semiconductor's online packaging area for the most recent package drawings:

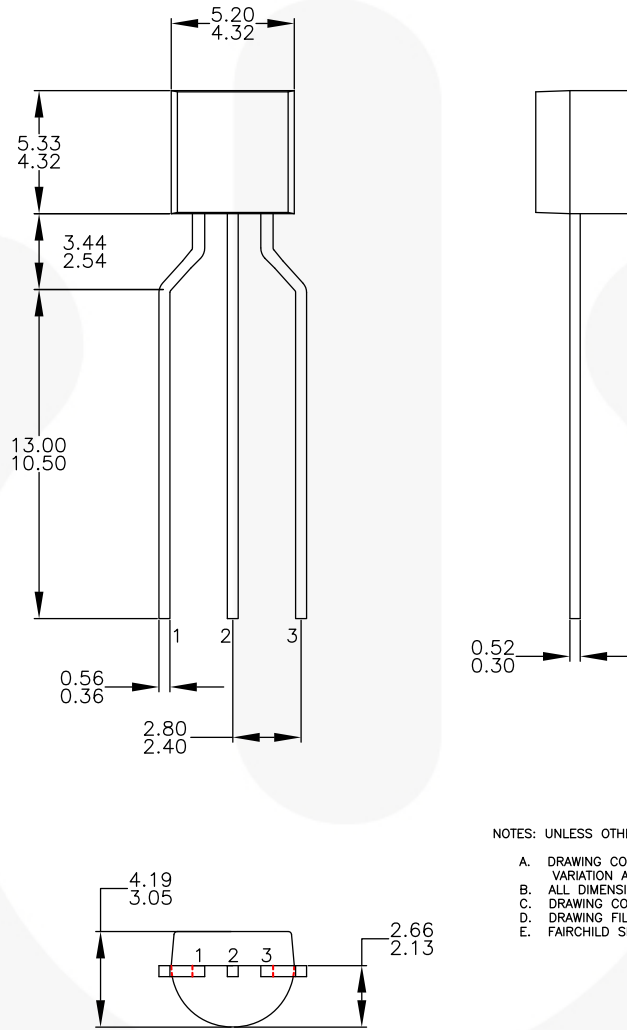
<http://www.fairchildsemi.com/dwg/ZA/ZA03D.pdf>

For current tape and reel specifications, visit Fairchild Semiconductor's online packaging area:

http://www.fairchildsemi.com/packaging_dwg/PKG-ZA03D_BK.pdf

Physical Dimensions (Continued)

TO-92 (Ammo, Tape and Reel)



NOTES: UNLESS OTHERWISE SPECIFIED

- A. DRAWING CONFORMS TO JEDEC MS-013, VARIATION AC.
- B. ALL DIMENSIONS ARE IN MILLIMETERS.
- C. DRAWING CONFORMS TO ASME Y14.5M-2009.
- D. DRAWING FILENAME: MKT-ZA03FREV3.
- E. FAIRCHILD SEMICONDUCTOR.

Figure 2. 3-LEAD, TO-92, MOLDED 0.200 IN LINE SPACING LEAD FORM (J61Z OPTION)

Package drawings are provided as a service to customers considering Fairchild components. Drawings may change in any manner without notice. Please note the revision and/or date on the drawing and contact a Fairchild Semiconductor representative to verify or obtain the most recent revision. Package specifications do not expand the terms of Fairchild's worldwide terms and conditions, specifically the warranty therein, which covers Fairchild products.






Always visit Fairchild Semiconductor's online packaging area for the most recent package drawings:
<http://www.fairchildsemi.com/dwg/ZA/ZA03F.pdf>

For current tape and reel specifications, visit Fairchild Semiconductor's online packaging area:
http://www.fairchildsemi.com/packing_dwg/PKG-ZA03F_BK.pdf



TRADEMARKS

The following includes registered and unregistered trademarks and service marks, owned by Fairchild Semiconductor and/or its global subsidiaries, and is not intended to be an exhaustive list of all such trademarks.

- | | | | |
|---|--|---|---|
| AccuPower™ | F-PFS™ |  |  |
| AX-CAP®* | FRFET® | PowerTrench® | TinyBoost® |
| BitSiC™ | Global Power Resource SM | PowerXS™ | TinyBuck® |
| Build it Now™ | GreenBridge™ | Programmable Active Droop™ | TinyCalc™ |
| CorePLUS™ | Green FPS™ | QFET® | TinyLogic® |
| CorePOWER™ | Green FPS™ e-Series™ | QS™ | TINYOPTO™ |
| CROSSVOLT™ | Gmax™ | Quiet Series™ | TinyPower™ |
| CTL™ | GTO™ | RapidConfigure™ | TinyPWM™ |
| Current Transfer Logic™ | IntelliMAX™ |  | TinyWire™ |
| DEUXPEED® | ISOPLANAR™ | Saving our world, 1mW/W/kW at a time™ | TranSiC™ |
| Dual Cool™ | Making Small Speakers Sound Louder and Better™ | SignalWise™ | TriFault Detect™ |
| EcoSPARK® | MegaBuck™ | SmartMax™ | TRUECURRENT®* |
| EfficientMax™ | MICROCOUPLER™ | SMART START™ | μSerDes™ |
| ESBC™ | MicroFET™ | Solutions for Your Success™ |  |
|  | MicroPak™ | SPM® | UHC® |
| Fairchild® | MicroPak2™ | STEALTH™ | Ultra FRFET™ |
| Fairchild Semiconductor® | MillerDrive™ | SuperFET® | UniFET™ |
| FACT Quiet Series™ | MotionMax™ | SuperSOT™-3 | VCX™ |
| FACT® | mWSaver® | SuperSOT™-6 | VisualMax™ |
| FAST® | OptoHiT™ | SuperSOT™-8 | VoltagePlus™ |
| FastvCore™ | OPTOLOGIC® | SupreMOS® | XS™ |
| FETBench™ | OPTOPLANAR® | SyncFET™ | 仙童™ |
| FPS™ | | Sync-Lock™ | |

* Trademarks of System General Corporation, used under license by Fairchild Semiconductor.

DISCLAIMER

FAIRCHILD SEMICONDUCTOR RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO ANY PRODUCTS HEREIN TO IMPROVE RELIABILITY, FUNCTION, OR DESIGN. FAIRCHILD DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN; NEITHER DOES IT CONVEY ANY LICENSE UNDER ITS PATENT RIGHTS, NOR THE RIGHTS OF OTHERS. THESE SPECIFICATIONS DO NOT EXPAND THE TERMS OF FAIRCHILD'S WORLDWIDE TERMS AND CONDITIONS, SPECIFICALLY THE WARRANTY THEREIN, WHICH COVERS THESE PRODUCTS.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF FAIRCHILD SEMICONDUCTOR CORPORATION.

As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury of the user.
2. A critical component in any component of a life support, device, or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

ANTI-COUNTERFEITING POLICY

Fairchild Semiconductor Corporation's Anti-Counterfeiting Policy. Fairchild's Anti-Counterfeiting Policy is also stated on our external website, www.fairchildsemi.com, under Sales Support.

Counterfeiting of semiconductor parts is a growing problem in the industry. All manufacturers of semiconductor products are experiencing counterfeiting of their parts. Customers who inadvertently purchase counterfeit parts experience many problems such as loss of brand reputation, substandard performance, failed applications, and increased cost of production and manufacturing delays. Fairchild is taking strong measures to protect ourselves and our customers from the proliferation of counterfeit parts. Fairchild strongly encourages customers to purchase Fairchild parts either directly from Fairchild or from Authorized Fairchild Distributors who are listed by country on our web page cited above. Products customers buy either from Fairchild directly or from Authorized Fairchild Distributors are genuine parts, have full traceability, meet Fairchild's quality standards for handling and storage and provide access to Fairchild's full range of up-to-date technical and product information. Fairchild and our Authorized Distributors will stand behind all warranties and will appropriately address any warranty issues that may arise. Fairchild will not provide any warranty coverage or other assistance for parts bought from Unauthorized Sources. Fairchild is committed to combat this global problem and encourage our customers to do their part in stopping this practice by buying direct or from authorized distributors.

PRODUCT STATUS DEFINITIONS

Definition of Terms

Datasheet Identification	Product Status	Definition
Advance Information	Formative / In Design	Datasheet contains the design specifications for product development. Specifications may change in any manner without notice.
Preliminary	First Production	Datasheet contains preliminary data; supplementary data will be published at a later date. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve design.
No Identification Needed	Full Production	Datasheet contains final specifications. Fairchild Semiconductor reserves the right to make changes at any time without notice to improve the design.
Obsolete	Not In Production	Datasheet contains specifications on a product that is discontinued by Fairchild Semiconductor. The datasheet is for reference information only.

Rev. I68

ANEXO 9. DATASHEET DEL PIC16F887



PIC16F882/883/884/886/887

Data Sheet

28/40/44-Pin, Enhanced Flash-Based 8-Bit
CMOS Microcontrollers with
nanoWatt Technology

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.


AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

© 2007, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

 Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

**QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
== ISO/TS 16949:2002 ==**



PIC16F882/883/884/886/887

28/40/44-Pin Flash-Based, 8-Bit CMOS Microcontrollers with nanoWatt Technology

High-Performance RISC CPU:

- Only 35 instructions to learn:
 - All single-cycle instructions except branches
- Operating speed:
 - DC – 20 MHz oscillator/clock input
 - DC – 200 ns instruction cycle
- Interrupt capability
- 8-level deep hardware stack
- Direct, Indirect and Relative Addressing modes

Special Microcontroller Features:

- Precision Internal Oscillator:
 - Factory calibrated to $\pm 1\%$
 - Software selectable frequency range of 8 MHz to 31 kHz
 - Software tunable
 - Two-Speed Start-up mode
 - Crystal fail detect for critical applications
 - Clock mode switching during operation for power savings
- Power-Saving Sleep mode
- Wide operating voltage range (2.0V-5.5V)
- Industrial and Extended Temperature range
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Brown-out Reset (BOR) with software control option
- Enhanced low-current Watchdog Timer (WDT) with on-chip oscillator (software selectable nominal 268 seconds with full prescaler) with software enable
- Multiplexed Master Clear with pull-up/input pin
- Programmable code protection
- High Endurance Flash/EEPROM cell:
 - 100,000 write Flash endurance
 - 1,000,000 write EEPROM endurance
 - Flash/Data EEPROM retention: > 40 years
- Program memory Read/Write during run time
- In-Circuit Debugger (on board)

Low-Power Features:

- Standby Current:
 - 50 nA @ 2.0V, typical
- Operating Current:
 - 11 μ A @ 32 kHz, 2.0V, typical
 - 220 μ A @ 4 MHz, 2.0V, typical
- Watchdog Timer Current:
 - 1 μ A @ 2.0V, typical

Peripheral Features:

- 24/35 I/O pins with individual direction control:
 - High current source/sink for direct LED drive
 - Interrupt-on-Change pin
 - Individually programmable weak pull-ups
 - Ultra Low-Power Wake-up (ULPWU)
- Analog Comparator module with:
 - Two analog comparators
 - Programmable on-chip voltage reference (CVREF) module (% of VDD)
 - Fixed voltage reference (0.6V)
 - Comparator inputs and outputs externally accessible
 - SR Latch mode
 - External Timer1 Gate (count enable)
- A/D Converter:
 - 10-bit resolution and 11/14 channels
- Timer0: 8-bit timer/counter with 8-bit programmable prescaler
- Enhanced Timer1:
 - 16-bit timer/counter with prescaler
 - External Gate Input mode
 - Dedicated low-power 32 kHz oscillator
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Enhanced Capture, Compare, PWM+ module:
 - 16-bit Capture, max. resolution 12.5 ns
 - Compare, max. resolution 200 ns
 - 10-bit PWM with 1, 2 or 4 output channels, programmable "dead time", max. frequency 20 kHz
 - PWM output steering control
- Capture, Compare, PWM module:
 - 16-bit Capture, max. resolution 12.5 ns
 - 16-bit Compare, max. resolution 200 ns
 - 10-bit PWM, max. frequency 20 kHz
- Enhanced USART module:
 - Supports RS-485, RS-232, and LIN 2.0
 - Auto-Baud Detect
 - Auto-Wake-Up on Start bit
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Master Synchronous Serial Port (MSSP) module supporting 3-wire SPI (all 4 modes) and I²C™ Master and Slave Modes with I²C address mask

PIC16F882/883/884/886/887

Device	Program Memory	Data Memory		I/O	10-bit A/D (ch)	ECCP/ CCP	EUSART	MSSP	Comparators	Timers 8/16-bit
	Flash (words)	SRAM (bytes)	EEPROM (bytes)							
PIC16F882	2048	128	128	28	11	1/1	1	1	2	2/1
PIC16F883	4096	256	256	24	11	1/1	1	1	2	2/1
PIC16F884	4096	256	256	35	14	1/1	1	1	2	2/1
PIC16F886	8192	368	256	24	11	1/1	1	1	2	2/1
PIC16F887	8192	368	256	35	14	1/1	1	1	2	2/1

PIC16F882/883/884/886/887

Pin Diagrams – PIC16F882/883/886, 28-Pin PDIP, SOIC, SSOP

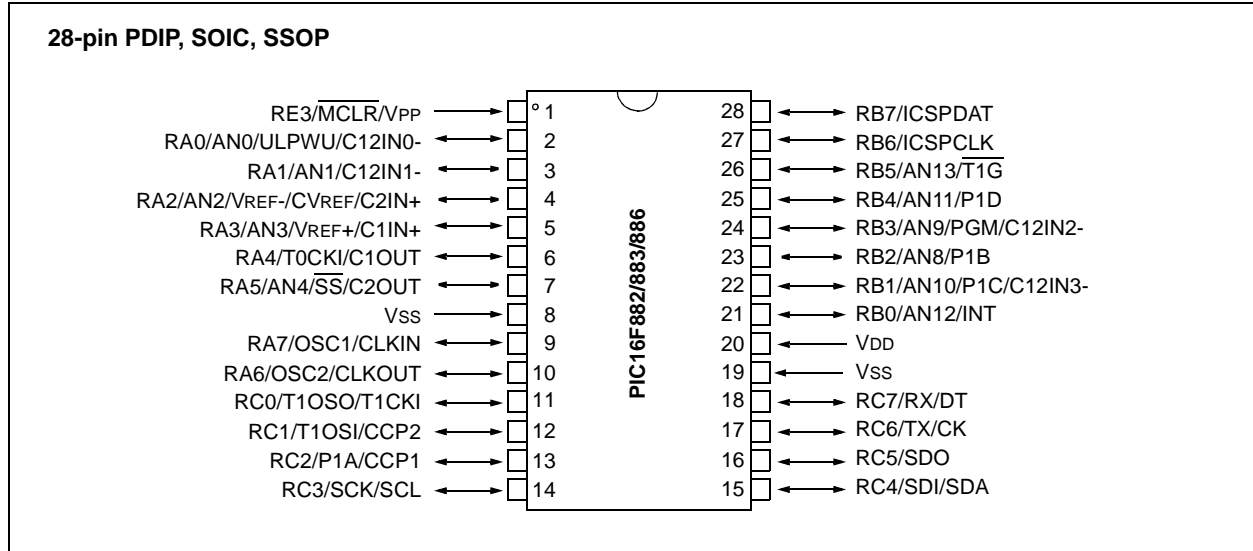


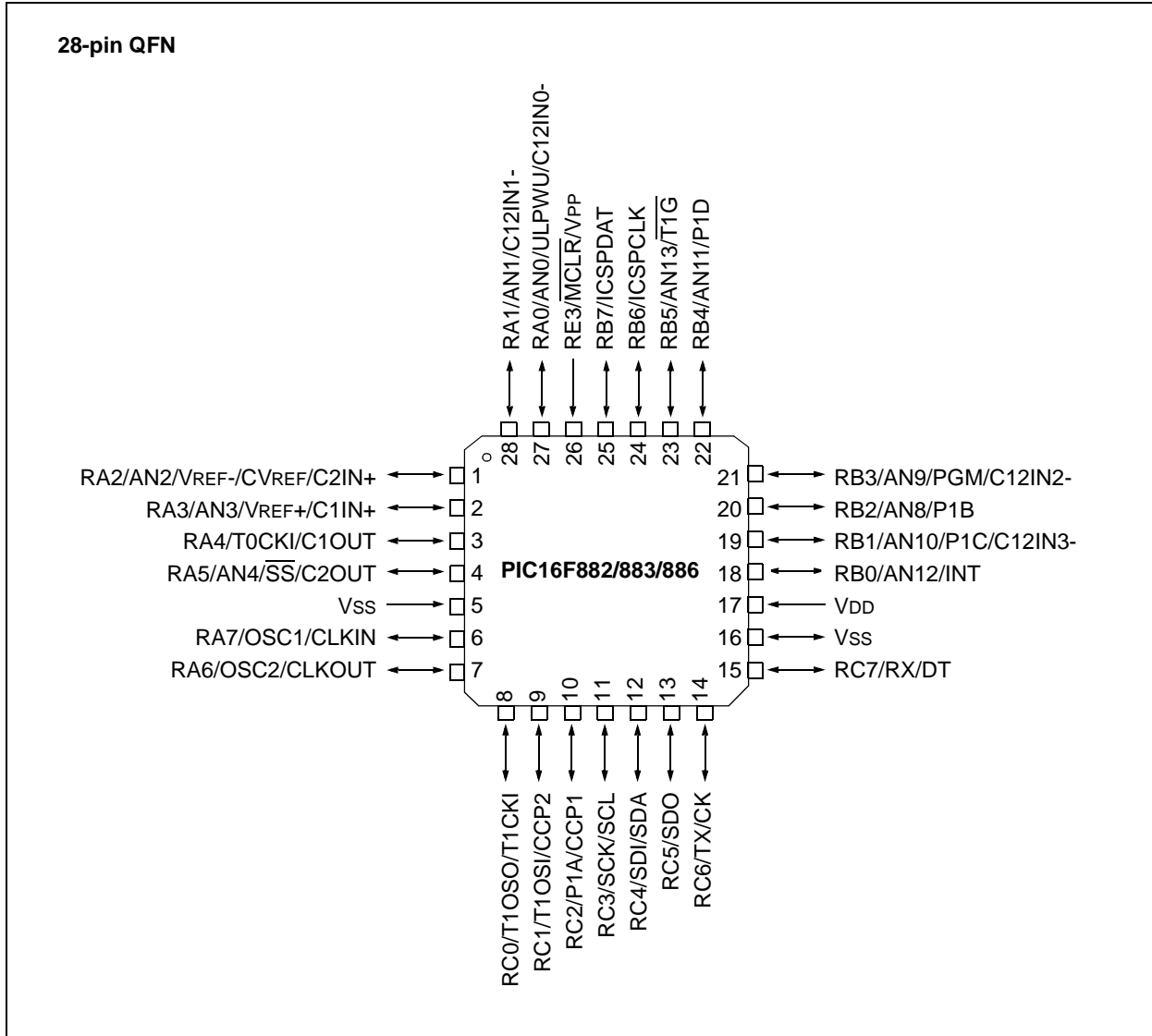
TABLE 1: PIC16F882/883/886 28-PIN SUMMARY (PDIP, SOIC, SSOP)

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	10	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	9	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	21	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	22	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	23	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	24	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	25	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	26	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	27	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	28	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	11	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	12	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	13	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	14	—	—	—	—	—	SCK/SCL	—	—	—
RC4	15	—	—	—	—	—	SDI/SDA	—	—	—
RC5	16	—	—	—	—	—	SDO	—	—	—
RC6	17	—	—	—	—	TX/CK	—	—	—	—
RC7	18	—	—	—	—	RX/DT	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	20	—	—	—	—	—	—	—	—	VDD
—	8	—	—	—	—	—	—	—	—	VSS
—	19	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.

PIC16F882/883/884/886/887

Pin Diagrams – PIC16F882/883/886, 28-Pin QFN



PIC16F882/883/884/886/887

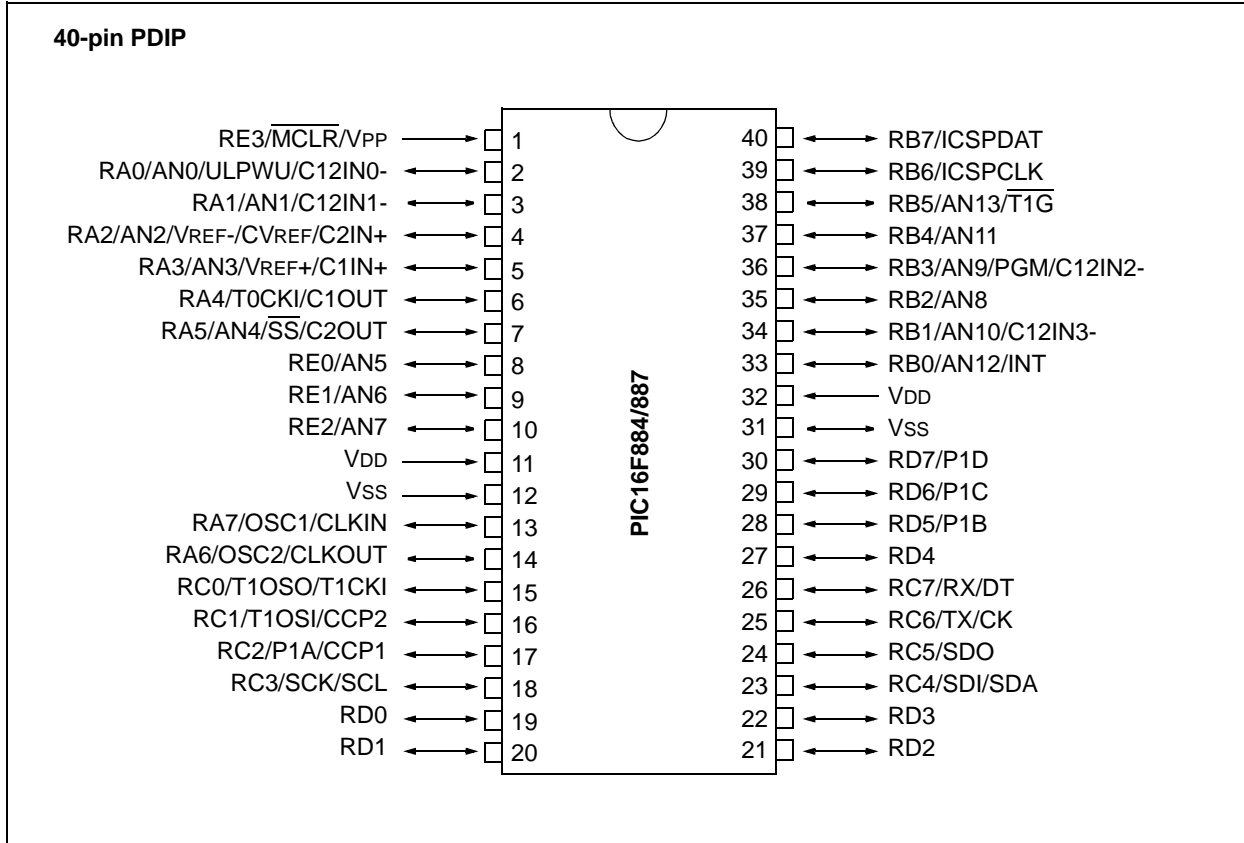
TABLE 2: PIC16F882/883/886 28-PIN SUMMARY (QFN)

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	27	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	28	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	1	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	2	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	3	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	4	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	7	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	6	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	18	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	19	AN10	C12IN3-	—	P1C	—	—	IOC	Y	—
RB2	20	AN8	—	—	P1B	—	—	IOC	Y	—
RB3	21	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	22	AN11	—	—	P1D	—	—	IOC	Y	—
RB5	23	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	24	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	25	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	8	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	9	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	10	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	11	—	—	—	—	—	SCK/SCL	—	—	—
RC4	12	—	—	—	—	—	SDI/SDA	—	—	—
RC5	13	—	—	—	—	—	SDO	—	—	—
RC6	14	—	—	—	—	TX/CK	—	—	—	—
RC7	15	—	—	—	—	RX/DT	—	—	—	—
RE3	26	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	17	—	—	—	—	—	—	—	—	VDD
—	5	—	—	—	—	—	—	—	—	VSS
—	16	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.

PIC16F882/883/884/886/887

Pin Diagrams – PIC16F884/887, 40-Pin PDIP



PIC16F882/883/884/886/887

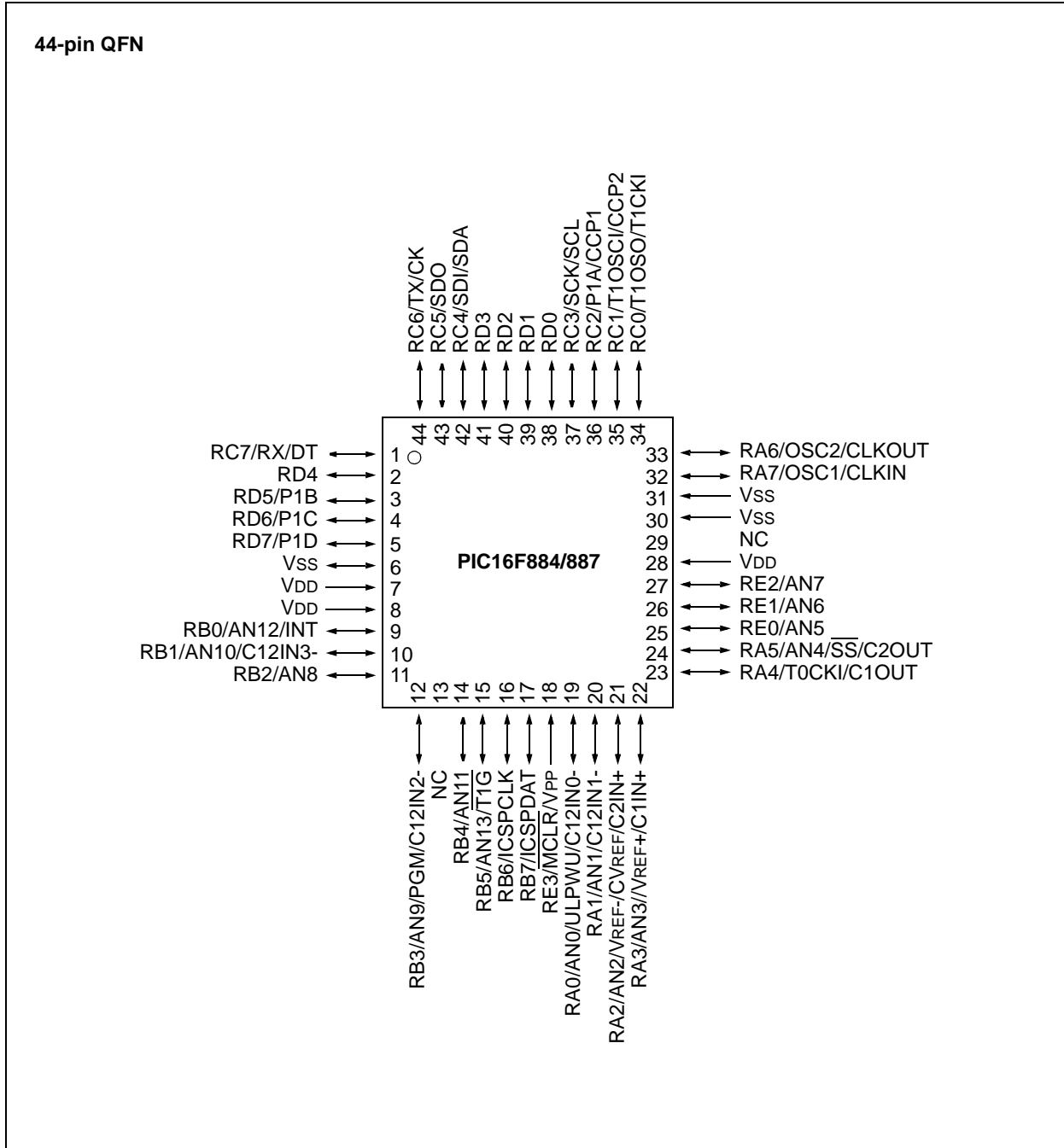
TABLE 3: PIC16F884/887 40-PIN SUMMARY (PDIP)

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	MSSP	Interrupt	Pull-up	Basic
RA0	2	AN0/ULPWU	C12IN0-	—	—	—	—	—	—	—
RA1	3	AN1	C12IN1-	—	—	—	—	—	—	—
RA2	4	AN2	C2IN+	—	—	—	—	—	—	VREF-/CVREF
RA3	5	AN3	C1IN+	—	—	—	—	—	—	VREF+
RA4	6	—	C1OUT	T0CKI	—	—	—	—	—	—
RA5	7	AN4	C2OUT	—	—	—	SS	—	—	—
RA6	14	—	—	—	—	—	—	—	—	OSC2/CLKOUT
RA7	13	—	—	—	—	—	—	—	—	OSC1/CLKIN
RB0	33	AN12	—	—	—	—	—	IOC/INT	Y	—
RB1	34	AN10	C12IN3-	—	—	—	—	IOC	Y	—
RB2	35	AN8	—	—	—	—	—	IOC	Y	—
RB3	36	AN9	C12IN2-	—	—	—	—	IOC	Y	PGM
RB4	37	AN11	—	—	—	—	—	IOC	Y	—
RB5	38	AN13	—	T1G	—	—	—	IOC	Y	—
RB6	39	—	—	—	—	—	—	IOC	Y	ICSPCLK
RB7	40	—	—	—	—	—	—	IOC	Y	ICSPDAT
RC0	15	—	—	T1OSO/T1CKI	—	—	—	—	—	—
RC1	16	—	—	T1OSI	CCP2	—	—	—	—	—
RC2	17	—	—	—	CCP1/P1A	—	—	—	—	—
RC3	18	—	—	—	—	—	SCK/SCL	—	—	—
RC4	23	—	—	—	—	—	SDI/SDA	—	—	—
RC5	24	—	—	—	—	—	SDO	—	—	—
RC6	25	—	—	—	—	TX/CK	—	—	—	—
RC7	26	—	—	—	—	RX/DT	—	—	—	—
RD0	19	—	—	—	—	—	—	—	—	—
RD1	20	—	—	—	—	—	—	—	—	—
RD2	21	—	—	—	—	—	—	—	—	—
RD3	22	—	—	—	—	—	—	—	—	—
RD4	27	—	—	—	—	—	—	—	—	—
RD5	28	—	—	—	P1B	—	—	—	—	—
RD6	29	—	—	—	P1C	—	—	—	—	—
RD7	30	—	—	—	P1D	—	—	—	—	—
RE0	8	AN5	—	—	—	—	—	—	—	—
RE1	9	AN6	—	—	—	—	—	—	—	—
RE2	10	AN7	—	—	—	—	—	—	—	—
RE3	1	—	—	—	—	—	—	—	Y ⁽¹⁾	MCLR/VPP
—	11	—	—	—	—	—	—	—	—	VDD
—	32	—	—	—	—	—	—	—	—	VDD
—	12	—	—	—	—	—	—	—	—	VSS
—	31	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.

PIC16F882/883/884/886/887

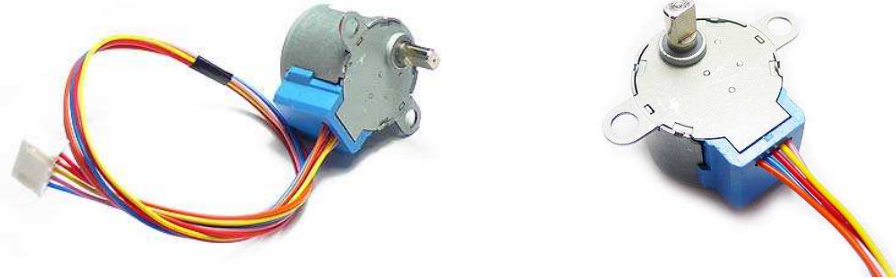
Pin Diagrams – PIC16F884/887, 44-Pin QFN



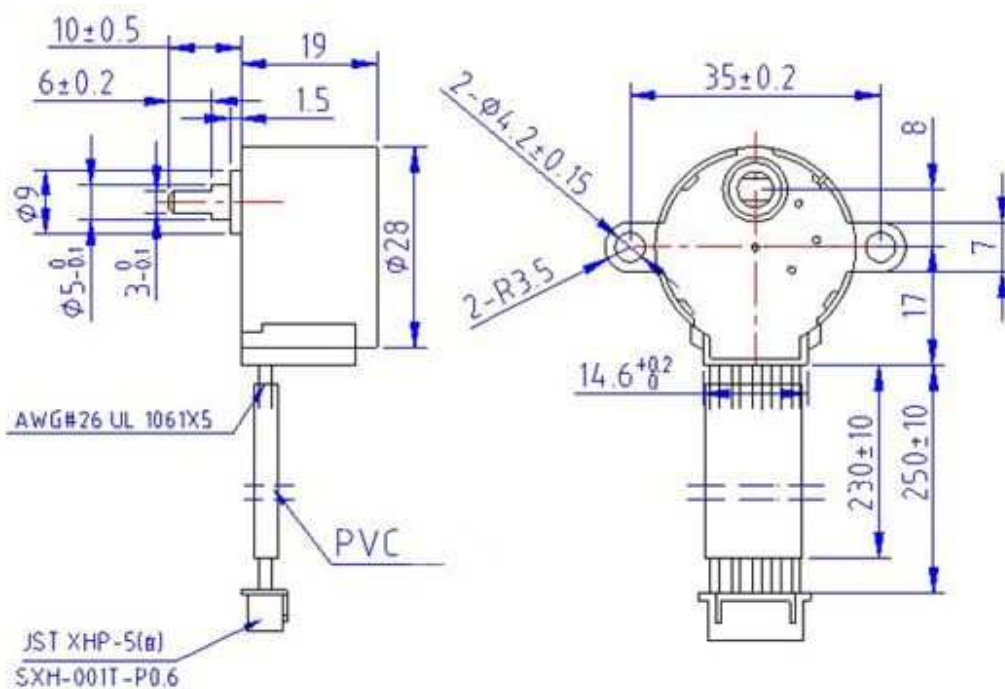
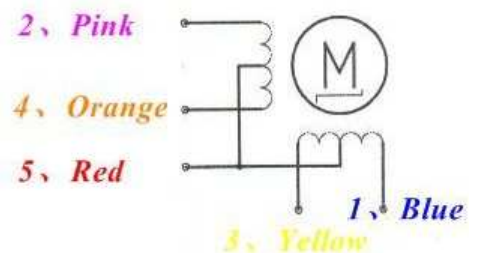
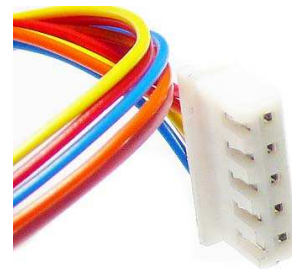
ANEXO 10. DATASHEET DEL INTEGRADO 28BYJ-48

28BYJ-48 – 5V Stepper Motor

The 28BYJ-48 is a small stepper motor suitable for a large range of applications.



Rated voltage :	5VDC
Number of Phase	4
Speed Variation Ratio	1/64
Stride Angle	5.625°/64
Frequency	100Hz
DC resistance	50Ω±7%(25°C)
Idle In-traction Frequency	> 600Hz
Idle Out-traction Frequency	> 1000Hz
In-traction Torque	>34.3mN.m(120Hz)
Self-positioning Torque	>34.3mN.m
Friction torque	600-1200 gf.cm
Pull in torque	300 gf.cm
Insulated resistance	>10MΩ(500V)
Insulated electricity power	600VAC/1mA/1s
Insulation grade	A
Rise in Temperature	<40K(120Hz)
Noise	<35dB(120Hz, No load, 10cm)
Model	28BYJ-48 – 5V



ANEXO 11. DATASHEET DEL ULN2003APG

TOSHIBA Bipolar Digital Integrated Circuit Silicon Monolithic

ULN2003APG,ULN2003AFWG ULN2004APG,ULN2004AFWG

7-ch Darlington Sink Driver

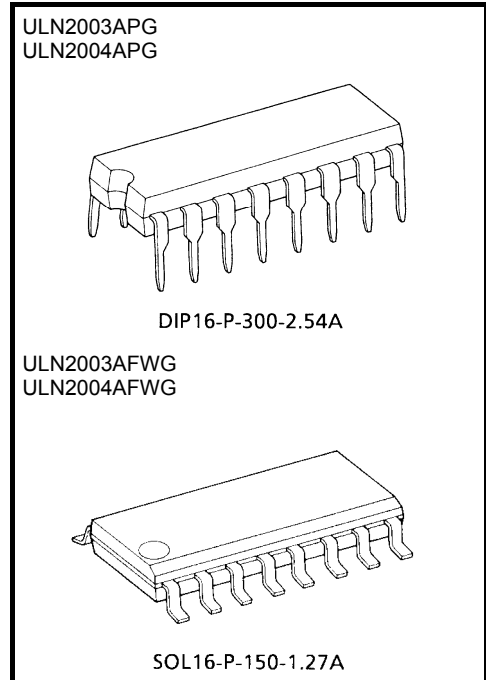
The ULN2003APG/AFWG Series are high-voltage, high-current darlington drivers comprised of seven NPN darlington pairs. All units feature integral clamp diodes for switching inductive loads.

Applications include relay, hammer, lamp and display (LED) drivers.

Features

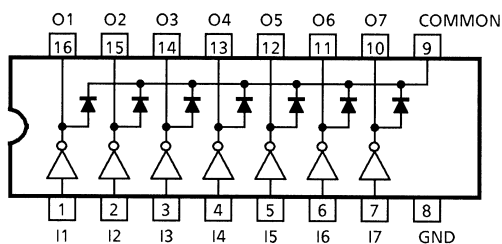
- Output current (single output): 500 mA max
- High sustaining voltage output: 50 V min
- Output clamp diodes
- Inputs compatible with various types of logic
- Package Type-APG: DIP-16pin
- Package Type-AFWG: SOL-16pin

Type	Input Base Resistor	Designation
ULN2003APG/AFWG	2.7 kΩ	TTL, 5 V CMOS
ULN2004APG/AFWG	10.5 kΩ	6 to 15 V PMOS, CMOS



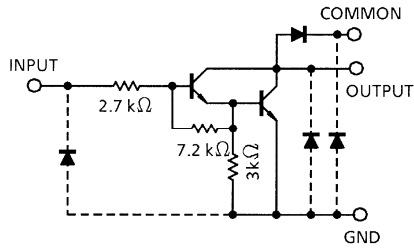
Weight
 DIP16-P-300-2.54A : 1.11 g (typ.)
 SOL16-P-150-1.27A: 0.15 g (typ.)

Pin Connection (top view)

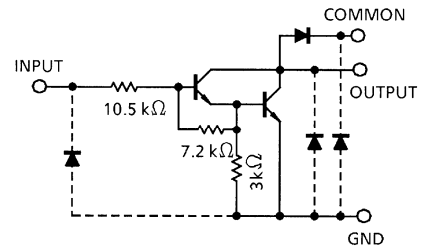


Schematics (each driver)

ULN2003APG/AFWG



ULN2004APG/AFWG



Note: The input and output parasitic diodes cannot be used as clamp diodes.

Absolute Maximum Ratings (Ta = 25°C)

Characteristic	Symbol	Rating	Unit
Output sustaining voltage	$V_{CE(SUS)}$	-0.5 to 50	V
Output current	I_{OUT}	500	mA/ch
Input voltage	V_{IN}	-0.5 to 30	V
Clamp diode reverse voltage	V_R	50	V
Clamp diode forward current	I_F	500	mA
Power dissipation	APG	1.47	W
	AFWG	1.25 (Note)	
Operating temperature	T_{opr}	-40 to 85	°C
Storage temperature	T_{stg}	-55 to 150	°C

Note: On PCB (Test Board: JEDEC 2s2p)

Recommended Operating Conditions (Ta = -40 to 85°C)

Characteristic		Symbol	Test Condition	Min	Typ.	Max	Unit	
Output sustaining voltage		V _{CE (SUS)}	—	0	—	50	V	
Output current	APG	I _{OUT}	t _{pw} = 25 ms 7 Circuits Ta = 85°C Tj = 120°C	Duty = 10%	0	—	350	mA/ch
				Duty = 50%	0	—	100	
	AFWG			Duty = 10%	0	—	300	
				Duty = 50%	0	—	90	
Input voltage		V _{IN}	—	0	—	24	V	
Input voltage (output on)	ULN2003A	V _{IN (ON)}	I _{OUT} = 400 mA h _{FE} = 800	2.8	—	24	V	
	ULN2004A			6.2	—	24		
Input voltage (output off)	ULN2003A	V _{IN (OFF)}	—	0	—	0.7	V	
	ULN2004A		—	0	—	1.0		
Clamp diode reverse voltage		V _R	—	—	—	50	V	
Clamp diode forward current		I _F	—	—	—	350	mA	
Power dissipation	APG	P _D	Ta = 85°C	—	—	0.76	W	
	AFWG		Ta = 85°C (Note)	—	—	0.65		

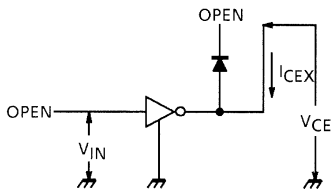
Note: On PCB (Test Board: JEDEC 2s2p)

Electrical Characteristics (Ta = 25°C unless otherwise noted)

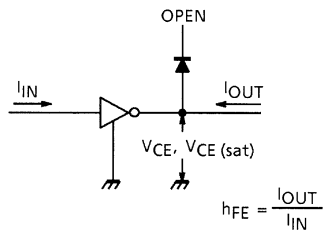
Characteristic	Symbol	Test Circuit	Test Condition	Min	Typ.	Max	Unit		
Output leakage current	I _{CEX}	1	V _{CE} = 50 V, Ta = 25°C	—	—	50	μA		
			V _{CE} = 50 V, Ta = 85°C	—	—	100			
Collector-emitter saturation voltage	V _{CE (sat)}	2	I _{OUT} = 350 mA, I _{IN} = 500 μA	—	1.3	1.6	V		
			I _{OUT} = 200 mA, I _{IN} = 350 μA	—	1.1	1.3			
			I _{OUT} = 100 mA, I _{IN} = 250 μA	—	0.9	1.1			
DC Current transfer ratio	h _{FE}	2	V _{CE} = 2 V, I _{OUT} = 350 mA	1000	—	—	—		
Input current (output on)	ULN2003A	I _{IN (ON)}	3	V _{IN} = 2.4 V, I _{OUT} = 350 mA	—	0.4	0.7	mA	
	ULN2004A								V _{IN} = 9.5 V, I _{OUT} = 350 mA
Input current (output off)	I _{IN (OFF)}	4	I _{OUT} = 500 μA, Ta = 85°C	50	65	—	μA		
Input voltage (output on)	ULN2003A	V _{IN (ON)}	5	V _{CE} = 2 V h _{FE} = 800	I _{OUT} = 350 mA	—	—	2.6	V
					I _{OUT} = 200 mA	—	—	2.0	
	ULN2004A				I _{OUT} = 350 mA	—	—	4.7	
					I _{OUT} = 200 mA	—	—	4.4	
Clamp diode reverse current	I _R	6	V _R = 50 V, Ta = 25°C	—	—	50	μA		
			V _R = 50 V, Ta = 85°C	—	—	100			
Clamp diode forward voltage	V _F	7	I _F = 350 mA	—	—	2.0	V		
Input capacitance	C _{IN}	—	—	—	15	—	pF		
Turn-on delay	t _{ON}	8	V _{OUT} = 50 V, R _L = 125 Ω C _L = 15 pF	—	0.1	—	μs		
Turn-off delay	t _{OFF}	8	V _{OUT} = 50 V, R _L = 125 Ω C _L = 15 pF	—	0.2	—			

Test Circuit

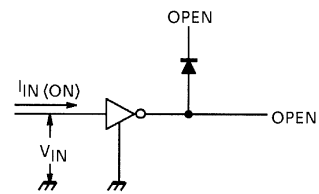
1. I_{CEX}



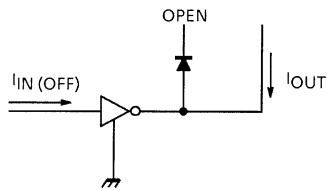
2. $V_{CE(sat)}$, h_{FE}



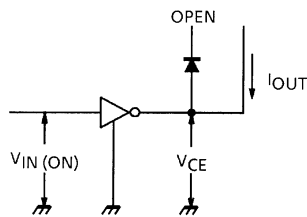
3. $I_{IN(ON)}$



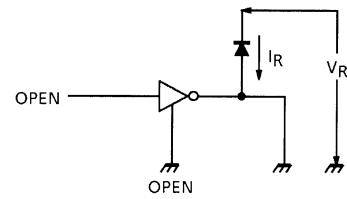
4. $I_{IN(OFF)}$



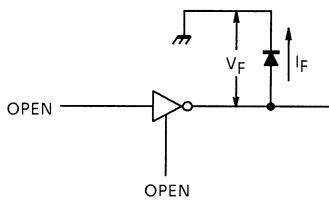
5. $V_{IN(ON)}$



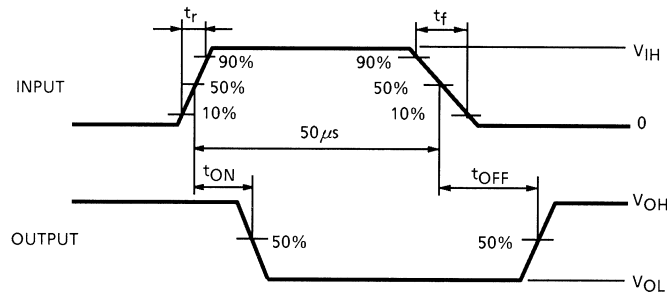
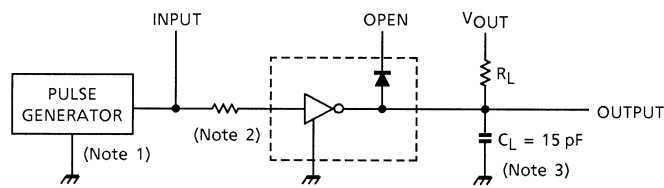
6. I_R



7. V_F



8. t_{ON} , t_{OFF}



Note 1: Pulse width 50 μ s, duty cycle 10%
Output impedance 50 Ω , $t_r \leq 5$ ns, $t_f \leq 10$ ns

Note 2: See below

Input Condition

Type Number	R1	V_{IH}
ULN2003A	0	3 V
ULN2004A	0	8 V

Note 3: C_L includes probe and jig capacitance.

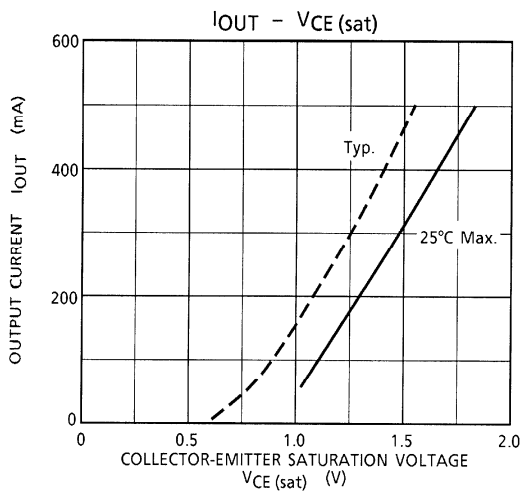
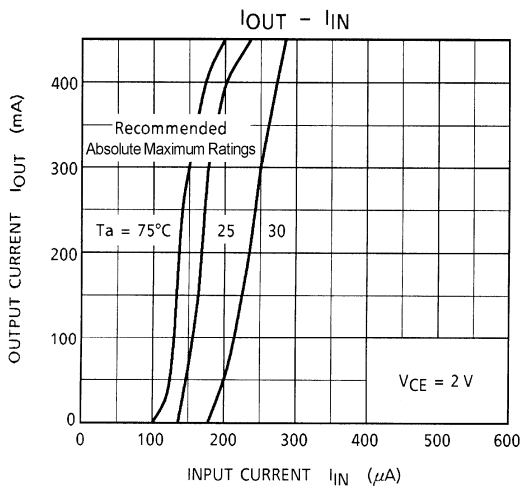
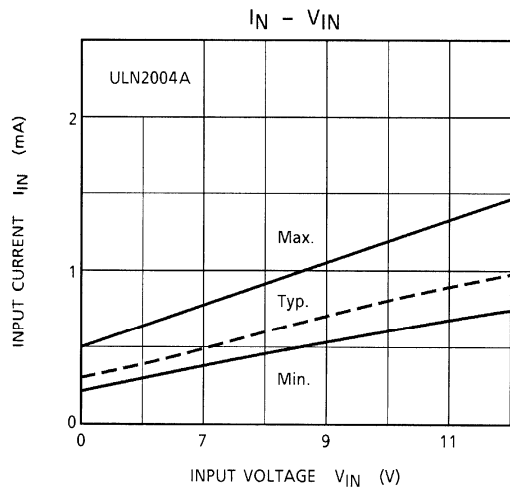
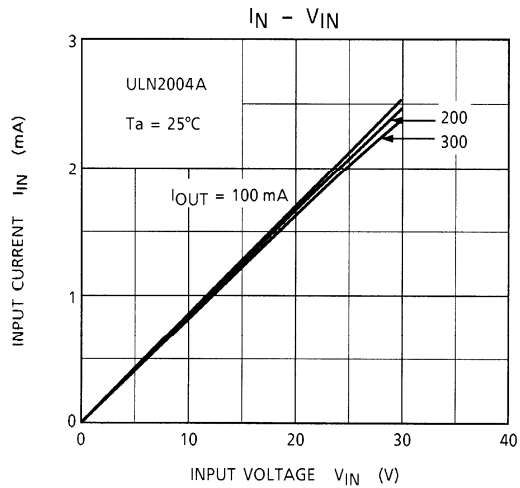
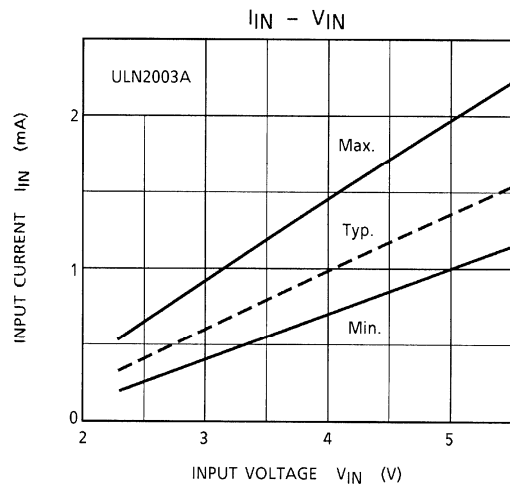
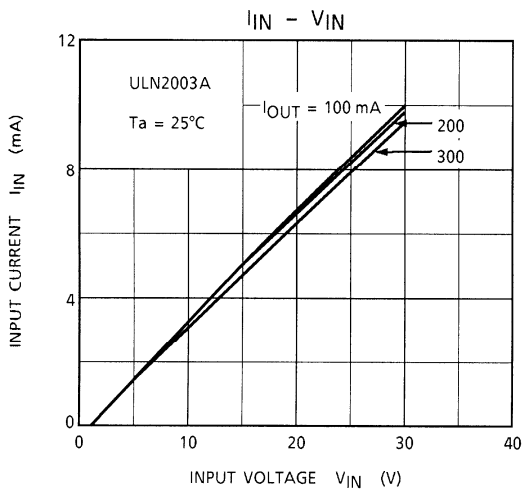
Precautions for Using

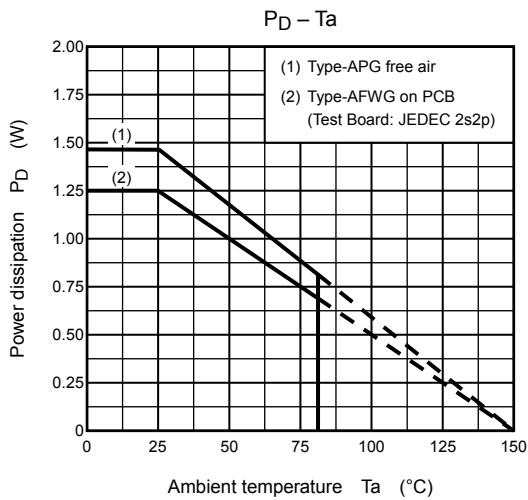
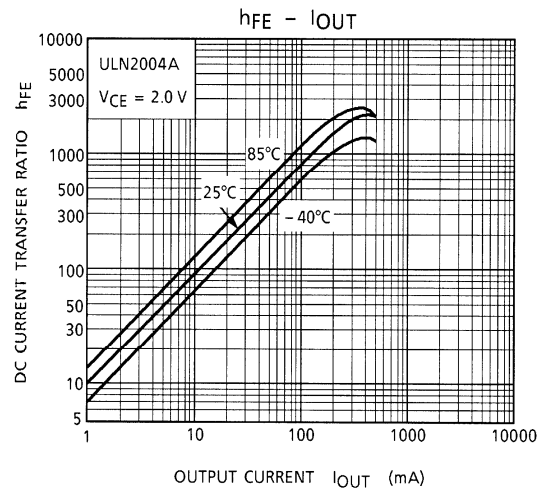
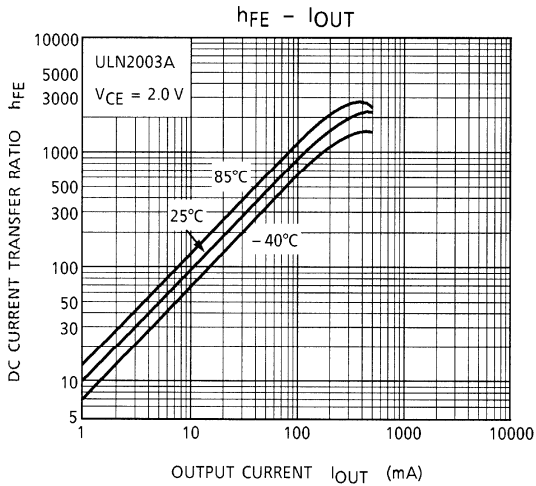
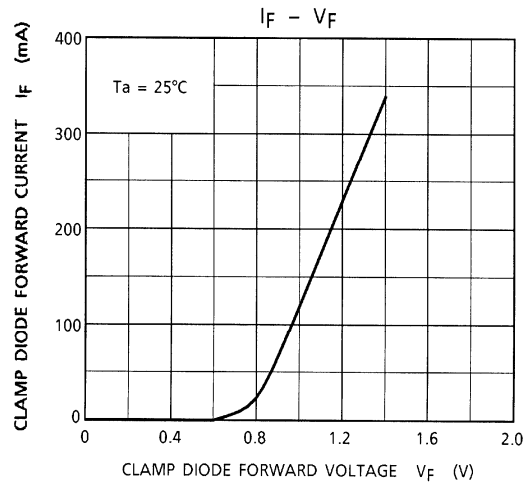
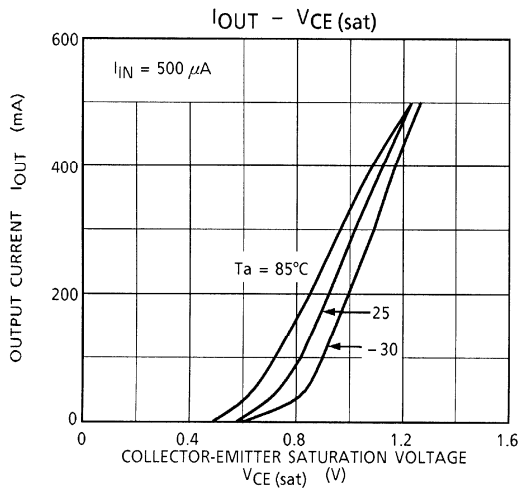
This IC does not include built-in protection circuits for excess current or overvoltage.

If this IC is subjected to excess current or overvoltage, it may be destroyed.

Hence, the utmost care must be taken when systems which incorporate this IC are designed.

Utmost care is necessary in the design of the output line, COMMON and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

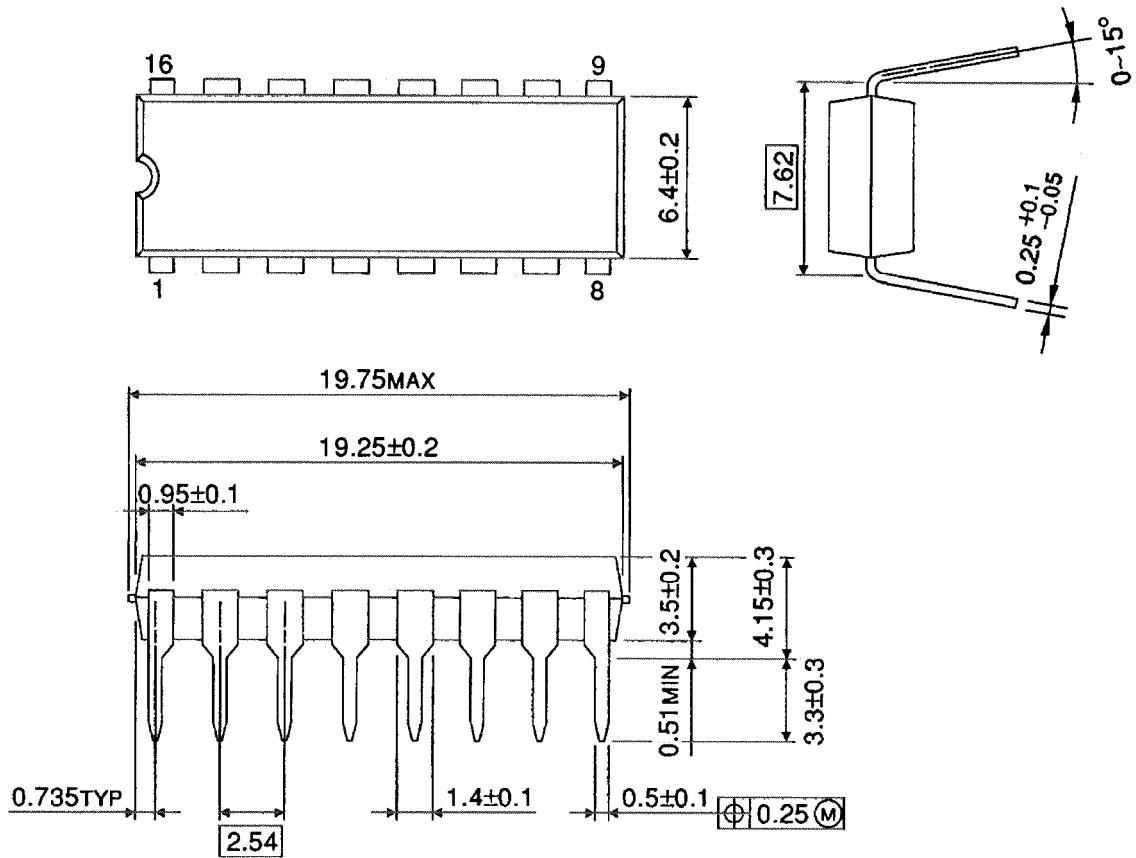




Package Dimensions

DIP16-P-300-2.54A

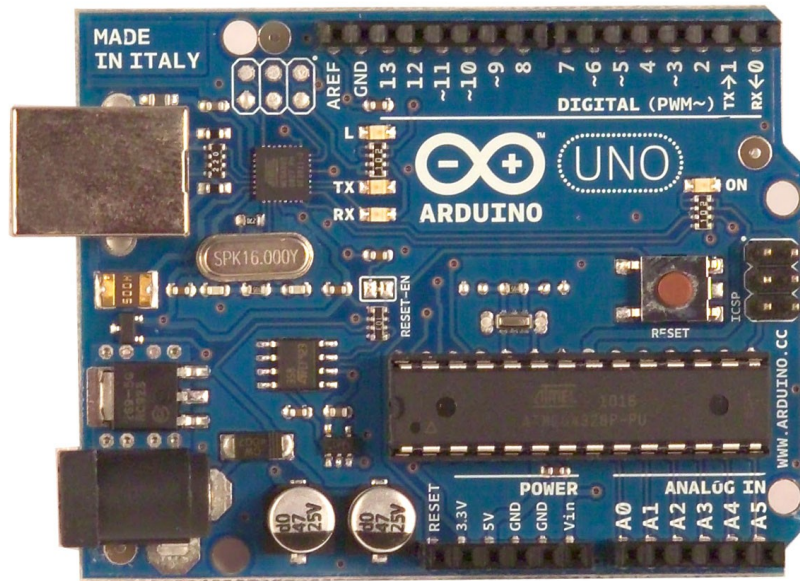
Unit : mm



Weight: 1.11 g (typ.)

ANEXO 12. DATASHEET DEL ARDUINO UNO

Arduino UNO



Product Overview

The Arduino Uno is a microcontroller board based on the ATmega328 ([datasheet](#)). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the [index of Arduino boards](#).

Index

Technical Specifications

Page 2

How to use Arduino
Programming Environment, Basic Tutorials

Page 6

Terms & Conditions

Page 7

Environmental Policies
half sqm of green via Impatto Zero®

Page 7



radiospares

RADIONICS



Technical Specification

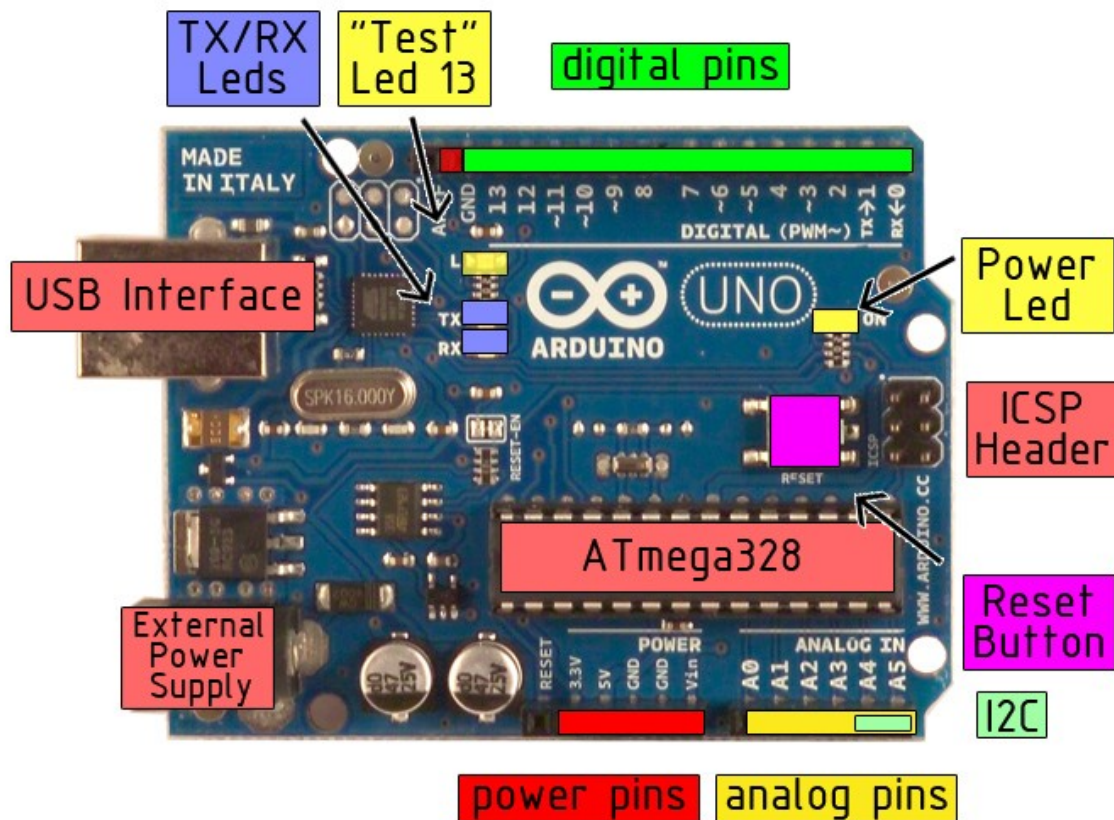


EAGLE files: [arduino-duemilanove-uno-design.zip](#) Schematic: [arduino-uno-schematic.pdf](#)

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

the board



radiospares

RADIONICS



Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0,5 KB is used for the bootloader); It has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip .
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.
- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.



radiospares

RADIONICS



The Uno has 6 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the [analogReference\(\)](#) function. Additionally, some pins have specialized functionality:

- **I²C: 4 (SDA) and 5 (SCL).** Support I²C (TWI) communication using the [Wire library](#).

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the [mapping between Arduino pins and Atmega328 ports](#).

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, an *.inf file is required..

The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Uno's digital pins.

The ATmega328 also support I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation](#) for details. To use the SPI communication, please see the ATmega328 datasheet.

Programming

The Arduino Uno can be programmed with the Arduino software ([download](#)). Select "Arduino Uno w/ ATmega328" from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the [reference](#) and [tutorials](#).

The ATmega328 on the Arduino Uno comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available . The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).



RADIOSPARES

RADIONICS



Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

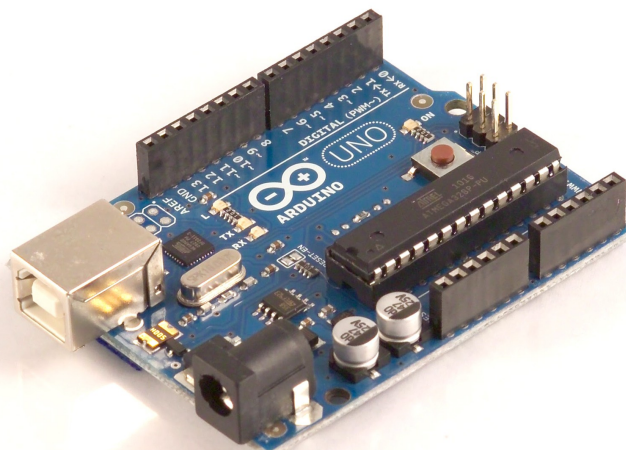
The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



RADIOSPARES

RADIONICS



How to use Arduino



Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the [Arduino programming language](#) (based on [Wiring](#)) and the Arduino development environment (based on [Processing](#)). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).

Arduino is a cross-platform program. You'll have to follow different instructions for your personal OS. Check on the [Arduino site](#) for the latest instructions. <http://arduino.cc/en/Guide/HomePage>

Linux Install

Windows Install

Mac Install

Once you have downloaded/unzipped the arduino IDE, you can Plug the Arduino to your PC via USB cable.

Blink led

Now you're actually ready to "burn" your first program on the arduino board. To select "blink led", the physical translation of the well known programming "hello world", select

**File>Sketchbook>
Arduino-0017>Examples>
Digital>Blink**

Once you have your sketch you'll see something very close to the screenshot on the right.

In **Tools>Board** select

Now you have to go to **Tools>SerialPort** and select the right serial port, the one arduino is attached to.

```
Blink | Arduino 0017
File Edit Sketch Tools Help
Blink $
int ledPin = 13; // LED connected to digital pin 13

// The setup() method runs once, when the sketch starts

void setup() {
  // initialize the digital pin as an output:
  pinMode(ledPin, OUTPUT);
}

// the loop() method runs over and over again,
// as long as the Arduino has power

void loop()
{
  digitalWrite(ledPin, HIGH); // set the LED on
  delay(1000); // wait for a second
  digitalWrite(ledPin, LOW); // set the LED off
  delay(1000); // wait for a second
}
```



Done compiling.

Press Compile button
(to check for errors)



Upload



TX RX Flashing



Blinking Led!

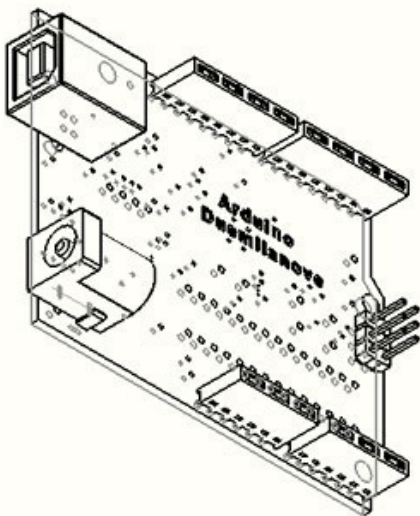
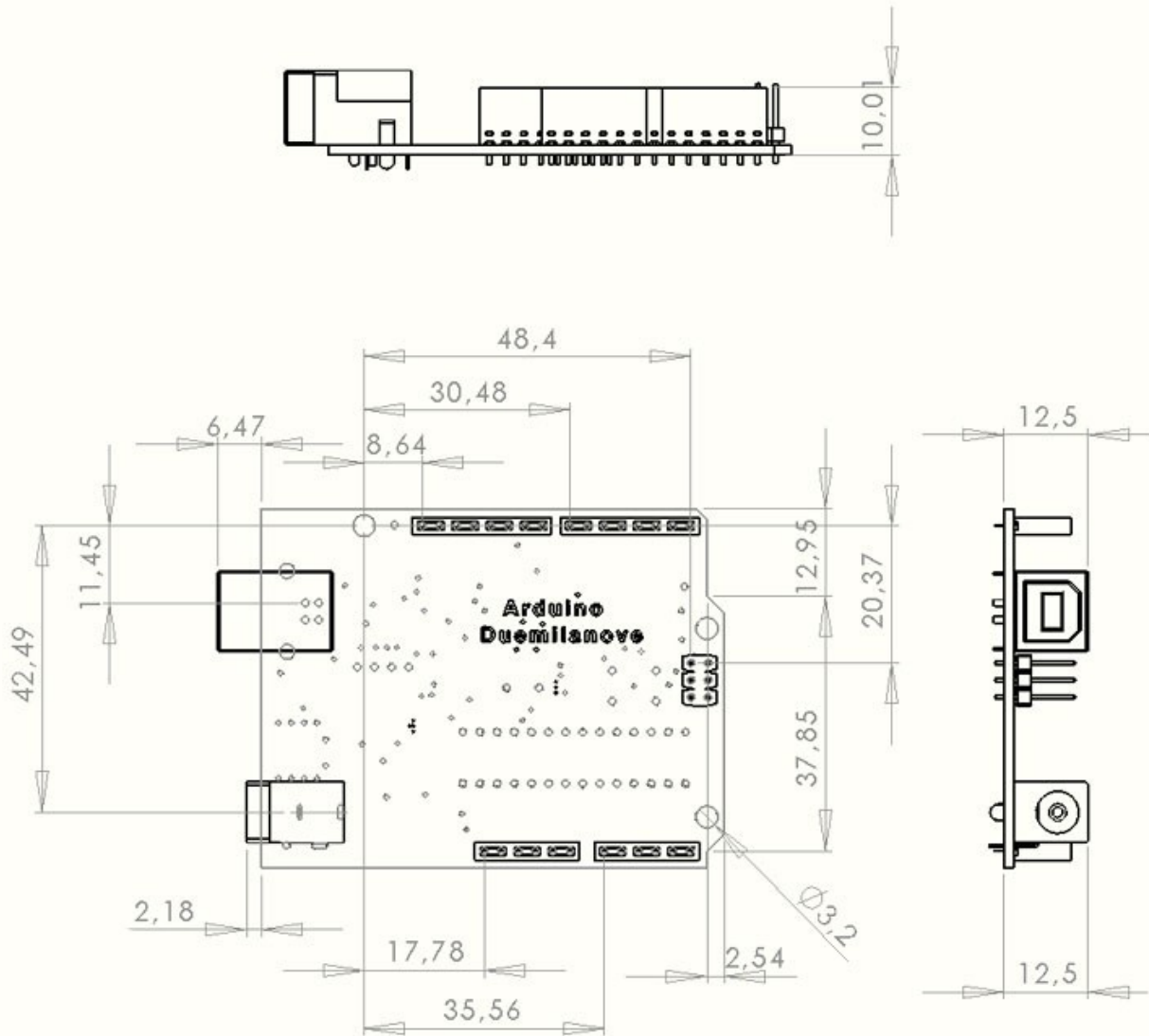


radiospares

RADIONICS



Dimensioned Drawing



radiospares

RADIONICS



Terms & Conditions



1. Warranties

1.1 The producer warrants that its products will conform to the Specifications. This warranty lasts for one (1) years from the date of the sale. The producer shall not be liable for any defects that are caused by neglect, misuse or mistreatment by the Customer, including improper installation or testing, or for any products that have been altered or modified in any way by a Customer. Moreover, The producer shall not be liable for any defects that result from Customer's design, specifications or instructions for such products. Testing and other quality control techniques are used to the extent the producer deems necessary.

1.2 If any products fail to conform to the warranty set forth above, the producer's sole liability shall be to replace such products. The producer's liability shall be limited to products that are determined by the producer not to conform to such warranty. If the producer elects to replace such products, the producer shall have a reasonable time to replacements. Replaced products shall be warranted for a new full warranty period.

1.3 EXCEPT AS SET FORTH ABOVE, PRODUCTS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS." THE PRODUCER DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE

1.4 Customer agrees that prior to using any systems that include the producer products, Customer will test such systems and the functionality of the products as used in such systems. The producer may provide technical, applications or design advice, quality characterization, reliability data or other services. Customer acknowledges and agrees that providing these services shall not expand or otherwise alter the producer's warranties, as set forth above, and no additional obligations or liabilities shall arise from the producer providing such services.

1.5 The Arduino™ products are not authorized for use in safety-critical applications where a failure of the product would reasonably be expected to cause severe personal injury or death. Safety-Critical Applications include, without limitation, life support devices and systems, equipment or systems for the operation of nuclear facilities and weapons systems. Arduino™ products are neither designed nor intended for use in military or aerospace applications or environments and for automotive applications or environment. Customer acknowledges and agrees that any such use of Arduino™ products which is solely at the Customer's risk, and that Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.6 Customer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products and any use of Arduino™ products in Customer's applications, notwithstanding any applications-related information or support that may be provided by the producer.

2. Indemnification

The Customer acknowledges and agrees to defend, indemnify and hold harmless the producer from and against any and all third-party losses, damages, liabilities and expenses it incurs to the extent directly caused by: (i) an actual breach by a Customer of the representation and warranties made under this terms and conditions or (ii) the gross negligence or willful misconduct by the Customer.

3. Consequential Damages Waiver

In no event the producer shall be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether the producer has been advised of the possibility of such damages. This section will survive the termination of the warranty period.

4. Changes to specifications

The producer may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The producer reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.



Environmental Policies



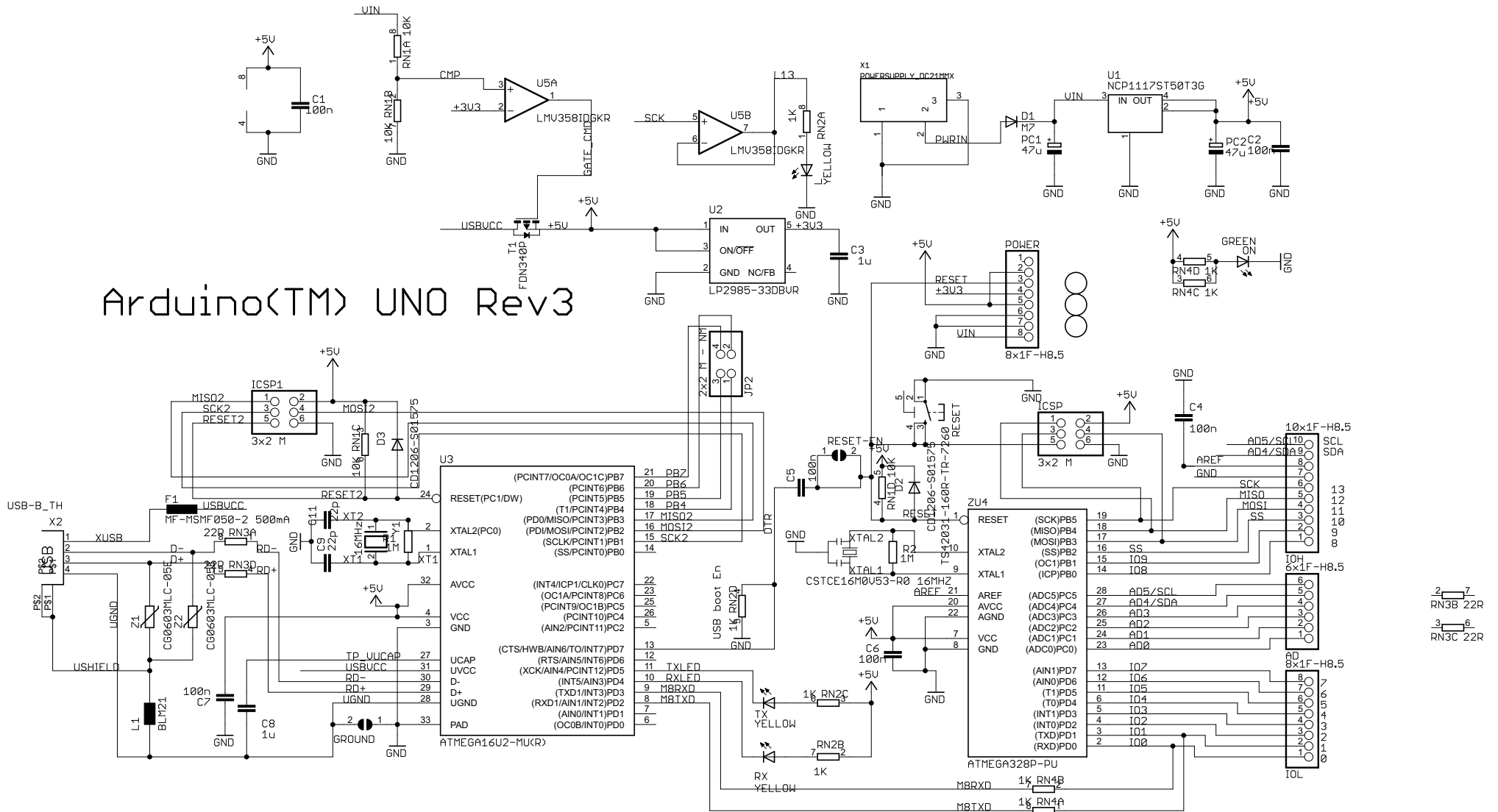
The producer of Arduino™ has joined the Impatto Zero® policy of LifeGate.it. For each Arduino board produced is created / looked after half squared Km of Costa Rica's forest's.



radiospares

RADIONICS





Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information.

ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

ANEXO 13. DATASHEET DEL PN532 NFC/RFID

1. Introduction

This document describes the NFC controller PN532. This document is a short form version; for full specification refer to the product data sheet.

2. General description

The PN532 is a highly integrated transmission module for contactless communication at 13.56 MHz including micro-controller functionality based on an 80C51 core. The transmission module utilises an outstanding modulation and demodulation concept completely integrated for different kinds of passive contactless communication methods and protocols at 13.56 MHz.

The PN532 support 4 different operating modes:

- Reader/writer mode supporting ISO 14443A / MIFARE® and FeliCa™ scheme
- ISO 14443B in reader/writer mode only.
- Card interface mode supporting ISO 14443A / MIFARE® and FeliCa™ scheme
- NFCIP-1 mode

Enabled in reader/ writer mode for ISO reader 14443A / MIFARE® and reader/writer mode for ISO 14443B, the PN532's internal transmitter part is able to drive a reader/writer antenna designed to communicate with ISO14443A /MIFARE® and ISO14443B cards and transponders without additional active circuitry.

The receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO 14443A / MIFARE® and ISO 14443B compatible cards and transponders. The digital part handles the complete ISO14443A framing and error detection (Parity & CRC).

The PN532 supports MIFARE® Classic (e.g. MIFARE® Standard) products. The PN532 supports contactless communication using MIFARE® Higher Baudrates up to 424kBaud in both directions.

Enabled in the reader/ writer mode for FeliCa™, the PN532 transmission module supports the FeliCa™ communication scheme. The receiver part provides a robust and efficient implementation of the demodulation and decoding circuitry for FeliCa™ coded signals. The digital part handles the FeliCa™ framing and error detection like CRC. The PN532 supports contactless communication using FeliCa™ Higher Baudrates up to 424 kbaud in both directions.

Enabled in card mode the PN532 transmission module is able to answer to a reader/writer command either according to FeliCa™ or ISO14443 A / MIFARE® card interface mode. The PN532 generates the digital load-modulated signals and in addition with an external circuit the answers can be send back to the reader/writer. A complete card functionality is only possible in combination with a secure memory IC.

Additionally, the PN532 transmission module offers the possibility to communicate directly to a second NFCIP-1 device in the NFCIP-1 mode. The NFCIP-1 mode offers different communication transfer speeds up to 424 kbit/s according to the ECMA 340 NFCIP-1 Standard. The digital part handles the complete NFCIP-1 framing and error detection. Transfer speeds on the RF interface above 424 kbit/s are supported by the digital part of the PN532 module. The modulation to transmit and the demodulation to receive data at transfer speeds has than to be done by an external circuit.

To make information exchange to the host systems several interfaces are implemented:

- SPI interface
- I²C interface
- Serial UART (similar to RS232 with 0 and PVDD voltage levels)

The PN532 embeds a low dropout voltage regulator allowing the device to be connected directly to a battery as well as a medium power switch to supply and control the power of the companion secure chip.

3. Features

- 80C51 micro controller core with 40 kbyte ROM and 1 kbyte RAM
- Highly integrated analog circuitry to demodulate and decode responses
- Buffered output drivers to connect an antenna with minimum number of external components
- Integrated RF Level detector
- Integrated data mode detector
- Supports ISO 14443A / MIFARE®
Supports ISO 14443B in reader/writer mode only
- Typical operating distance in reader/writer mode for communication to a ISO14443A/MIFARE®, ISO14443B or FeliCa™ card up to 50 mm depending on the antenna size and tuning
- Typical operating distance in NFCIP-1 mode up to 50 mm depending on the antenna size and tuning and power supply
- Typical operating distance in ISO14443A / MIFARE® card or FeliCa™ card interface mode of about 100 mm depending on the antenna size and tuning and the external field strength
- Supports MIFARE® Classic encryption in reader/writer mode and MIFARE® higher transfer speed communication at 212 kbit/s and 424 kbit/s
- Supports contactless communication according to the FeliCa™ scheme at 212 kbaud and 424 kbaud
- Integrated RF interface for NFCIP-1 up to 424 kBaud
- Possibility to communicate on the RF interface above 424 kbaud using external analog circuitry

- Supported host interfaces
 - ◆ SPI interface
 - ◆ I²C interface
 - ◆ High Speed Serial UART (similar to RS232 with 0 and PVDD voltage levels)
- Flexible interrupt using IRQ pin
- Hard reset with low power function
- Power down mode per embedded firmware

Automatic wake up on the I²C, HSU and SPI interfaces when device is in power down mode
- Programmable timer
- Internal oscillator to connect 27.12 MHz crystal
- 2.7 to 5.4V power supply
- Power Switch for external secure companion chip.
- Specific IO ports for external devices control

Embedded test of absence of antenna and/or antenna tuning components by detection of significant load impedance deviation resulting in high power consumption increase.

4. Applications

- Mobile and portable devices
- PC world
- Consumer application

5. Quick reference data

Table 1: Quick reference data

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V _{BAT}	Battery Supply Voltage		2.7		5.4	V
ICVDD	LDO output voltage	VSS = 0V VBAT > 3.3V	[1] 2.7	3.0	3.3	V
PVDD	Supply Voltage for host interface	VSS = 0V PVDD < VBAT	[2] 1.6		3.6	V
SVDD	Supply Voltage for SAM interface	VSS = 0V VBAT > 3.3V (SVDD Switch Enabled)	2.7	3.0	3.3	V
I _{HPD}	Hard Power Down Current	VBAT=5V, RF level detector off			2	μA
I _{SPD}	Soft Power down Current	VBAT=5V, RF level detector on			10	μA
I _{CVDD}	Digital Supply Current	VBAT=5V, RF level detector on, SVDD switch off	[1]	25		mA

Table 1: Quick reference data ...continued

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I _{SVDD}	SVDD Supply Current	V _{BAT} =5V, SVDD switch on			30	mA
I _{AVDD}	Analog Supply Current	I _{VBAT} =5V, RF level detector on	[1]	6		mA
I _{TVDD}	Transmitter (TX) Supply Current	During RF V _{BAT} =5V, 40Ω typical TX Z _{load} (min. tbd)	[1]	60	100	mA
I _{VBAT}	continuous total current consumption	T _{amb} = -30 to +85 °C, 40Ω typical TX Z _{load} (min tbd), excluding the secure companion chip	[3]	91	140	mA
T _{amb}	operating ambient temperature		-30		+85	°C

[1] DVDD, AVDD and TVDD shall always be connected together.

[2] It is not allowed to have PVDD above V_{BAT}

[3] The total current consumption depends also on the firmware version (different internal IC clock speed)

6. Ordering information

Table 2: Ordering information

Type number	Package		
	Name	Description	Version
PN5320A3HN/C101 [1]	HVQFN40	plastic, heatsink very thin quad flat package; no leads; 40terminals; body 6x 6x 0.85mm	SOT618-1
PN5321A3HN/C101 [2]	HVQFN40	plastic, heatsink very thin quad flat package; no leads; 40terminals; body 6x 6x 0.85mm. Type B SW is enable.	SOT618-1

[1] 01 is the reference of the romcode version.

[2] A purchaser of this Philips IC has to take care for appropriate third party patent license.

7. Block diagram

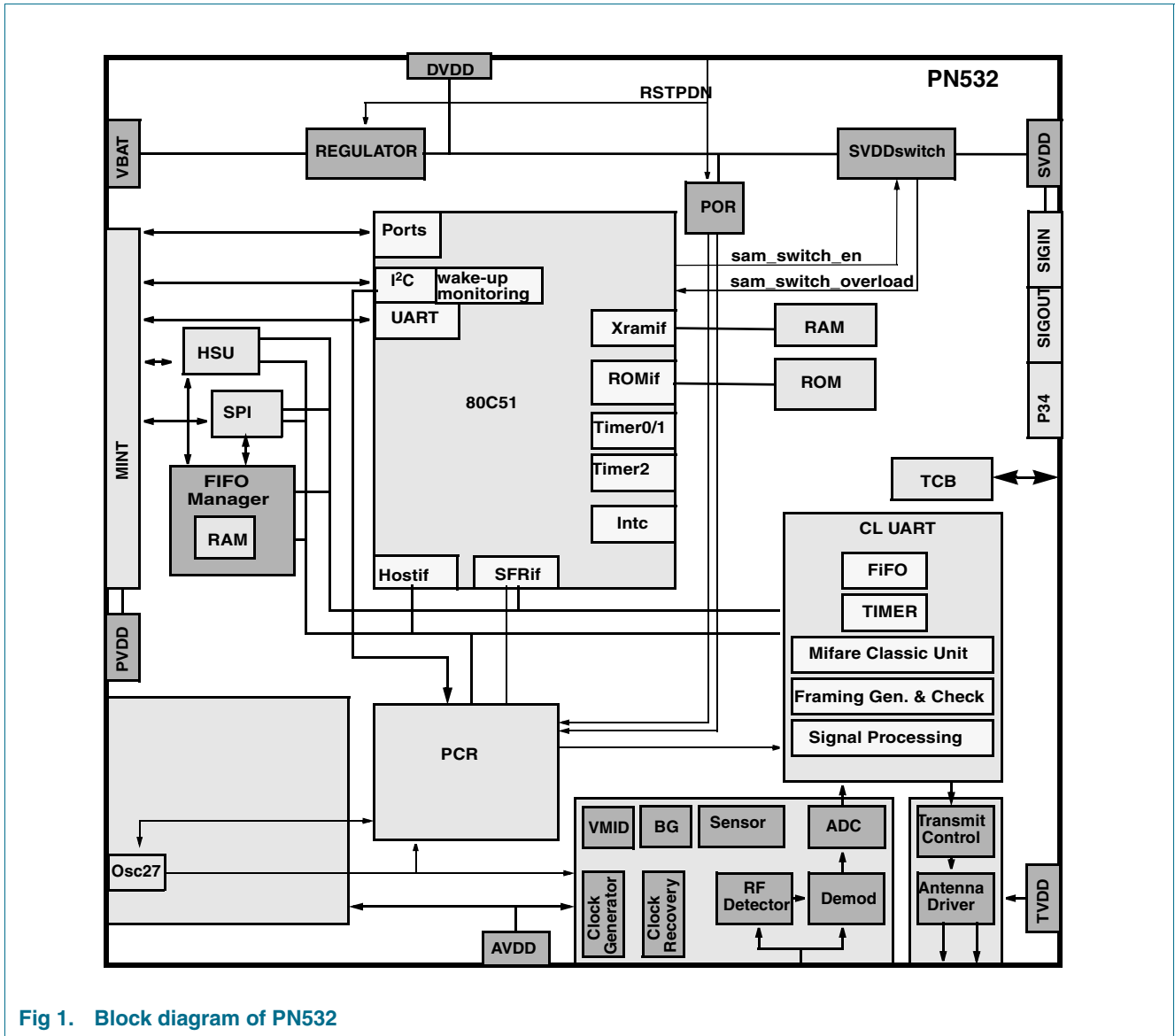


Fig 1. Block diagram of PN532

8. Pinning information

8.1 Pin description

Table 3: PN532 Pin description

Symbol	Pin	Type	Pad Ref Voltage	Description
DVSS	1	PWR		Digital Ground
LOADMOD	2	O	DVDD	Load Modulation output provides digital signal for FeliCa™ and MIFARE® card operating mode
TVSS1	3	PWR		Transmitter Ground: supplies the output stage of TX1 and TX2
TX1	4	O	TVDD	Transmitter 1: delivers the modulated 13.56 MHz energy carrier
TVDD	5	PWR		Internal Transmitter power supply: supplies the output stage of TX1 and TX2
TX2	6	O	TVDD	Transmitter 2: delivers the modulated 13.56 MHz energy carrier
TVSS2	7	PWR		Transmitter Ground: supplies the output stage of TX1 and TX2
AVDD	8	PWR		Internal Analog Power Supply
VMID	9	O	AVDD	Internal Reference Voltage: This pin delivers the internal reference voltage.
RX	10	I	AVDD	Receiver Input: Input pin for the reception signal, which is the load modulated 13.56 MHz energy carrier from the antenna circuit.
AVSS	11	PWR		Analog Ground
AUX1	12	O	AVDD	Auxiliary Output: This pin delivers analog and digital test signals.
AUX2	13	O	AVDD	Auxiliary Output: This pin delivers analog and digital test signals.
OSCIN	14	I	AVDD	Crystal Oscillator Input: input to the inverting amplifier of the oscillator. This pin is also the input for an externally generated clock (fosc = 27.12 MHz).
OSCOU	15	O	AVDD	Crystal Oscillator output: Output of the inverting amplifier of the oscillator.
I0	16	I	DVDD	General purpose IO signal Can be used by the embedded firmware to select the used host interface.
I1	17	I	DVDD	General purpose IO signal Can be used by the embedded firmware to select the used host interface.
TESTEN	18	I	DVDD	Test enable pin: When set to 1 enable the test mode. When set to 0 reset the TCB and disable the access to the test mode.
P35	19	IO	DVDD	General purpose IO signal
NC	20			
NC	21			
NC	22			
PVDD	23	PWR		Pad power supply
P30	24	IO	PVDD	General purpose IO signal. Can be configured to act either as RX line of the second serial interface or general purpose IO. In test mode this signal is used as input and output test signal.
IRQ	25	O	PVDD	Interrupt request: Output to signal an interrupt event to the host (Port 7 bit 0)
RSTOUTN	26	IO	PVDD	Output reset signal. When Low it indicates that the circuit is in reset state.
NSS	27	IO	PVDD	Not Slave Select .
MOSI	28	IO	PVDD	Master Out Slave In.
MISO	29	IO	PVDD	Master In Slave Out .
SCK	30	IO	PVDD	

Table 3: PN532 Pin description ...continued

Symbol	Pin	Type	Pad Ref Voltage	Description
P31	31	IO	PVDD	General purpose IO signal. Can be configured to act either as TX line of the second serial interface or general purpose IO. In test mode this signal is used as input and output test signal.
P32_INT0	32	IO	PVDD	General purpose IO signal. Can be used to generate an HZ state on the output of the selected interface for the Host communication and to enter PN532 into powerdown mode without resetting the internal state of PN532. In test mode this signal is used as input and output test signal.
P33_INT1	33	IO	PVDD	General purpose IO signal. Can also be used as an interrupt source In test mode this signal is used as input and output test signal.
P34	34	IO	SVDD	General purpose IO signal or clk signal for the SAM
SIGOUT	35	O	SVDD	Contactless communication interface output: delivers a serial data stream according to NFCIP-1 and output signal for the SAM. In test mode this signal is used as test signal output.
SIGIN	36	I	SVDD	Contactless communication interface input: accepts a digital, serial data stream according to NFCIP-1 and input signal from the SAM. In test mode this signal is used as test signal input.
SVDD	37	O		Output power for SAM power supply. Switched on by Firmware with an overload detection. Used as a reference voltage for SAM communication.
RSTPDN	38	I	PVDD	Reset and Power Down: When Low, internal current sources are switched off, the oscillator is inhibited, and the input pads are disconnected from the outside world. With a negative edge on this pin the internal reset phase starts.
DVDD	39	PWR		Internal Digital Power Supply
VBAT	40	PWR		Main external power supply.

9. Functional description

9.1 CONTACT LESS MODULE

The PN532 includes a highly integrated transmission/reception module for contactless communication at 13.56 MHz. This transmission/reception contact less (CL) module utilises an outstanding modulation and demodulation concept completely integrated for different kinds of contactless communication methods and protocols at 13.56 MHz.

The CL module support 4 different operating modes

- reader / writer mode supporting ISO 14443A / MIFARE® and FeliCa™ scheme
- reader / writer mode supporting ISO 14443B
- card operation mode supporting ISO 14443A / MIFARE® and FeliCa™ scheme
- NFCIP-1 mode

Enabled in reader / writer mode for ISO 14443A / MIFARE®, the CL module transmitter part is able to drive a reader / writer antenna designed to communicate with ISO 14443A / MIFARE® cards and transponders without additional active circuitry. The CL module receiver part provides a robust and efficient implementation of a demodulation and decoding circuitry for signals from ISO 14443A / MIFARE® compatible cards and transponders. The CL module handles the complete ISO 14443A framing and error detection (Parity & CRC). The CL module supports MIFARE® Classic (e.g. MIFARE® Standard) products. The CL module supports contactless communication using MIFARE® Higher transfer speeds up to 424 kbit/s in both directions.

Enabled in reader / writer mode for FeliCa™, the CL module supports the FeliCa™ communication scheme. The CL module receiver part provides a robust and efficient implementation of the demodulation and decoding circuitry for FeliCa™ coded signals. The CL module digital part handles the FeliCa™ framing and error detection like CRC. The CL module supports contactless communication using FeliCa™ Higher transfer speeds up to 424 kbit/s in both directions.

The CL module supports all layers of the ISO/IEC 14443 B reader / writer communication scheme, given correct implementation of additional components, like oscillator, power supply, coil etc. and provided that standardised protocols, e.g. like ISO/IEC 14443-4 and/or ISO/IEC 14443 B anticollision are correctly implemented. The use of this Philips IC according to ISO/IEC 14443 B might infringe third party patent rights. A purchaser of this Philips IC has to take care for appropriate third party patent licenses.

In card operation mode, the CL module is able to answer to a reader / writer command either according to the FeliCa™ or ISO 14443A / MIFARE® card interface scheme. The CL module generates the digital load modulated signals and in addition with an external circuit the answer can be sent back to the reader / writer. A complete card functionality is only possible in combination with a secure core IC using the S2C interface.

Additionally, the CL module offers the possibility to communicate directly to an NFCIP-1 device in the NFCIP-1 mode. The NFCIP-1 mode offers different communication modes and transfer speeds up to 424kbit/s according to the Ecma 340 NFCIP-1 Standard. The CL module digital part handles the complete NFCIP-1 framing and error detection.

9.1.1 Simplify block diagram

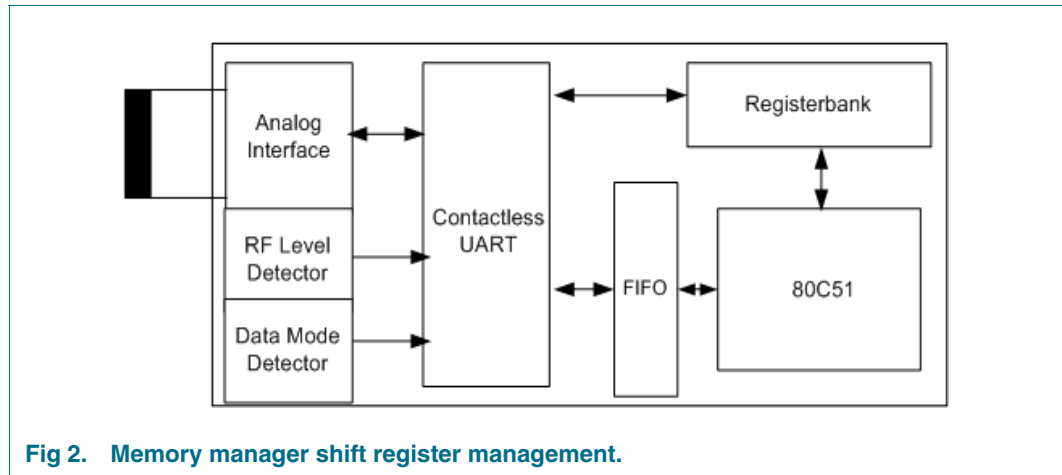


Fig 2. Memory manager shift register management.

The Analog interface handles the modulation and demodulation of the analog signals according to the card receiving mode, reader / writer mode and NFCIP-1 mode communication scheme.

The RF level detector detects the presence of an external RF-field delivered by the antenna to the RX pin.

The data mode detector detects a MIFARE®, FeliCa™ or NFCIP-1 mode in order to prepare the internal receiver to demodulate signals, which are sent to the PN512.

The communication (S2C) interface provides digital signals to support communication for transfer speeds above 424 kbit/s and digital signals to communicate to a secure core IC.

The contactless UART handles the protocol requirements for the communication schemes in co-operation with the host. The comfortable FIFO buffer allows a fast and convenient data transfer from the host to the contactless UART and vice versa.

9.1.2 Feature list

- Close communication link to the analog circuitry to demodulate and decode card's response
- Typical MOVX access to non critical registers
- SFR register map for high frequency register access (16 Registers)
- Integrated data mode detector
- Supports ISO 14443A / MIFARE®
- Supports ISO 14443 B reader / writer functionality
- Adjustable parameters to optimize the reception according to the antenna configuration
- Adjustable parameters to optimize the transmission according to the antenna configuration and characteristics.
- typical operating distance in reader / writer mode for communication to a ISO 14443A/ MIFARE® or FeliCa™ card up to 50 mm depending on the antenna size, tuning and power supply

- typical operating distance in NFCIP-1 mode up to 50 mm depending on the antenna size and tuning and power supply
- typical operating distance in ISO 14443A / MIFARE® card or FeliCa™ card operation mode of about 100 mm depending on the antenna size and tuning and the external field strength
- Supports MIFARE® Classic encryption in reader / writer mode
- Supports ISO 14443A higher transfer speed communication at 212 kbit/s and 424 kbit/s
- Supports contactless communication according to the FeliCa™ scheme at 212 kbit/s and 424 kbit/s
- Integrated RF interface for NFCIP-1 up to 424 kbit/s
- Possibility to communicate on the RF interface above 424 kbit/s using external analog circuitry
- Support of the S2C interface
- 64 byte send and receive FIFO-buffer
- Programmable timer
- CRC Co-processor
- internal self test
- 2 interrupt sources
- Integrated RF Level detector
- Integrated RF interface for NFCIP-1 up to 424 kbit/s

9.1.3 Operating Modes

The CL module support the following operating modes:

- Reader/writer mode supporting ISO14443A / MIFARE®, Felica™ and ISO14443B schemes.
- Card operation mode supporting ISO14443A / MIFARE® and Felica™ schemes
- NFCIP-1 mode

The modes support different transfer speeds and modulation schemes. The following chapters will explain the different modes more in detail.

Note: All indicated modulation indexes and modes in this chapter are system parameters. This means that beside the IC settings a suitable antenna tuning is required to achieve the optimal performance.

9.1.3.1 Reader / Writer mode

Generally 3 reader/writer-operating modes are supported. The PN532 can act as a reader / writer for ISO14443A / MIFARE®, FeliCa™ and ISO14443B cards.

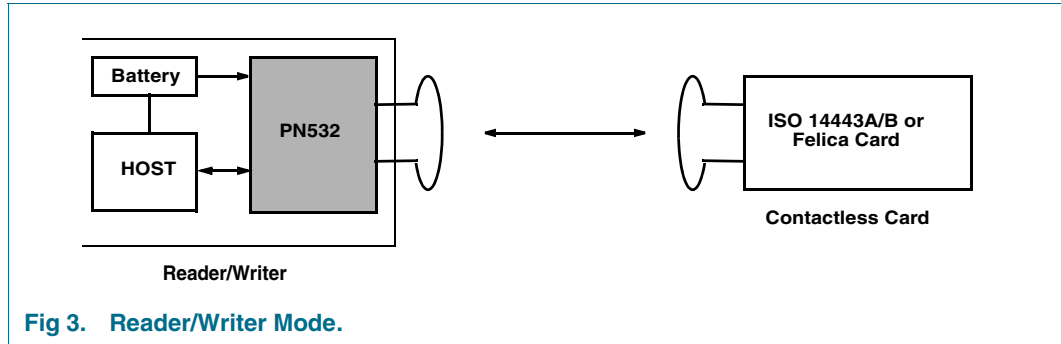


Fig 3. Reader/Writer Mode.

9.1.3.2 ISO14443A Reader / Writer Functionality

The ISO14443A / MIFARE® reader / writer mode is the general reader / writer to card communication scheme according to the ISO14443A / MIFARE® specification. The following diagram describes the communication on a physical level.

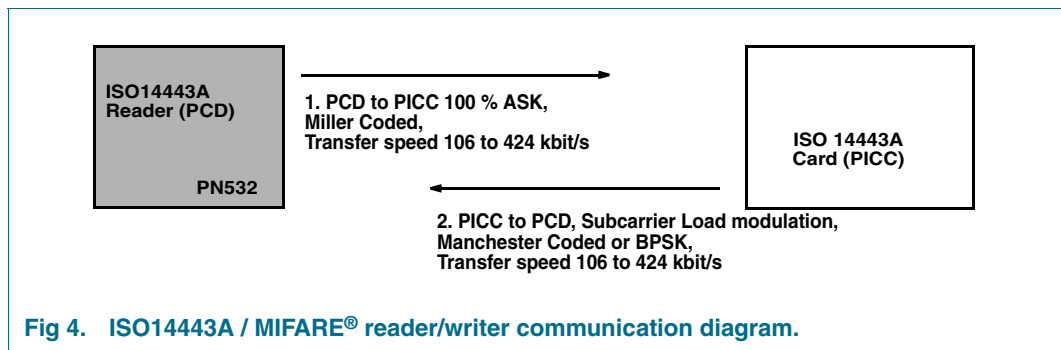


Fig 4. ISO14443A / MIFARE® reader/writer communication diagram.

Table 4: Communication overview for ISO14443A / MIFARE® reader/writer

Communication direction	Baudrate	MIFARE® / ISO14443A	MIFARE® Higher transfer speed	
		106kbaud	212 Kbaud	424kbaud
PN532 → card	Modulation on reader side	100 % ASK	100 % ASK	100 % ASK
	bit coding	Modified Miller coding	Modified Miller coding	Modified Miller coding
	Bitlength	$128/13.56 = 9.44 \mu\text{s}$	$64/13.56 = 9.44\mu\text{s}/2$	$32/13.56 = 9.44\mu\text{s}/4$
Card → PN532	Modulation on card side	Subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	Subcarrier frequency	$13.56\text{MHz}/16$	$13.56\text{MHz}/16$	$13.56\text{MHz}/16$
	bit coding	Manchester coding	BPSK	BPSK

The contactless UART, in cooperation with the internal micro-controller of PN532 and the external host handle the complete MIFARE® / ISO14443 A protocol.

The internal CRC coprocessor calculates the CRC value according to the definitions given in the ISO 14443A part 3.

9.1.3.3 FeliCa™ Reader/Writer Functionality

The FeliCa™ mode is the general reader / writer to card communication scheme according to the FeliCa™ specification. The following diagram describes the communication on a physical level.

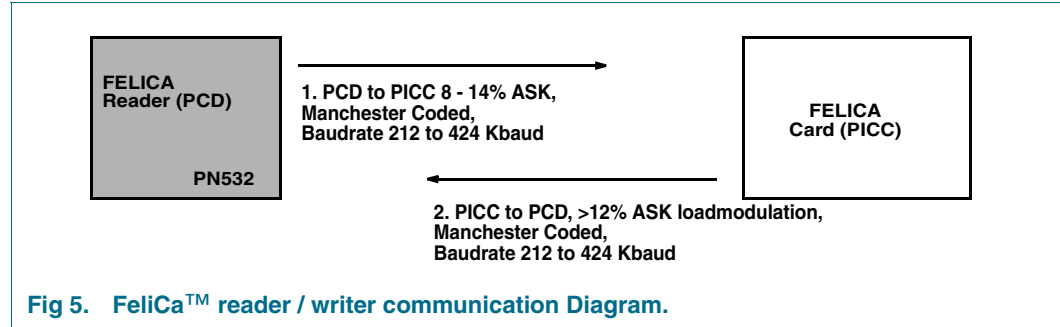


Fig 5. FeliCa™ reader / writer communication Diagram.

Table 5: Communication Overview for FeliCa™ reader/writer functionality

Communication direction	Baudrate	FeliCa™	
		212 Kbaud	FeliCa™ Higher Baudrate 424kBaud
PN532 ->card	Modulation on reader side	8 - 14 % ASK	8 - 14 % ASK
	bit coding	Manchester coding	Manchester coding
	Bitlength	$64/_{13.56} = 9.44\mu s/2$	$32/_{13.56} = 9.44\mu s/4$
Card->PN532	Modulation on card side	>12% ASK	>12% ASK
	bit coding	Manchester coding	Manchester coding

The internal contactless UART, the internal µC of PN532 and the external host handle the FeliCa™ protocol.

The Framing and coding of the FeliCa™ should be according the following table:

Table 6: FeliCa™ Framing and Coding

Preamble						Sync	Len	n-Data				CRC	
00	00	00	00	00	00	B2	4D						

To enable the FeliCa™ communication a 6 bytes long preamble and 2 bytes Sync bytes are sent in order to synchronise the internal receiver. The Len byte is an indicator for the length of the sent data bytes plus the n-data bytes. The CRC calculation is done according to the FeliCa™ definitions with the MSB first.

To transmit data on the RF interface, the host has to send the Preamble-, Syn-, Len- and data- bytes to the PN532. Only the internal CRC calculation is made and added internally of the PN532

The starting value for the CRC Polynomial is 2 null bytes: (0x00), (0x00)

Example of frame sent to the field:

Table 7: FeliCa™ Framing and Coding

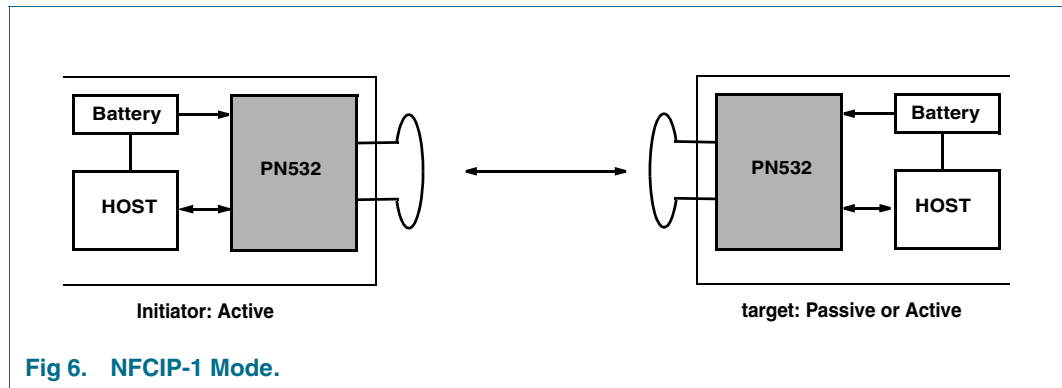
Preamble						Sync	Len	2 Data Bytes		CRC		
00	00	00	00	00	00	B2	4D	03	AB	CD	90	35

9.1.4 NFCIP-1 MODE

The NFCIP-1 communication differentiates between an active and a passive communication mode.

- Active Communication Mode means both the initiator and the target are using their own RF field to transmit data
- Passive Communication Mode means that the target answers to an initiator command in a load modulation scheme. The initiator is active in terms of generating the RF field.
- Initiator: generates RF field @ 13.56 MHz and starts the NFCIP
- Target: responds to initiator command either in a load modulation scheme for passive communication mode or using a self generated and self modulated RF field for active communication mode.

In order to fully support the NFCIP-1 standard the PN532 supports the active and passive communication mode at the transfer speeds 106 kbit/s, 212 kbit/s and 424 kbit/s as defined in the NFCIP-1 standard



9.1.4.1 ACTIVE Communication mode

Active Communication Mode means both the initiator and the target are using their own RF field to enable the communication.

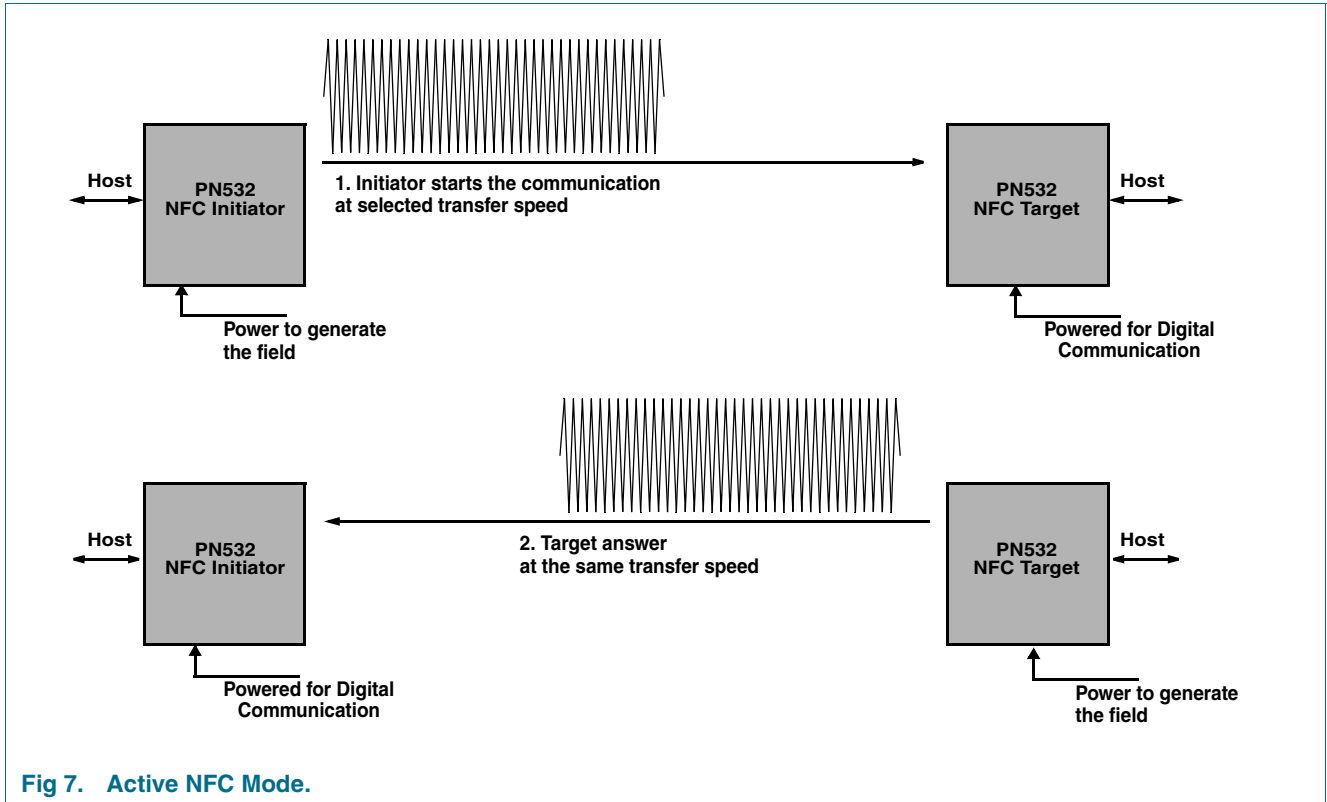


Fig 7. Active NFC Mode.

The following table gives an overview of the active communication modes:

Table 8: Communication Overview for active NFC

Communication direction	106 kbit/s	212 kbit/s	424 kbit/s	848 kbit/s	1.69 Mbit/s 3.39 Mbit/s
Initiator -> Target	According to ISO14443A 100% ASK, Miller Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	digital capability to handle this communication	according to the NFC mode
Target -> Initiator	According to ISO14443A 100% ASK, Miller Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	digital capability to handle this communication	according to the NFC mode

Note: Transfer speed above 424 kbit/s are not defined in the NFCIP-1. The PN532 supports these transfer speeds only with dedicated external circuitry.

9.1.4.2 PASSIVE Communication mode

Passive Communication Mode means that the target answers to an initiator command in a load modulation scheme. The initiator is active meaning generating the RF field.

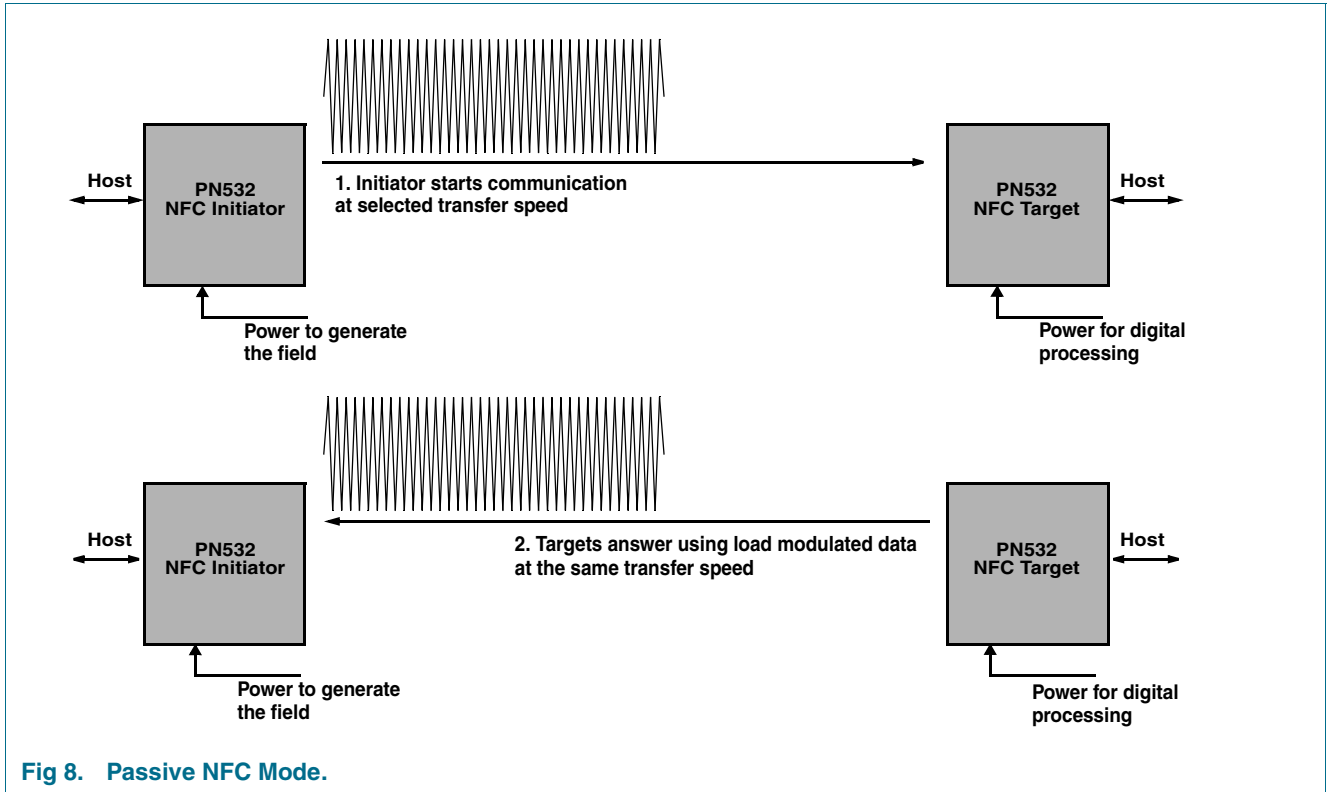


Fig 8. Passive NFC Mode.

The following table gives an overview of the active communication modes:

Table 9: Communication Overview for passive NFC

Communication direction	106 kbit/s	212 kbit/s	424 kbit/s	848 kbit/s	1.69 Mbit/s 3.39 Mbit/s
Initiator -> Target	According to ISO14443A 100% ASK, Miller Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	According to FeliCa™, 8-30 %ASK Manchester Coded	digital capability to handle this communication	according to the NFC mode
Target -> Initiator	according to ISO14443 A subcarrier load modulation, Manchester Coded	according to FeliCa™, >12 % ASK, Manchester Coded	according to FeliCa™, >12 % ASK, Manchester Coded	digital capability to handle this communication	according to the NFC mode

Note: Transfer speed above 424 kbit/s are not defined in the NFCIP-1. The PN532 supports these transfer speeds only with dedicated external circuitry.

9.1.4.3 NFC FRAMING AND CODING

The NFCIP-1 framing and coding in active and passive communication modes are defined in the NFCIP-1 standard.

Table 10: NFC Framing and Coding Overview

Baudrate	Framing and Coding
106 kbaud	According to the ISO 14443A / MIFARE® scheme
212 kbaud	According to the FeliCa™ scheme
424 kbaud or higher	According to the FeliCa™ scheme

9.1.4.4 NFC Protocol Support

The NFCIP-1 protocol is not completely described in this document. For detailed explanation of the protocol refer to the NFCIP-1 standard. However the datalink layer is according to the following policy:

- Speed shall not be changed while continuum data exchange in a transaction.
- More than one transaction at a time in the same operation field is prohibited.
- Transaction includes initialisation and anticollision methods and data exchange (in continuous way, meaning no interruption by another transaction).

In order not to disturb current infrastructure based on 13.56 MHz general rules to start NFC communication are defined in the following way.

- Per default NFCIP-1 device is in target mode, meaning its RF field is switched off.
- The RF level detector is active
- Only if application requires the NFC device shall switch to initiator mode
- Initiator shall only switch on RF if no external RF field is detected by RF Level detector during a time of T_{IDT} .
- The initiator performs initialisation according to the selected mode.

9.1.5 Card operation mode

The PN532 can be addressed like a FeliCa™ or ISO 14443A / MIFARE® card. This means that the PN532 can generate an answer in a load modulation scheme according to the ISO 14443A / MIFARE® or FeliCa™ interface description.

Remark: The PN532 does not support a complete card protocol. This has to be handled by a dedicated card SAM or a micro-controller. The SAM is optional.

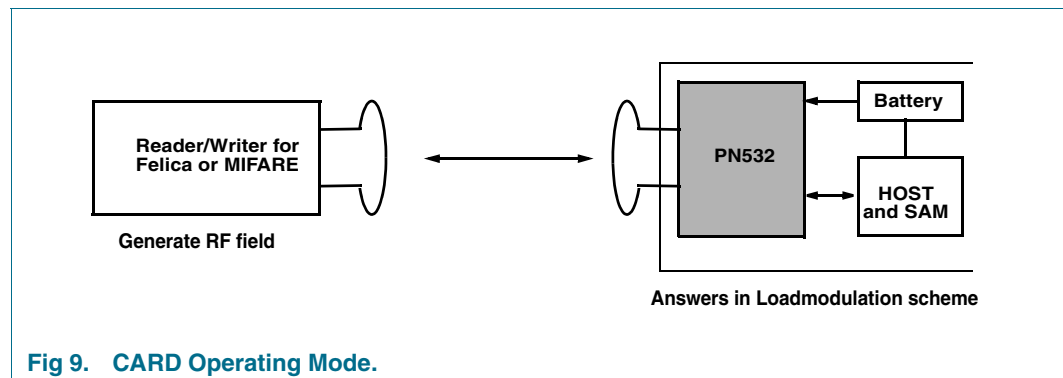


Fig 9. CARD Operating Mode.

9.1.5.1 MIFARE® card interface mode

Table 11: MIFARE® CARD operating mode

Communication direction	Transfer speed	MIFARE® / ISO14443A	MIFARE® Higher Baudrates	
		106kbit/s	212 kbit/s	424 kbit/s
PN532 receiving data from the reader / writer	Modulation on reader side	100 % ASK	100 % ASK	100 % ASK
	bit coding	Modified Miller coding	Modified Miller coding	Modified Miller coding
	Bitlength	$128/13.56 = 9.44 \mu\text{s}$	$64/13.56 = 9.44\mu\text{s}/2$	$32/13.56 = 9.44\mu\text{s}/4$
PN532 sending data back to the reader / writer	Modulation on PN532 side	Subcarrier load modulation	subcarrier load modulation	subcarrier load modulation
	Subcarrier frequency	$13.56\text{MHz}/16$	$13.56\text{MHz}/16$	$13.56\text{MHz}/16$
	bit coding	Manchester coding	BPSK	BPSK

9.1.5.2 FeliCa™ card interface mode

Table 12: FeliCa™ CARD operating mode

Communication direction	Baudrate	FeliCa™	FeliCa™ Higher Baudrates
		212kbaud	424kBaud
PN532 receiving data from the reader / writer	Modulation on reader side	8-14 % ASK	8-14 % ASK
	bit coding	Manchester Coding	Manchester Coding
	Bitlength	$64/13.56\mu\text{s}$	$32/13.56\mu\text{s}$
PN532 sending data back to the reader / writer	Modulation on PN532 side	>12% ASK, loadmodulation	>12% ASK, load modulation
	bit coding	Manchester coding	Manchester coding

10. Limiting values

Table 13: Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions	Min	Max	Unit
PVDD	Supply Voltage		-0.5	4	V
VBAT	Battery Supply Voltage		-0.5	6.0	V
P _{tot}	Total power dissipation			tbd	mW
I _{TX1}	Maximum current in transmitter TX1		-100	100	mA
I _{TX2}	Maximum current in transmitter TX2		-100	100	mA
T _{stg}	Storage temperature		-55	150	°C
T _j	Junction temperature			100	°C

Table 14: ESD Characteristics

Symbol	Parameter	Conditions	Specification	Value
ESDH	ESD Susceptibility (Human Body model)	1500 Ohm, 100pF	JESD22-A114-B	2 KV
ESDM	ESD Susceptibility (Machine model)	0.75 μH, 200 pF	JESD22-A114-A	200 V
ESDC	ESD Susceptibility (Charge Device model)	Field induced model	JESC22-C101-A	1 KV

11. Recommended operating conditions

Table 15: Operating conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
T _{amb}	Ambiant Temperature		-30	+25	+85	°C
VBAT	Battery Supply Voltage	VSS = 0V	[1] , [2] 2.7	5	5.4	V
PVDD	Supply voltage from host interface	VSS=0V	1.6	1.8-3.3	3.6	V

[1] VSS represents DVSS, TVSS1, TVSS2, AVSS.

[2] Supply voltage of VBAT below 3.3 V reduces the performance (e.g. the achievable operating distance).

12. Thermal characteristics

Table 16: Thermal characteristics

Symbol	Parameter	Conditions	Typ	Unit
R _{thj-a}	thermal resistance from junction to ambient (for HVQFN40 package)	in free air with exposed pad soldered on a 4 layer Jedec PCB-0.5	35	K/W

13. Characteristics

Table 17: Current Consumption

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Ihpd	Hard Power Down Current	PVDD=3V, RF level detector off	[5]			10	mA
ISPD	Soft Power down Current	PVDD=3V, RF level detector on	[5]			35	mA
IAVDD	Analog Supply Current	VBAT = 5V PVDD=3V, RF level detector on			tbd	6	mA
IAVDDrcvoff	Analog Supply Current	VBAT = 5V PVDD=3V, RF level detector off			3	5	mA
IPVDD	Pad Supply Current		[2]			tbd	mA
ISVDD	Output Supply Current for SAM	sam_switch_en set to 1	[3]			30	mA
ITVDD1,4	Transmitter Supply Current	Continuous Wave, VBAT = 5V	[1][4]		602	100	mA
IVBAT	Total Supply Current	Continuous Wave, VBAT = 5V	[1][4]		76,5	tbd	mA

[1] ITVDD depends on TVDD and the external circuitry connected to Tx1 and Tx2.

[2] IPVDD depends on the overall load at the digital pins.

[3] ISVDD depends on the overall load on SVDD pad.

[4] During operation with a typical circuitry the overall current is below 100 mA.

[5] ISPD and IHPD are the total currents over all supplies.

[6] Typical value using a complementary driver configuration and an antenna matched to 40 Ohm between TX1 and TX2 at 13.56 MHz.

14. Application information

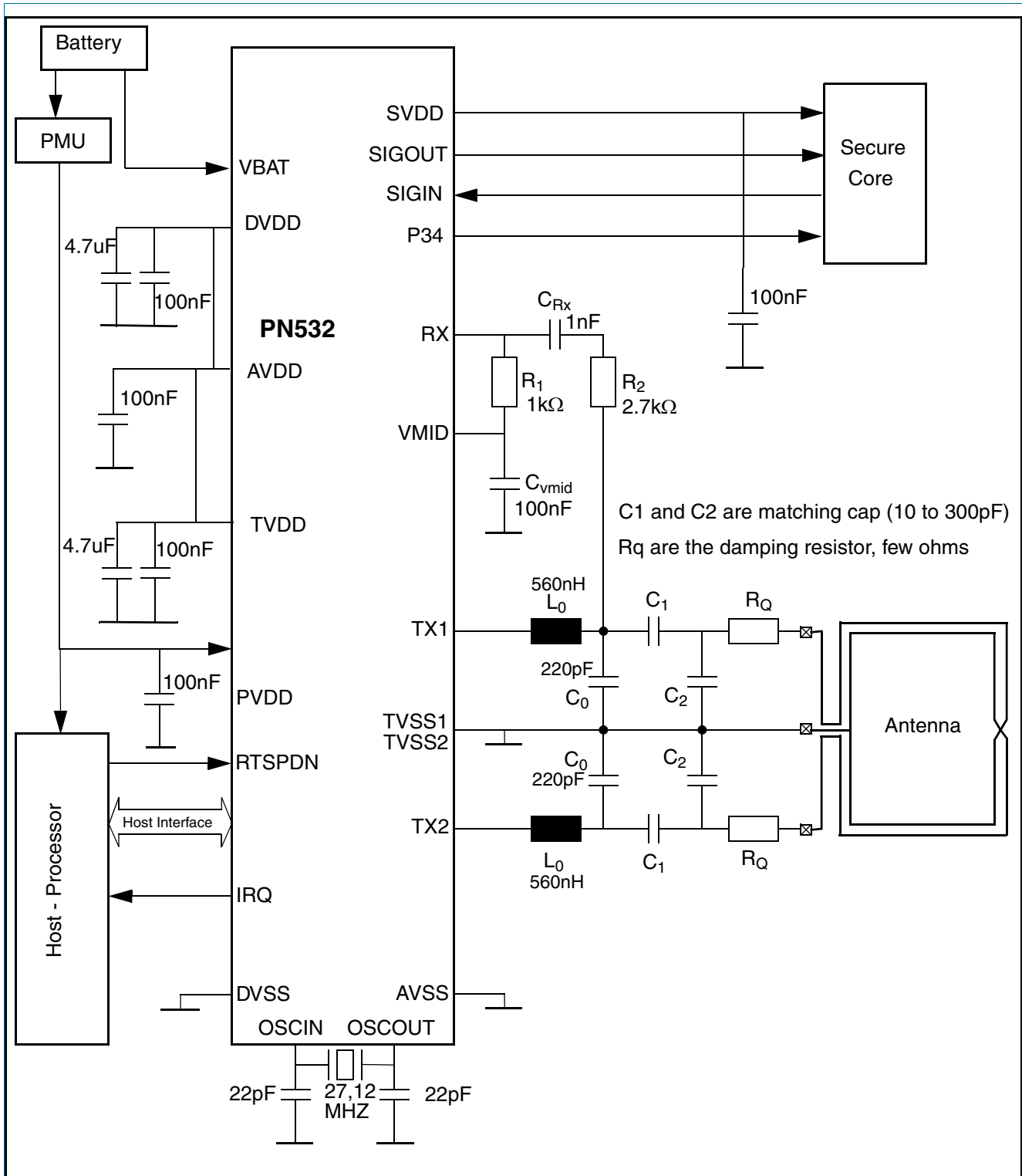


Fig 10. Application diagram of PN532

In the example the 27.12MHz quartz is a TAS-3225A, SMD

15. Package outline

HVQFN40: plastic thermal enhanced very thin quad flat package; no leads; 40 terminals; body 6 x 6 x 0.85 mm

SOT618-1

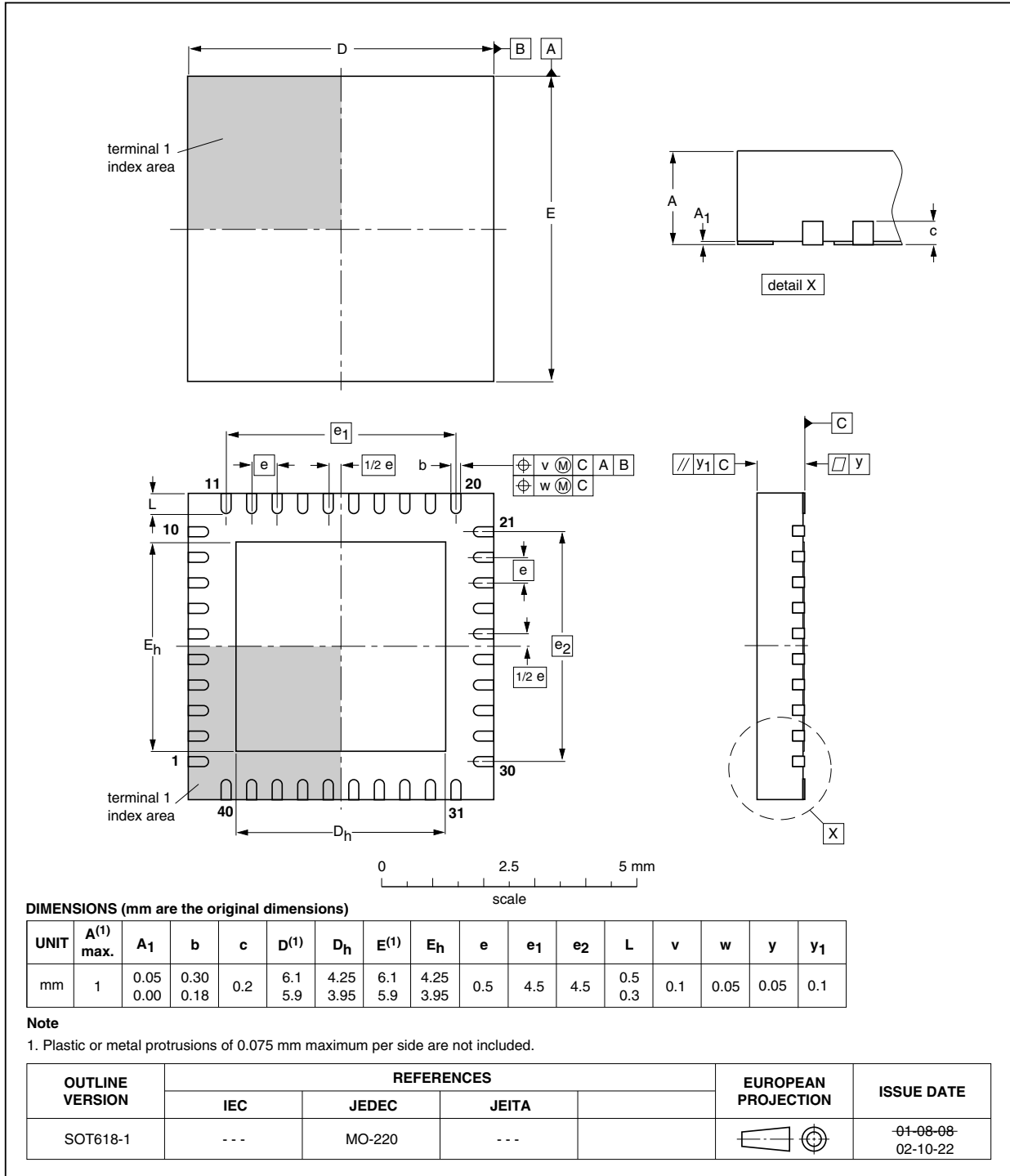


Fig 11. Package outline HVQFN40 (SOT618-1)

16. Abbreviations

Table 18: Abbreviations

Acronym	Description
ASK	Amplitude Shift keying
PCD	Proximity Coupling Device. Definition for a Card Reader/ Writer according to the ISO 14443 Specification
PICC	Proximity Cards. Definition for a contactless Smart Card according to the ISO14443 specification
PCD -> PICC	Communication flow between a PCD and a PICC according to the ISO14443A/MIFARE®
PICC -> PCD	Communication flow between a PICC and a PCD according to the ISO14443A/MIFARE®
Initiator	Generates RF field @ 13.56 MHZ and starts the NFCIP-1 communication.
Modulation Index	The modulation index is defined as the voltage ratio $(V_{max} - V_{min}) / (V_{max} + V_{min})$.
Loadmodulation Index	The load modulation index is defined as the card's voltage ratio $(V_{max} - V_{min}) / (V_{max} + V_{min})$ measured at the card's coil.
Target	Responds to initiator command either using load modulation scheme (RF field generated by Initiator) or using modulation of self generated RF field (no RF field generated by initiator).

17. Revision history

Table 19: Revision history

Document ID	Release date	Data sheet status	Change notice	Doc. number	Supersedes
	2006.01.08	short form data sheet		Draft 1.2	Initial version

18. Data sheet status

Level	Data sheet status ^[1]	Product status ^[2] ^[3]	Definition
I	Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
II	Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
III	Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Relevant changes will be communicated via a Customer Product/Process Change Notification (CPCN).

[1] Please consult the most recently issued data sheet before initiating or completing a design.

[2] The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL <http://www.semiconductors.philips.com>.

[3] For data sheets describing multiple type numbers, the highest-level product status determines the data sheet status.

19. Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

20. Disclaimers

Life support — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes in the products - including circuits, standard cells, and/or software - described or contained herein in order to improve design and/or performance. When the product is in full production (status 'Production'), relevant changes will be communicated via a Customer Product/Process

Change Notification (CPCN). Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

21. Licenses

Purchase of Philips I²C-bus components



Purchase of Philips I²C-bus components conveys a license under the Philips' I²C-bus patent to use the components in the I²C-bus system provided the system conforms to the I²C-bus specification defined by Koninklijke Philips Electronics N.V. This specification can be ordered using the code 9398 393 40011.

Purchase of Philips RC5 components

Purchase of Philips RC5 components conveys a license under the Philips RC5 patent to use the components in RC5 system products conforming to the RC5 standard UATM-5000 for allocation of remote control commands defined by Koninklijke Philips Electronics N.V.

Purchase of NXP ICs with NFC technology

Purchase of an NXP Semiconductors IC that complies with one of the Near Field Communication (NFC) standards ISO/IEC 18092 and ISO/IEC 21481 does not convey an implied license under any patent right infringed by implementation of any of those standards. A license for the patents portfolio of NXP B.V. for the NFC standards needs to be obtained at Via Licensing, the pool agent of the NFC Patent Pool, e-mail: info@vialicensing.com.

22. Contact information

For additional information, please visit: <http://www.semiconductors.philips.com>

For sales office addresses, send an email to: sales.addresses@www.semiconductors.philips.com

23. Contents

1	Introduction	1
2	General description	1
3	Features	2
4	Applications	3
5	Quick reference data	3
6	Ordering information	4
7	Block diagram	5
8	Pinning information	6
8.1	Pin description	6
9	Functional description	8
9.1	CONTACT LESS MODULE	8
9.1.1	Simplify block diagram	9
9.1.2	Feature list	9
9.1.3	Operating Modes	10
9.1.4	NFCIP-1 MODE	13
9.1.5	Card operation mode	16
10	Limiting values	18
11	Recommended operating conditions	18
12	Thermal characteristics	18
13	Characteristics	19
14	Application information	20
15	Package outline	21
16	Abbreviations	22
17	Revision history	23
18	Data sheet status	24
19	Definitions	24
20	Disclaimers	24
21	Licenses	24
22	Contact information	24

ANEXO 14. DATASHEET DEL EASYVR



EasyVR

Datasheet

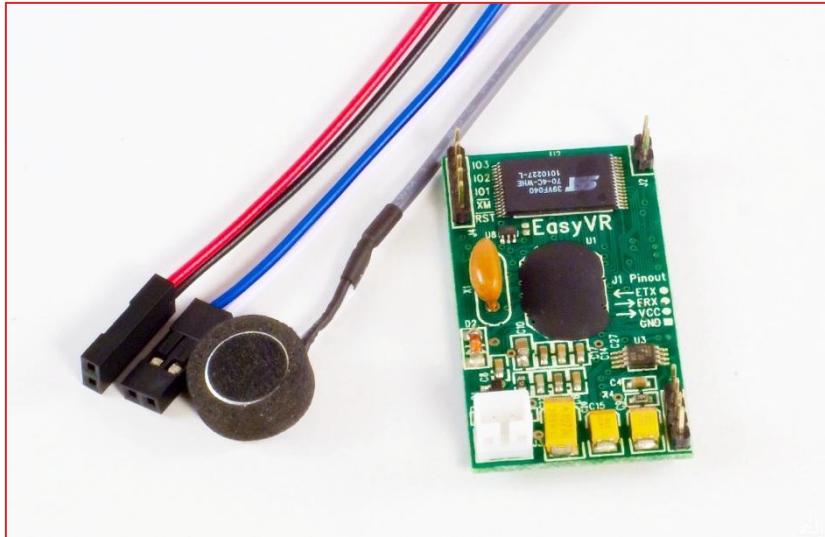
Release 2.0



Product Description

EasyVR is the second generation version of the successful VRbot Module. It is a multi-purpose speech recognition module designed to easily add versatile, robust and cost effective speech recognition capabilities to virtually any application.

The EasyVR module can be used with any host with an UART interface powered at 3.3V – 5V, such as PIC and Arduino boards. Some application examples include home automation, such as voice controlled light switches, locks or beds, or adding “hearing” to robots such as ROBONOVA-I and POP Bot.



EasyVR features

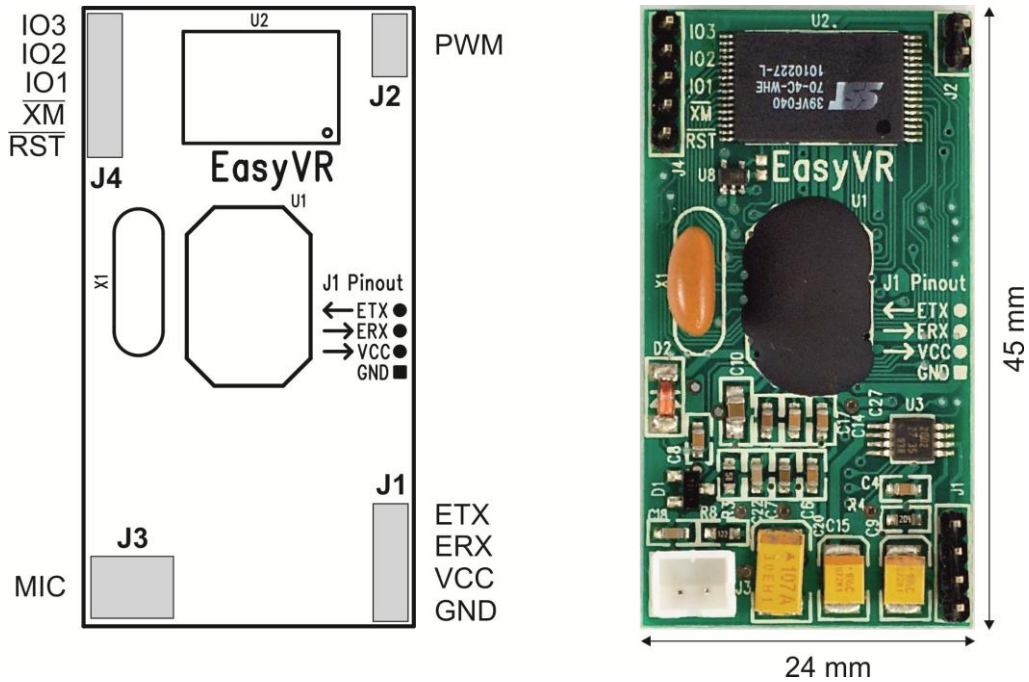
- A host of built-in speaker independent (SI) commands for ready to run basic controls
- Supports up to 32 user-defined Speaker Dependent (SD) triggers or commands as well as Voice Passwords. SD custom commands can be spoken in ANY language.
- Easy-to-use and simple Graphical User Interface to program Voice Commands
- Languages currently supported for SI commands: English U.S., Italian, Japanese, German, Spanish and French. More languages could be available in future.
- Module can be used with any host with an UART interface (powered at 3.3V - 5V)
- Simple and robust documented serial protocol to access and program through the host board

New features on EasyVR

- Two new SI languages: Spanish and French
- 3 GPIO lines (IO1, IO2, IO3) that can be controlled by new protocol commands
- (Note: the GPIO are at nominal 3.0VDC level. do not connect 5VDC to these pins!)
- Audio output that supports 8 ohm speakers
- Firmware update capability with two additional lines (/XM, /RST)
- Sound playback feature:
 - You can make your own sound tables using Sensory QuickSynthesis4 tool
 - The new EasyVR GUI includes a command to process and download custom sound tables to the module (overwriting existing sound table)
 - NOTE: default firmware has no sound table, but can Beep using sound index 0 – always available. Custom sounds start at index 1.
 - The VoiceGP DevBoard (available separately) is required for programming the EasyVR flash.

Technical specifications

Physical dimensions and pin assignment



Connector	Number	Name	Type	Description
J1	1	GND	-	Ground
	2	VCC	I	Voltage DC input
	3	ERX	I	Serial Port Receive Data (TTL level)
	4	ETX	O	Serial Port Transmit Data (TTL level)
J2	1-2	PWM	O	Differential audio output (can directly drive 8Ω speaker)
J3	1-2	MIC	I	Microphone input
J4	1	/RST	I	Reset
	2	/XM	I	Reserved – leave unconnected
	3	IO1	I/O	General purpose I/O (3.0 VDC TTL level)
	4	IO2	I/O	General purpose I/O (3.0 VDC TTL level)
	5	IO3	I/O	General purpose I/O (3.0 VDC TTL level)

Note: the GPIO (J4.3/4/5) are at nominal 3.0VDC level. Do not connect 5VDC to these pins!

Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VCC	Voltage DC Input	3.3	5.0	5.5	V
Ta	Ambient Operating Temperature Range	0	25	70	°C
ERX	Serial Port Receive Data	0	-	VCC	V
ETX	Serial Port Transmit Data	0	-	VCC	V

Power Supply Requirements

Symbol	Parameter	Min	Typ	Max	Unit
I _{Sleep}	Sleep current		< 1		mA
I _{Oper}	Operating current		12		mA
I _{Speaker}	Audio playback current (with 8ohm speaker)		180		mA

Communications

Adjustable Asynchronous Serial Communication:

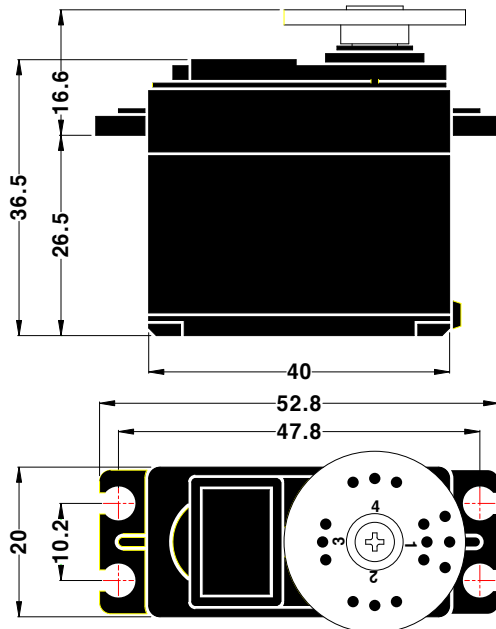
- Baud Rate: **9600** (default), 19200, 38700, 57600, 115200
- 8 Data bits
- No parity
- 1 Stop bit

ANEXO 15. DATASHEET HITEC HS-311

ANNOUNCED SPECIFICATION OF HS-311 STANDARD SERVO

1. TECHNICAL VALUE

CONTROL SYSTEM	: +PULSE WIDTH CONTROL 1500usec NEUTRAL	
OPERATING VOLTAGE RANGE	: 4.8V TO 6.0V	
TEST VOLTAGE	: AT 4.8V	AT 6.0V
OPERATING SPEED	: 0.19sec/60° AT NO LOAD	0.15sec/60° AT NO LOAD
STALL TORQUE	: 3.0kg.cm(42oz.in)	3.5kg.cm(48.60oz.in)
IDLE CURRENT	: 7.4mA AT STOPPED	7.7mA AT STOPPED
RUNNING CURRENT	: 160mA/60° AT NO LOAD	180mA/60° AT NO LOAD
STALL CURRENT	: 700mA	800mA
DEAD BAND WIDTH	: 5usec	5usec
OPERATING TRAVEL	: 40°/ONE SIDE PULSE TRAVELING 400usec	
DIRECTION	: CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec	
MOTOR TYPE	: CORED METAL BRUSH	
POTENTIOMETER TYPE	: 4 SLIDER/DIRECT DRIVE	
AMPLIFIER TYPE	: ANALOG CONTROLLER & TRANSISTOR DRIVER	
DIMENSIONS	: 40x20x36.5mm(1.57x0.78x1.43in)	
WEIGHT	: 43g(1.51oz)	
BALL BEARING	: TOP/RESIN BUSHING	
GEAR MATERIAL	: RESIN	
HORN GEAR SPLINE	: 24 SEGMENTS/□5.76	
SPLINED HORNS	: SUPER/R-XA	
CONNECTOR WIRE LENGTH	: 300mm(11.81in)	
CONNECTOR WIRE STRAND COUNTER	: 40EA	
CONNECTOR WIRE GAUGE		



2. FEATURES

LONG LIFE POTENTIOMETER, TOP RESIN BUSHING

3. APPLICATIONS

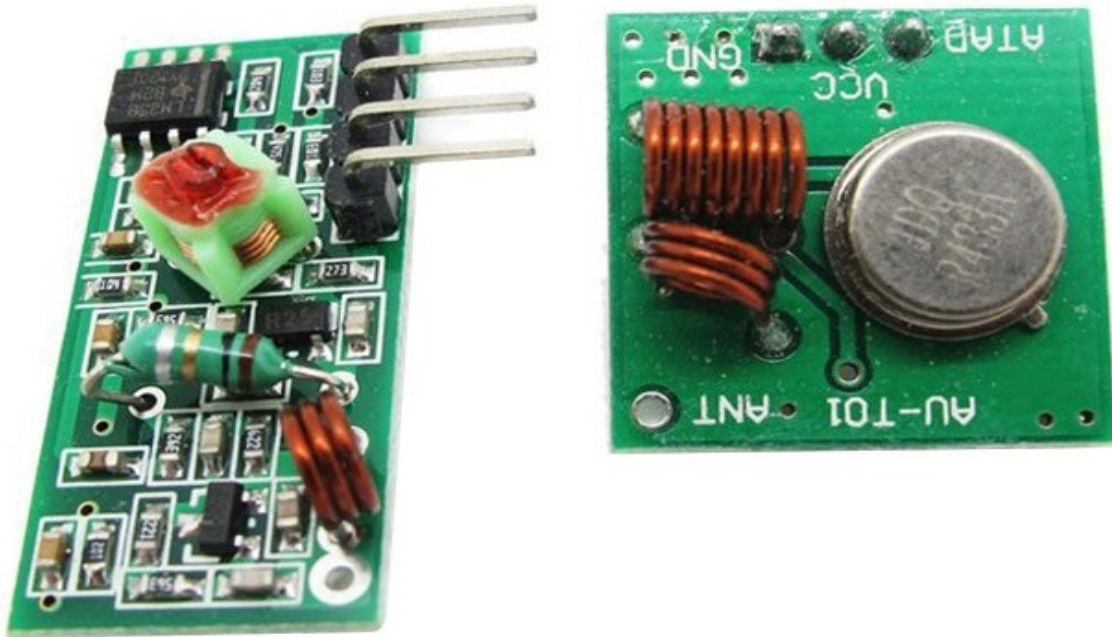
AIRCRAFT 20-40 SIZE, STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS

4. ACCESSORY & OPTION

CASE SET/	GEAR SET/	HORN SET/
HS322T:1EA	HS322G1:1EA	R-XA:1EA
HS322M:1EA	HS322G2:1EA	
HS322L:1EA	HS322G3:1EA	
PH/T-2 2x30 NI:4EA	HS322G4:1EA	
	HS300RB:1EA	

ANEXO 16. MÓDULO DE RADIOFRECUENCIA

**RB-Ite-108
433 Mhz RF Link Kit**

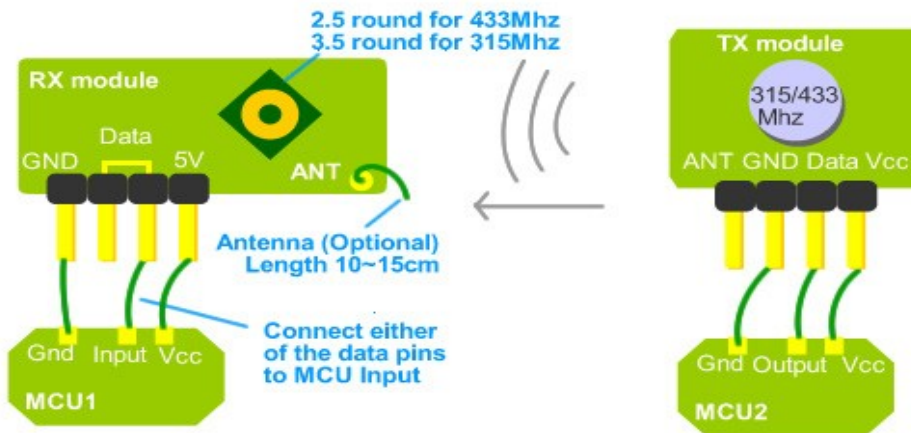


The 433MHz RF link kit is consisted of transmitter and receiver, popularly used for remote control.

Features

- Frequency: 433Mhz
- Modulation: ASK
- Receiver Data Output: High - 1/2 Vcc, Low - 0.7v
- Transmitter Input Voltage: 3-12V (high voltage = more transmitting power)

Usage



Demostration scheme of 433/315Mhz RF kit

The popular link is like this: MCU -> Encoder -> Transmitter ----- Receiver -> Decoder -> MCU

Encoder and Decoder are optional, their existence is to

ANEXO 17. DATASHEET DEL INTEGRADO HT12E

Features

- Operating voltage
 - 2.4V~5V for the HT12A
 - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1μA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
 - Four words for the HT12E
 - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- Pair with Holtek's 2¹² series of decoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2¹² encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12-N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header

bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a \overline{TE} trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2¹² series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

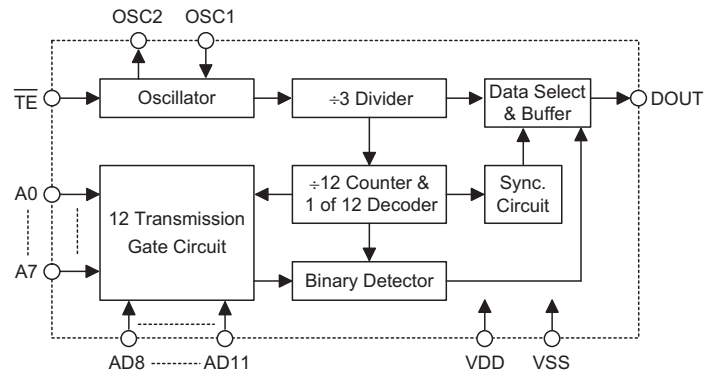
Function Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Carrier Output	Negative Polarity	Package
HT12A	8	0	4	455kHz resonator	D8~D11	38kHz	No	18DIP, 20SOP
HT12E	8	4	0	RC oscillator	\overline{TE}	No	No	18DIP, 20SOP

Note: Address/Data represents pins that can be either address or data according to the application requirement.

Block Diagram

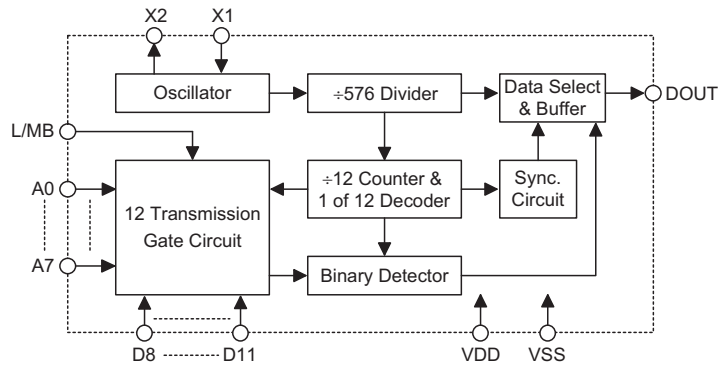
\overline{TE} Trigger

HT12E



DATA Trigger

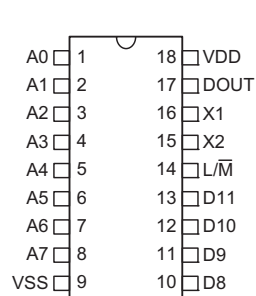
HT12A



Note: The address data pins are available in various combinations (refer to the address/data table).

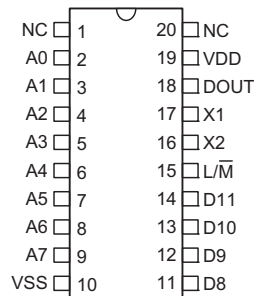
Pin Assignment

**8-Address
4-Data**



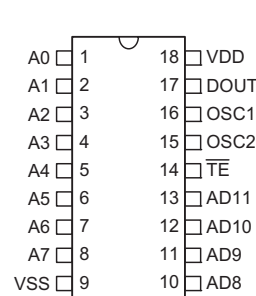
**HT12A
-18 DIP-A**

**8-Address
4-Data**



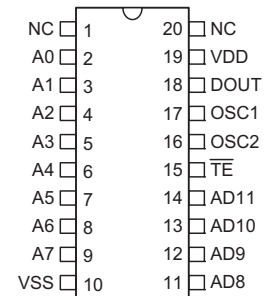
**HT12A
-20 SOP-A**

**8-Address
4-Address/Data**



**HT12E
-18 DIP-A**

**8-Address
4-Address/Data**



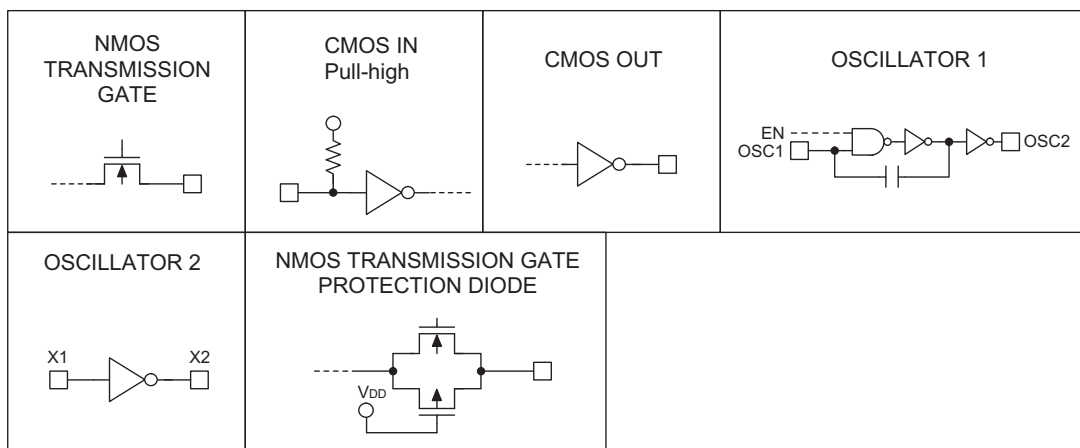
**HT12E
-20 SOP-A**

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A7	I	CMOS IN Pull-high (HT12A)	Input pins for address A0~A7 setting These pins can be externally set to VSS or left open
		NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	
AD8~AD11	I	NMOS TRANSMISSION GATE PROTECTION DIODE (HT12E)	Input pins for address/data AD8~AD11 setting These pins can be externally set to VSS or left open
D8~D11	I	CMOS IN Pull-high	Input pins for data D8~D11 setting and transmission enable, active low These pins should be externally set to VSS or left open (see Note)
DOUT	O	CMOS OUT	Encoder data serial transmission output
$\overline{L/M}$	I	CMOS IN Pull-high	Latch/Momentary transmission format selection pin: Latch: Floating or VDD Momentary: VSS
\overline{TE}	I	CMOS IN Pull-high	Transmission enable, active low (see Note)
OSC1	I	OSCILLATOR 1	Oscillator input pin
OSC2	O	OSCILLATOR 1	Oscillator output pin
X1	I	OSCILLATOR 2	455kHz resonator oscillator input
X2	O	OSCILLATOR 2	455kHz resonator oscillator output
VSS	I	—	Negative power supply, ground
VDD	I	—	Positive power supply

Note: D8~D11 are all data input and transmission enable pins of the HT12A.

\overline{TE} is a transmission enable pin of the HT12E.

Approximate Internal Connections


Absolute Maximum Ratings

Supply Voltage (HT12A)	$V_{SS}-0.3V$ to $V_{SS}+5.5V$	Supply Voltage (HT12E)	$-0.3V$ to $13V$
Input Voltage	$V_{SS}-0.3$ to $V_{DD}+0.3V$	Storage Temperature	$-50^{\circ}C$ to $125^{\circ}C$
Operating Temperature	$-20^{\circ}C$ to $75^{\circ}C$		

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics
HT12A
 $T_a=25^{\circ}C$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	3	5	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		5V		—	0.1	1	μA
I_{DD}	Operating Current	3V	No load $f_{OSC}=455kHz$	—	200	400	μA
		5V		—	400	800	μA
I_{DOUT}	Output Drive Current	5V	$V_{OH}=0.9V_{DD}$ (Source)	-1	-1.6	—	mA
			$V_{OL}=0.1V_{DD}$ (Sink)	2	3.2	—	mA
V_{IH}	"H" Input Voltage	—	—	$0.8V_{DD}$	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	$0.2V_{DD}$	V
R_{DATA}	D8-D11 Pull-high Resistance	5V	$V_{DATA}=0V$	—	150	300	$k\Omega$

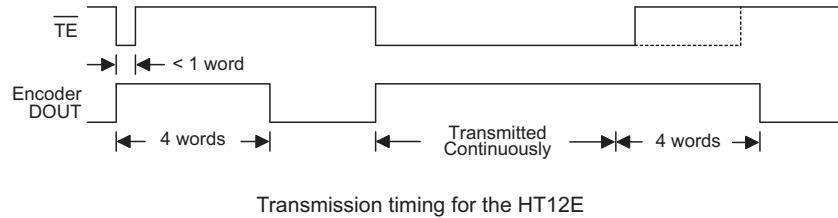
HT12E
 $T_a=25^{\circ}C$

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	3V	No load, $f_{OSC}=3kHz$	—	40	80	μA
		12V		—	150	300	μA
I_{DOUT}	Output Drive Current	5V	$V_{OH}=0.9V_{DD}$ (Source)	-1	-1.6	—	mA
			$V_{OL}=0.1V_{DD}$ (Sink)	1	1.6	—	mA
V_{IH}	"H" Input Voltage	—	—	$0.8V_{DD}$	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	$0.2V_{DD}$	V
f_{OSC}	Oscillator Frequency	5V	$R_{OSC}=1.1M\Omega$	—	3	—	kHz
R_{TE}	\overline{TE} Pull-high Resistance	5V	$V_{TE}=0V$	—	1.5	3	$M\Omega$

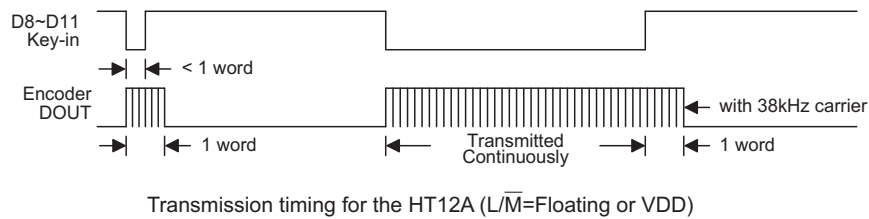
Functional Description

Operation

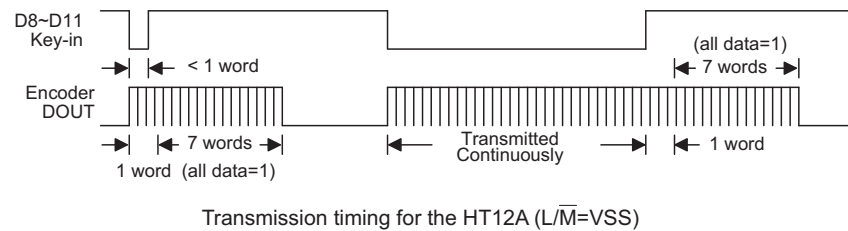
The 2¹² series of encoders begin a 4-word transmission cycle upon receipt of a transmission enable (\overline{TE} for the HT12E or D8~D11 for the HT12A, active low). This cycle will repeat itself as long as the transmission enable (\overline{TE} or D8~D11) is held low. Once the transmission enable returns high the encoder output completes its final cycle and then stops as shown below.



Transmission timing for the HT12E



Transmission timing for the HT12A (L/\overline{M} =Floating or VDD)

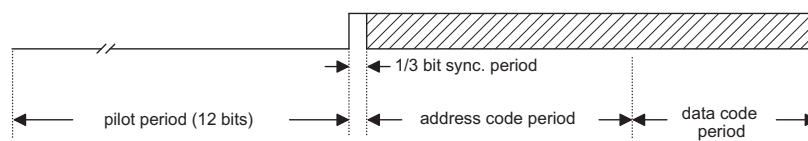


Transmission timing for the HT12A (L/\overline{M} =VSS)

Information Word

If $L/\overline{M}=1$ the device is in the latch mode (for use with the latch type of data decoders). When the transmission enable is removed during a transmission, the DOUT pin outputs a complete word and then stops. On the other hand, if $L/\overline{M}=0$ the device is in the momentary mode (for use with the momentary type of data decoders). When the transmission enable is removed during a transmission, the DOUT outputs a complete word and then adds 7 words all with the "1" data code.

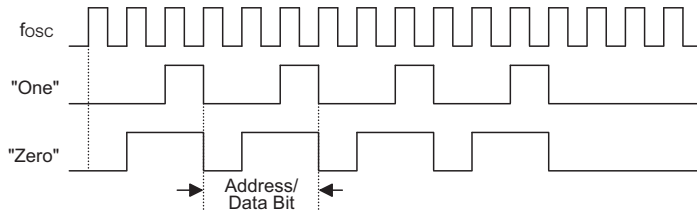
An information word consists of 4 periods as illustrated below.



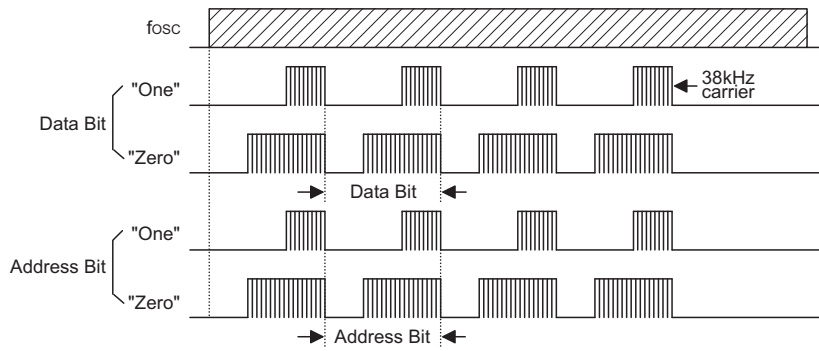
Composition of information

Address/Data Waveform

Each programmable address/data pin can be externally set to one of the following two logic states as shown below.



Address/Data bit waveform for the HT12E



Address/Data bit waveform for the HT12A

The address/data bits of the HT12A are transmitted with a 38kHz carrier for infrared remote controller flexibility.

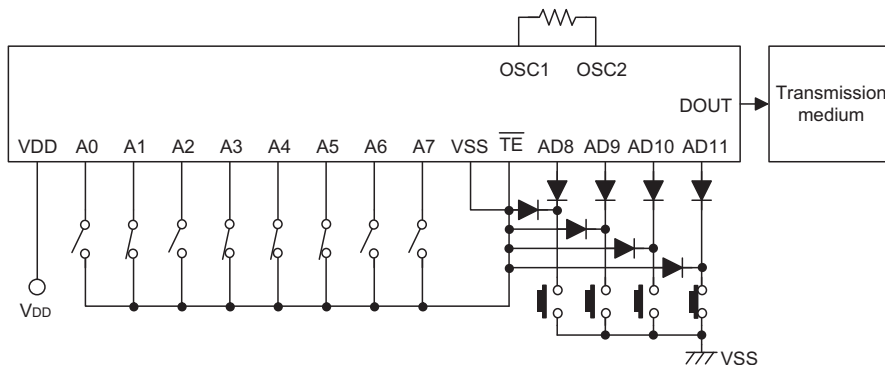
Address/Data Programming (Preset)

The status of each address/data pin can be individually pre-set to logic "high" or "low". If a transmission-enable signal is applied, the encoder scans and transmits the status of the 12 bits of address/data serially in the order A0 to AD11 for the HT12E encoder and A0 to D11 for the HT12A encoder.

During information transmission these bits are transmitted with a preceding synchronization bit. If the trigger signal is not applied, the chip enters the standby mode and consumes a reduced current of less than $1\mu A$ for a supply voltage of 5V.

Usual applications preset the address pins with individual security codes using DIP switches or PCB wiring, while the data is selected by push buttons or electronic switches.

The following figure shows an application using the HT12E:



The transmitted information is as shown:

Pilot & Sync.	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11
	1	0	1	0	0	0	1	1	1	1	1	0

Address/Data Sequence

The following provides the address/data sequence table for various models of the 2¹² series of encoders.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12A	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12E	A0	A1	A2	A3	A4	A5	A6	A7	AD8	AD9	AD10	AD11

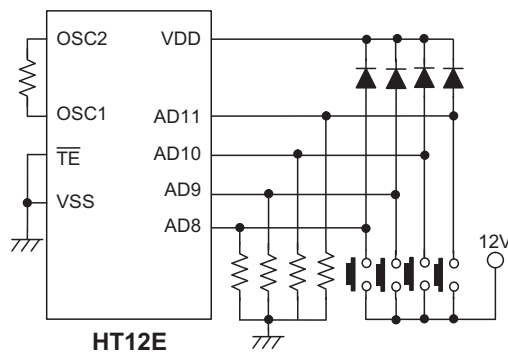
Transmission Enable

For the HT12E encoders, transmission is enabled by applying a low signal to the \overline{TE} pin. For the HT12A encoders, transmission is enabled by applying a low signal to one of the data pins D8~D11.

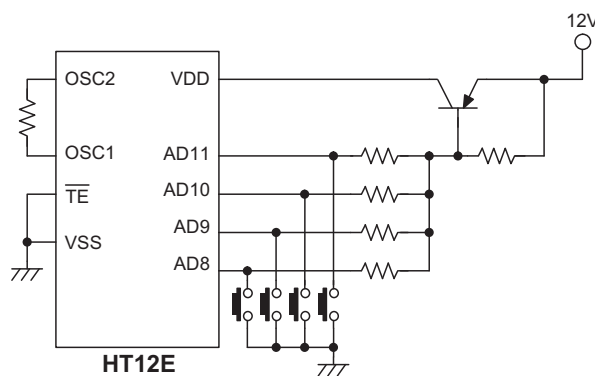
Two Erroneous HT12E Application Circuits

The HT12E must follow closely the application circuits provided by Holtek (see the "Application circuits").

- Error: AD8~AD11 pins input voltage > V_{DD}+0.3V



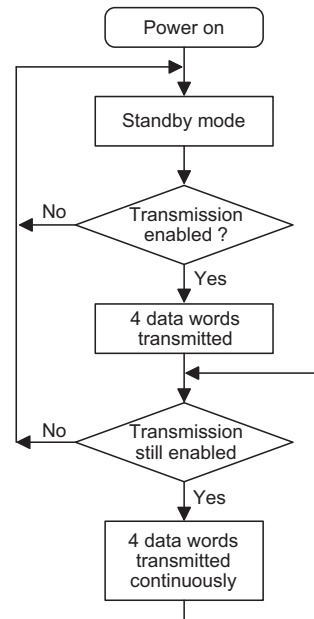
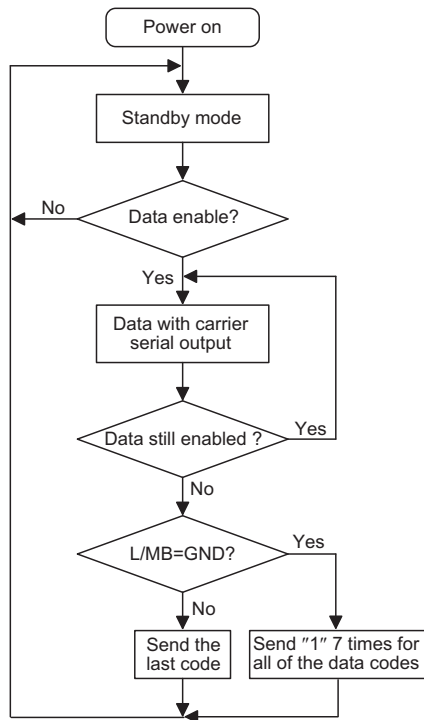
- Error: The IC's power source is activated by pins AD8~AD11



Flowchart

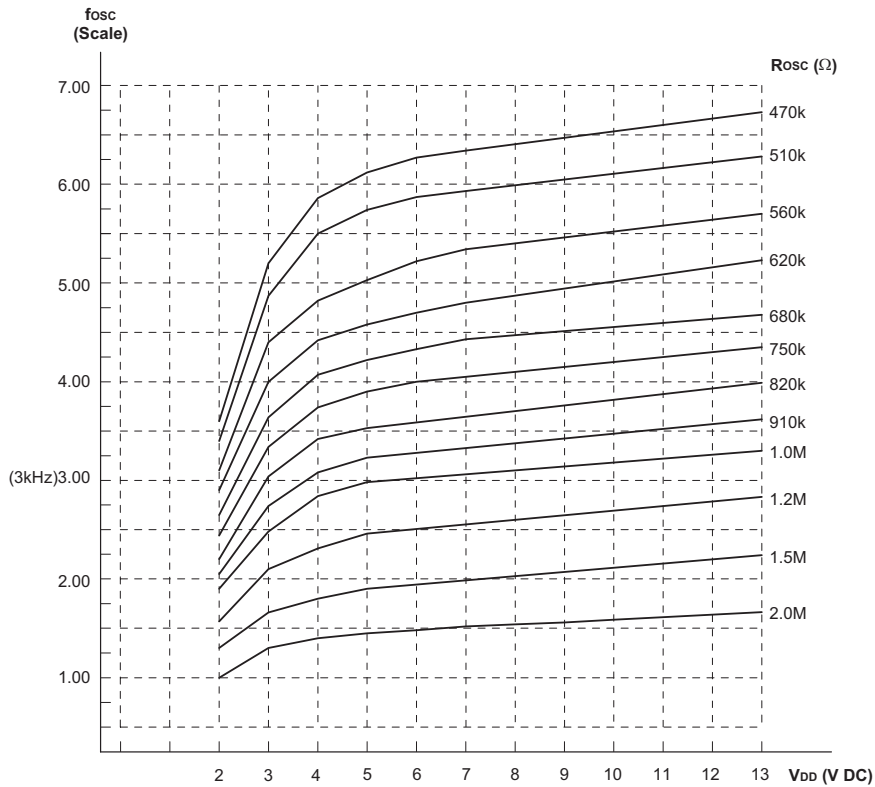
• HT12A

• HT12E



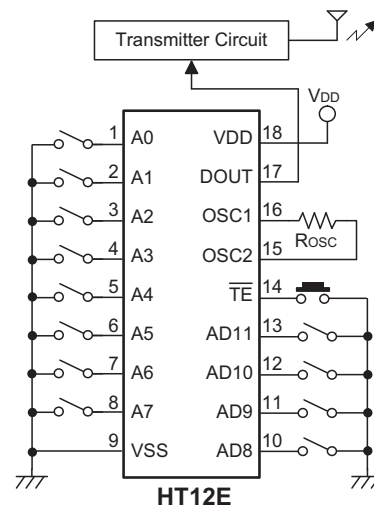
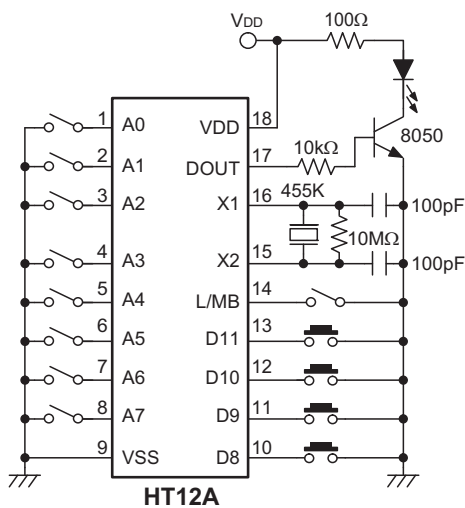
Note: D8~D11 are transmission enables of the HT12A.
 $\overline{\text{TE}}$ is the transmission enable of the HT12E.

Oscillator Frequency vs. Supply Voltage



The recommended oscillator frequency is $f_{OSC D} (\text{decoder}) \cong 50 f_{OSCE} (\text{HT12E encoder})$
 $\cong \frac{1}{3} f_{OSCE} (\text{HT12A encoder})$

Application Circuits



ANEXO 18. DATASHEET DEL INTEGRADO HT12D

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
 - HT12D: 8 address bits and 4 data bits
 - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components
- Pair with Holtek's 2¹² series of encoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2¹² decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2¹² series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2¹² series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continu-

ously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

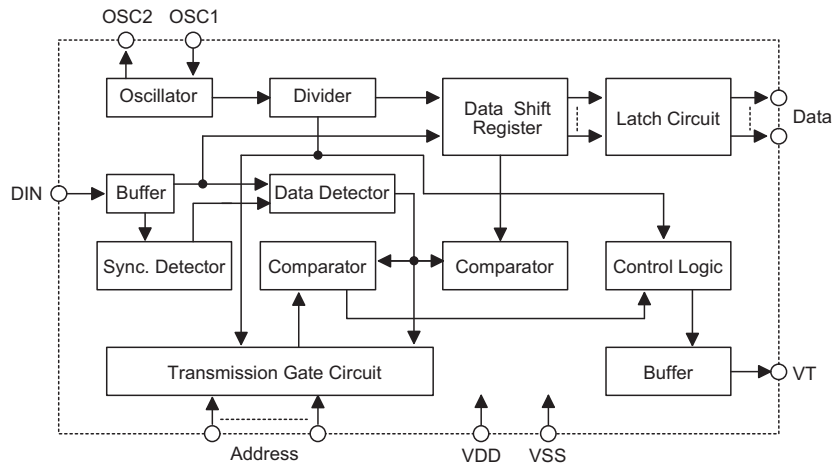
The 2¹² series of decoders are capable of decoding informations that consist of N bits of address and 12–N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

Selection Table

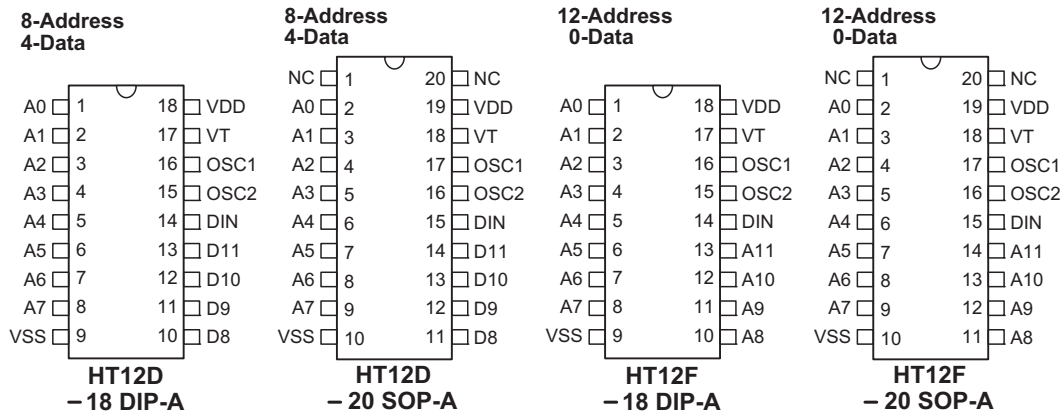
Function Part No.	Address No.	Data		VT	Oscillator	Trigger	Package
		No.	Type				
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP

Notes: Data type: L stands for latch type data output.

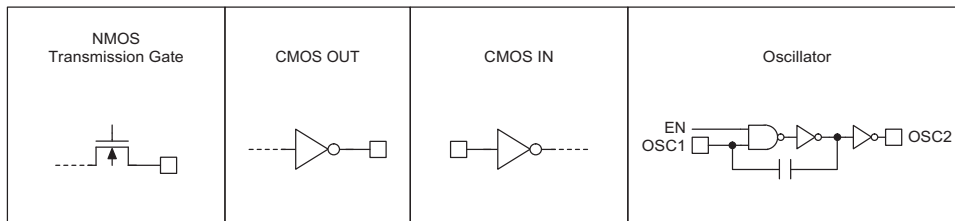
VT can be used as a momentary data output.

Block Diagram


Note: The address/data pins are available in various combinations (see the address/data table).

Pin Assignment

Pin Description

Pin Name	I/O	Internal Connection	Description
A0~A11 (HT12F)	I	NMOS Transmission Gate	Input pins for address A0~A11 setting These pins can be externally set to VSS or left open.
A0~A7 (HT12D)			Input pins for address A0~A7 setting These pins can be externally set to VSS or left open.
D8~D11 (HT12D)	O	CMOS OUT	Output data pins, power-on state is low.
DIN	I	CMOS IN	Serial data input pin
VT	O	CMOS OUT	Valid transmission, active high
OSC1	I	Oscillator	Oscillator input pin
OSC2	O	Oscillator	Oscillator output pin
VSS	—	—	Negative power supply, ground
VDD	—	—	Positive power supply

Approximate internal connection circuits

Absolute Maximum Ratings

Supply Voltage	-0.3V to 13V	Storage Temperature	-50°C to 125°C
Input Voltage	$V_{SS}-0.3$ to $V_{DD}+0.3V$	Operating Temperature	-20°C to 75°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

Electrical Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V _{DD}	Conditions				
V _{DD}	Operating Voltage	—	—	2.4	5	12	V
I _{STB}	Standby Current	5V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I _{DD}	Operating Current	5V	No load, f _{OSC} =150kHz	—	200	400	μA
I _O	Data Output Source Current (D8~D11)	5V	V _{OH} =4.5V	-1	-1.6	—	mA
	Data Output Sink Current (D8~D11)	5V	V _{OL} =0.5V	1	1.6	—	mA
I _{VT}	VT Output Source Current	5V	V _{OH} =4.5V	-1	-1.6	—	mA
	VT Output Sink Current		V _{OL} =0.5V	1	1.6	—	mA
V _{IH}	"H" Input Voltage	5V	—	3.5	—	5	V
V _{IL}	"L" Input Voltage	5V	—	0	—	1	V
f _{OSC}	Oscillator Frequency	5V	R _{OSC} =51kΩ	—	150	—	kHz

Functional Description

Operation

The 2¹² series of decoders provides various combinations of addresses and data pins in different packages so as to pair with the 2¹² series of encoders.

The decoders receive data that are transmitted by an encoder and interpret the first N bits of code period as addresses and the last 12-N bits as data, where N is the address code number. A signal on the DIN pin activates the oscillator which in turn decodes the incoming address and data. The decoders will then check the received address three times continuously. If the received address codes all match the contents of the decoder's local address, the 12-N bits of data are decoded to activate the output pins and the VT pin is set high to indicate a valid transmission. This will last unless the address code is incorrect or no signal is received.

The output of the VT pin is high only when the transmission is valid. Otherwise it is always low.

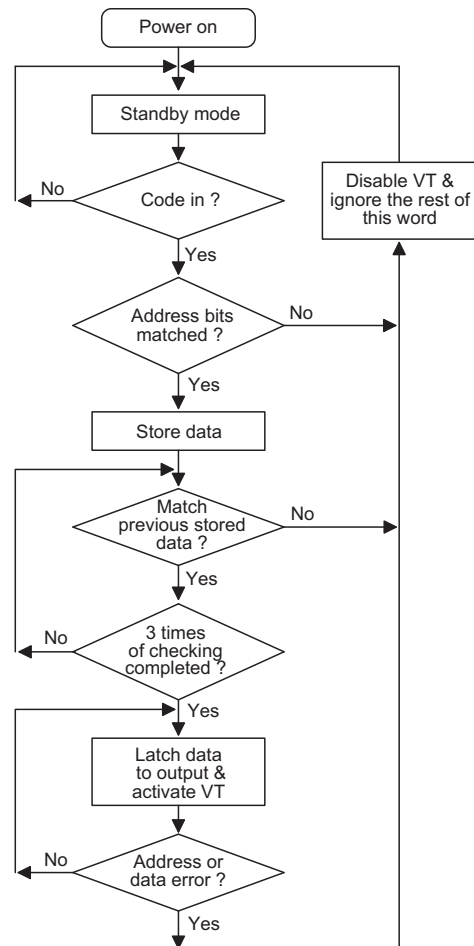
Output Type

Of the 2¹² series of decoders, the HT12F has no data output pin but its VT pin can be used as a momentary data output. The HT12D, on the other hand, provides 4 latch type data pins whose data remain unchanged until new data are received.

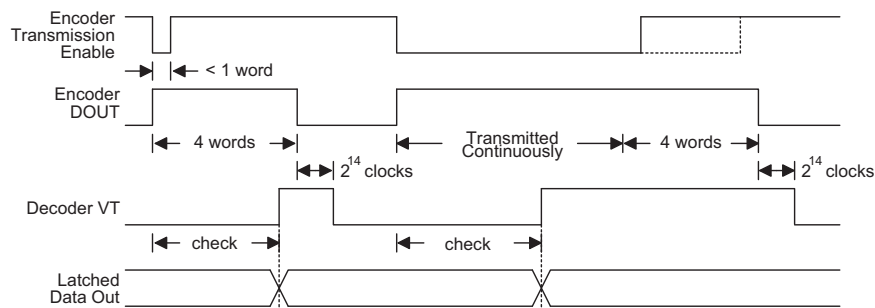
Part No.	Data Pins	Address Pins	Output Type	Operating Voltage
HT12D	4	8	Latch	2.4V~12V
HT12F	0	12	—	2.4V~12V

Flowchart

The oscillator is disabled in the standby state and activated when a logic "high" signal applies to the DIN pin. That is to say, the DIN should be kept low if there is no signal input.



Decoder Timing



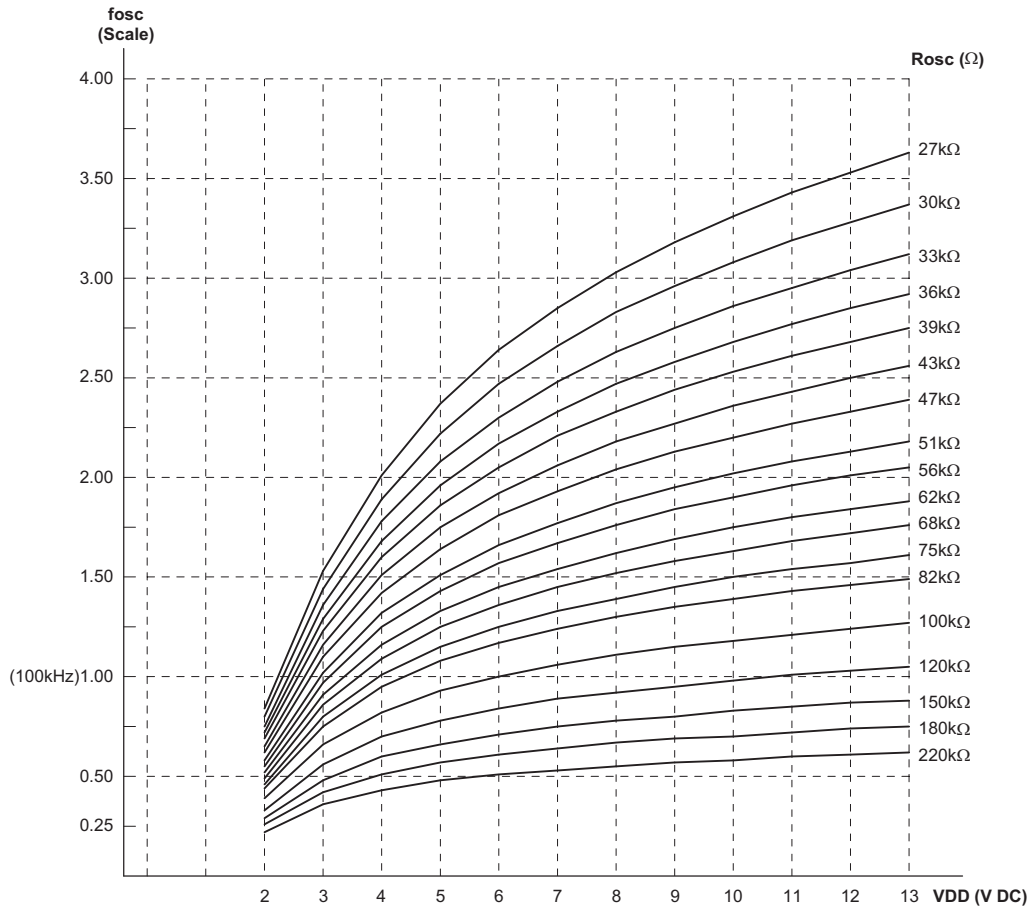
Encoder/Decoder Cross Reference Table

Decoders Part No.	Data Pins	Address Pins	VT	Pair Encoder	Package			
					Encoder		Decoder	
					DIP	SOP	DIP	SOP
HT12D	4	8	√	HT12A HT12E	18	20	18	20
HT12F	0	12	√	HT12A HT12E	18	20	18	20

Address/Data Sequence

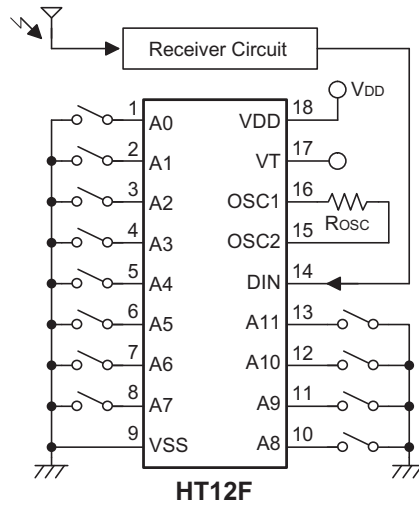
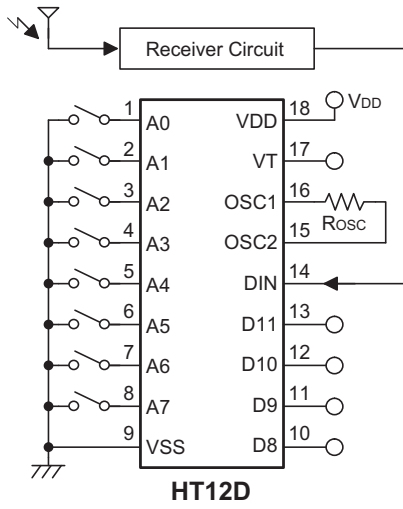
The following table provides address/data sequence for various models of the 2¹² series of decoders.

Part No.	Address/Data Bits											
	0	1	2	3	4	5	6	7	8	9	10	11
HT12D	A0	A1	A2	A3	A4	A5	A6	A7	D8	D9	D10	D11
HT12F	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11

Oscillator Frequency Vs Supply Voltage


Note: The recommended oscillator frequency is f_{oscD} (decoder) $\cong 50 f_{oscE}$ (HT12E encoder)
 $\cong \frac{1}{3} f_{oscE}$ (HT12A encoder).

Application Circuits



ANEXO 19. DATASHEET WS2812B



WS2812

Intelligent control LED
integrated light source

Features and Benefits

- Control circuit and RGB chip are integrated in a package of 5050 components, form a complete control of pixel point.
- Built-in signal reshaping circuit, after wave reshaping to the next driver, ensure wave-form distortion not accumulate.
- Built-in electric reset circuit and power lost reset circuit.
- Each pixel of the three primary color can achieve 256 brightness display, completed 16777216 color full color display, and scan frequency not less than 400Hz/s.
- Cascading port transmission signal by single line.
- Any two point the distance more than 5m transmission signal without any increase circuit.
- When the refresh rate is 30fps, low speed model cascade number are not less than 512 points, high speed mode not less than 1024 points.
- Send data at speeds of 800Kbps.
- The color of the light were highly consistent, cost-effective..

Applications

- Full-color module, Full color soft lights a lamp strip.
- LED decorative lighting, Indoor/outdoor LED video irregular screen.

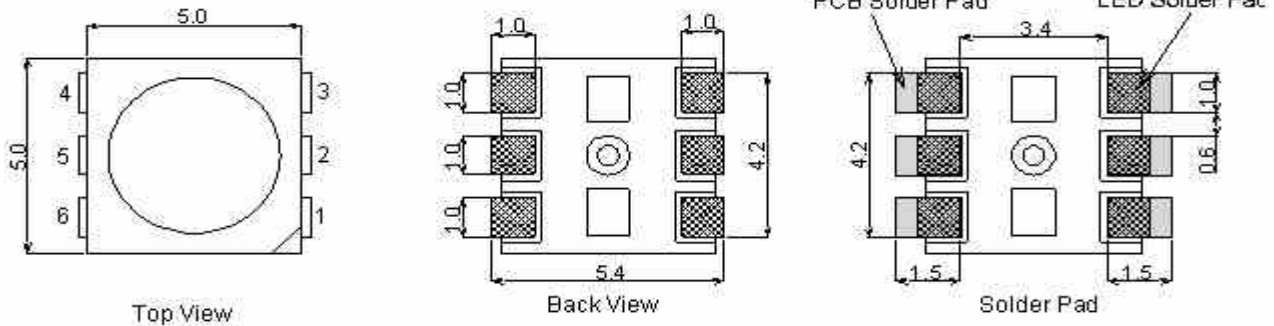
General description

WS2812 is a intelligent control LED light source that the control circuit and RGB chip are integrated in a package of 5050 components. It internal include intelligent digital port data latch and signal reshaping amplification drive circuit. Also include a precision internal oscillator and a 12V voltage programmable constant current control part, effectively ensuring the pixel point light color height consistent.

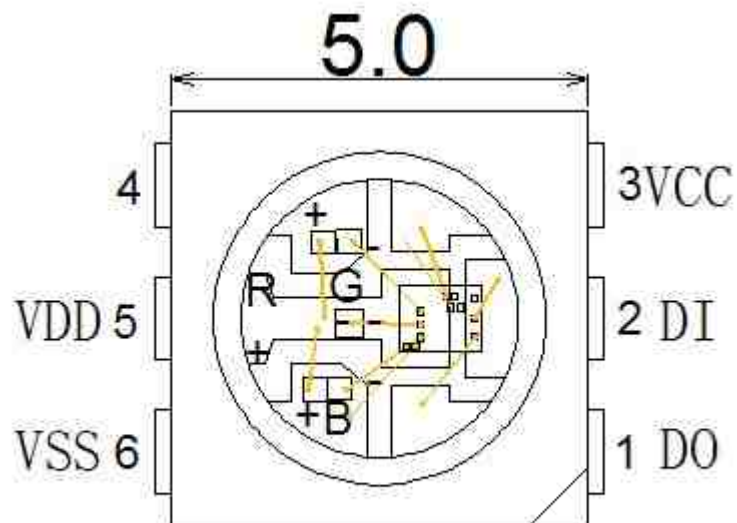
The data transfer protocol use single NZR communication mode. After the pixel power-on reset, the DIN port receive data from controller, the first pixel collect initial 24bit data then sent to the internal data latch, the other data which reshaping by the internal signal reshaping amplification circuit sent to the next cascade pixel through the DO port. After transmission for each pixel, the signal to reduce 24bit. pixel adopt auto reshaping transmit technology, making the pixel cascade number is not limited the signal transmission, only depend on the speed of signal transmission.

LED with low driving voltage, environmental protection and energy saving, high brightness, scattering angle is large, good consistency, low power, long life and other advantages. The control chip integrated in LED above becoming more simple circuit, small volume, convenient installation.

Mechanical Dimensions



PIN configuration



PIN function

NO.	Symbol	Function description
1	DOUT	Control data signal output
2	DIN	Control data signal input
3	VCC	Power supply control circuit
4	NC	
5	VDD	Power supply LED
6	VSS	Ground

Absolute Maximum Ratings

<http://www.world-semi.com>



WS2812

Intelligent control LED
integrated light source

Parameter	Symbol	Ratings	Unit
Power supply voltage	V _{CC}	+6.0~+7.0	V
Power supply voltage	V _{DD}	+6.0~+7.0	V
Input voltage	V _I	-0.5~V _{DD} +0.5	V
Operation junction temperature	T _{opt}	-25~+80	°C
Storage temperature range	T _{stg}	-55~+150	°C

Electrical Characteristics (T_A=-20~+70°C, V_{DD}=4.5~5.5V, V_{SS}=0V, unless otherwise specified)

Parameter	Symbol	conditions	Min	Tpy	Max	Unit
Low voltage output current	I _{OL}	R _{OUT}	—	18.5	—	mA
	I _{dout}	V _O =0.4V, D _{OUT}	10	—	—	mA
Input current	I _I	V _I =V _{DD} /V _{SS}	—	—	±1	μA
Input voltage level	V _{IH}	D _{IN} , SET	0.7V _{DD}	—	—	V
	V _{IL}	D _{IN} , SET	—	—	0.3 V _{DD}	V
Hysteresis voltage	V _H	D _{IN} , SET	—	0.35	—	V

Switching characteristics (T_A=-20~+70°C, V_{DD}=4.5~5.5V, V_{SS}=0V, unless otherwise specified)

Parameter	Symbol	Condition	Min	Tpy	Max	Unit
Operation frequency	F _{osc2}	—	—	800	—	KHz
Transmission delay time	t _{PLZ}	CL=15pF, DIN→DOUT, RL=10KΩ	—	—	300	ns
Fall time	t _{THZ}	CL=300pF, O _{UTR} /O _U TG/O _{UTB}	—	—	120	μs
Data transmission rate	F _{MAX}	Duty ratio 50%	400	—	—	Kbps
Input capacity	C _I	—	—	—	15	pF



WS2812

Intelligent control LED
integrated light source

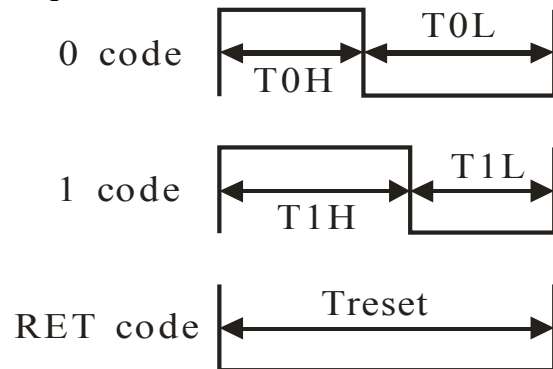
LED characteristic parameter

Emitting color	Wavelength(nm)	Luminous intensity(mcd)	Current(mA)	Voltage(V)
Red	620-630	550-700	20	1.8-2.2
Green	515-530	1100-1400	20	3.0-3.2
Blue	465-475	200-400	20	3.2-3.4

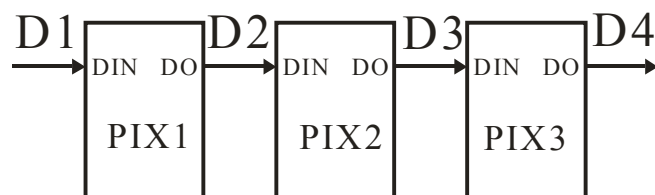
Data transfer time(TH+TL=1.25μs±600ns)

T0H	0 code ,high voltage time	0.35us	±150ns
T1H	1 code ,high voltage time	0.7us	±150ns
T0L	0 code , low voltage time	0.8us	±150ns
T1L	1 code ,low voltage time	0.6us	±150ns
RES	low voltage time	Above 50μs	

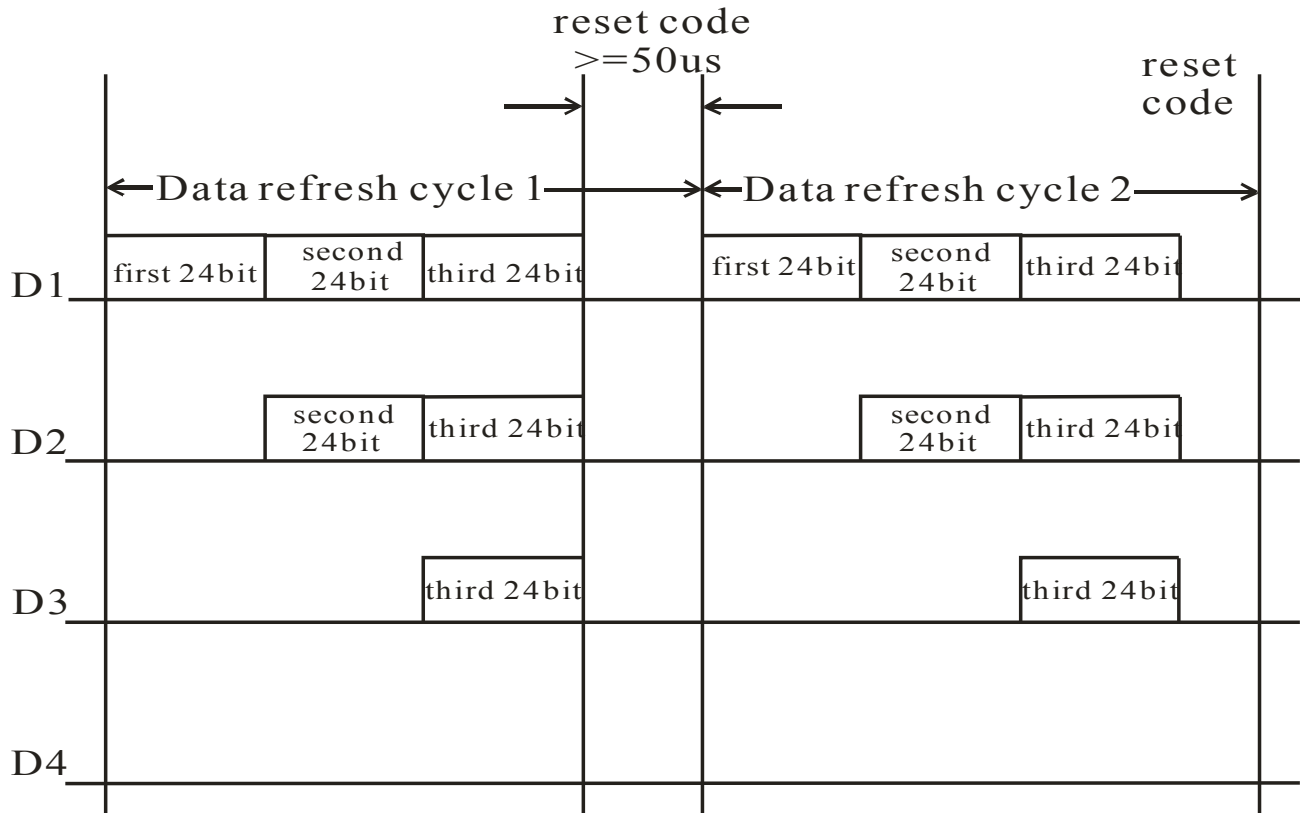
Sequence chart:



Cascade method:



Data transmission method:



Note: The data of D1 is send by MCU, and D2, D3, D4 through pixel internal reshaping amplification to transmit.

Composition of 24bit data:

G7	G6	G5	G4	G3	G2	G1	G0	R7	R6	R5	R4	R3	R2	R1	R0	B7	B6	B5	B4	B3	B2	B1	B0
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Note: Follow the order of GRB to sent data and the high bit sent at first.

Typical application circuit:

