



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“SISTEMA DE CONTROL DE ACCESO A
TERMINAL DE PASAJEROS INTERPROVINCIAL”

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del TÍTULO de:

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

**JONATHAN JAVIER VELA FAJARDO
RAUL ALEJANDRO CEBALLOS CEDEÑO**

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTO

Queremos dedicar este espacio para todos aquellos que han estado junto a nosotros durante una larga y grata etapa politécnica, en las buenas y en las malas, en las farras y en las amanecidas de estudio, en proyectos y en almuerzos, en tardes deportivas y horas de exámenes.

A aquellos profesores que tuvieron la paciencia necesaria y suficiente para poner su granito de conocimiento, y aportar a los hombres que somos hoy en día.

A nuestro exigente y bien intencionado tutor de tesis, y a la gran familia que nos precede.

DEDICATORIA

Este espacio muy especial se lo dedico de entero a Dios quien a pesar de mis flaquezas, errores, desganos y caídas supo hacerme entender que sin él no somos nadie, supo hacerme ver que su existencia es pilar fundamental en mis triunfos, victorias y mi realización como profesional, padre de familia, cabeza de un hogar, buen hijo y sobretodo hombre de bien.

Quiero extender mis consideraciones a mi compañero Jonathan Vela quien en su forma de ver la vida me recordó siempre que el trabajo a presión es bueno y que el cumplimiento diario de nuestras obligaciones es importante en el progreso de nuestra meta final.

Un gran cariño a mi tutor de tesis, quien supo comprendernos y corregirnos cuando se debe, a mis hermanos que siempre confiaron en mi e incluso cuando pensé que solo perdía mi tiempo, y a mi padre, madre y tía que nunca perdieron las esperanzas de verme realizado en esta meta cumplida. Pero en especial a mi hijo.

Raúl Alejandro Ceballos Cedeño

A la gracia del azar que sin darse cuenta nos brinda más y más tiempo para poco a poco cumplir con los objetivos que nos planteamos.

A todos quienes permitieron que este día sea posible. A mis entrañables amigos.

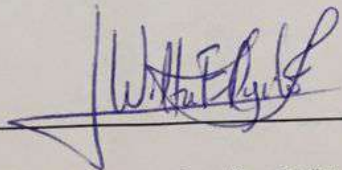
Y a estas dos frases del gran Никола Тесла

“La vida es y siempre seguirá siendo una ecuación incapaz de resolver, pero tiene ciertos factores que conocemos”.

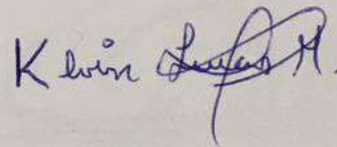
“El futuro mostrará los resultados y juzgará a cada uno de acuerdo a sus logros”.

Jonathan Javier Vela Fajardo

COMUNIDAD EDUCATIVA
CORONA
TRIBUNAL DE EVALUACIÓN




Ph. D. Wilton Agila Gálvez
PROFESOR EVALUADOR



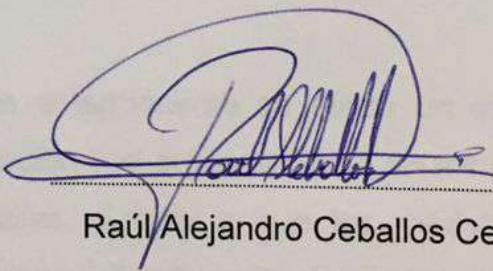
Ing. Kevin Lucas Marcillo
PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Jonathan Javier Vela Fajardo



Raúl Alejandro Ceballos Cedeño

RESUMEN

El control de paso de personas se ha convertido con el tiempo en una necesidad para muchos recintos, entre ellos los terminales terrestres. Los torniquetes facilitan el control de acceso de personas, los mismos que son un método seguro y rápido para garantizar la seguridad en el recinto, evitando el paso a personas no autorizadas.

Este proyecto trata de la implementación de controladoras en reemplazo de las ya existentes en el terminal de Duran, las cuales después de haber sido instalados hace 4 años empezaron a presentar inconvenientes en su funcionamiento por razones que se tratarán en este documento, entre ellas la utilización de elementos con poca duración de vida útil, soldaduras mal efectuadas, y nulo mantenimiento del hardware.

Vale destacar que en las tarjetas diseñadas anteriormente se utilizó un módulo genérico, el cual tenía embebidas la comunicación y el procesamiento de los datos, entre otras funciones adicionales desconocidas, debido a que los módulos no contienen ningún serial, entender y reprogramar el funcionamiento de los mismos era una tarea complicada.

En base a lo mencionado anteriormente, se optó por diseñar e implementar nuevas controladoras que trabajen teniendo como plataforma el software que se vendió junto con las controladoras, y de esta forma evitar el trabajar en el software. Para la implementación de estas nuevas controladoras se utilizan tecnologías existentes poco implementadas como LCD con módulos I2C, chip ATMEGA de la casa ATMEL y Netburner.

ÍNDICE

AGRADECIMIENTO	II
DEDICATORIA	III
TRIBUNAL DE EVALUACIÓN	III
DECLARACIÓN EXPRESA	V
RESUMEN	VI
CAPÍTULO 1	1
1. ANTECEDENTES Y JUSTIFICACIÓN.....	1
1.1 Descripción del problema	1
1.2 Solución propuesta	2
1.3 Justificación del proyecto.....	3
1.4 Objetivos del proyecto	3
CAPÍTULO 2.....	5
2. DISEÑO DEL C-TTD	5
2.1 Arquitectura del C-TTD.....	5
2.1.1 Controladora Electrónica.....	5
2.1.2 Comunicación.....	6
2.1.3 Controlador ATMEGA328P	7
2.2 Características.....	8
2.2.1 Comunicación Serial RS-232	8
2.2.2 MAX3232CSE	9
2.2.3 ATMEL ATMEGA328P-PU.....	10
2.2.4 Convertidor de voltaje DC-DC módulo mp1584	11
2.2.5 Relés	13
2.2.6 Final de Carrera	13
2.2.7 Luces piloto	14
2.2.8 Solenoide	15
2.2.9 Lector de código de barras.....	16

2.2.10 Integrado ULN2003A.....	18
2.2.11 NetBurner SBL2E de modulo Ethernet.....	19
2.2.12 Display LCD Serial I2C 2x16	21
2.3 Arquitectura del Software	22
2.3.1 Recepción de datos.....	22
2.3.2 Comparación de datos	23
2.3.3 Envío de datos	23
2.4 Integración e implementación del Sistema	24
2.4.1 Bloque de alimentación	24
2.4.2 Bloque de comunicación serial.....	25
2.4.3 Bloque de comunicación red	27
2.4.4 Bloque del microcontrolador Atmel atmega328p-pu.....	28
2.4.5 Bloque de comunicación I2C.....	30
2.4.6 Bloque de sensores.....	30
2.4.7 Bloque controlador de salidas	31
2.4.8 Bloque de actuadores o relay.....	32
CAPÍTULO 3.....	34
3. PRUEBAS Y VALIDACIÓN DEL SISTEMA.....	34
3.1 Pruebas de laboratorio	34
3.1.1 Pruebas de indicadores de finales de carrera	34
3.1.2 Pruebas de indicadores de accionamiento de solenoides.....	35
3.1.3 Pruebas de comunicación computador- Controladora TTD.....	36
3.1.4 Pruebas de mensajes en el display	36
3.1.5 Pruebas de funcionamiento del sistema.....	39
3.2 Pruebas y validación del sistema.....	41
3.2.1 Pruebas de finales de carrera	41
3.2.2 Pruebas de accionamiento de solenoides	42
3.2.3 Pruebas de comunicación área de sistemas - Controladora TTD .	44
3.2.4 Pruebas de luces piloto	44
3.2.5 Pruebas de funcionamiento del sistema implementado	45

CONCLUSIONES Y RECOMENDACIONES	49
BIBLIOGRAFÍA.....	50

CAPÍTULO 1

1. ANTECEDENTES Y JUSTIFICACIÓN

Generar un control de paso de personas en el terminal de Duran nos lleva a exponer los siguientes puntos:

1.1 Descripción del problema

Debido a que en Durán convergen gran cantidad de vías se construye en el 2011 el TTD para satisfacer el déficit de servicio de transporte terrestre de los habitantes de dicho cantón, el cual garantiza la movilización de las personas y los bienes, en condiciones de calidad y seguridad.

Con el pasar del tiempo se han ido ampliando el número de líneas dentro del terminal y con esto el número de usuarios que usan estos servicios, por lo tanto la infraestructura ha tenido que ir creciendo y con ello los sistemas de seguridad, marcación, facturación Y control de flujos de personas han tenido que ser modificados para que se adapten a los nuevos requerimientos.

En muchos casos el cambio de integradores de sistemas electrónicos y mecánicos crea una gran desventaja ya que debido a estos cambios se pierde información del sistema en general causando un problema al momento de querer diseñar un nuevo requerimiento o corregir un error, por lo tanto no se puede hacer el sistema más grande o más óptimo y a su vez que permita un mantenimiento y control anti-fallas de manera rápida y sencilla.

En el sistema anterior de acceso en el TTD, el controlador tenía embebido la adquisición, procesamiento y envía de datos en un IC, siendo esto un problema al momento de diseñar o corregir el procesamiento de datos ya que no se conoce la arquitectura de la tarjeta; siendo más sencillo el cambio del controlador por completo.

Otra de las problemáticas que presentaba la tarjeta es conocida como soldadura fría, que quiere decir soldadura en mal estado, esto significa que la soldadura

ha perdido sus características de cuando se realizó en estado caliente. El principal problema de las soldaduras frías es que el circuito puede funcionar correctamente durante un tiempo, de incluso años, esto se produce debido a que se generan espacios huecos entre la soldadura y los contactos soldados lo que genera ruido el cual es fundamental en un sistema que depende de la comunicación. En ocasiones las soldaduras frías permiten un contacto entre la patilla y la pista de circuito impreso, a pesar de no estar realmente soldadas. De esta forma las soldaduras pueden ser válidas durante un período de tiempo más o menos largo, dependiendo de las condiciones físicas y ambientales del entorno en las que se encuentre el circuito: frío, calor, vibraciones mecánicas, etc.

También se observó una disminución en los faradios de los capacitores utilizados en la comunicación del lector con la tarjeta, así como de un alto factor de disipación en los capacitores. El factor de disipación se define como la relación de la resistencia equivalente en serie (E.S.R) con respecto a la reactancia capacitiva (X_c). A mayor resistencia, mayor factor de disipación, y en general peor es el capacitor. Debido a que los buenos capacitores tienen el factor de disipación con una cifra muy pequeña, frecuentemente se expresa en porcentajes.

$$\text{Factor disipacion} = \frac{\text{E.S.R}}{X_c} \quad (1.1)$$

1.2 Solución propuesta

De acuerdo al análisis de la problemática actual, se propone el diseño de un sistema de control centralizado escalable para el control de paso a las personas.

El sistema estará conformado por un servidor, el cual contendrá el software que permite enlazarse con la base de datos para identificar si el boleto leído por el lector RFID es válido y a su vez se encuentra ubicado en el ANDEN correcto y de esta forma permitir o no el paso al usuario. Por otro lado el torniquete será

controlado por una tarjeta C-TTD, la cual verifica mediante vía Ethernet si el código de barras adquirido por el lector es válido, en caso de serlo activa los solenoides permitiendo el paso al usuario y sino envía por la pantalla LCD el motivo por el cual no se habilita el acceso.

1.3 Justificación del proyecto

El fin de este proyecto es incorporar una solución óptima para el control de paso de las personas a los ANDENES del TTD, mediante la utilización de un servidor y sistemas electrónicos, además de esto que el sistema esté abierto a cambios en el futuro.

Por estas razones se diseñarán nuevas tarjetas controladoras de los torniquetes, las cuales van a corregir los errores de las antiguas, optimizar el proceso, y a su vez se les añadirán protecciones necesarias para el cuidado de los elementos electrónicos, también se incluirán indicadores para verificar el correcto funcionamiento de los dispositivos eléctricos y electrónicos inmersos en el sistema del torniquete y un indicador anti-falla de la comunicación con el servidor. Lo cual ahorrará tiempo, dinero de mantenimiento y actualizaciones de los mismos.

1.4 Objetivos del proyecto

Son los siguientes:

1.4.1 Objetivos generales

- Controlar el acceso completo de personas del TTD, adaptándolo a las operaciones de adquisición y presentación de datos apropiados para cada una de los torniquetes del TTD.
- Brindar capacidad para que los operarios encargados del sistema puedan verificar el funcionamiento correcto del torniquete.

- Diseñar un sistema escalable para cubrir el incremento del número de personas acorde a las necesidades futuras del TTD, dado el crecimiento de la población.

1.4.2 Objetivos específicos

- Diseñar y construir un controlador que interactúe con un servidor y los periféricos para permitir el acceso de las personas a los andenes.
- Diseñar y construir una red de comunicación de la tarjeta con el servidor para entregar y/o recibir los datos tanto del servidor como del lector RFID.

CAPÍTULO 2

2. DISEÑO DEL C-TTD

El siguiente capítulo tiene como objetivo brindar información de la operación de la tarjeta Controlador TTD además de explicar su funcionamiento en conjunto con el sistema autónomo que operan los torniquetes.

2.1 Arquitectura del C-TTD

El hardware requerido para cualquier sistema de control de acceso de persona consiste en un lector, en nuestro caso lector de código de barras, un controlador que procesa los datos y permite la comunicación con la base de datos para la identificación del ticket. El controlador contiene un juego integrado de reglas que le brinda la habilidad de controlar el hardware (periféricos) desde un entorno de programación. Desde el entorno de programación, usted puede visualizar y registrar datos, automatizar pruebas, enviar alarmas y habilitar salidas.

2.1.1 Controladora Electrónica

La controladora electrónica es una tarjeta especialmente diseñada para trabajar en conjunto con el sistema mecánico del torniquete y con la red hacia el software de gestión de tickets.

La controladora usa los periféricos mostrados en la Figura 2.1.

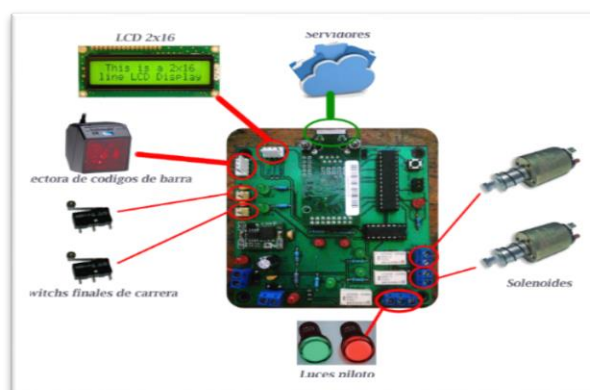


Figura 2.1 Controladora TTD.

En el LCD se mostraran los mensajes provistos por el software que se aloja en el servidor.

La lectora de códigos de barra leerá los tickets para poder suministrarlos al software del servidor.

Los switches finales de carrera indicaran cuando una persona pase por el torniquete.

Los solenoides trabajarán en conjunto con la parte mecánica y permitirán que el torniquete se libere o se cierre.

Las luces pilotos informaran cuando es factible pasar o no por el torniquete.

2.1.2 Comunicación

La tarjeta electrónica Controlador TTD fue diseñada específicamente para poder gestionar órdenes externas a través de una conexión de red de área local y poder controlar la operación de los torniquetes BOODOOM, además puede transmitir eventos con la misma red hacia el software de control.

Para ello se define esquema de funcionamiento mostrado en la Figura 2.2:

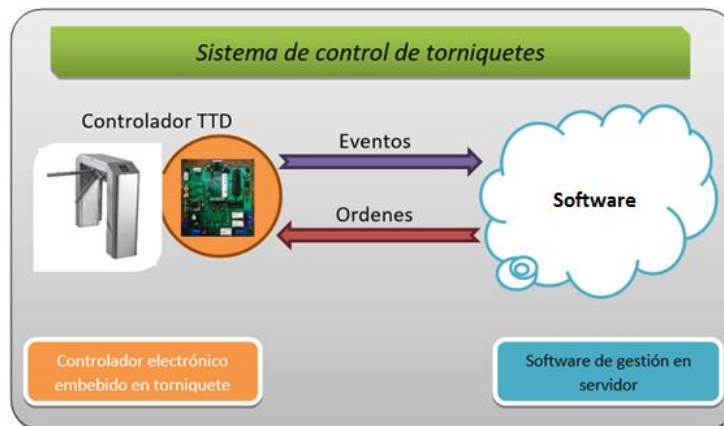


Figura 2.2 Interacción C-TTD con el Software

El esquema muestra el sistema de control de torniquetes en conjunto con el software de gestión el cual debe estar vinculada con una base de datos para poder administrar el paso de los usuarios a partir de tickets compuestos en su esencia por un código de barra.

2.1.3 Controlador ATMEGA328P

Son de alto rendimiento de los dispositivos que combinan la memoria de 32KB destello de la ISP con capacidades de lectura, mientras que de lectura-escritura, de 1 KB de EEPROM, 2 KB de SRAM RISC-basados, 23 de propósito general líneas I / O, de 32 años de uso general de trabajo registros, USART programable de serie, y mucho más. Atmel ATmega328 MCUs ejecutar instrucciones de gran alcance en un solo ciclo de reloj, permitiendo que el dispositivo para lograr rendimientos se acerca a 1 MIPS por MHz, mientras que el equilibrio de consumo de energía y velocidad de procesamiento.

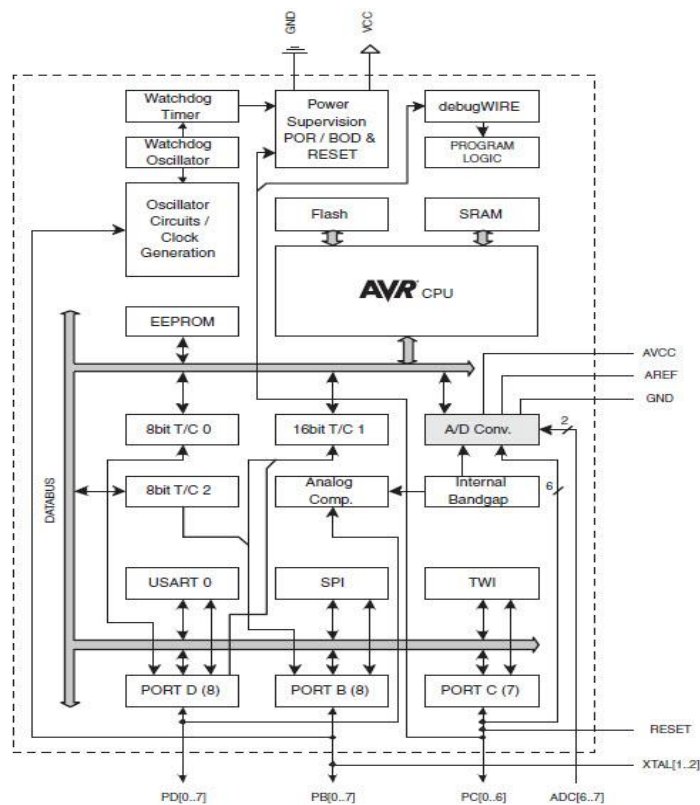


Figura 2.3 Arquitectura ATMEMEGA 328P

2.2 Características

A continuación se exponen las características más importantes de este dispositivo:

2.2.1 Comunicación Serial RS-232

Una comunicación serial RS-232 es una de las normas más utilizadas para establecer interfaz de comunicación en modo asincrónico con velocidades bajas y medias. Popularmente es conocida como RS-232C y es utilizada para la conexión entre equipos de datos a cortas distancias (originalmente menos de 16.4m) en un entorno ruidoso.

La norma RS-232 generalmente comunica un equipo terminal de datos (DTE) y el equipo de comunicación de datos (DCE). El esquema general de un sistema de comunicación de datos es el ilustrado a continuación:

Los DTE son los equipos que producen la señal de datos o son los receptores finales de la señal de datos.

Los DCE son los equipos que adecuan la señal de datos al medio de transmisión utilizado o reciben esta señal del medio de transmisión ofreciéndola de forma apropiada al receptor final.

La comunicación serial asincrónica se caracteriza por introducir un elemento de sincronización en cada dato transmitido, que consiste en un bit con valor '0' para indicar el comienzo de cada palabra y otro bit con valor '1' para indicar el final de las palabras. El 0 inicial se denomina bit de inicio o start y el 1 final se denomina bit de parada o stop. Cuando el transmisor hace una pausa porque no tiene palabras para transmitir, mantiene en su terminal de salida una secuencia de bits de parada, es decir, la salida permanece en 1 mientras dura la parada

El bit de paridad o parity a diferencia de los bits de inicio y parada, es un parámetro opcional (el ultimo bit después de los bits de datos) que se utiliza en las comunicaciones serie para determinar si el dispositivo remoto está recibiendo correctamente el carácter de datos que se

transmite, es decir, permite la detección de errores pero no los corrige. Existen dos variantes de este método, bit de paridad par y bit de paridad impar. Para que la lectura de los datos sea correcta ambos equipos deben estar configurados a la misma velocidad y demás parámetros y no exceder más allá de los 2 metros, pasado esta distancia los datos recibidos pueden no ser correctos debido a la pérdida de voltaje en el cable, ruido, etc.

2.2.2 MAX3232CSE

Las placas MAX3232CSE se utilizan para conectar un microcontrolador a los dispositivos RS-232 y tal como puertos serie de la PC.

Sus características principales son:

- Comunicación UART serial
- Bajo consumo de energía
- Alimentación entre 3.3v y 5v.

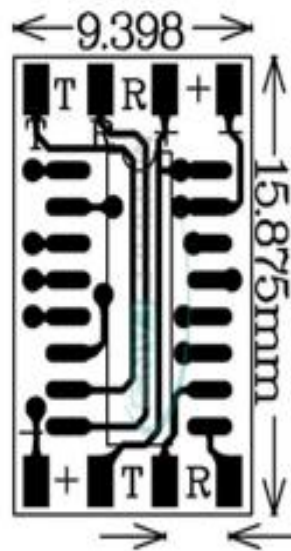


Figura 2.4 Integrado Max3232CSC.

Las características específicas del dispositivo son las siguientes:

- Modelo de la viruta: MAX3232CSE
- Rango de voltaje: 3 ~ 5 V
- Máxima Velocidad de transmisión: No más de 120 kbps
- Dimensión: 15.875mm x 9.398mm x 2.54mm
- Puerto Serie Mini RS232 a TTL Convertidor Adaptador Junta Módulo MAX3232 115200bps

2.2.3 ATMEL ATMEGA328P-PU

Estos Atmel MCU están diseñados para su uso en la automatización industrial y la automatización de viviendas y edificios, funcionan entre 1.8 y 5.5 voltios.

Las características más importantes son las siguientes:

- *Alto rendimiento, bajo consumo AVR microcontrolador de 8 bits.*
- *Advanced RISC Architecture.*
- *131 Instrucciones de gran alcance.*
- *Hasta 20 MIPS de rendimiento a 20MHz.*
- *En el chip multiplicador de 2 tiempos.*
- *Número de pines: 28/32.*
- *Max I/O Pins: 23.*
- *SPI: 2.*
- *UART: 1.*
- *ADC: 8 canales, resolución de 10 bits.*
- *Comparadores analógicos: 1.*
- *Flash (Kbytes): 32.*
- *EEPROM (Kbytes): 1.*
- *SRAM (Kbytes): 2.*
- *Temperatura. Rango: -40 a 85 ° C.*

Atmel ofrece una variedad de herramientas de desarrollo para la depuración y el dispositivo de programación. El AVR Dragón es un depurador En-Sistema de Atmel 8 bits y 32 bits Microcontroladores AVR

con con el SoC de depuración capacidad (TOC). Se puede realizar una depuración simbólica en todos los dispositivos con TOC con SPI, JTAG, PDI (dispositivos seleccionados), programación en serie de alta tensión, la programación en paralelo, y los modos aWire, y apoya la depuración utilizando interfaces SPI, JTAG, PDI. Un área de desarrollo permite a los diseñadores crear sus propios circuitos o incorpora puntos de la huella dispositivo deseado.



Figura 2.5 Integrado ATMEGA328P-PU

2.2.4 Convertidor de voltaje DC-DC módulo mp1584

Este circuito te permite tener un voltaje regulado a partir de una fuente de alimentación con un voltaje mayor, por ejemplo si tienes una fuente de 12V puedes regularlos a 5V, 3.3V, 2.2V, etc, para el uso con microcontroladores, Arduino, PICs, Raspberry Pi, fuentes variables, drivers para leds, etc.

Este módulo está basado en el Regulador DC-DC Step Down LM2596 que es un circuito integrado monolítico adecuado para el diseño fácil y conveniente de una fuente de conmutación tipo buck. Es capaz de conducir una corriente de hasta 3A. Maneja una carga con excelente regulación de línea y bajo voltaje de rizado. Este dispositivo está disponible con voltaje de salida ajustable. El módulo reduce al mínimo el uso de componentes externos para simplificar el diseño de fuentes de alimentación.

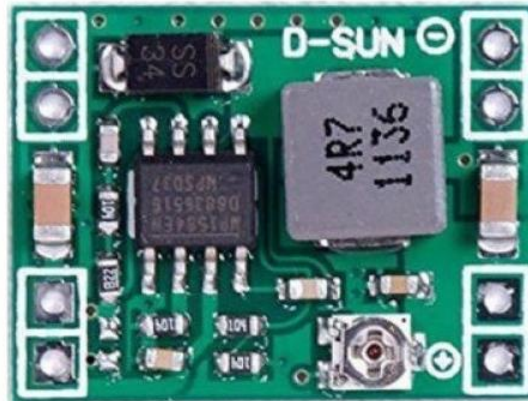


Figura 2.6 Modulo LM2596

Especificaciones:

- Temperatura de funcionamiento: -40 Celsius to +85 Celsius
- Se puede personalizar: Sí
- Número de modelo: Ultra-small DC-DC step-down module
- Voltaje de alimentación: 4.5-28V
- Tipo: Regulador de voltaje
- Aplicación: Varias
- Conversion efficiency: 96% (highest)
- Output ripple: 30mV (MAX)
- Voltaje de entrada: 4.5 ~ 28 V
- Tensión de salida: 0.8 V ~ 20 V
- Corriente de salida: 3A (Max)
- Eficiencia: 92% (Max)
- Frecuencia De conmutación: 1.5 MHz (Max), 1 MHz (Typ)
- Temperatura de funcionamiento: -20 a + 85
- Tamaño: 22mm * 17mm * 4mm

2.2.5 Relés

Un relé, de modo sencillo, es un interruptor accionado electrónicamente y se ubica en un circuito eléctrico para interrumpir o no el flujo de corriente eléctrica, según un circuito de control.

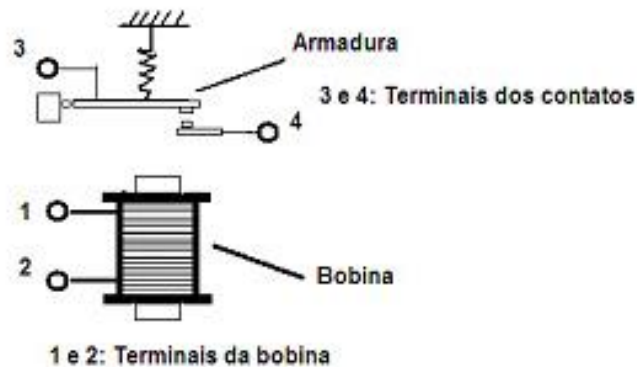


Figura 2.7 Esquema de un relé.

2.2.6 Final de Carrera

Dentro de los componentes electrónicos, se encuentra el final de carrera o sensor de contacto (también conocido como "interruptor de límite"), son dispositivos eléctricos, neumáticos o mecánicos situados al final del recorrido o de un elemento móvil, como por ejemplo una cinta transportadora, con el objetivo de enviar señales que puedan modificar el estado de un circuito.

Internamente pueden contener interruptores normalmente abiertos (NA), cerrados (NC) o conmutadores dependiendo de la operación que cumplan al ser accionados, de ahí la gran variedad de finales de carrera que existen en mercado.

.Un final de carrera se lo coloca, justamente, al final de un desplazamiento mecánico, y antes que se active el último interruptor que es la parada de emergencia.



Figura 2.8 Final de carrera.

2.2.7 Luces piloto



Figura 2.9 Luz piloto.

Especificaciones del artículo

- Tipo de artículo: Luces de indicador.
- Tiempo de vida continua: ≥ 30000 H.
- Color: rojos, verdes, azules, blancos y amarillos.
- Voltaje: AC 110V.
- Numero de modelo: AD56-22S.
- Frecuencia de trabajo: AC 50-60 Hz.
- Sobre voltaje permitido: 20%.
- Resistencia de aislamiento: $\geq 2M \Omega$.

2.2.8 Solenoide

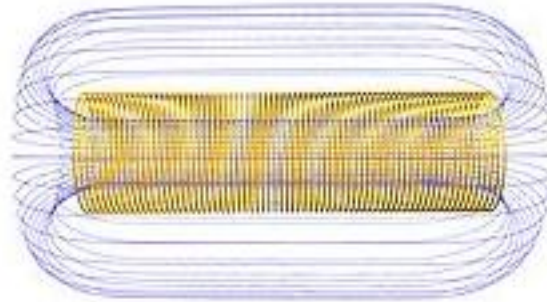


Figura 2.10 Solenoide atravesado por una corriente. *(Las líneas azules representan las líneas del campo magnético).*

Un solenoide (del griego, «solen», 'tubo', 'conducto', y «eidos», 'en forma de') es cualquier dispositivo físico capaz de crear un campo magnético sumamente uniforme e intenso en su interior, y muy débil en el exterior. Un ejemplo teórico es el de una bobina de hilo conductor aislado y enrollado helicoidalmente, de longitud indeterminada. En ese caso ideal el campo magnético sería uniforme en su interior y, como consecuencia, afuera sería nulo.

En la práctica, una aproximación real a un solenoide es un alambre aislado, de longitud finita, enrollado en forma de hélice (bobina) o un número de espirales con un paso acorde a las necesidades, por el que circula una corriente eléctrica. Cuando esto sucede, se genera un campo magnético dentro de la bobina tanto más uniforme cuanto más larga sea la bobina. La ventaja del solenoide radica en esa uniformidad que a veces se requiere en algunos experimentos de física. Pero también tiene inconvenientes: es más engorroso que las bobinas de Helmholtz y no puede producir un campo magnético elevado sin un equipo costoso y un sistema de refrigeración. André-Marie Ampère inventó en 1820 el nombre de solenoide, en un experimento en las corrientes circulares.

La bobina con un núcleo apropiado, se convierte en un electroimán. Se utiliza en gran medida para generar un campo magnético uniforme.

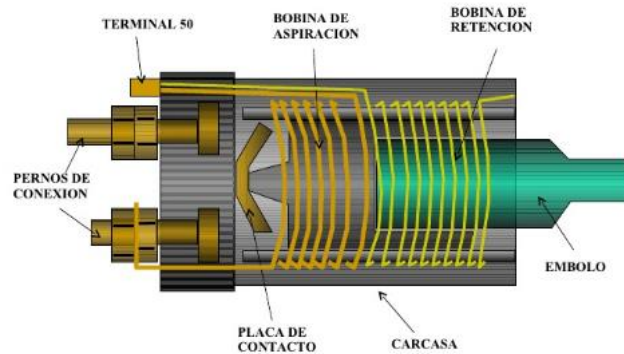


Figura 2.11 Estructura interna de un solenoide de doble bobina

2.2.9 Lector de código de barras



Figura 2.12 Lector de código de barras

Un lector de códigos de barras es un dispositivo electrónico que por medio de un láser lee el código de barras y emite el número que muestra el código de barras, no la imagen. Básicamente, consiste en el escáner propiamente dicho (que mediante un láser lee el código), un

decodificador y un cable o antena wifi que actúa como interfaz entre el decodificador y el terminal o la computadora.

La función del escáner es leer el símbolo del código de barras y proporcionar una salida eléctrica a la computadora, correspondiente a las barras y espacios del código de barras. Sin embargo, es el decodificador el que reconoce la simbología del código de barras, analiza el contenido del código de barras leído y transmite dichos datos a la computadora en un formato de datos tradicional.

Lectura en los códigos de barras

Los códigos de barras se leen pasando un pequeño punto de luz sobre el símbolo del código de barras impreso. Solo se ve una fina línea roja emitida desde el escáner láser. Pero lo que pasa es que las barras oscuras absorben la fuente de luz del escáner y la misma se refleja en los espacios luminosos. Un dispositivo del escáner toma la luz reflejada y la convierte en una señal eléctrica.

El láser del escáner (fuente de luz) comienza a leer el código de barras en un espacio blanco (la zona fija) antes de la primera barra y continúa pasando hasta la última línea, para finalizar en el espacio blanco que sigue a ésta. Debido a que el código no se puede leer si se pasa el escáner fuera de la zona del símbolo, las alturas de las barras se eligen de manera tal de permitir que la zona de lectura se mantenga dentro del área del código de barras. Mientras más larga sea la información a codificar, más largo será el código de barras necesario. A medida que la longitud se incrementa, también lo hace la altura de las barras y los espacios a leer.

Interfaces de los lectores de códigos de barras

Todas las aplicaciones pueden aceptar la salida que produce un lector de código de barras, siempre y cuando se posea el equipo necesario. Los lectores de códigos de barras se encuentran con distintas interfaces de conexión al PC.

Existen modelos de lectores que tienen solamente una interfaz integrada, pero hay algunos de ellos que aceptan varias interfaces. Basta con un simple cambio de cables y una reconfiguración para utilizar una interfaz u otra.

2.2.10 Integrado ULN2003A

El ULN2002A, ULN2003A, ULN2003AI, ULN2004A, ULQ2003A y ULQ2004A son matrices de transistores Darlington de alta tensión y alta corriente, cada uno consta de siete pares NPN Darlington que cuentan con salidas de alto voltaje con diodos de cátodo común para la conmutación de cargas inductivas. Los pares Darlington pueden conectarse en paralelo para una mayor capacidad de corriente.

El ULN2003A y ULQ2003A tienen una resistencia de base de la serie de $2,7\text{ k}\Omega$ para cada par Darlington para la operación directamente con TTL o dispositivos de CMOS.

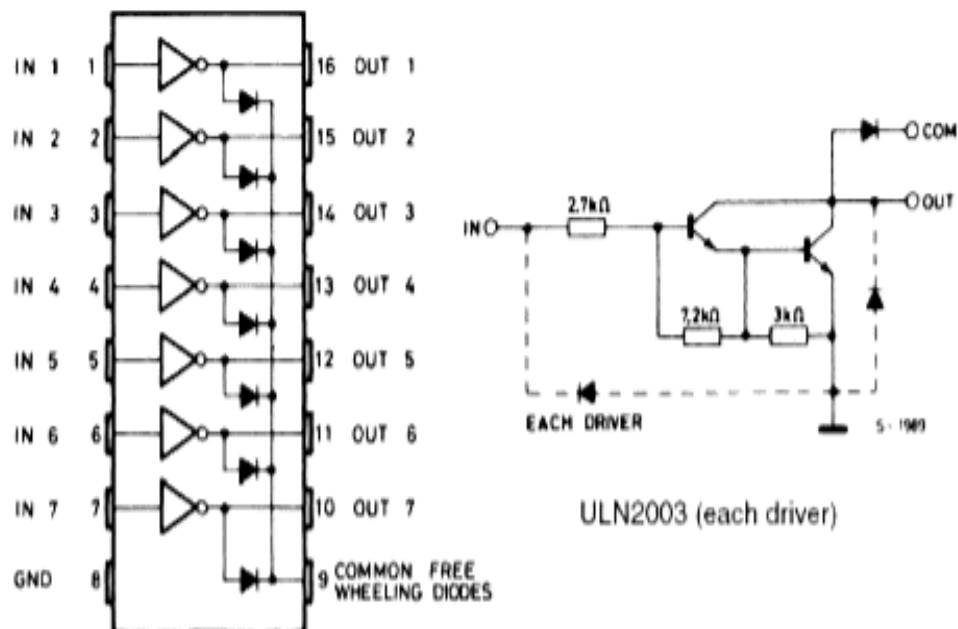


Figura 2.13 Integrado ULN2003A

Características Hardware:

- Compatible con Arduino
- Voltaje de alimentación de 5V a 12V
- Entradas compatibles con varios tipos de lógica.
- Salida protegida con diodos.
- Indicador de encendido.
- Indicadores de funcionamiento de 4 salidas.

2.2.11 NetBurner SBL2E de modulo Ethernet

NetBurner SBL2e serie de módulos Ethernet, diseñado en torno a la MCF52x Freescale MCU ColdFire, proporcionan conectividad Ethernet simple para dispositivos serie. Módulos compactos, multifuncionales SBL2e red permiten a las aplicaciones existentes, proporcionando 10/100 Ethernet, dos canales serie, cuatro analógico de 12 bits a-digitales entradas, y hasta el 10 de propósito general de E / S de señales.

La serie NetBurner SBL2e-100IR a la placa Ethernet cuenta con un conector Ethernet de a bordo. El SBL2e-200IR proporciona una cabecera conector Ethernet de 10 pines. Estos módulos NetBurner proporcionan una conexión de red independiente que le permite leer y escribir con el propósito general de las señales E / S y leer las entradas de analógico al digital. Esto es independiente de los corrientes seriales para que tenga acceso directo y rápido para controlar el dispositivo.

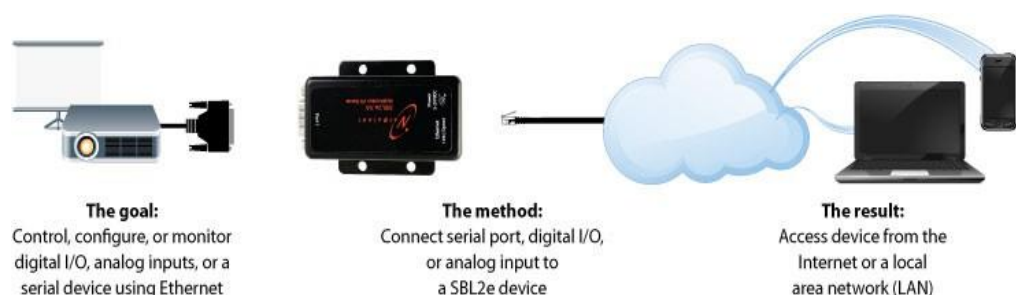


Figura 2.14 Comunicación por NetBurner SBL2e-100IR

❖ Características:

- Conectividad Ethernet simple para dispositivos serie
- Proporciona la funcionalidad de serie a Ethernet, GPIO y ADC de doble
- No se requiere programación
- RS-232, RS422 / 485 listo (requiere cambiador de nivel externo)
- 10/100 Ethernet
- Soporte de DHCP / IP estática
- Las opciones de personalización de paquetización de serie
- Control de flujo por hardware y software
- Configuración basada en web

❖ Partes que lo conforman:

- Procesador: de 32 bits de Freescale ColdFire 52236 funcionando a 50 MHz
- Cuatro analógico de 12 bits a las entradas digitales
- Hasta 10 señales de uso general de E / S
- UART Interfaces: dos interfaces serie TTL
- UART máxima Velocidad de transmisión: hasta 115.200 bps
- Protocolos de red soportados: TCP, UDP, Telnet, HTTP, DHCP

❖ Características físicas:

- Dimensiones: 1.2 "x 2" (LxA)
- Peso: 1 onza
- Los agujeros de montaje: 3 x 0.125 "de diámetro
- Alimentación: CC Voltaje de entrada: 3,3 V @ 300 mA máx
- Medio Ambiente, Temperatura de funcionamiento: -40 ° a 85 ° C



Figura 2.15 NetBurner SBL2e

2.2.12 Display LCD Serial I2C 2x16

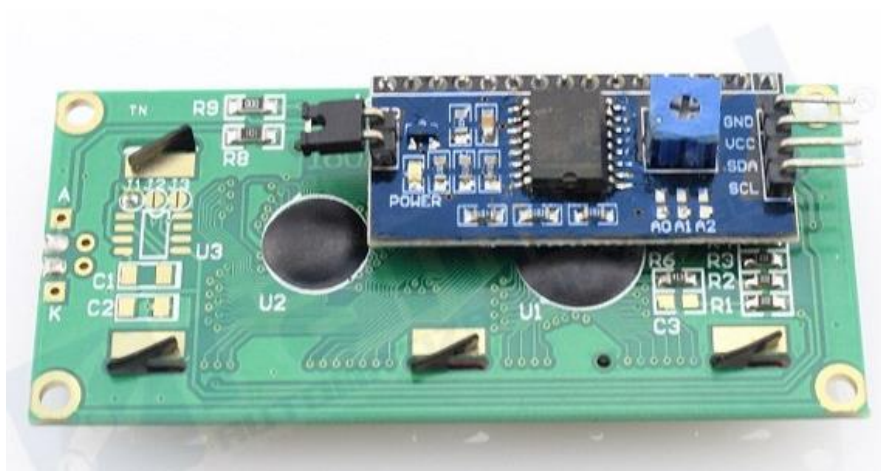


Figura 2.16 Display LCD Serial I2C 2x16

Compatible con numerosos microcontroladores y en especial con Arduino. Posee menos pines que las pantallas paralelas, sin embargo permite que los proyectos se ejecuten de manera más rápida, en comparación a muchas pantallas LCD.

La pantalla LCD es capaz de mostrar un máximo de 32 caracteres, en 16 columnas por 2 filas (16X2).

Una de las características principales es su interfaz I2C, capaz de mostrar la información a través de solo 2 líneas I2C. Si ya se cuenta con algún dispositivo I2C en el proyecto no debemos preocuparnos, esta pantalla no consume suficientes recursos en absoluto. La dirección a utilizar se puede ajustar de 0x20 a 0x27.

Junto a la ayuda del bus conversor I2C y las librerías necesarias nosotros fácilmente podemos usar este módulo con tan solo dos cables, ya que esta pantalla cuenta con 4 pines de comunicación donde 2 son de alimentación (5V y GND) y los otros dos son de comunicación I2C (SDA y SCL). Esta es una gran ventaja en comparación a una pantalla convencional, ya que estas últimas necesitan 14 pines (incluyendo alimentación y comunicación) donde incluso debemos colocar un potenciómetro de ajuste de contraste, donde la pantalla serial I2C, ya lo tiene incluido.

CARACTERÍSTICAS:

- Dirección I2C: 0x20 - 0x27 (0x20 por defecto)
- Retroiluminación: Azul con blanco carbón
- Voltaje de alimentación: 5V
- Interfaz: I2C / TWI x1
- Contraste ajustable con potenciómetro interno.
- Tamaño: 82mm x 35mm x 18mm
- Angulo de visión: 6H

2.3 Arquitectura del Software

A continuación indicamos detalles de esta arquitectura.

2.3.1 Recepción de datos

El software se comunica mediante puerto Ethernet donde recibe el dato proveniente del microcontrolador. Los datos que se reciben son los

códigos de barras transformados en tramas IP y también recibe los estados de los finales de carrera.

2.3.2 Comparación de datos

Una vez recibido el dato suministrado por el lector de código de barras, el software realiza una comparación entre dicho código y una base de datos en sietab, lugar en donde todos los datos de cada uno de los boletos emitidos por las diferentes cooperativas son almacenados de forma automática e independiente del software, estos datos se encuentran almacenados y organizados en la base de datos antes mencionada, lo que permite que el software solamente verifique si la información coincide con la suministrada.

2.3.3 Envío de datos

La controladora trabaja con un software de gestión embebido en un servidor, pero básicamente posee unos comandos para poder realizar las operaciones anteriormente descritas, para ello se debe tener en cuenta que para comunicarse con la controladora posee una IP y tres puertos para la gestión de los comandos.

- PORT0: encargado de la gestión de los códigos de barra de los tickets.
- PORT1: encargado de los mensajes que se mostraran en el LCD.
- PORT_GPIO: encargado de la operación con el torniquete para suministrar la orden de paso o cierre del torniquete y además por donde se interpretara cuando un usuario paso o no por el mismo.

Para poder transmitir un mensaje desde un servidor se debe utilizar el puerto PORT1 en donde básicamente se escribe el mensaje que se quiere mostrar anteponiendo el signo + y finalizando un enter '\n' o un fin de string '\0'

Ejemplo: < +Hola mundo\n >.

Para poder dar la orden de apertura de torniquete se usa el puerto PORT_GPIO, en el cual se debe transmitir el comando <P07=1 > y pasado un pequeño tiempo < P07=0 >, de tal forma que la tarjeta controladora responderá con < 0,ok > luego de esto un usuario pasara por el torniquete y la tarjeta electrónica responderá al software < P09=1 > en un pequeño lapso de tiempo en donde otra vez volverá a su estado original < P09=0 >.

2.4 Integración e implementación del Sistema

Indicaremos bloque por bloque el funcionamiento del sistema propuesto

2.4.1 Bloque de alimentación

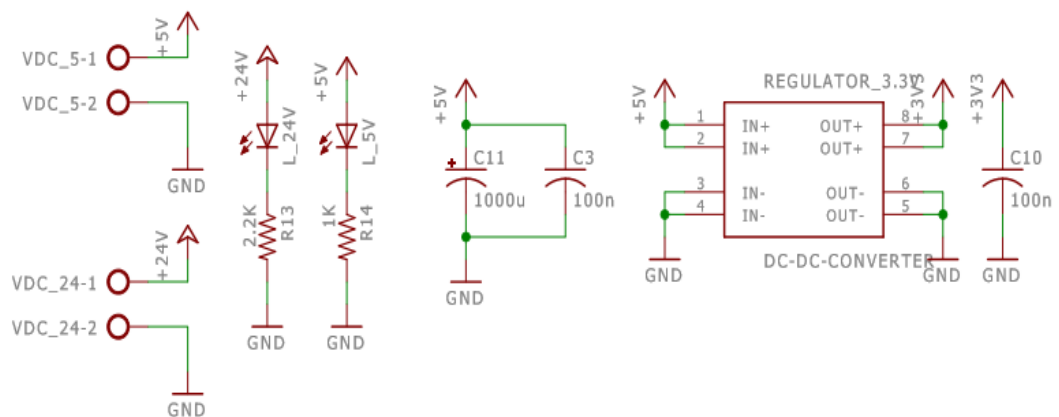


Figura 2.17 Bloque de alimentación.

Este bloque maneja tres voltajes:

- 5v que le llegan regulados a la tarjeta,
- 24v que también llegan regulados a la tarjeta y
- 3.3v que es la trasformación de los 5v regulados por medio del bloque de regulación el mismo que sirve de alimentación para el NETBURNER SBL2E

Parte de 5v.- deben entrar 5v regulados los cuales van a trabajar con el microcontrolador ATMEL ATMEGA328P-PU

Parte de 24v.- alimenta a los solenoides. En ese puerto de 24 v deben entrar 24v regulados para hacer funcionar los solenoides

En este mismo bloque existe un bloque de regulación que usa un módulo mp1584 el cual es conmutado siendo más eficiente, ya que son más pequeños y aprovechan casi toda la energía que entra disipando una pequeña cantidad en calor por lo que no necesitan disipador a diferencia de los normales. Trabajan por modulación de ancho de pulso. Con este regulador transformamos los 5v de entrada a 3.3v de salida para alimentar el NETBURNER SBL2E.

Los bloques de capacitores de 100n trabajan como filtros o estabilizadores, se los ubica para tener una señal más limpia. Si se agregan más capacitores en un circuito puede ser una desventaja debido a la tardanza en los tiempos de carga al inicio por el τ ($T=RC$) causando un retraso en los demás elementos o circuitos conectados en cascada, aunque para nuestro proyecto no influye en nada ya que para el paso de una tasa (persona por el torniquete) va a pasar cerca de 1-2 o 3 segundos lo cual es irrelevante ya que se manejan tiempos del orden de los mA y uA.

2.4.2 Bloque de comunicación serial

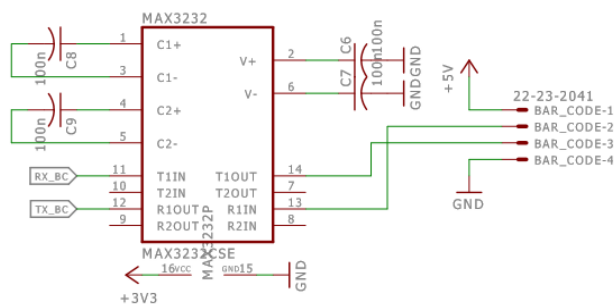


Figura 2.18 Bloque de comunicación serial

Este bloque usa un chip de comunicación serial, el MAX3232CSE, se encargar de acondicionar dos protocolos de comunicación seriales.

Este bloque se comunica con la lectoría de códigos de barras mediante comunicación serial, usando el protocolo RS 232, a través del cual se envían los niveles de voltajes de $\pm 12V$ o $\pm 24V$, siendo este protocolo mejor que el CMOS, ya que si usáramos niveles de voltajes CMOS (0 a 3.3V) por las distancias estos niveles se atenuarían y la comunicación se perdería, por otro lado los niveles de voltajes RS 232 dan más seguridad al momento de establecer comunicación a largas distancias (entiéndase por largas 3-4-5 metros).

Una vez que los datos leídos por el lector de código de barras entran al chip, este lo que hace es transformar el voltaje de $-12V$ o $12V$ (RS 232) a 0 y 3.3V (CMOS) que es el requerido por el NETBURNER SBL2E (si no se hace esto se quemaría el NETBURNER SBL2E ya que no está diseñado para aguantar niveles de voltajes superiores a los de CMOS).

El MAX3232CSE tiene una característica muy especial, está diseñado para aguantar descargas electrostáticas (ESD). Esto sirve como protección ya que si se recibe una gran descarga electrostática solo se dañaría El MAX3232CSE evitando que las demás piezas de las tarjetas sufran daño alguno.

La ruta que se sigue en este bloque es la mostrada en la Figura 2.19:

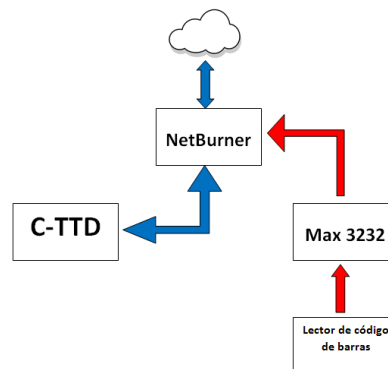


Figura 2.19 Diagrama de bloque del sistema

1. Lectura de código de barras
2. Luego la información es dirigida a la entrada del MAX3232CSE a través de comunicación serial RS 232.
3. El MAX3232CSE convierte los niveles de voltajes RS 232 a niveles de voltaje CMOS
4. Luego esta conversión CMOS entra al NETBURNER SBL2E
5. El NETBURNER SBL2E recibe la información ya convertida en niveles de voltaje CMOS y la envía vía red a un servidor o computador donde está instalado el Software que gobierna todo el comportamiento del sistema.

2.4.3 Bloque de comunicación red

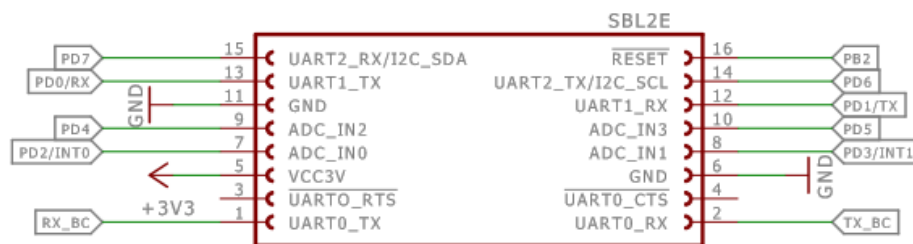


Figura 2.20 Bloque de comunicación red

El NETBURNER SBL2E se puede conectar a una red de datos que maneja comunicación IP, y estos datos son entendidos a través de una serie de parámetros de comandos que él entiende y transformados a señales electrónicas, y viceversa.

El NETBURNER tiene dos grupos de pines, un grupo que es de comunicación serial RX y TX que van dirigidos al MAX3232CSE y otro grupo de pines que solo actúan como 0 y 1 que van dirigidas al microcontrolador ATMEL ATMEGA328P-PU. A su vez tiene un microcontrolador que trabaja a 90 MHz (manejo de instrucciones súper rápido) el mismo que está diseñado con arquitectura ARM. Dentro de este micro hay dos grupos o bloques de códigos: uno que maneja el

hardware electrónico (entiéndase como tal los pines y el modulo serial) y otro que maneja la comunicación WEB o IP que entra o sale de él y es lo que va a pasar por el cable de red (conector RJ45).

El NETBURNER tiene embebido dentro una página web, es decir, que dentro hay datos y estos abren una página web, haciendo factible al conectarlo a una computadora a través de cable de red configurarlo por medio de un programa y así setear IP, mascara de subred, GUETWEY, DNS, dándole una dirección física y ahí se abre la página web donde lo configuramos según nuestra necesidad. La configuración hay que hacerla según como indica el manual.

2.4.4 Bloque del microcontrolador Atmel atmega328p-pu

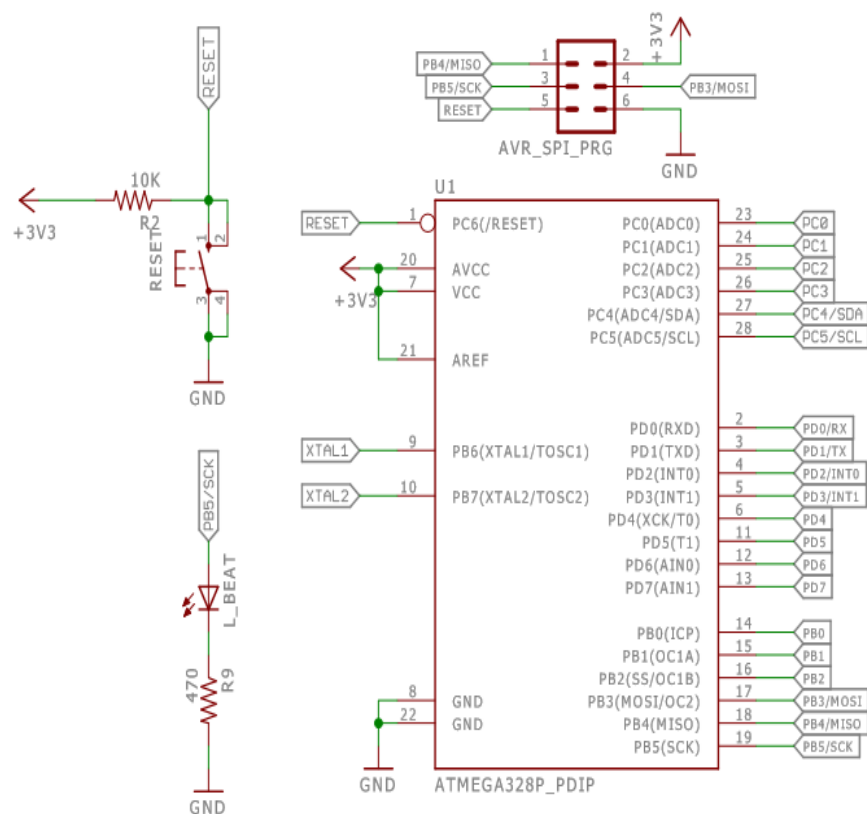


Figura 2.21 Bloque del microcontrolador

Algo importante en este micro es que trabaja con instrucciones en PROGRAMACIÓN C permitiendo optimizar recursos y tiempo. Tiene un grupo de patas llamado AVR_SPI Programming que fue usado para programar el chip, también sirve para depuración.

En el proyecto solo hemos usado 8 Kb para la codificación, comunicación I2C, comunicación serial e interrupciones. Este microcontrolador se encarga de gobernar, controlar y manejar los siguientes dispositivos:

1. Display LCD
2. Lector de sensores del torniquete
3. Los solenoides (los cuales permiten la apertura y cierre del torniquete)

Se usan 7 puertos de salida, 3 para controlar los relés que controlan los solenoides y 4 que controlan los Leds que indican que pasa en la tarjeta.

El microcontrolador se comunica con el display LCD a través de protocolo I2C por medio de los puertos de comunicación del micro.

Se usaron 5 Puertos adicionales que provienen del NETBURNER, dos de entrada, dos de salida y uno para reseteo del NETBURNER (este se ejecuta desde el micro desde la programación). Los de entrada PD3IN1 y PD0IN0 son dos puertos en interrupciones que provienen desde el NETBURNER y llega el ATMEGA, es decir, por ejemplo: si por el NETBURNER envía la orden de apertura del torniquete en el sentido derecho sucede que por un puerto el NETBURNER lleva la comunicación al ATMEGA el cual está usando puerto de interrupción externa, el mismo que recibe en un bloque de interrupción externa en el cual se encuentra lo necesario para poder interactuar con los solenoides y hacer ejecutar la orden sobre el torniquete.

2.4.5 Bloque de comunicación I2C

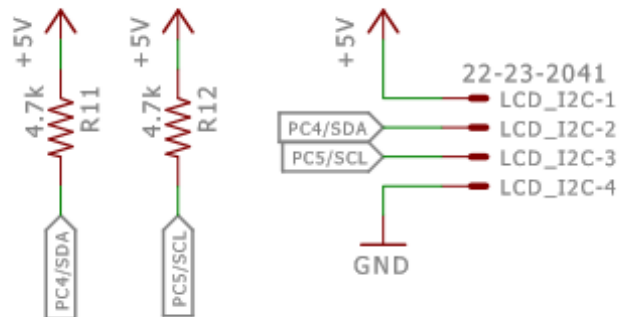


Figura 2.22 Bloque de comunicación I2C

Consiste de una resistencia para hacer PUSH-POOL porque así lo dictamina el protocolo I2C, y luego las salidas van a llegar al LCD, dos para alimentación y dos para datos.

2.4.6 Bloque de sensores

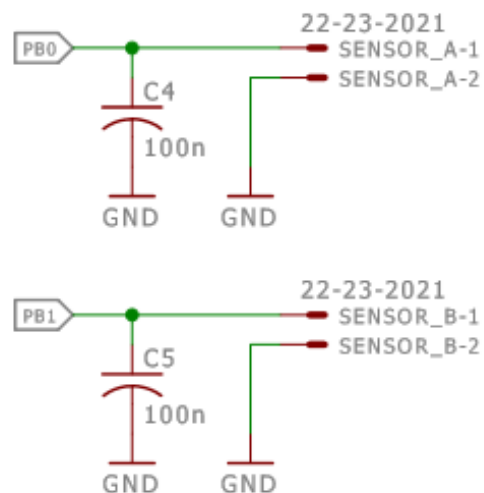


Figura 2.23 Bloque de sensores

Para la parte del sensado se utilizaron finales de carrera que nos indican si el tambor del torniquete ha girado.

El microcontrolador tiene activado el PUSH-POOL interno por esa razón no se utilizan resistencias externas, el nivel lógico del pin es alto y al cerrarse el final de carrera el circuito se cierra aterrizándose lo que cambia el nivel lógico pasando de alto a bajo, permitiendo de esta manera sensor el giro del torniquete.

Aquí el capacitor es importante para filtrar los ruidos debido a que los finales de carrera son instrumentos mecánicos.

2.4.7 Bloque controlador de salidas

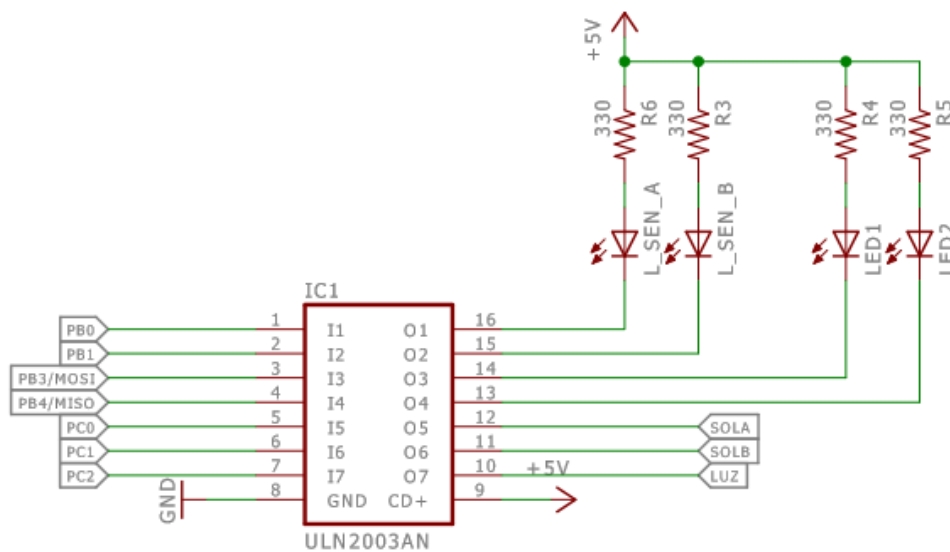


Figura 2.24 Bloque del controlador de salida

Todo lo que uso el microcontrolador como salidas, entiéndase como tal los solenoides, los leds y los relés de control de luz, fueron enviados a la tarjeta o chip ULN2003AN la cual tiene internamente transistores

Darlington para que responda a las señales TTL o niveles lógicos CMOS y suministre energía a los periféricos que requieran más corriente hasta 1A.

Los leds presentes son manejados también desde este chip aprovechando los pines y el uso del mismo Darlington, por protección. Así se evita estresar o dar más trabajo al ATMEGA.

Hay un pin que controla el relé para luz piloto como se observa en la gráfica. Aquí también tenemos dos pines que son para controlar los relés de solenoides y son SOLA y SOLB. Estos trabajan con lógica negada.

2.4.8 Bloque de actuadores o relay

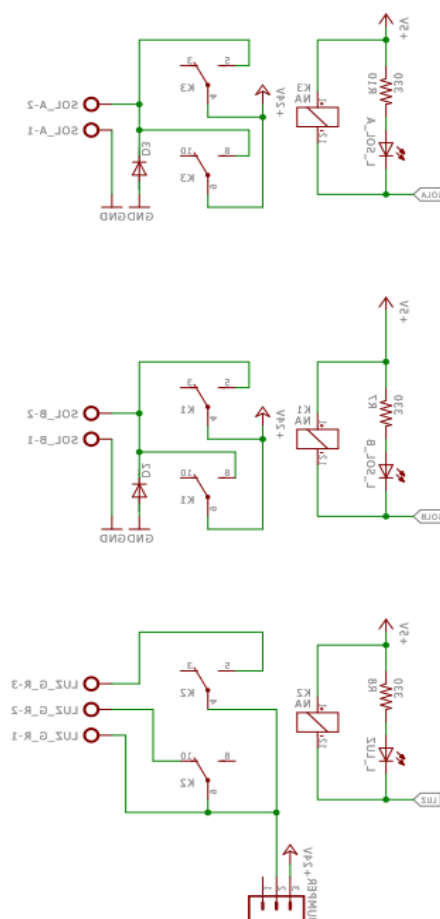


Figura 2.25 Bloque de actuadores o relay

Aquí se tiene dos relés con resistencias en paralelo, cada uno respectivamente, enviando la orden en lógica negativa, es decir a 0V se activa y a 5V se desactiva. En este circuito tenemos solenoides los cuales por inducción producen corrientes en sentido contrario debido a su naturaleza propia por lo cual la cortocircuitamos a través de un diodo de protección en sentido contrario para así evitar que se quemara la fuente. Solo se coloca este diodo de un lado, ya que el chip ULN2003AN ya posee diodos de protección interna que se los hace interactuar con los relés.

El bloque de control de luz trabaja en dos sentidos, es decir que cuando está apagado (un alto, 5V) por default deja una línea de luz y cuando está encendido (un bajo, 0V) otra línea de luz. Aquí se hace uso de un jumper el mismo que fue creado para poder alimentar las luces pilotos con 110V.

CAPÍTULO 3

3. PRUEBAS Y VALIDACIÓN DEL SISTEMA

El siguiente capítulo tiene como objetivo brindar información de la operación de la tarjeta Controlador TTD además de explicar su funcionamiento en conjunto con el sistema autónomo que operan los torniquetes.

3.1 Pruebas de laboratorio

Las pruebas que se presentaran a continuación, son pruebas que se realizaron en nuestros domicilios.

Estas pruebas tienen como fin verificar el correcto funcionamiento de las E/S que controlan los diferentes elementos periféricos utilizados en el proyecto, así como también probar el correcto funcionamiento de la comunicación por medio del Netburner.

3.1.1 Pruebas de indicadores de finales de carrera

En la tarjeta se implementaron leds indicadores para simular el estado de los finales de carrera. Solo probaremos si al cerrarse el circuito el led enciende, y estas pruebas son visibles en las Figuras 3.1 y 3.2



Figura 3.1 Led final de carrera 1



Figura 3.2 Led final de carrera 2

3.1.2 Pruebas de indicadores de accionamiento de solenoides

En la tarjeta se implementaron leds indicadores para el estado de la salida a los relés que habilitan el voltaje necesario para la activación de los solenoides, los cuales activan un sistema mecánico el cual se mostrara en las pruebas realizadas en el terminal terrestre de duran.

Adicional a los indicadores leds se realizó una prueba de niveles de voltajes a la salida que alimenta a los solenoides como se muestra en la Figura 3.3.

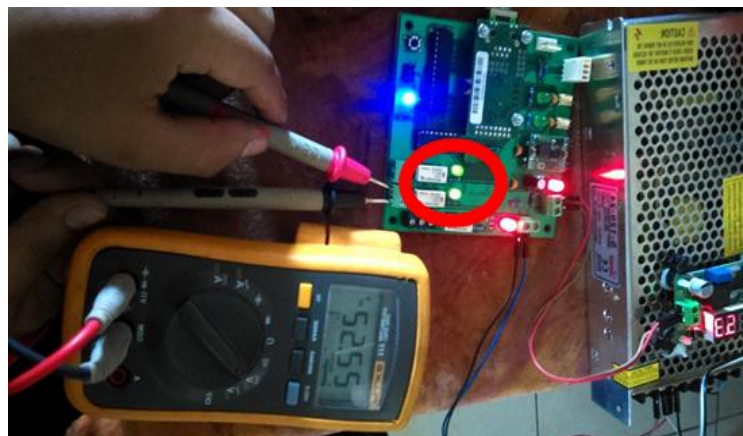


Figura 3.3 Voltaje de salida a los solenoides

3.1.3 Pruebas de comunicación computador- Controladora TTD

Esta prueba se la realizara por medio de un programa que se llama IPSetup el cual verifica la existencia de IP estáticas.

Al darle doble clic se abre la ventana mostrada en la Figura 3.4:

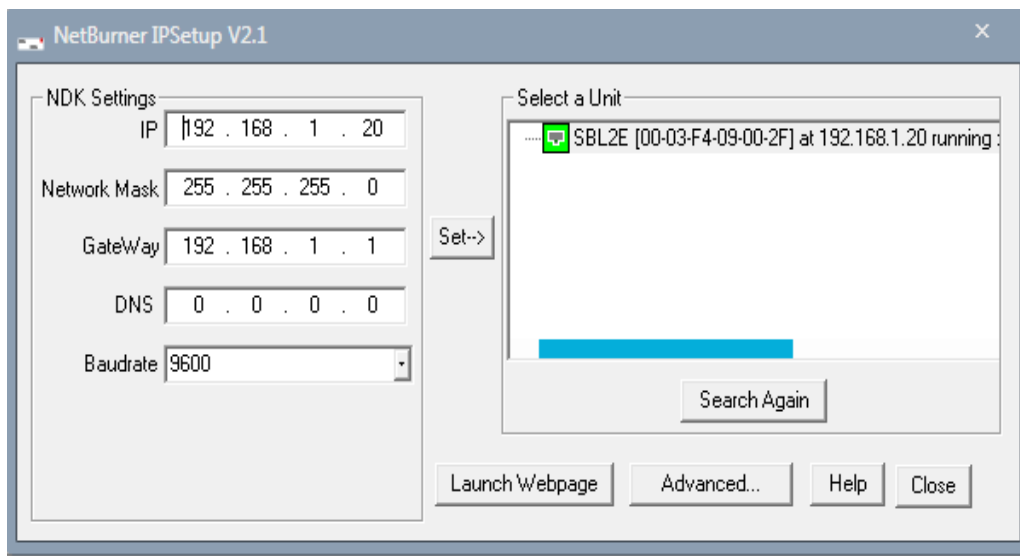


Figura 3.4 Ventana de programa identificador de IP Netburner IPSetup V2.1

Esta ventana busca los IP estáticos asociados a la red a la cual esté conectado de forma alámbrica o inalámbrica, en nuestro caso inalámbrica.

Como notamos la IP asociada a nuestra controladora es la 192.168.1.20.

3.1.4 Pruebas de mensajes en el display

Para probar esta etapa de la tarjeta enviaremos un mensaje de la computadora a la controladora por medio del router al igual que lo hicimos anteriormente.

Las pruebas del display se las realizó verificando dos situaciones:

1. Al encenderse: La controladora por default muestra un mensaje de la compañía que nos alberga “Código & Silicio” como se observa en la Figura 3.5.



Figura 3.5 Mensaje en display 1

2. Enviando un mensaje del computador hacia la controladora: Para poder conectarnos con la controladora es necesario la dirección, en nuestro caso 192.168.1.20 y saber el puerto por donde se enviara el mensaje, siendo este el 25 con tipo de conexión Telnet. Esta acción se realiza por medio de un programa llamado PuTTY el cual se encarga ser la interfaz entre el software y el controlador.

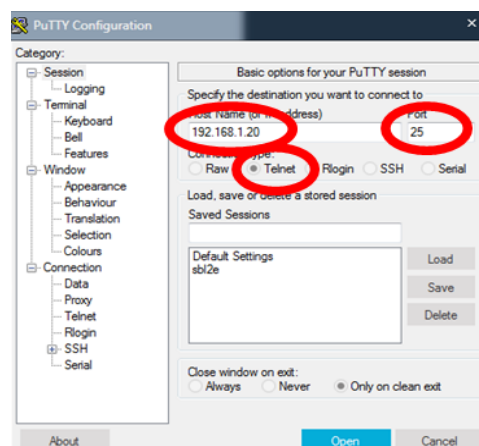


Figura 3.6 Ventana Programa PuTTY a

Al hacer open en esta ventana nos aparece la ventana de la figura 3.7 que permite comunicarnos directamente con la controladora.

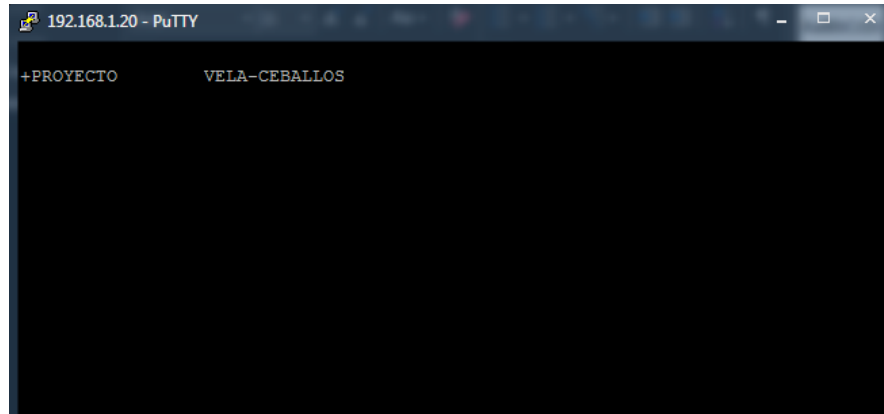


Figura 3.7 Ventana de comunicación puerto 25

En esta ventana escribimos el mensaje que queremos enviar a la computadora anteponiendo el signo más y finalizando con un Enter lo cual da como resultado el mensaje mostrado en la Figura 3.8



Figura 3.8 Mensaje den display 2

3.1.5 Pruebas de funcionamiento del sistema

Estas pruebas tienen como finalidad verificar la orden de apertura enviada del computador a la controladora por medio de un router.

Para esto nuevamente utilizaremos el programa Putty de igual forma que lo hicimos la vez anterior con la diferencia de que en esta ocasión haremos uso del puerto 1001 o puerto PORT_GPIO, puerto por el cual se envía el dato que permite la habilitación del giro del molinete (Figura 3.9).

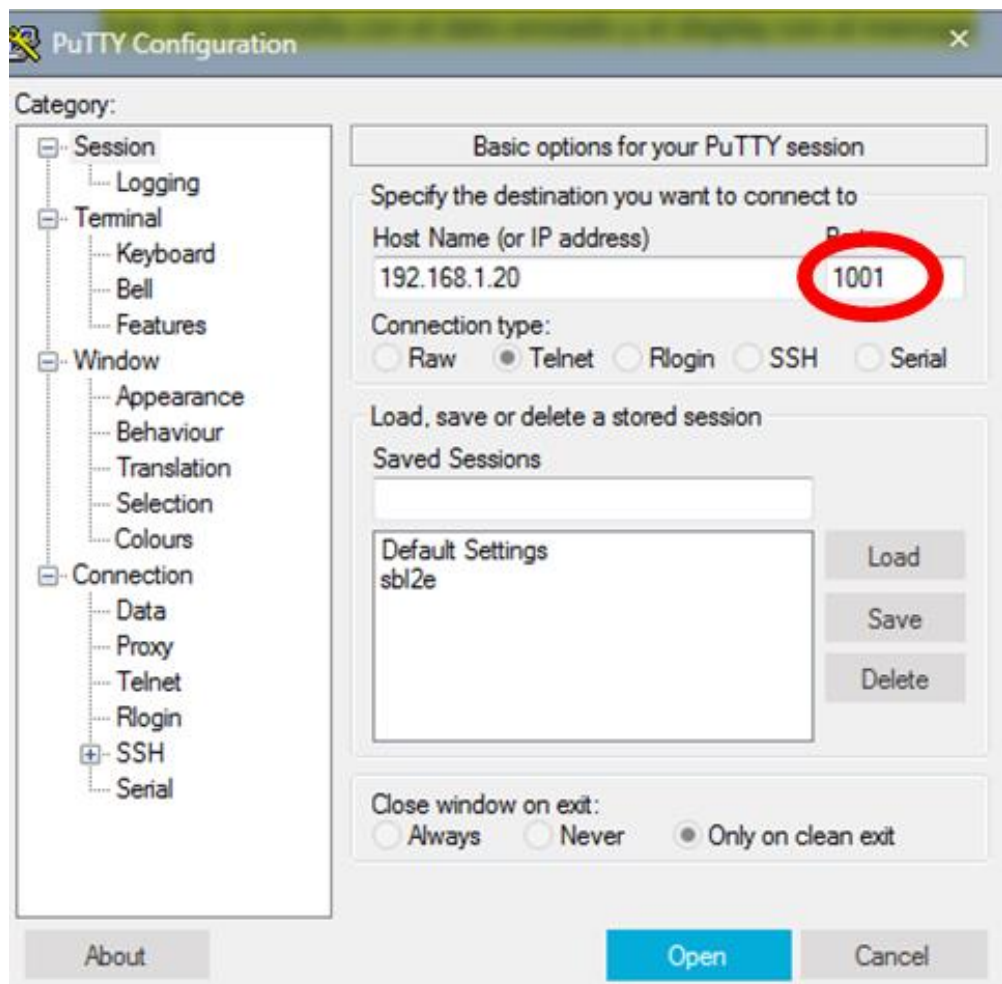
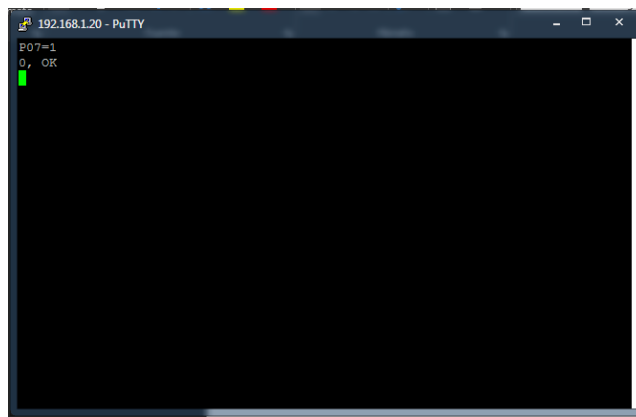


Figura 3.9 Configuración de ventana PuTTY

Una vez dado clic en open se abrirá la ventana en donde realizaremos la siguiente acción.

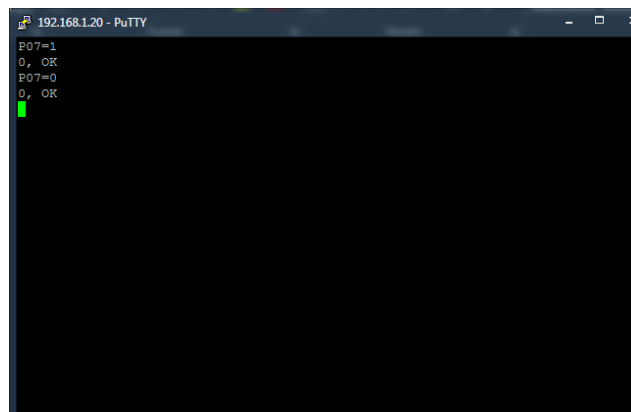
Enviamos los comandos <P07=1> a través del puerto GPIO, a lo cual la tarjeta controladora deberá responder con <0, ok>. Esta orden <P07=1> es la orden que el software envía a la controladora para habilitar el giro en un sentido en el torniquete, y la respuesta <0, ok> es solo la respuesta que realiza la controladora para indicar que recibió el dato tal y cual se observa en la Figura 3.10.



```
192.168.1.20 - PuTTY
P07=1
0, OK
```

Figura 3.10 Ventana de comunicación puerto 1001 a

Luego de esto es necesario enviar los comandos <P07=0>, para que la controladora cuente un franco de reloj para que quede en espera de otro nuevo aviso de apertura, nuevamente la controladora contestara con <0, ok>, tal y cual se observa en la Figura 3.11.



```
192.168.1.20 - PuTTY
P07=1
0, OK
P07=0
0, OK
```

Figura 3.11 Ventana de comunicación puerto 1001 b

3.2 Pruebas y validación del sistema

Esta parte del proyecto tiene como fin la demostración de la implementación de las controladoras TTD en el terminal terrestre de duran.

3.2.1 Pruebas de finales de carrera

Como notamos en las figuras 3.12 y 3.13, al activarse el final de carrera se activa el indicador de la tarjeta lo que muestra un funcionamiento correcto de los mismos.

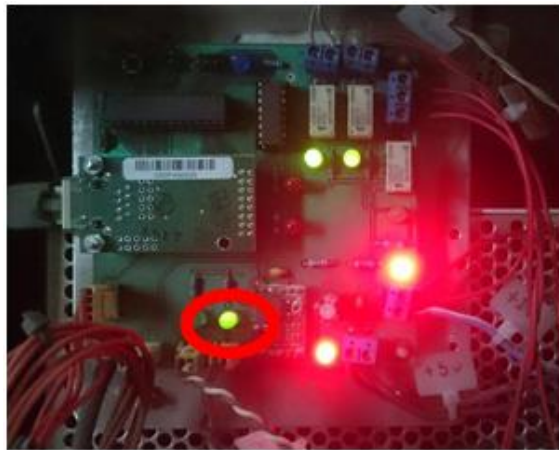


Figura 3.12 Activación de luz piloto según el giro 1

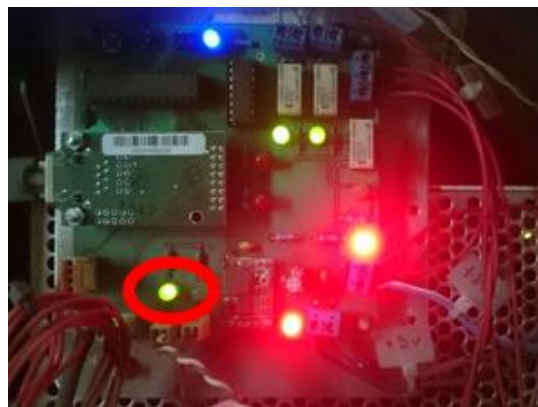


Figura 3.13 Activación de luz piloto según el giro 2

El giro del tambor permite el cierre del circuito por medio de las cuñas de la pieza blanca y el muelle propio del final de carrera como se visualiza en la Figura 3.14.

Es válido destacar que se utilizaron dos finales de carrera para permitir el conocimiento del sentido del giro, para de esta forma identificar si el usuario sale o ingresa.

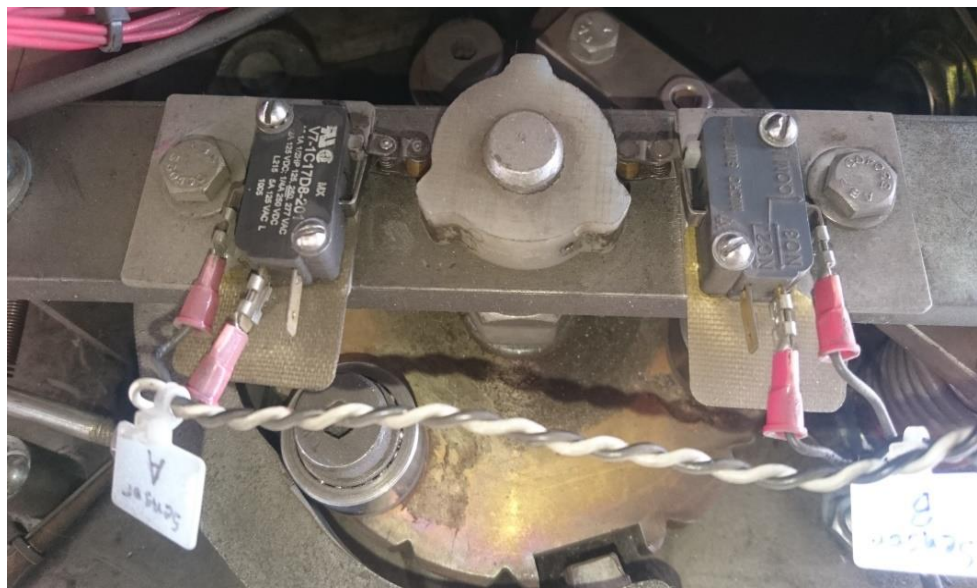


Figura 3.14 Sistema de indicativo del giro del molinillo.

3.2.2 Pruebas de accionamiento de solenoides

El que los indicadores de accionamiento en la tarjeta funcionen apropiadamente denota que el sistema funciona de manera apropiada, así que esta parte se centrara en el funcionamiento mecánico de los solenoides.(Figuras 3.15 y 3.16)



Figura 3.15 Desactivación del solenoide



Figura 3.16 Activación del solenoide

3.2.3 Pruebas de comunicación área de sistemas - Controladora TTD

Para verificar que los torniquetes estén funcionando de forma lógica se monitorea mediante la siguiente aplicación llamada NetBurner IPSetup, se observa en la Figura 3.17 que todos los torniquetes están activos.

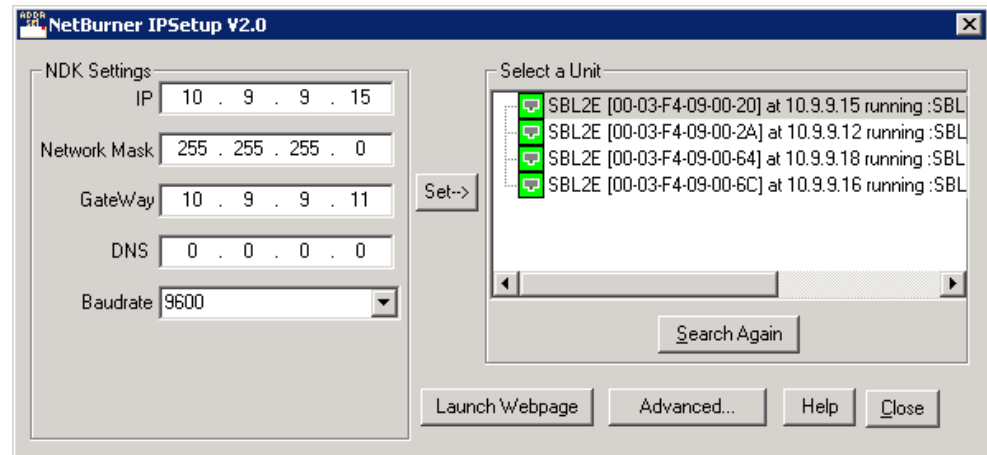


Figura 3.17 Ventana Programa PuTTY b

3.2.4 Pruebas de luces piloto

Por default la luz piloto encendida es la roja como indicativo de acceso no permitido, cuando se habilita el paso en caso de ser correcto el boleto se habilita la luz verde comúnmente utilizada como acceso permitido. (Figuras 3.18 y 3.19)

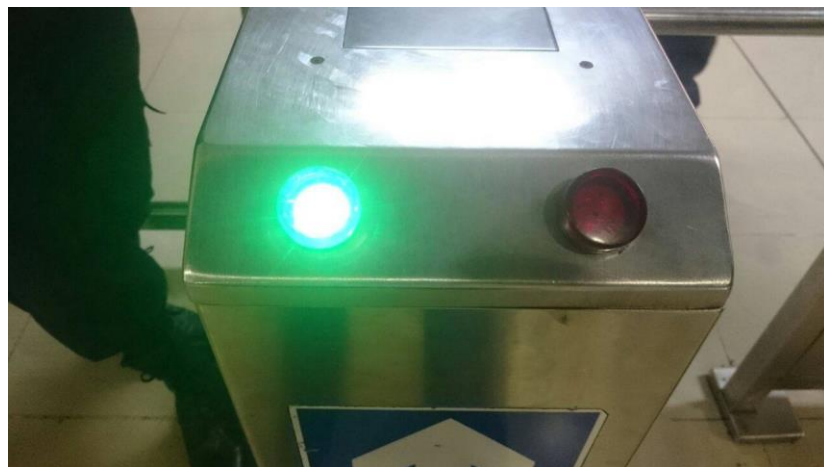


Figura 3.18 Luz piloto verde



Figura 3.19 Luz piloto roja

3.2.5 Pruebas de funcionamiento del sistema implementado

Esta parte tiene como fin la muestra final de la implementación del sistema en el terminal terrestre de duran.

En el título de la ventana se observa una nomenclatura P1T1, la cual significa Puerta 1 Torniquete 1 y se muestra su respectiva dirección IP.

En este caso sería 10.9.9.16. (Figura 3.20)

```

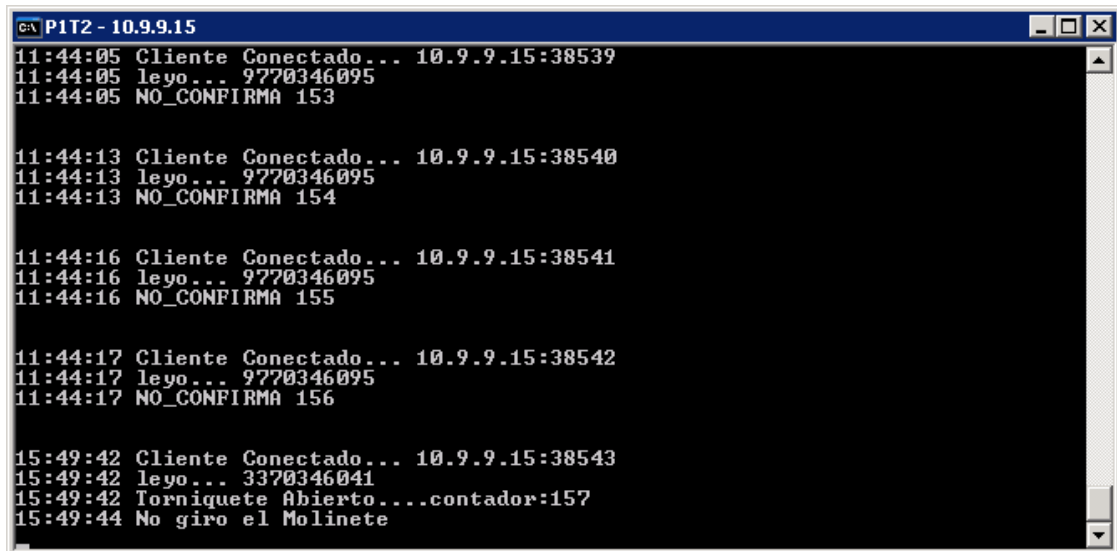
c:\ P1T1 - 10.9.9.16
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at torniquete.Torniquete.display(Torniquete.java:310)
at torniquete.Hilo.run(Hilo.java:51)

jul 19, 2016 5:02:07 PM torniquete.Torniquete display
SEVERE: null
java.net.ConnectException: Connection timed out: connect
at java.net.IwoStacksPlainSocketImpl.socketConnect(Native Method)
at java.net.AbstractPlainSocketImpl.doConnect(Unknown Source)
at java.net.AbstractPlainSocketImpl.connectToAddress(Unknown Source)
at java.net.AbstractPlainSocketImpl.connect(Unknown Source)
at java.net.PlainSocketImpl.connect(Unknown Source)
at java.net.SocksSocketImpl.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.connect(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at java.net.Socket.<init>(Unknown Source)
at torniquete.Torniquete.display(Torniquete.java:310)
at torniquete.Hilo.run(Hilo.java:51)

```

Figura 3.20 Ventana P1T1

P1T2, la cual significa Puerta 1 Torniquete 2 cuya IP 10.9.9.15 (Figura 3.21)



```

C:\> P1T2 - 10.9.9.15
11:44:05 Cliente Conectado... 10.9.9.15:38539
11:44:05 leyo... 9770346095
11:44:05 NO_CONFIRMA 153

11:44:13 Cliente Conectado... 10.9.9.15:38540
11:44:13 leyo... 9770346095
11:44:13 NO_CONFIRMA 154

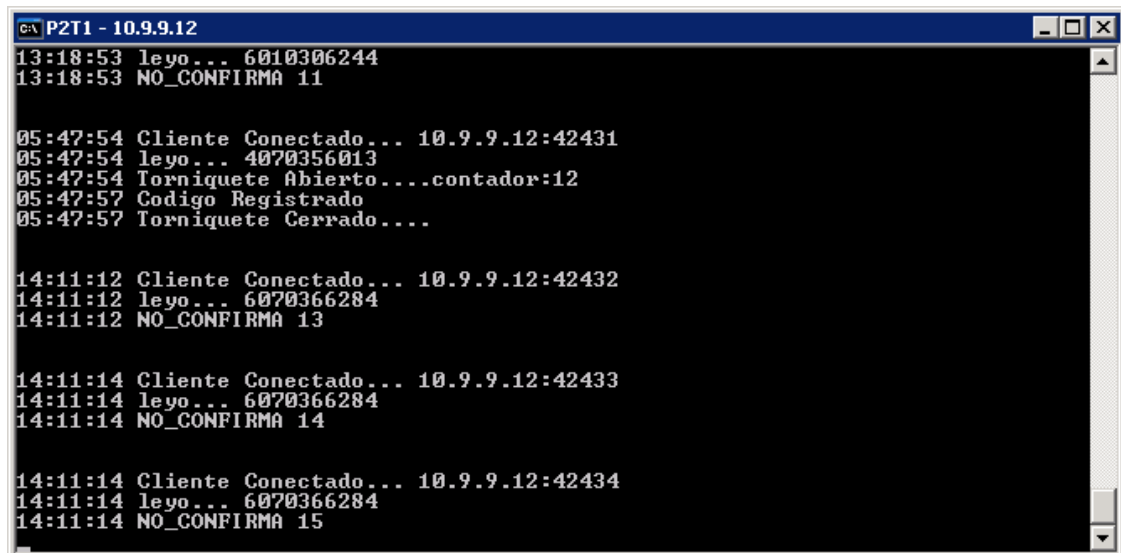
11:44:16 Cliente Conectado... 10.9.9.15:38541
11:44:16 leyo... 9770346095
11:44:16 NO_CONFIRMA 155

11:44:17 Cliente Conectado... 10.9.9.15:38542
11:44:17 leyo... 9770346095
11:44:17 NO_CONFIRMA 156

15:49:42 Cliente Conectado... 10.9.9.15:38543
15:49:42 leyo... 3370346041
15:49:42 Torniquete Abierto...contador:157
15:49:44 No giro el Molinete
  
```

Figura 3.21 Ventana P1T2

P2T1, la cual significa Puerta 2 Torniquete 1 cuya IP 10.9.9.12 (Figura 3.22)



```

C:\> P2T1 - 10.9.9.12
13:18:53 leyo... 6010306244
13:18:53 NO_CONFIRMA 11

05:47:54 Cliente Conectado... 10.9.9.12:42431
05:47:54 leyo... 4070356013
05:47:54 Torniquete Abierto...contador:12
05:47:57Codigo Registrado
05:47:57 Torniquete Cerrado....

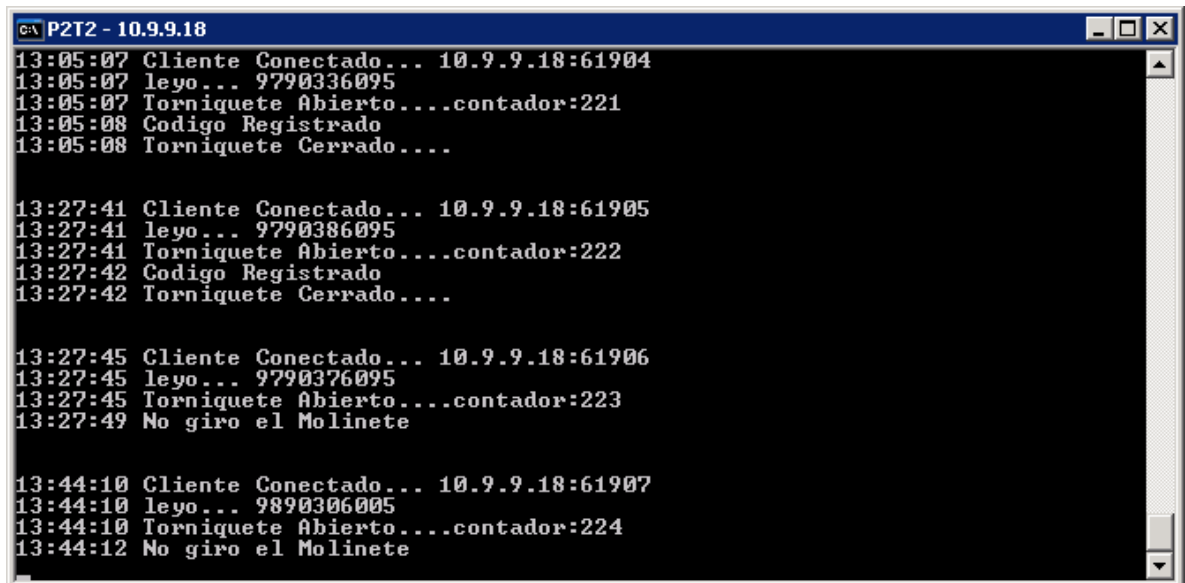
14:11:12 Cliente Conectado... 10.9.9.12:42432
14:11:12 leyo... 6070366284
14:11:12 NO_CONFIRMA 13

14:11:14 Cliente Conectado... 10.9.9.12:42433
14:11:14 leyo... 6070366284
14:11:14 NO_CONFIRMA 14

14:11:14 Cliente Conectado... 10.9.9.12:42434
14:11:14 leyo... 6070366284
14:11:14 NO_CONFIRMA 15
  
```

Figura 3.22 Ventana P2T1

P2T2, la cual significa Puerta 2 Torniquete 2 cuya IP 10.9.9.18 (Figura 3.23)



```
P2T2 - 10.9.9.18
13:05:07 Cliente Conectado... 10.9.9.18:61904
13:05:07 leyo... 9790336095
13:05:07 Torniquete Abierto...contador:221
13:05:08Codigo Registrado
13:05:08 Torniquete Cerrado....

13:27:41 Cliente Conectado... 10.9.9.18:61905
13:27:41 leyo... 9790386095
13:27:41 Torniquete Abierto...contador:222
13:27:42Codigo Registrado
13:27:42 Torniquete Cerrado....

13:27:45 Cliente Conectado... 10.9.9.18:61906
13:27:45 leyo... 9790376095
13:27:45 Torniquete Abierto...contador:223
13:27:49 No giro el Molinete

13:44:10 Cliente Conectado... 10.9.9.18:61907
13:44:10 leyo... 9890306005
13:44:10 Torniquete Abierto...contador:224
13:44:12 No giro el Molinete
```

Figura 3.23 Ventana P2T2

En la sección 3.1.5 Pruebas de funcionamiento del sistema, se detalló los comandos enviados por el programa PuTTY, todo lo antes mencionado allí dará como resultado que en la controladora al identificar el comando enviado por el puerto GPIO cambie el estado de los solenoides y active el led indicativo de apertura del torniquete, de igual forma activara el cambio de luz piloto roja por default a verde.

Al mismo momento en que se envía el comando de apertura se envía el dato por el puerto 25 con el mensaje de “PASE POR FAVOR” como se ve en la Figura 3.24.



Figura 3.24 Mensaje LCD 1

Los otros posibles escenarios presentados serían los observados en las Figuras 3.25 y 3.26



Figura 3.25 Mensaje LCD 2



Figura 3.26 Mensaje LCD 3

CONCLUSIONES Y RECOMENDACIONES

La elaboración de este proyecto permitió la creación de un sistema de control, con capacidad de garantizar el acceso de personas de forma automática y/o redirigirla al lugar adecuado.

El sistema implementado permite acceder de manera remota al servidor que contiene todos los tickets adquiridos por los usuarios asegurando la validez de los mismos.

El sistema facilita las tareas de revisión de equipos de una manera rápida ya que consta de indicadores anti-fallas.

Se recomienda mantener las lectoras limpias todo el tiempo puesto que la interrupción del haz de luz por intervención de partículas de polvo puede generar la adquisición de señales erróneas, o en el peor de los casos la no adquisición de las mismas.

Otro de los fallos que podría presentar el sistema tiene que ver con la calidad de las impresiones de los tiquetes, los cuales sino presentan una impresión clara en su código de barras, generarían la adquisición de señales erróneas o en el peor de los casos la no adquisición de las señales de lectura esperadas.

BIBLIOGRAFÍA

- [1] MANUAL CONTROLADOR MURPHY TTD
<http://www.fae.co.za/fwmurphy/pdf/00-02-0329.pdf>
- [2] Controlador ATMEGA328P
<http://www.atmel.com/devices/atmega328p.aspx>
- [3] Las placas MAX3232CSE
<https://datasheets.maximintegrated.com/en/ds/MAX3222-MAX3241.pdf>
- [4] Hoja de datos del ATMEL ATMEGA328P-PU
<http://www.mouser.com/ds/2/36/atmel-8271-8-bit-avr-microcontroller-atmega48a-48p-365589.pdf>
- [5] Convertidor de voltaje DC-DC módulo mp1584
https://www.monolithicpower.com/Portals/0/Documents/Products/Documents/datasheet/MP1584_r1.0.pdf
- [6] Integrado ULN2003A
<http://www.ti.com/lit/ds/symlink/uln2003a.pdf>
- [7] NetBurner SBL2E de modulo Ethernet
<http://www.netburner.com/index.php/support/documents/sbl2e/340-1-9/file>
- [8] RS232 Y MAX2323
<https://www.sparkfun.com/tutorials/215>

ANEXOS

ABREVIATURAS

TTD	Terminal Terrestre de Durán
RFID	Identificación por Radio Frecuencia
C-TTD	Controladora Terminal Terrestre de Duran


```

volatile unsigned char time_out;

volatile unsigned char inBuffer[34]= "Codigo & Silicio FW_2.00";

ISR (INT0_vect){ // habilitar paso en sentido a
    if( !B_NEWPASS ){
        LUZ_hi;
        B_NEWPASS = TRUE;
        B_OPEN_A = TRUE;
    }
}

ISR (INT1_vect){ // habilitar paso en sentido b
    if( !B_NEWPASS ){
        LUZ_hi;
        B_NEWPASS = TRUE;
        B_OPEN_B = TRUE;
    }
}

ISR (PCINT0_vect){ // sensores switch de torniquete
    sensorA_now = SENSORA_status;
    sensorB_now = SENSORB_status;

    if( sensorA_old == LOW && sensorA_now == HIGH ){
        B_PASS_COMPLETED_A = TRUE;
    }else if( sensorB_old == LOW && sensorB_now == HIGH ){
        B_PASS_COMPLETED_B = TRUE;
    }

    sensorA_old = sensorA_now;
    sensorB_old = sensorB_now;
}

ISR (TIMER1_COMPA_vect){ // alive de tarjeta y timeout de paso
    LEDbeat_to;

    if( B_NEWPASS ){
        if( time_out++ == 160){ // 20 segundos
            time_out = 0;
            B_OPEN_A = FALSE;
            B_OPEN_B = FALSE;
            B_NEWPASS = FALSE;
            LUZ_lo;
        }
    }else{
        time_out = 0;
    }
}

ISR (USART_RX_vect){ // comunicacion sistemas tarjeta
    static bool flg_msg, flg_cmd;

```

```

static char inChar;
static int index;

inChar = UDR0;      // Read data from the RX buffer

if( B_NEW_MSG )
    return;

if( index >= 33 ){
    index = 33;
    inChar = '\r';
}

switch ( inChar ){
case '$':
    index = 0;
    flg_cmd = TRUE;
    flg_msg = FALSE;
    break;
case '+':
    index = 0;
    flg_msg = TRUE;
    flg_cmd = FALSE;
    break;
case '\n':
case '\r':
    if( flg_msg ){
        inBuffer[index] = '\0';
        index = 0;
        flg_msg = FALSE;

        B_NEW_MSG = TRUE;
    }else if( flg_cmd ){
        inBuffer[ index ] = '\0';
        index = 0;
        flg_cmd = FALSE;
        //if(inBuffer[0]=='r')
        //reset_cpu();
    }
    break;
default:
    if( flg_msg || flg_cmd )
        inBuffer[index++] = inChar;
    break;
}
}

void setup();
void setupTimer();
void setupUsart();

void loop();

void HANDLE_LCD();

```

```

void HANDLE_TORNIQUETE();
void HANDLE_NETBURNER();

int main(void)
{
    setup();

    while (1)
    {
        loop();
    }
}

void loop(){
    HANDLE_LCD();
    HANDLE_TORNIQUETE();
    HANDLE_NETBURNER();
}

void setup(){
    //SALIDAS-----
    // cpu
    set_bit( DDRB, LED1);
    set_bit( DDRB, LED2);

    LED1_lo;
    LED2_lo;

    set_bit( DDRB, LEDbeat);

    LEDbeat_lo;

    // torniquete
    set_bit( DDRC, SOLENA);
    set_bit( DDRC, SOLENB);

    SOLENA_open;
    SOLENB_close;

    set_bit( DDRC, LUZ);

    LUZ_lo;

    // netburner
    set_bit( DDRB, RESETnetburner);
    set_bit( DDRD, PASScompla);
    set_bit( DDRD, PASScomplB);

    RESETnetburner_hi;
    PASScompla_lo;
    PASScomplB_lo;

    //ENTRADAS-----
    // jumper

```

```

clear_bit( DDRC, JUMPER);
set_bit( PORTC, JUMPER); //pullup para jumper

// sensores con interrupcion pcint
clear_bit( DDRB, SENSORA);
clear_bit( DDRB, SENSORB);

set_bit( PORTB, SENSORA); //pullup para sensor a
set_bit( PORTB, SENSORB); //pullup para sensor b

set_bit( PCICR, PCIE0); // enable pc0 vector
set_bit( PCMSK0, PCINT0); // pcint0 = sensor a
set_bit( PCMSK0, PCINT1); // pcint1 = sensor b

//desbloquear torniquete netburner
clear_bit( DDRD, PASS_DirA);
clear_bit( DDRD, PASS_DirB);

set_bit( EICRA, ISC00);
set_bit( EICRA, ISC01); // 00 y 01 para rising (int0)
set_bit( EIMSK, INT0); // enable interrupcion externa int0

set_bit( EICRA, ISC10);
set_bit( EICRA, ISC11); // 00 y 01 para rising (int1)
set_bit( EIMSK, INT1); // enable interrupcion externa int1

// configurar timer-----
setupTimer();

// configurar usart-----
setupUsart();

// configurar i2c init-----
i2c_init();

// lcd init-----
LCDInit();
B_NEW_MSG = TRUE;
HANDLE_LCD();

// init status de sensores-----
sensorA_old = SENSORA_status;
sensorB_old = SENSORB_status;

// enable interrupts-----
sei();
}

void setupTimer(){

```

```

        // OCRn = [ (clock_speed 8mhz / Prescaler_value 1024 ) *
Desired_time_in_Seconds ] - 1
        OCR1A = 0x03D0; // cada 0.125 seg
        TCCR1B |= (1 << WGM12); // Mode 4, CTC on OCR1A
        TIMSK1 |= (1 << OCIE1A); //Set interrupt on compare match
        TCCR1B |= (1 << CS12) | (1 << CS10); // set prescaler to 1024 and
start the timer
    }

void setupUsart(){
    /*Set baud rate */
    UBRRE0H = (MYUBRR) >> (8);
    UBRRE0L = MYUBRR;

    UCSRB |= (1 << RXEN0); // Enable
receiver and transmitter
    UCSRB |= (1 << RXCIE0); // enable
interrupt
    UCSRC |= (1 << UCSZ01) | (1 << UCSZ00); // Set frame: 8data, 1 stp
}

void HANDLE_LCD(){
    if( B_NEW_MSG ){
        LCDInit();
        LCDWriteTextFill(inBuffer);
        B_NEW_MSG = FALSE;
    }
}

void HANDLE_TORNIQUETE(){
    if( B_OPEN_A ){ // dejar abierto en sentido A,
cerrado en B
        SOLENA_open;
        SOLENB_close;
    }else if( B_OPEN_B ){ // dejar abierto en sentido B, cerrado
en A
        SOLENB_open;
        SOLENA_close;
    }else{
        if(JUMPER_status == LOW){
            // dejar cerrado en ambos sentidos
            SOLENA_open;
            SOLENB_open;
        } else {
            // dejar abierto en sentido B, cerrado en A
            SOLENB_open;
            SOLENA_close;
        }
    }
}

void HANDLE_NETBURNER(){
    if( B_PASS_COMPLETED_A ){
        if(B_OPEN_A){
            B_NEWPASS = FALSE;

```

```
        B_OPEN_A = FALSE;
        B_OPEN_B = FALSE;
        time_out = 0;
        LUZ_lo;

        PASScomplA_hi;
        LED1_hi;

        _delay_ms(500);

        PASScomplA_lo;
        LED1_lo;
    }

    B_PASS_COMPLETED_A = FALSE;
}else if(B_PASS_COMPLETED_B ){
    if(B_OPEN_B){
        B_NEWPASS = FALSE;
        B_OPEN_A = FALSE;
        B_OPEN_B = FALSE;
        time_out = 0;
        LUZ_lo;
    }

    PASScomplB_hi;
    LED2_hi;

    _delay_ms(500);

    PASScomplB_lo;
    LED2_lo;

    B_PASS_COMPLETED_B = FALSE;
}
}
```