

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACIÓN

“Control de Robot Oruga con controlador Orangutan SV-328 mediante mensajes de texto utilizando interfaz con el dispositivo Narobo Drone Cell”

TESINA DE SEMINARIO

Previa la obtención del Título de:

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Tatiana Estefanía Silva Ríos

Julio César Jaramillo Tinoco

GUAYAQUIL – ECUADOR

AÑO 2011

AGRADECIMIENTO

A Dios por ser él quien me guía cuando me encuentro perdida, sin rumbo. A mi compañero y amigo Julio por permitirme instruirme junto a él, por compartir conocimientos conmigo y por apoyarme en cada momento.

A mis padres, Edgar y Alba, por ser el pilar fundamental en la realización de mis sueños, son ellos quienes aportan consejos y acciones en mi vida personal y profesional.

A mis hermanos, María Eugenia y Edgar, por sus abrazos y palabras de aliento. Gracias por su comprensión y voluntad. A Carlos Yoong, sus consejos, palabras, fe y amor me han llevado a pensar que en esta vida no se es nada sin creer en lo imposible.

Tatiana Estefanía Silva Ríos.

Quisiera agradecer en primer lugar a Dios y segundo a mi madre la cual siempre ha sido mi apoyo incondicional, ella me ha enseñado que la constancia y la perseverancia siempre harán que cualquier persona cumpla sus sueños y metas, soy una persona sumamente afortunada ya que sin sus sabias palabras y consejos probablemente no estaría escribiendo estas líneas.

También le doy las gracias a mi papa, a mi hermana y amigos, que siempre me han escuchado y ayudado en todo momento.

Al Ing. Carlos Valdivieso por su esfuerzo y confianza en la culminación del presente informe

Julio César Jaramillo Tinoco.

DEDICATORIAS

Dedico todo mi esfuerzo y perseverancia a
Dios, porque con él todo es posible.

Tatiana Estefanía Silva Ríos.

A todos los que están junto a mí, en
especial a mi madre ya que sin su apoyo y
consejos no habría conseguido esta meta.

Julio César Jaramillo Tinoco.

TRIBUNAL DE SUSTENTACIÓN



Ing. Carlos Valdivieso

PROFESOR DEL SEMINARIO DE GRADUACIÓN



Ing. Hugo Villavicencio

PROFESOR DELEGADO POR EL DECANO DE LA FACULTAD

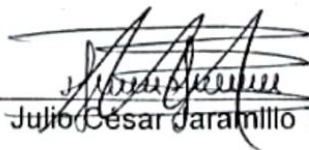
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este trabajo, nos corresponde exclusivamente; y el patrimonio intelectual del mismo a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL".

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Tatiana Estefanía Silva Ríos



Julio César Jaramillo Tinoco

RESUMEN

El presente trabajo “Control de Robot Oruga con Controlador Orangutan SV-328 mediante mensajes de texto utilizando interfaz con el dispositivo Narobo Drone Cell”, es parte del Seminario de Graduación “Microcontroladores Avanzados”, tiene como objeto el dar a conocer la manera como los microcontroladores revolucionan de una forma bastante sencilla y ágil la forma de trabajar con circuitos eléctricos y aplicaciones específicas para diferentes áreas.

Estos dispositivos permiten reemplazar gran cantidad de circuitos digitales en un solo chip, con lo cual se agilizan las etapas de desarrollo y aplicación así como reducen radicalmente los costos y el tiempo de desarrollo en la implementación de diseños de miles de compuertas lógicas.

La programación la realizaremos en un lenguaje denominado GCC que aunque no es el lenguaje nativo del microcontrolador, sin embargo permite un desarrollo mucho más flexible y da la facilidad de utilizar códigos abiertos para desarrollar cualquier proyecto con un ahorro de tiempo mayor.

INDICE GENERAL

RESUMEN

INDICE GENERAL

INDICE DE FIGURAS

INTRODUCCIÓN

1.	GENERALIDADES	1
1.1	Descripción del Proyecto	1
1.2	Alcance y limitaciones del proyecto	3
1.3	Objetivos.....	4
1.3.1	Objetivos Generales	4
1.3.2	Objetivo Específico	5
1.4	Antecedentes.....	5
1.5	Identificación del Problema.....	6
1.6	Descripción breve de la posible solución.....	7

2.	FUNDAMENTO TEORICO	8
2.1	Herramientas de Software	8
2.1.1	AVR Studio	8
2.1.2	ATMEL AVR Assembler	8
2.2	Herramientas de Hardware.....	10
2.2.1	Microcontrolador ATMEGA328P	10
2.2.2	Orangutan SV-328 Robot Controller.....	10
2.2.3	Narobo Drone Cell.....	12
2.2.4	Comandos AT.....	13
2.2.5	Robot Chassis RP5-CH02.....	15
2.2.6	Telefonia Celular	17
3.	DISEÑO E IMPLEMENTACIoN	19
3.1	Diagrama de Bloques del Robot Oruga.....	19
3.2	Descripción Detallada de Cada Bloque	20
3.2.1	Bloque de Presentación de Datos	21
3.3	Descripción del Funcionamiento.....	21

3.3.1	Proceso de Recepción de un SMS en el Narobo Drone Cell ..	22
3.4	Diagrama de Flujo General del Proyecto	23
3.4.1	Descripción de Subrutinas.....	24
3.5	Formato del Mensaje de Texto	26
4.	SIMULACIÓN Y RESULTADOS.....	28
4.1	Simulación de la Transmisión de Datos del Módulo Orangutan	28
4.2	Verificación de la Transmisión de Datos del Narobo Drone Cell	30
4.3	Resultados Experimentales con el Robot Oruga	34
CONCLUSIONES		
RECOMENDACIONES		
ANEXOS		
REFERENCIAS BIBLIOGRÁFICAS		

ABREVIATURAS

SMS	Short Message System
IDE	Integrated Development Environment
MTSO	Mobile Telephone Switching Office
BS	Base Station
UART	Universal asynchronous receiver/ transmitter
SIM	Subscriber Identity Module
LCD	Liquid Crystal Display
USB	Universal Serial Bus
RS-232	Recommended Standard 232
TTL	Transistor-Transistor Logic
SV-328	Spectra Video 328
AT	Attention
MAX 232	Conversor TTL a RS-232

MODEM	Modulador Demodulador
GNU	Acrónimo Recursivo
RISC	Reduced Instruction Set Computer
ISP	In System Programation
EEPROM	Electrically-Erasable Programmable Read Only Memory
SRAM	Static Radom Access Memory
SPI	Serial Peripheral Interface
A/D	Conversión Analógica a Digital

INDICE DE FIGURAS

Figura 1-1 Arquitectura del Proyecto	2
Figura 2-1 Controlador de Robots Orangutan SV-328.....	11
Figura 2-2 NaroBo Drone Cell.....	12
Figura 2-3 Chassis RP5-CH02.....	16
Figura 2-4 Escritura de un SMS.....	17
Figura 2-5 Teclado común de teléfono móvil.	18
Figura 3-1 Diagrama de Bloques del Robot Oruga	19
Figura 3-2 Ejemplo de Visualización de comando en curso	21
Figura 3-3 Formato de Letras y Números para el recorrido	27
Figura 4-1 Simulación de Secuencia de Inicialización	28
Figura 4-2 Envío de Comandos para la Recepción de Mensajes de Texto	30

Figura 4-3 Encendido del Módulo Narobo Drone Cell.....	31
Figura 4-4 Recepción de SMS desde el Módulo Narobo Drone Cell	32
Figura 4-5 Envío de SMS desde el Módulo Narobo Drone Cell.....	33
Figura 4-6 Robot Oruga Funcionando en Espera de el SMS.....	34

INDICE DE TABLAS

Tabla 1-1 Distancia máxima soportada por las diversas tecnologías	6
Tabla 2-1 Comandos AT principales.....	15
Tabla 3-1 Descripción de las Letras en el Recorrido	27
Tabla 4-1 Comandos utilizados para recibir Mensajes de Texto.....	29
Tabla 4-2 Comandos AT que Intervienen en el envío de un SMS	33

INTRODUCCIÓN

La tecnología ha venido cambiando, evolucionando y con ella se han dado muchas mejoras en los diferentes campos de la misma. En este trabajo se estudiará y analizará sobre un desarrollo específico que se le puede dar a un microcontrolador denominado ATMEGA128P, pero que será de gran utilidad en la vida cotidiana y tendría múltiples usos con pequeños cambios.

Un Microcontrolador permite al usuario escoger o crear el código del programa a ser usado en el dispositivo, en lugar de ser definido por el fabricante.

La funcionalidad que le daremos a este dispositivo será la de controlar un Robot tipo Oruga utilizando los diferentes recursos que nos ofrece.

También estudiaremos brevemente acerca de los dispositivos como son el NaroBo Drone Cell que es capaz de permitirnos la conexión con el sistema de telefonía celular. Todos estos conceptos serán ampliados en el presente trabajo y le daremos una aplicación.

El proyecto se ha estructurado en 4 capítulos que los detallamos a continuación:

En el capítulo 1, encontramos la descripción del proyecto, donde se explica de forma general el funcionamiento, sus alcances, limitaciones, los antecedentes y los problemas que solucionará.

El capítulo 2, trata todo lo que es el marco teórico, de todos los dispositivos utilizados en el proyecto, protocolos de comunicación utilizados, y todos los conceptos claves en base a los cuales se ha trabajado.

El capítulo 3, trata el diseño e implementación del proyecto, el cual se va a orientar a un ambiente de laboratorio, se realiza un diagrama de bloques básico y general para luego ir explicando el funcionamiento detallado de cada uno de los componentes.

En el capítulo 4, se realizan las pruebas y se analizan los resultados en base a los parámetros establecidos, se detallan las ventajas y desventajas del mismo, así como también las recomendaciones en caso de posibles fallas presentadas por un problema físico o de configuración de los dispositivos.

CAPITULO 1

1. GENERALIDADES

En este capítulo se aborda el planteamiento del proyecto, así como sus alcances y limitaciones. Se expone además la razón de esta implementación y se explica la solución propuesta al problema planteado.

1.1 Descripción del Proyecto

El proyecto consiste en controlar la ruta que recorre un robot tipo Oruga mediante mensajes de texto cortos (SMS) desde cualquier teléfono celular o dispositivo que posea esta tecnología.

El Robot Oruga consiste de un chasis que tiene dos Orugas para su desplazamiento, el cual es guiado a su vez por un controlador de robots denominado Orangutan SV-328. Este controlador tiene dos funciones, la de direccionar la ruta por la cual el robot se mueve, y la segunda que es la de intervenir en la comunicación con otro dispositivo que es el NaroBo Drone Cell el cual es el encargado de conectar al robot con el sistema de telefonía celular.

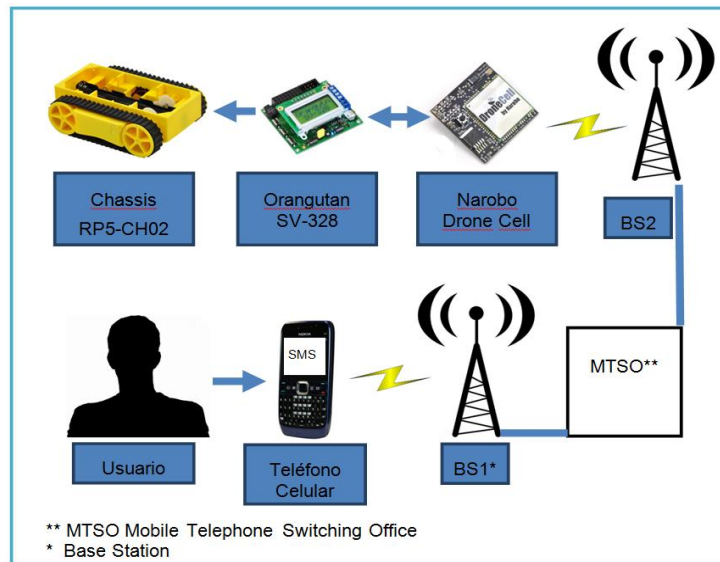


Figura 1-1 Arquitectura del Proyecto

Como se puede observar en la Figura 1-1 Arquitectura del Proyecto, se muestra la forma como están conectados cada uno de los elementos del proyecto. Tenemos al usuario que es quien envía la ruta que desea que recorra el Robot Oruga a través de un mensaje de texto corto escrito sobre cualquier teléfono celular, este mensaje se envía a través del sistema de Telefonía Celular, primeramente a través de sus estaciones base (BS) para luego pasar a través de la Oficina de Conmutación de Telefonía Móvil (MTSO), después a la estación base más cercana al dispositivo Narobo Drone Cell que es quién finalmente recibe el mensaje SMS, este lo almacena para luego enviarlo al controlador principal, que es el microcontrolador ATMEGA328P que se encuentra en el Interior del controlador de robots Orangutan SV-328. La comunicación entre el dispositivo Narobo Drone Cell y el Orangutan SV-328 es mediante UART o comunicación serial asíncrona. El microcontrolador antes mencionado es quién se encargaría de interpretar el mensaje recibido y ejecutar el recorrido que tendría el Robot Oruga finalmente.

1.2 Alcance y limitaciones del proyecto

El desarrollo del proyecto se realiza en un ambiente de Laboratorio, en el cual utilizamos un software específico proveído por el fabricante del microcontrolador.

Este proyecto posee las siguientes limitaciones:

- El área de trabajo del Robot Oruga en lo que es la comunicación, depende directamente de la cobertura que tenga la operadora celular con la que se está trabajando, por lo que el área de funcionamiento del proyecto tanto en lo que es la parte de control y el robot podrían funcionar a escasos centímetros entre sí, o de un país a otro.
- El tiempo de trabajo del Robot Oruga está dado por el tipo de batería que se utilice y la carga con la que éstas cuenten. Para incrementar este tiempo se utilizan en el proyecto pilas recargables AA de Níquel-Hidruro.
- El recorrido que cumpla el robot está limitado a la cantidad de caracteres que se ingresan en un mensaje de texto corto que estaría en máximo 140 caracteres que es lo que normalmente utilizan la operadoras de telefonía celular.
- No se puede tener certeza de la ubicación del Robot ya que no se poseen sensores de ubicación, pero al ser controlado por un microcontrolador reprogramable este proyecto podría perfectamente ser mejorado con la implementación de estos sensores especializados.

1.3 Objetivos

1.3.1 Objetivos Generales

Los objetivos generales con los cuales se realizó el trabajo son los siguientes:

- Familiarizarse con el uso de microcontroladores de la familia ATMEL y su ambiente de trabajo.
- Aprender acerca del funcionamiento de los Robots Oruga y como controlar a los mismos a través de módulos especializados como lo es el Orangutan SV-328.
- Conocer el uso de las librerías para lenguaje GCC disponibles y desarrollar destrezas que nos permitan desplegar cualquier tipo de aplicación a futuro.
- Familiarizarnos con dispositivos de comunicación inalámbrica como lo es el NaroBo Drone Cell.
- Implementar el proyecto, orientándolo a un ambiente de laboratorio y en relación con los resultados obtener conclusiones y recomendaciones acerca de las pruebas realizadas.

1.3.2 Objetivo Específico

El objetivo principal de este proyecto es desarrollar un código en lenguaje C para controlar un Robot tipo Oruga (RP5-CH02) mediante mensajes SMS, con el uso del controlador Orangutan SV-328 y con el dispositivo de comunicación Narobo Drone Cell.

1.4 Antecedentes

Desde hace mucho tiempo atrás el hombre ha querido controlar todos los dispositivos de una forma remota, para lo cual se hicieron una gran cantidad de desarrollos entre los más notables están el que fue desarrollado por Nikola Tesla en el año de 1893, él publicó un artículo llamado “Método de un aparato para el mecanismo de control de vehículo o vehículos en movimiento”. Le seguiría Leonardo Torres Quevedo quién en 1903 ya presentaría una patente denominada “Telekino” y que se consideraría como el primer control remoto que utilizaba radiofrecuencia en el mundo. Desde este punto en adelante los avances de la ciencia hicieron que cada día aparezcan nuevos y mejores dispositivos para el control remoto de diversas máquinas.

Actualmente encontramos una gran variedad de estos dispositivos, para distintas aplicaciones no solo en control de movimiento, y que utilizan diversas tecnología, desde las que son hechas en base al infrarrojo, ultrasonido, radio frecuencia, entre otras.

1.5 Identificación del Problema

Para el control de dispositivos y robots es común utilizar varias tecnologías desde las que encontramos, infrarrojo, ultrasonido (antiguamente), las que son hechas a partir de ondas electromagnéticas utilizando diversas modulaciones como son ASK, FSK, entre otras, o la mezcla de estas en tecnologías más elaboradas como son el Bluetooth, WiMAX, Wi-Fi, ZigBee, entre otras.

Tabla 1-1 Distancia máxima soportada por las diversas tecnologías

Tipo de Comunicación	Distancia Máxima [metros]
Infrarrojo	Pocos metros
Bluetooth Clase 1	100
Zig-Bee	1500
WiMAX	80 000
Wi-Fi	95

De acuerdo a la Tabla 1-1 Distancia máxima soportada por las diversas tecnologías, tenemos que todas estas solucionarían cualquier problema para el control a una corta distancia, siempre y cuando no existan obstáculos entre el control y el robot que podrían cortar el paso de la señal entre ambos, pero si se requiere que el control no dependa de la distancia que exista entre ambos, que sin importar donde se encuentren tanto el robot como el control estén en comunicación permanente ambos.

1.6 Descripción breve de la posible solución

Para solucionar el problema anterior, en este proyecto se expone la idea de utilizar el sistema de telefonía celular, que está disponible en casi cualquier parte del mundo, entonces el control del Robot ya no estaría limitado por su distancia física, resolviendo así el problema planteado anteriormente.

Con el uso de módulos como el Narobo Drone Cell se tiene el acceso a la red de telefonía celular y con el controlador de Robots Orangutan SV-328 tendríamos una solución completa para el mando de robots desde casi cualquier ubicación del mundo.

El único mecanismo que se requerirá para controlar el Robot será un teléfono celular con capacidad de mensajes de texto cortos o algún dispositivo que tenga esa característica.

CAPITULO 2

2. FUNDAMENTO TEORICO

En este capítulo se presentan los principales conceptos que vamos a utilizar en este proyecto. Para de esta manera familiarizarnos con cada uno de ellos.

2.1 Herramientas de Software

2.1.1 AVR Studio

AVR Studio es un Ambiente de Desarrollo Integrado (IDE) para escribir y correr aplicaciones de AVR en Windows 9x/ ME/ NT/ 2000/ XP/ VISTA/ 7. AVR Studio provee herramientas de trabajo, editor de archivos fuente, simulador, assembler, C/C++, programación, emulación y depuración en el microcontrolador; todo esto para microcontroladores fabricados por la empresa ATMEL.

2.1.2 ATMEL AVR Assembler

El ensamblador traduce conjuntos de código fuente en código máquina. El código máquina generado se puede utilizar como entrada a un simulador como el simulador de ATMEL AVR o un emulador como el AVR ATMEL emulador en circuito. El ensamblador también genera un

código PROMable que se puede programar directamente en la memoria del programa de un microcontrolador AVR.

El ensamblador genera un código que fija las asignaciones.

El ensamblador se ejecuta en Microsoft Windows 3.11, Windows 95 y Microsoft Windows NT.

2.1.3.- ATMEL AVR GCC

AVR-GCC es un compilador de lenguaje C que tiene un código de alto nivel y crea un archivo binario que se puede programar en su propio AVR.

AVR-GCC se basa técnicamente sólo en lenguaje de compilación C y C++. Es un puerto de GCC la colección de compiladores de GNU con el procesador del regulador.

2.1.2.1 Pololu AVR Library

Las librerías utilizadas en el AVR Pololu son de gran ayuda para desempeñar operaciones para diferentes proyectos en los que se requiera la utilización del Orangutan SV-328 y el Pololu 3pi.

El popular compilador gratuito GNU C/C++ trabaja perfectamente con el 3pi, Atmel AVR Studio provee un cómodo ambiente de desarrollo y el extenso set de librerías de Pololu hace verdaderamente fácil la interface con todo tipo de hardware.

2.2 Herramientas de Hardware

2.2.1 Microcontrolador ATMEGA328P

El alto rendimiento pico Power Atmel de 8 bits del microcontrolador AVR RISC combina 16KB de memoria flash, ISP con lectura y escritura, 512KB de EEPROM, 1 KB de SRAM, 23 registros de entrada/salida de propósito general, 32 registros de propósito general de trabajo, 3 temporizadores/contadores para comparar los modos, las interrupciones internas y externas, programación serial USART, un byte orientado a interfaz de dos hilos de serie, puerto serial SPI, 6 canales de 10 bits convertidor A/D, temporizador de vigilancia programable con el oscilador interno, y cinco modos seleccionables de software de ahorro de energía. El dispositivo funciona entre 2.7 a 5.5 Voltios.

Mediante la ejecución de instrucciones de gran alcance en un sólo ciclo de reloj, el dispositivo logra tasas de transferencia cerca de 1 MIPS por MHz, equilibrando el consumo de energía y velocidad de procesamiento. Mayor detalle del microcontrolador se encuentra en el ANEXO A.

2.2.2 Orangutan SV-328 Robot Controller

El Robot Controlador Orangután SV-328 es una solución completa para el control de pequeños robots funcionando de 6 a 13.5 Voltios. El módulo es pequeño (2.15 pulgadas x 1.9 pulgadas) e incluye un potente

microcontrolador Atmel AVR ATmega328P, dos puertos bidireccionales de motor, cada uno capaz de proporcionar una corriente continua de 1 Ampere, un LCD de 8 caracteres x 2 líneas, un zumbador, tres botones para el usuario, y dos LED.

Ocho líneas de entrada/salida de propósito general con hasta ocho canales de entrada analógicos permiten la adición de sensores o la expansión del sistema. Estas líneas son presentadas a 0.1 pulgadas como se muestra en la imagen de abajo, de izquierda a derecha, la fila inferior de pines son: PC5, PC4, PC3, PC2, PC1, PC0, PD1 y PD0. El Orangután SV-328 cuenta con un regulador de conmutación de 5 Voltios capaz de transportar hasta 3 Ampere de corriente. Se puede acceder a esta fuente de voltaje a través de los pines de alimentación y de tierra dedicadas.

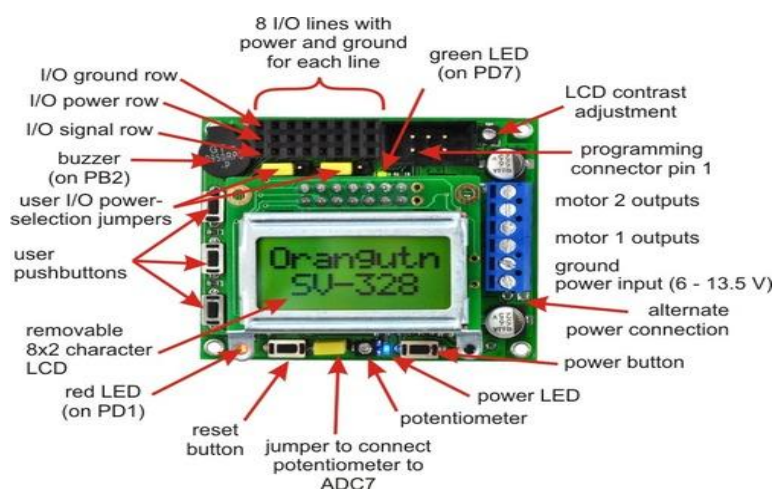


Figura 2-1 Controlador de Robots Orangutan SV-328

El Orangután SV-328 cuya imagen se muestra en la Figura 2-1 Controlador de Robots Orangutan SV-328 se basa en el microcontrolador Atmel AVR mega328 mencionado anteriormente, que funciona a 20 MHz y dispone de 32 Kbytes de memoria Flash de programa, 2 Kbytes de SRAM, y 1024 bytes de memoria EEPROM. Debido a que el usuario tiene acceso directo al microcontrolador, cualquier desarrollo de software para microcontroladores AVR de Atmel, incluyendo Atmel AVR Studio y WinAVR GCC/C++ Compiler, es compatible con el Orangután SV-328. Un programador en circuito, tales como nuestro USB programador AVR, se requiere para la programación del Orangután SV-328.

2.2.3 Narobo Drone Cell

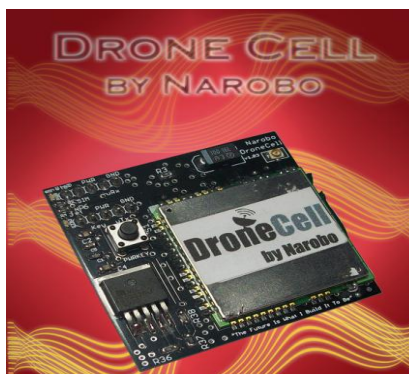


Figura 2-2 Narobo Drone Cell

El módulo Narobo Drone Cell que se muestra en la Figura 2-2 Narobo Drone Cell es un dispositivo con tecnología celular para usos en

robótica, electrónica y minicomputadoras. Su funcionalidad abarca un todo en dispositivos de comunicación: comunicar a través de textos, llamadas telefónicas, o incluso a través de Internet. Cualquier dispositivo con una UART TTL puede comunicarse con el Narobo Drone Cell.

Se trabajará en cualquier lugar en el que haya una antena de telefonía móvil. Controla un robot en Los Ángeles desde tú computadora portátil en Nueva York. Imagínese el envío de un mensaje de texto a su circuito a medida que comienza a rodar su coche. O el envío de datos de forma inalámbrica a través de Internet a tú computadora mientras vuelas sobre Afganistán. O conseguir GPS y datos meteorológicos de su globo de aire de alta altitud. O incluso que su mayordomo robotdial, su teléfono celular y correo electrónico cuando se detecta un intruso. Mayor detalle en el ANEXO B.

2.2.4 Comandos AT

Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM.

Los comandos AT fueron desarrollados en 1977 por Dennis Hayes como un interfaz de comunicación con un MODEM para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Más adelante, con el avance del baudio, fueron las compañías

Microcom y US Robotics las que siguieron desarrollando y expandiendo el juego de comandos hasta universalizarlo.

Los comandos AT se denominan así por la abreviatura de attention. Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil GSM también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles GSM poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a los terminales, permiten acciones tales como realizar llamadas de datos o de voz, leer y escribir en la agenda de contactos y enviar mensajes SMS, además de muchas otras opciones de configuración del terminal.

Es claro que la implementación de los comandos AT corresponde a los dispositivos GSM y no depende del canal de comunicación a través del cual estos comandos sean enviados, ya sea cable de serie, canal Infrarrojos, Bluetooth, etc.

Una lista de los comandos principales utilizados en este proyecto se pueden encontrar en la Tabla 2-1 Comandos AT principales, en la cual tenemos los comandos para el manejo de mensajes de texto SMS utilizados en este proyecto.

Tabla 2-1 Comandos AT principales

Comandos generales	
Nombre	Descripción
AT+CGMI	Identificación del fabricante
AT+CGSN	Obtener número de serie
AT+CIMI	Obtener el IMSI
AT+CPAS	Leer estado del modem
Comandos del servicio de red	
Nombre	Descripción
AT+CSQ	Obtener calidad de la señal
AT+COPS	Selección de un operador
AT+CREG	Registrarse en una red
AT+WOPN	Leer nombre del operador
Comandos AT para SMS	
AT+CPMS	Seleccionar lugar de almacenamiento de los SMS
AT+CMGF	Seleccionar formato de los mensajes SMS
AT+CMGR	Leer un mensaje SMS almacenado
AT+CMGL	Listar los mensajes almacenados
AT+CMGS	Enviar mensaje SMS
AT+CMGW	Almacenar mensaje en memoria
AT+CMSS	Enviar mensaje almacenado
AT+CSCA	Establecer el Centro de mensajes a usar
AT+WMSC	Modificar el estado de un mensaje

2.2.5 Robot Chassis RP5-CH02

El chassis RP5 mostrado en la Figura 2-3 Chassis RP5-CH02, es una plataforma de robot móvil que utiliza orugas para conducirse sobre una variedad de superficies, este maneja sin problemas pequeños golpes y obstáculos. Sólo se tiene que conectar un controlador de robot o un controlador de motor al soporte de la batería incluida de 6 AA y dos motores de corriente continua para obtener la plataforma móvil.



Figura 2-3 Chassis RP5-CH02

El chasis cuenta con un cuerpo durable de plástico amarillo con capacidad para albergar el soporte de la batería incluida y algo de electrónica adicional, bandas de rodadura de goma para hacer frente a superficies irregulares, y motores de gran alcance que son lo suficientemente fuertes como para levantar su propio peso y se deja alcanzar velocidades tan altas como 6 m/s. Las dimensiones exteriores del chasis son de aproximadamente 7 cm de largo, 5.5 cm de ancho y 2.5 cm de altura. Uno de los engranajes en cada tren de potencia tiene un agujero que se puede utilizar para el seguimiento de la rotación del motor con una resolución de 4 conteos por revolución del eje de salida, que corresponde a la distancia lineal de 1.5 pulgadas.

2.2.6 Telefonía Celular

2.2.6.1 Short Message System

El servicio de mensajes cortos (SMS) es el componente de servicio de comunicación de texto de teléfono, web o sistemas de comunicaciones móviles, utilizando protocolos de comunicación estandarizados que permiten el intercambio de mensajes de texto cortos entre líneas fijas o teléfonos móviles. Mensajería de texto SMS es la aplicación más utilizada de datos en el mundo, con 2.4 millones de usuarios activos, el 74% de todos los abonados a la telefonía móvil. El término de SMS se utiliza como sinónimo de todos los tipos de mensajes cortos de texto.



Figura 2-4 Escritura de un SMS

En la Figura 2-4 Escritura de un SMS se distingue un teléfono celular que posee esta tecnología y que es muy fácil de utilizar.

SMS como se usa en los teléfonos modernos se originó a partir de radiotelegrafía y más tarde se define como parte del Sistema Global para Comunicaciones Móviles (GSM). En las normas de 1985 se define como un medio para enviar mensajes de hasta 160 caracteres, desde y hacia teléfonos móviles GSM. Desde entonces, el apoyo para el servicio se ha ampliado para incluir otras tecnologías móviles tales como ANSI redes CDMA y AMPS digital, así como las redes de satélites y terrestres. La mayoría de los mensajes SMS son mensajes de texto de móvil a móvil, aunque la norma admite otros tipos de emisión de mensajes también.



Figura 2-5 Teclado común de teléfono móvil

La gran mayoría de teléfonos celulares poseen un teclado con botones que están ordenados como se muestra en la Figura 2-5

Teclado común de teléfono móvil

CAPITULO 3

3. DISEÑO E IMPLEMENTACION

En este capítulo se procederá al diseño e implementación del proyecto. Para la explicación se ha elaborado un diagrama de bloques así como también su diagrama de flujo. Las explicaciones detalladas se darán a continuación.

3.1 Diagrama de Bloques del Robot Oruga

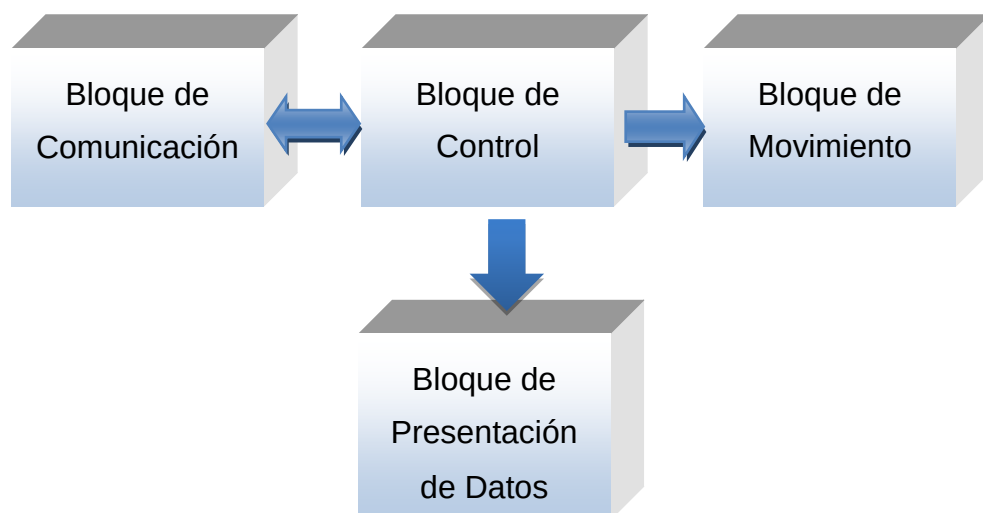


Figura 3-1 Diagrama de Bloques del Robot Oruga

En el diagrama podemos observar que se encuentra distribuido en cuatro partes principales las cuales se detallarán a continuación.

3.2 Descripción Detallada de Cada Bloque

3.2.1 Bloque de Comunicación

El Bloque de comunicación está formado por el Narobo Drone Cell y el Microcontrolador ATMEGA328P los cuales son los encargados de esta función primaria. El primero ya que permite la interconexión con la red de telefonía celular, y el segundo que es el que envía instrucciones con las cuales se pueden enviar y recibir los mensajes.

3.2.2 Bloque de Control

Está conformado principalmente por el microcontrolador que es el encargado de controlar todos los dispositivos que se conectan a él, como son: la LCD de 8 caracteres y 2 líneas, el Narobo y el driver para el control de los motores que mueven al Robot.

3.2.3 Bloque de Movimiento

En este bloque intervienen los motores, el driver para los motores y el microcontrolador que en su interior posee un bloque especializado que permite generar PWM lo que hace posible la manipulación de la velocidad y aceleración que se desee que los motores alcancen.

3.2.4 Bloque de Presentación de Datos

Aquí encontramos una LCD de 8 caracteres y dos líneas con un microcontrolador interno denominado Hitachi hd44780, el cual nos muestra la condición actual en la que se encuentra el Robot Oruga, es decir que comando se está ejecutando, como por ejemplo en la Figura 3-2 Ejemplo de Visualización de comando en curso podemos observar Virar a la Izq. Entre otros.

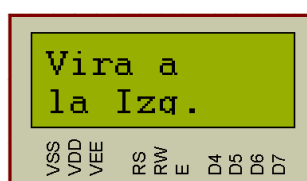


Figura 3-2 Ejemplo de Visualización de comando en curso

3.3 Descripción del Funcionamiento

Como se vio en el capítulo 2, para que el Narobo Drone Cell trabaje con el microcontrolador del Orangutan SV-328, se necesita comunicación RS-232, con 8 bits de datos, un bit de parada y un bit de paridad, a una velocidad de 115200 baudios o sino a 38400 baudios y utiliza comandos AT para su configuración y uso, por lo que se procedió a utilizar una librería especializada entregada por el mismo fabricante. A continuación se muestra el proceso para recibir un mensaje de texto desde el Narobo, el cual es una parte fundamental en el funcionamiento del proyecto.

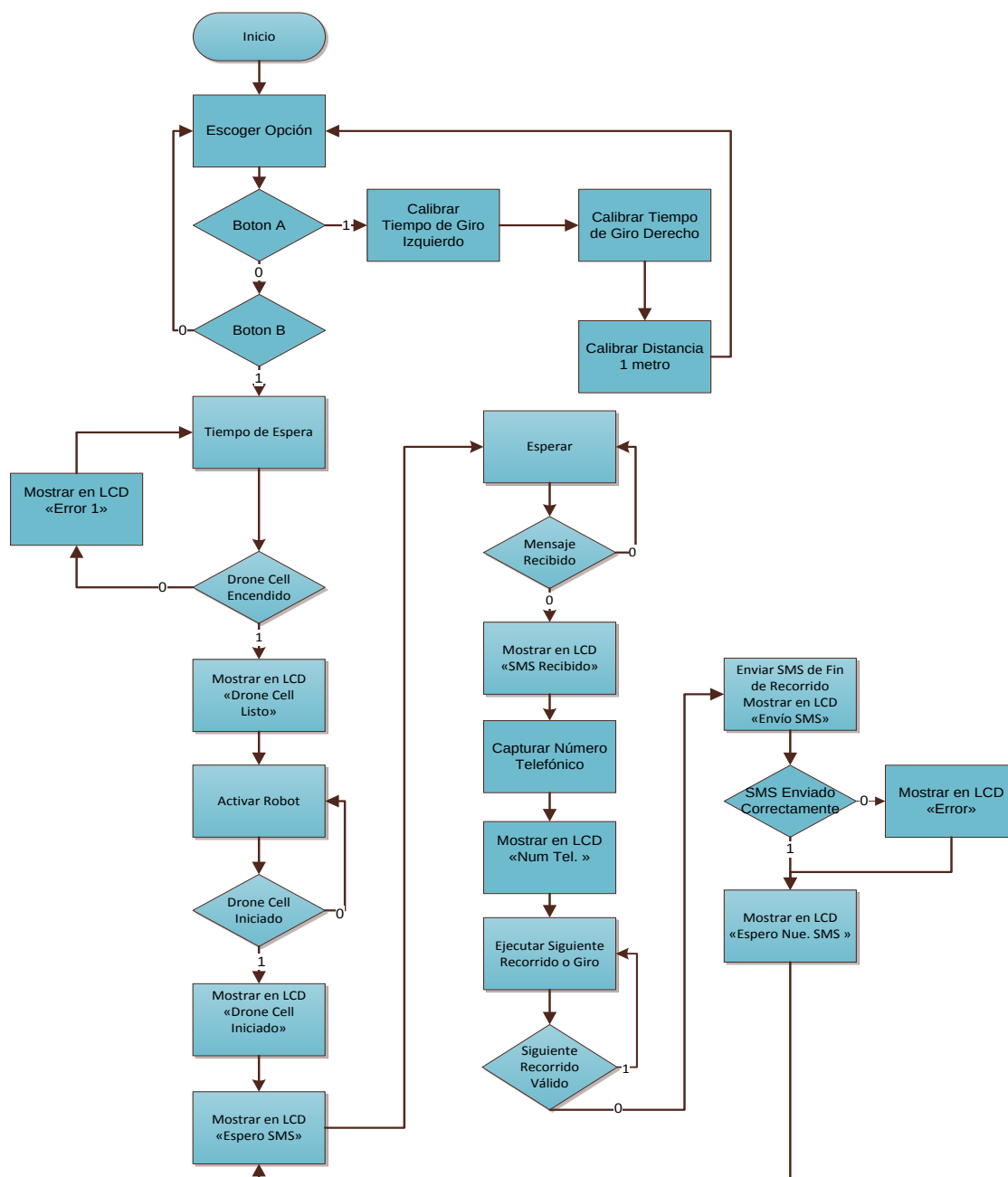
3.3.1 Proceso de Recepción de un SMS en el Narobo Drone Cell

El proceso es el siguiente:

- Enviar el comando `AT+CMGF=1\r`, con lo que le indicamos al módulo que configure el formato de los mensajes. El módulo responderá con “OK” en caso de haber recibido correctamente el comando, caso contrario “ERROR”.
- Luego enviamos `AT+CMGDA=“DEL ALL”`, para borrar todos los mensajes que se encuentren almacenados en la tarjeta SIM, y así poder utilizar los que lleguen luego.
- El siguiente paso es enviar `AT+CNMI=0,2` para deshabilitar códigos de error no solicitados y que el mensaje recibido por el Narobo Drone Cell sea enviado inmediatamente al microcontrolador.
- Finalmente la instrucción `AT+CMGR=1` permite leer el mensaje de entrada, en el Capítulo 4 se puede apreciar de mejor manera este proceso a través del uso de un terminal de computadora conectado directamente al Narobo Drone Cell.

Todo este procedimiento, lo realiza el microcontrolador utilizando la librería de control del Narobo.

3.4 Diagrama de Flujo General del Proyecto



El código principal se lo puede observar en el Anexo C.

3.4.1 Descripción de Subrutinas

Inicialización de Parámetros

Aquí se ejecutan las funciones encargadas de inicializar los parámetros en el microcontrolador como son la velocidad de transmisión serial, asignación de entradas, salidas, así como las funciones que inicializan el Display LCD y el Narobo Drone Cell.

Drone Cell Iniciado

Cuando esta bandera se enciende es porque no ha existido ningún inconveniente en la inicialización, por lo que el programa podrá seguir con normalidad.

Mostrar en LCD “Mensaje...”

Esta función imprime en la pantalla de la LCD una cadena de caracteres que está escrita en el código del microcontrolador que es el “Mensaje...”.

Botón A, B

Estos son los botones que ya vienen incluidos en el controlador de Robots Orangutan SV-328 para interactuar con el usuario.

Activar Robot

Al presionar el botón B el robot entra en modo de funcionamiento con lo que podrá perfectamente recibir órdenes desde cualquier celular.

Enviar SMS

Esta función envía un mensaje de texto corto al celular que esté asignado internamente en el microcontrolador. Se utiliza principalmente para que el usuario que controle el robot esté al tanto de la situación en la que se encuentra el robot.

Esperar

En esta parte del código el robot simplemente espera a que alguna orden llegue, sin ejecutar ninguna acción adicional.

Calibrar Tiempo de Giro Izquierdo, Derecho, 1 Metro

Estas funciones se encargan de calibrar el tiempo que se demora en girar a la derecha, Izquierda o en recorrer un metro. Aquí interviene el usuario.

Ejecutar siguiente recorrido o giro

Esta subrutina ejecuta el siguiente recorrido o giro que debe realizar el robot, si el mensaje contiene caracteres inválidos, sale inmediatamente del lazo y con eso concluye el recorrido del robot.

Capturar número telefónico

En esta subrutina el microcontrolador captura el número telefónico de quién envió el mensaje para luego utilizarlo al momento de finalizar el recorrido con un mensaje de confirmación.

3.5 Formato del Mensaje de Texto

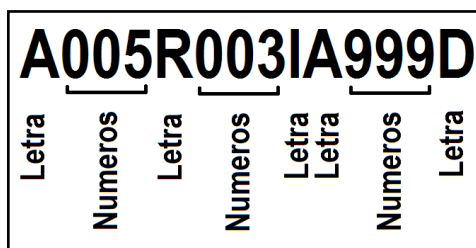
El formato que debe tener el mensaje de texto para que el robot siga la ruta deseada es el siguiente:

- Primeramente las únicas letras que acepta el robot son: A, R, I, D y los números, las letras corresponden a la dirección por ejemplo I es Izquierda, D derecha, en la Tabla 3-1 Descripción de las Letras en el Recorrido, se encuentra más detallada la función de cada una de las letras.
- Los números indican la distancia en centímetros que se desea que recorra. La distancia máxima que puede recorrer el robot en un solo comando es de 999 centímetros, más adelante se detalla el por qué.

Tabla 3-1 Descripción de las Letras en el Recorrido

Letras	Descripción
A, a	Indica al robot que siga hacia adelante
R, r	Indica al robot que siga hacia atrás
I, i	Giro a la izquierda 90 grados
D, d	Giro a la derecha 90 grados

- Luego de cada letra A, a o R, r va necesariamente la distancia que se desea que recorra en centímetros. Siempre debe existir una cantidad de 3 números consecutivos después de la letra como se aprecia en la Figura 3-3 Formato de Letras y Números para el recorrido, se le pide al robot que avance 5 centímetros hacia adelante luego que retroceda 3 centímetros, que gire a la izquierda, después que avance 999 centímetros y finalmente que gire a la derecha.

**Figura 3-3 Formato de Letras y Números para el recorrido**

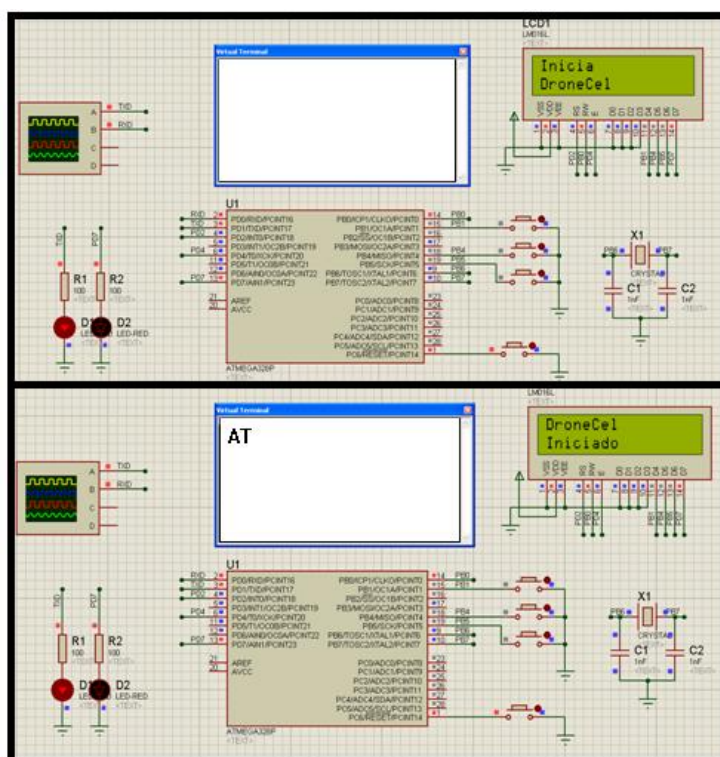
- No debe existir espacio entre letras y números. No importa que las letras sean mayúsculas o minúsculas 'A' equivale a 'a'.
- Si existe una letra errónea el robot inmediatamente terminará su recorrido.

CAPITULO 4

4. SIMULACIÓN Y RESULTADOS

4.1 Simulación de la Transmisión de Datos del Módulo Orangutan

Para verificar que el módulo Orangutan transmita los datos correctamente se utilizó un simulador para tener la certeza de que funcionaría correctamente cuando ya se lo implemente físicamente. Para la simulación se utilizó el programa Proteus en su versión 7.7.



**Figura 4-1 Simulación de Secuencia de Inicialización
Enviada por el Microcontrolador ATMEGA328P**

En la Simulación de Secuencia de Inicialización se puede observar como empieza el trabajo del microcontrolador, mostrando en la pantalla LCD el Mensaje “Inicia Drone Cell”, luego de que el Drone Cell se enciende, envía el Comando AT para verificar que esté funcionando, con lo que el módulo responderá con “0\r” que significa “OK” o correcto, caso contrario “4\r” que representa “Error”.

Una vez que el módulo confirma que se encuentra en funcionamiento correcto, el microcontrolador envía los parámetros para recibir mensajes correctamente, con los comandos mostrados en la Tabla 4-1 Comandos utilizados para recibir Mensajes de Texto.

Tabla 4-1 Comandos utilizados para recibir Mensajes de Texto

Comando	Descripción
AT+CMGF=1\r	Selecciona el formato de mensaje que se utiliza.
AT+CMGDA="DEL ALL"\r	Borra todos los mensajes que se encuentren en la SIM.
AT+CNMI=0,2\r	Informa acerca del ingreso de un nuevo SMS, inmediatamente.

La simulación de este proceso se observa en la Figura 4-2 Envío de Comandos para la Recepción de Mensajes de Texto, al responder positivamente todos los comandos enviados desde el microcontrolador presentará en la pantalla de la LCD “Espero el SMS”, con lo que espera que el módulo le envíe la trama que contiene el mensaje de texto recibido para procesarlo y poner en marcha al robot con la ruta especificada por el usuario.

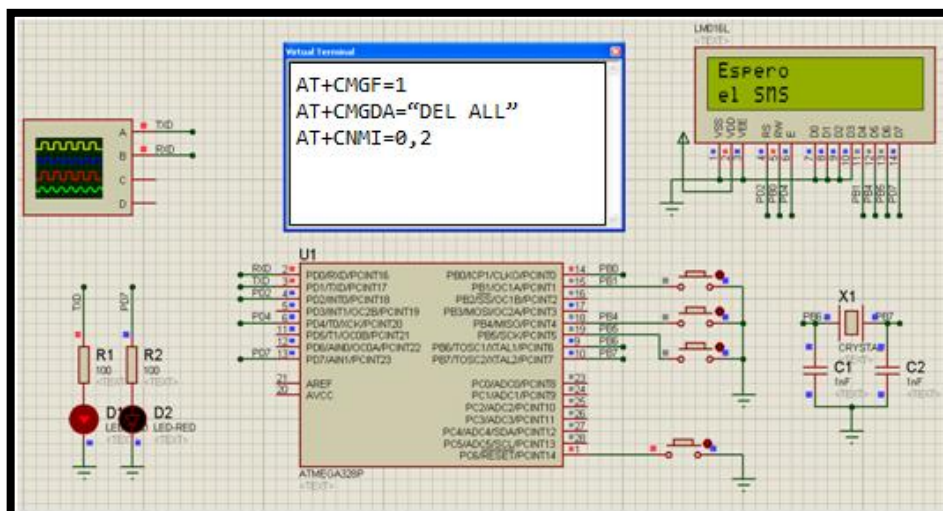


Figura 4-2 Envío de Comandos para la Recepción de Mensajes de Texto

4.2 Verificación de la Transmisión de Datos del Narobo Drone Cell

Para comprobar el funcionamiento del módulo Narobo Drone Cell, se procedió a conectar este con una computadora a través de un puerto USB de la misma, utilizando como intermediarios un cable especial que convierte USB a RS-232 y el integrado MAX 232 que convierte señales tipo RS-232 a TTL.

Una vez que se realizó la conexión física con la computadora y el módulo, el programa que se escogió para realizar la comunicación entre ambos fue el "Terminal V1.9b", el que permite que la conexión entre ambos se realice de una manera sencilla. Para lograr una comunicación exitosa, se deben poner los parámetros de velocidad de transmisión o Baud Rate a 115200, el tamaño del dato debe ser de 8 bits, sin bit de paridad y con un bit de parada.

Cuando se enciende el módulo Narobo Drone Cell, este envía el siguiente Mensaje “RDY\r +CFUN\r +CPN: Ready\r Call Ready\r” como se puede apreciar en la Figura 4-3 Encendido del Módulo Narobo Drone Cell.

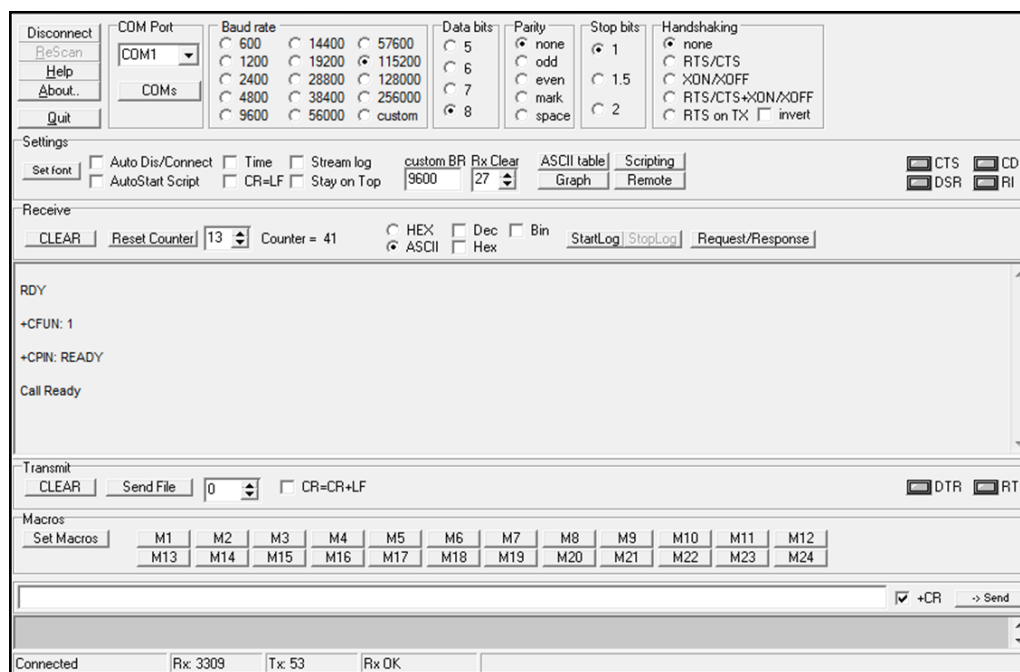


Figura 4-3 Encendido del Módulo Narobo Drone Cell

Una vez confirmado que el módulo está funcionando correctamente se procedió a enviar los comandos AT respectivos para recibir un SMS desde el módulo. En la Figura 4-4 Recepción de SMS desde el Módulo Narobo Drone Cell se observa el envío de los comandos y la respuesta que da el módulo. El mensaje recibido en este caso fue “Prueba recepción de mensaje”, se identifica también el número telefónico que envió el mensaje que en este caso es el “+59397438027”, número que utiliza el proyecto para enviar un mensaje de confirmación una vez que ha recorrido la ruta exitosamente el Robot

Oruga, los comandos utilizados son los ya descritos en la Tabla 4-1
Comandos utilizados para recibir Mensajes de Texto.

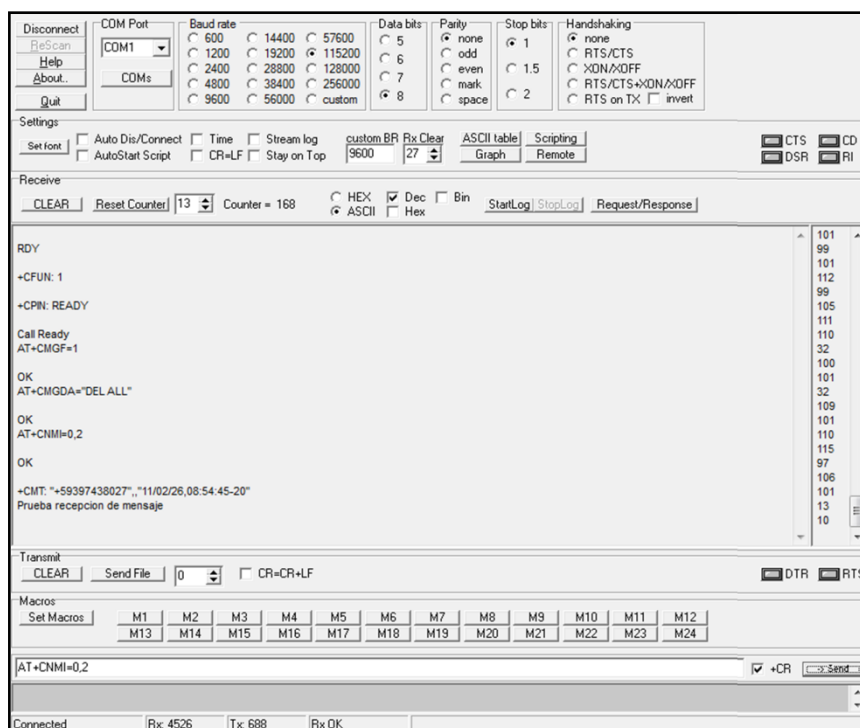


Figura 4-4 Recepci3n de SMS desde el M3dulo Narobo Drone Cell

El env3o de mensajes tambi3n fue ensayado como se aprecia en la Figura 4-5
Env3o de SMS desde el M3dulo Narobo Drone Cell, se puede apreciar que el
mensaje enviado es “Esto es una prueba de env3o de Mensaje” al n3mero
“097438027”.

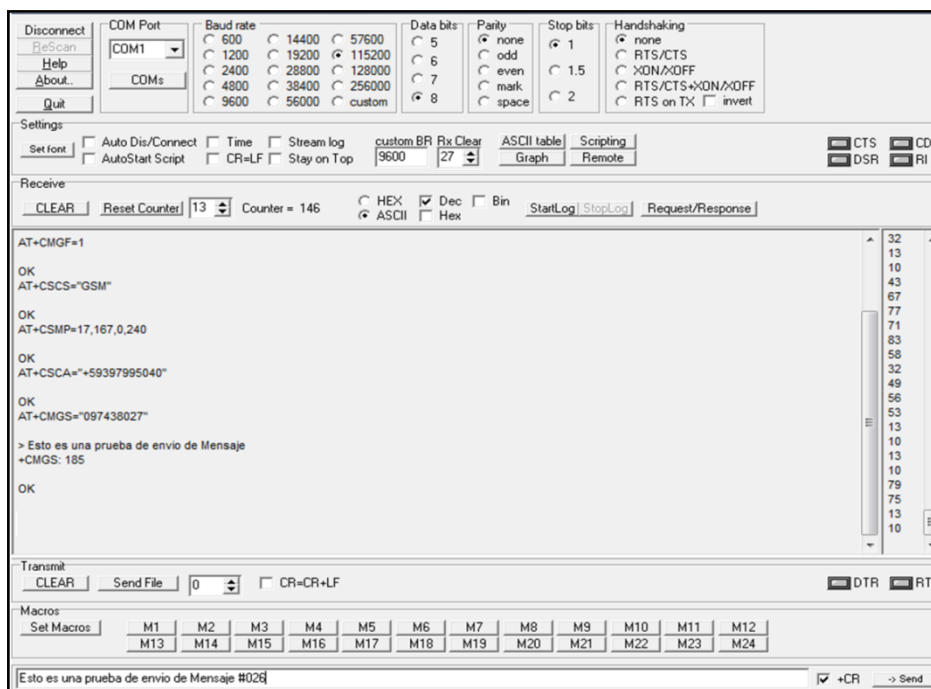


Figura 4-5 Envío de SMS desde el Módulo Narobo Drone Cell

En el envío de mensajes intervienen otros comandos a diferencia de la recepción, en la Tabla 4-2 Comandos AT que Intervienen en el envío de un SMS se describen cada uno de ellos.

Tabla 4-2 Comandos AT que Intervienen en el envío de un SMS

Comando	Descripción
AT+CMGF=1\r	Selecciona el formato de mensaje que se utiliza.
AT+CSCS="GSM"\r	Selecciona el set de caracteres de GSM por defecto.
AT+CSMP=17,167,0,240\r	Cambiar los parámetros para los SMS como número de caracteres y otros.
AT+CSCA="NUMERO"	Comando para ingresar el número del centro de mensajes que utilizará el Narobo Drone Cell.
AT+CMGS	Con este comando se ingresa el número telefónico al cual se enviaría el SMS seguido por el mensaje.

4.3 Resultados Experimentales con el Robot Oruga

Los resultados experimentales obtenidos con el robot Oruga fueron los esperados, se comportó de la misma forma que lo hizo en las simulaciones.

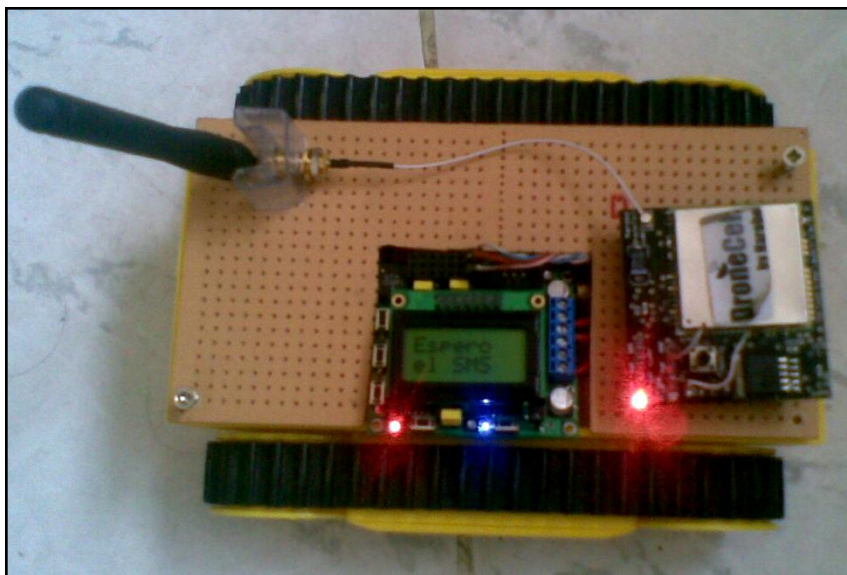


Figura 4-6 Robot Oruga Funcionando en Espera del SMS

En cuanto a la carga de energía que utiliza el Robot, las seis pilas recargables de Hidruro de Niquel fueron suficientes para mantener en funcionamiento tanto al Narobo Drone Cell, el controlador de Robots Orangutan y los motores que mueven al Robot completo.

Calibrando de una buena manera los tiempos de giro y el del recorrido de un metro, se obtuvieron resultados muy aceptables y próximos a los requeridos por el usuario.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

1.- Se concluye que el uso de librerías realizadas específicamente para cada microcontrolador y usadas de manera correcta, dan al diseñador un ahorro de tiempo permitiendo desarrollar aplicaciones de mayor complejidad en funcionamiento pero más comprensibles y sencillas para el diseñador y para quien quiera hacerle mejoras a los proyectos, tal como se hizo en este proyecto al utilizar la librería de funciones que proporciona la empresa Pololu para el controlador de robots Orangutan y la librería del módulo Narobo Drone Cell.

2.- Se concluye también que el conocimiento que se tenga sobre el microcontrolador a utilizar es el pilar con el que se sostiene el proyecto a realizar. Un conocimiento profundo del microcontrolador da las pautas para que el código sea el más eficiente, y permite que se puedan adaptar librerías de microcontroladores cercanos al utilizado, como se lo realizó en este proyecto al adaptar la librería de funciones que venía con el Narobo Drone Cell al microcontrolador con el que se trabajó este proyecto.

3.- El uso de módulos como el Narobo Drone Cell en unión con microcontroladores permiten realizar proyectos, los cuales no poseen limitaciones de distancia, a diferencia de los típicos módulos de radio frecuencia, por lo que el uso de estos podrían ser una perfecta solución para

otro tipo de aplicaciones como son domótica y seguridad en los cuales su uso sería aún más apropiado.

4.- Utilizar módulos para controlar Robots como el Orangutan SV-328, estos proporcionan una gran flexibilidad para realizar este tipo de proyectos, ya que a más de proporcionarnos una completa gama de sensores, nos facilita también todas las librerías con funciones completas para que el desarrollo de un robot sea algo muy sencillo y rápido.

RECOMENDACIONES

1.- Recomendamos investigar acerca de las librerías de funciones disponibles para cada microcontrolador con el que se trabaje. Al trabajar con librerías específicas se obtiene un gran ahorro de tiempo y esfuerzo.

2.- Recomendamos también descargar ejemplos sencillos para probar de manera correcta el software del Orangutan SV-328 y Narobo Drone Cell. Averiguar con qué frecuencia de reloj trabajan ambos, en qué modo y cuál es su voltaje de alimentación.

3.- Es importante verificar el funcionamiento del Narobo Drone Cell, esto se puede realizar comunicando el puerto serial que posee el dispositivo, un circuito con el integrado MAX-232 y una computadora con puerto serial, la computadora debe tener algún tipo de Terminal que me permita visualizar la información del Narobo Drone Cell.

4.-Para las personas que requieran utilizar un módulo similar al Narobo Drone Cell es recomendable que conozcan de antemano los comandos AT, los cuales por ser un estándar no solo sirven para enviar mensajes de texto, sino que también sirven para controlar cualquier tipo de comunicación, llamadas, entre otros. Independientemente del medio o tecnología con la que se trabaje, por ejemplo se utilizan estos comandos para trabajar con bluetooth.

5.- Se recomienda tener especial cuidado con el chasis RP5 y con el controlador de robots Orangutan SV-328, debido a que si el chasis se quedara estancado en algún sitio con los motores funcionando se podría producir una corriente superior a los tres amperios que soporta el controlador.

ANEXOS

Anexo A

Hoja de Datos

Microcontrolador

ATMEGA328P

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory
 - 256/512/1K Bytes EEPROM
 - 512/1K/1K/2K Bytes Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V
- Temperature Range:
 - -40°C to 85°C
- Speed Grade:
 - 0 - 4 MHz@1.8 - 5.5V, 0 - 10 MHz@2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



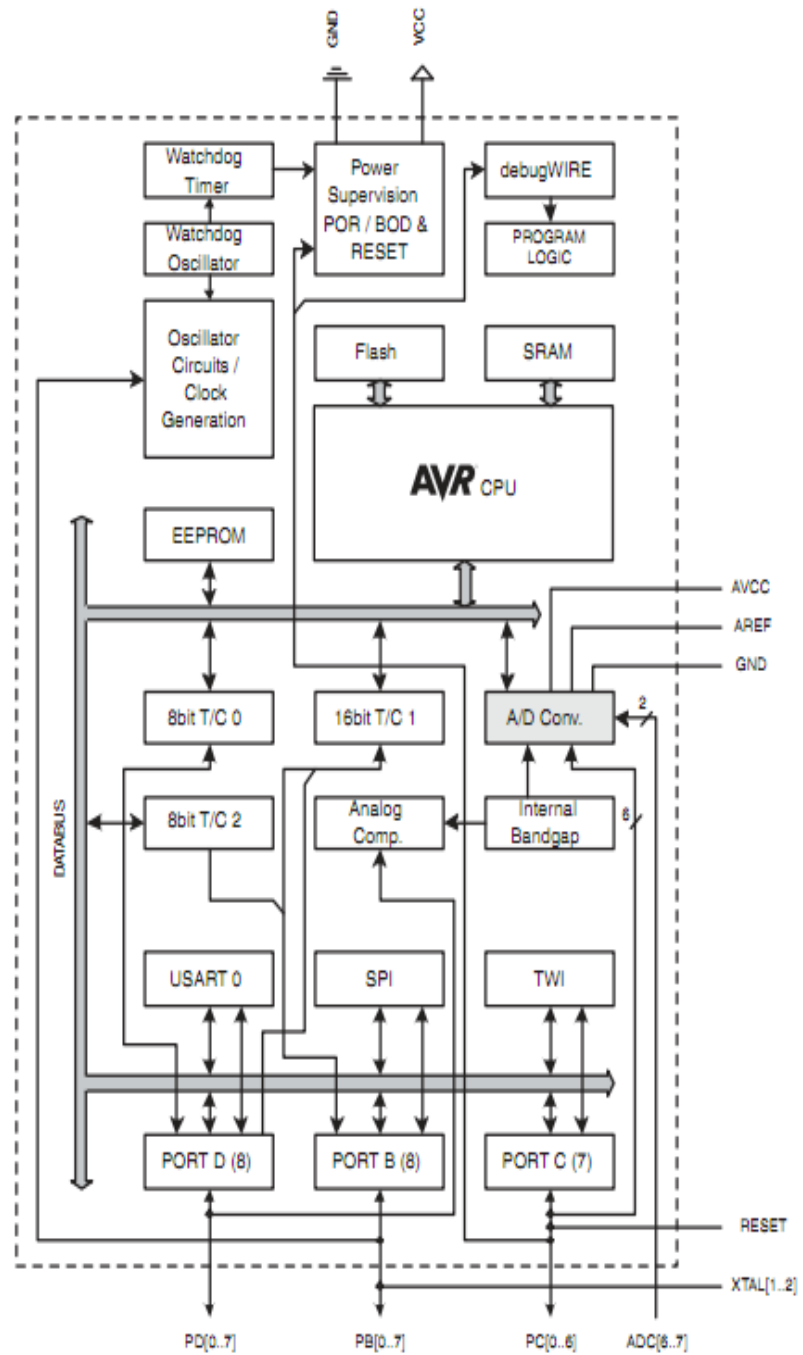
8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48A
ATmega48PA
ATmega88A
ATmega88PA
ATmega168A
ATmega168PA
ATmega328
ATmega328P

Summary

Block Diagram

Figure 2-1. Block Diagram



5. Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd,K	Add Immediate to Word	$RdH:RdL \leftarrow RdH:RdL + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd,K	Subtract Immediate from Word	$RdH:RdL \leftarrow RdH:RdL - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \cdot (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
MUL	Rd, Rr	Multiply Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multiply Signed	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiply Signed with Unsigned	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Fractional Multiply Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Fractional Multiply Signed	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Fractional Multiply Signed with Unsigned	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2

Anexo B

Hoja de Datos

SIMCom GSM GPRS Module

2 Product concept

Designed for global market, SIM300DZ is tri-band GSM/GPRS engine that works on frequencies, GSM 900 MHz, DCS 1800 MHz and PCS1900 MHz. SIM340DZ is quad-band GSM/GPRS engine that works on frequencies, GSM/GPRS 900/1800 MHz and 850/1900MHz. SIM3XXDZ series features GPRS multi-slot class 10 /Class 8^① capability and supports the GPRS coding schemes CS-1, CS-2, CS-3 and CS-4.

① SIM3XXDZ also provides GPRS multi-slot class 8 and the default is class 10.

With a tiny configuration of 33mm x 33mm x 3 mm, SIM3XXDZ can meet almost all the space requirement in your application, such as M2M, smart phone, PDA phone, Car Phone , Wireless PSTN , and other mobile device.

The hardware package of 48 pins

- 9 GND PINS and 2 VBAT pins
- 1 Pin is programmable as General Purpose I/O .This gives you the flexibility to develop customized applications.
- Serial port and Debug port can help you easily develop your applications. But they can not work at the same time.
- Two audio channels include two microphone inputs and two speakers' outputs. This can be easily configured by AT command.

With the charge circuit integrated inside the SIM3XXDZ, it is very suitable for the battery power application.

The SIM3XXDZ provides RF antenna interface. And customer's antenna should be located in the customer's mainboard and connect to module's antenna pad through micro strip line or other type RF traces whose impedance must be controlled in 50Ω.

The SIM3XXDZ is designed with power saving technique, so that the current consumption is as low as 2.5mA in SLEEP mode (BS-PA-MFRMS=5).

The SIM3XXDZ is integrated with the TCP/IP protocol, Extended TCP/IP AT commands are developed for customers to use the TCP/IP protocol easily, which is useful for those data transfer applications.

2.1 SIM3XXDZ key features at a glance

Table 3: SIM3XXDZ key features

Feature	Implementation
Power supply	Single supply voltage 3.4V – 4.5V
Power saving	Typical power consumption in SLEEP mode to 2.5mA (BS-PA-MFRMS=5)
Charging	Supports charging control for Li-Ion battery
Frequency Bands	● SIM300DZ tri-band: GSM 900, DCS 1800, PCS 1900. The SIM300DZ can search the 3 frequency bands automatically. The frequency bands also can be set by AT command. ● SIM340DZ quad-band: GSM/GPRS 900/1800 MHz and 850/1900MHz. The SIM340DZ can search the 4 frequency bands automatically. The frequency bands also can be set by AT command. ● Compliant to GSM Phase 2/2+
GSM class	Small MS
Transmitting power	● Class 4 (2W) at EGSM900/GSM850 ● Class 1 (1W) at DCS1800/PCS1900
GPRS connectivity	● GPRS multi-slot class 8 (optional) ● GPRS multi-slot class 10 (default) ● GPRS mobile station class B
Temperature range	● Normal operation: -30°C to +80°C ● Restricted operation: -40°C to -30°C and +80°C to +85°C⁽¹⁾ ● Storage temperature: -45°C to +90°C
DATA GPRS:	● GPRS data downlink transfer: max. 85.6 kbps ● GPRS data uplink transfer: max. 42.8 kbps ● Coding scheme: CS-1, CS-2, CS-3 and CS-4 ● SIM3XXDZ supports the protocols PAP (Password Authentication Protocol) usually used for PPP connections. ● The SIM3XXDZ integrates the TCP/IP protocol. ● Support Packet Switched Broadcast Control Channel (PBCCH)
CSD:	● CSD transmission rates: 2.4, 4.8, 9.6, 14.4 kbps, non-transparent ● Unstructured Supplementary Services Data (USSD) support
SMS	● MT, MO, CB, Text and PDU mode ● SMS storage: SIM card
FAX	Group 3 Class 1
SIM interface	Support SIM card: 1.8V ,3V
External antenna	Connected via 50 Ohm antenna connector or antenna pad

Audio features	Speech codec modes: ● Half Rate (ETS 06.20) ● Full Rate (ETS 06.10) ● Enhanced Full Rate (ETS 06.50 / 06.60 / 06.80) ● Adaptive multi rate (AMR) ● Echo Cancellation
Serial interface and Debug interface	● Serial Port: Seven lines on Serial Port Interface ● Serial Port can be used for CSD FAX, GPRS service and sending AT command of controlling module. ● Autobauding supports baud rates from 1200 bps to 115200bps. ● Debug port : provide two lines on Serial Port Interface /TXD and /RXD ● Debug port is only used for debugging
Phonebook management	Support phonebook types: SM, FD, LD, MC, RC, ON, ME,BN,VM,LA,DC,SD
SIM Application Toolkit	Support SAT class 3, GSM 11.14 Release 99
Real time clock	Implemented
Timer function	Programmable via AT command
Physical characteristics	Size: $33\pm0.15 \times 33\pm0.15 \times 3\pm0.3$ mm Weight: 8g
Firmware upgrade	Firmware upgrade over serial interface

Note: The Adaptive multi rate (AMR) of Speech codec mode and Echo Cancellation function only supported on the Release 16 firmware.

(1) The SIM3XXDZ does works, but deviations from the GSM specification may error, For example, both the frequency error and the phase error will be large.

2.2 SIM3XXDZ functional diagram

The following figure shows a functional diagram of the SIM3XXDZ and illustrates the mainly functional part:

- The GSM baseband engine
- Flash and SRAM

SIM3XXDZ_HD_V2.07

14

30.03.2009

SIM3XXDZ Hardware Design



- The GSM radio frequency part
- The antenna interface
- The external interface

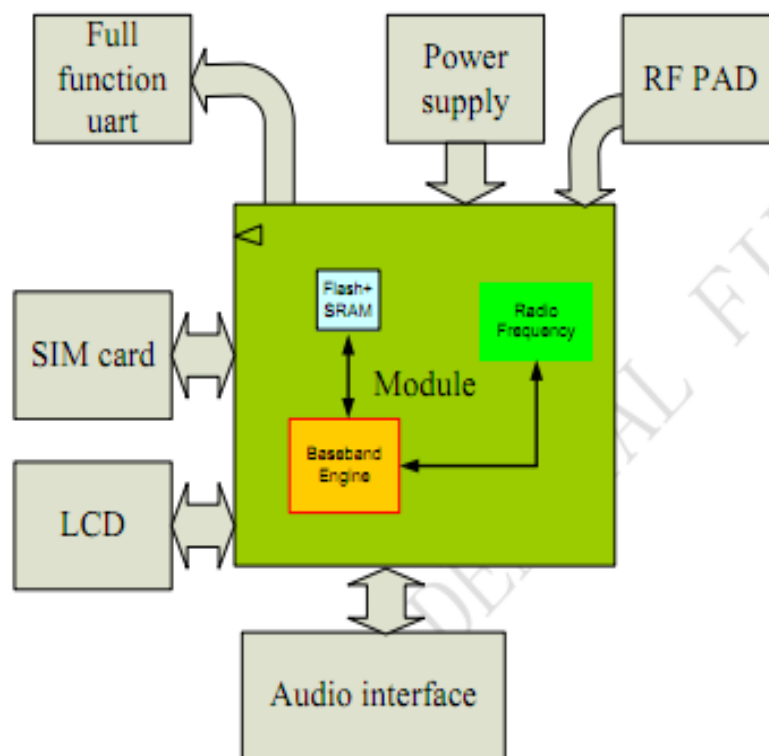


Figure 2: SIM3XXDZ functional diagram

3 Application interface

All hardware interfaces are described in detail in following chapters:

- Power supply and charging control (see Chapters 3.3 and 3.5)
- Provide serial interface and Debug interface (see chapter 3.9)
- Two analog audio interfaces (see chapter 3.10)
- SIM interface (see chapter 3.11)

3.1 SIM3XXDZ Pin description

Table 5: Pin description

Power Supply				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
VBAT		2 VBAT pins are dedicated to connect the supply voltage. The power supply of SIM3XXDZ has to be a single voltage source of VBAT= 3.4V..4.5V. It must be able to provide sufficient current in a transmit burst which typically rises to 2A.mostly, these 2 pins are voltage input, however ,when use the charge circuit to charge the battery ,these pins become the current output, select one of these pins as the charge current output Pin	Vmax= 4.5V Vmin=3.4V Vnorm=4.0V	
VRTC	I/O	Current input for RTC when the battery is not supplied for the system. Current output for backup	Vmax=2.0V Vmin=1.2V Vnorm=1.8V Iout(max)= 20uA	Do not keep Pin open, it should be connected to a battery or a

		battery when the main battery is present and the backup battery is in low voltage state.	$I_{in}=5\text{ uA}$	capacitor.
VCHG	I	Voltage input for the charge circuit, as the signal for detecting the charger connecting	$V_{max}=5.25V$ $V_{min}=1.1$ * VBAT $V_{norm}=5.1V$ $I_{min}=650mA$	If unused keep Pin open
GND		Digital ground		

Power on or power off

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
PWRKEY	I	Voltage input for power on key. Press the key , the PWRKEY get a low level voltage for user to power on or power off the system, the user should keep pressing the key for a moment when power on or power off the system. Because the system need margin time assert the software.	$V_{ILmax}=0.2*V_{BAT}$ $V_{IHmin}=0.6*V_{BAT}$ $V_{Imax}=V_{BAT}$	Pull up to VBAT inside.

Audio interfaces

PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
MIC1P MIC1N	I	Positive and negative voiceband input	Audio DC Characteristics refer to chapter 3.10	If unused keep Pin open
MIC2P MIC2N	I	Auxiliary positive and negative voiceband input		If unused keep Pin open
SPK1P SPK1N	O	Positive and negative voiceband output		If unused keep Pin open
SPK2P SPK2N	O	Auxiliary positive and negative voiceband output		If unused keep Pin open
AGND		Analog ground		Separate ground connection for external audio

				circuits. If unused keep Pin open
GERNERAL PURPOSE input/output				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
STATUS	O	Indicate work status	VILmin=0V VILmax=0.3*2.93V	If unused keep pins open
GPO1	O	Normal Output Port	VIHmin=0.7*2.93V	If unused keep pins open
DISP_DATA	I/O /4 mA	Display interface	VIHmax=2.93V+0.3	If unused keep pins open
DISP_CLK	O/4 mA		VOLmin=GND	
DISP_CS	O/4 mA		VOLmax=0.2V	
DISP_D/C	O/4 mA		VOHmin=2.93V-0.2	
DISP_RST	O/4 mA		VOHmax=2.93V	
KBR0	I/4 mA			Pull up inside, if unused keep pins open
Serial Interface				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERISTICS	COMMENT
RXD	I/8 mA	Receive data	VILmin=0V VILmax=0.3*2.93V	If use only TXD, RXD GND three pins to communicate, RTS Pin connect to GND directly. DTR Pin is pulled up inside. If unused keep pins open
DTR	I/8 mA	Data terminal Ready	VIHmin=0.7*2.93V	
TXD	O/8 mA	Transmit data	VIHmax=2.93V+0.3	
RTS	I/8 mA	Request to send	VOLmin=GND	
CTS	O/8 mA	Clear to send	VOLmax=0.2V	
RI	O/8 mA	Ring indicator	VOHmin=2.93V-0.2 VOHmax=2.93V	

DCD	O/4 mA			
Debug interface				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERI STICS	COMMENT
DBG_TXD	O/4 mA	Serial interface for debugging only		If unused keep pins open
DBG_RXD	I/4 mA			
SIM interface				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERI STICS	COMMENT
SIM_VDD	O	Voltage supply for SIM card	The voltage can be select by software either 1.8v or 3V	
SIM_DATA	I/O /4 mA	SIM data output	VILmin=0V VILmax=0.3*SI M_VDD	All signals of SIM interface are protected against ESD with a TVS diode array. Maximum cable length 200mm from the module connector to SIM card holder.
SIM_CLK	O/4 mA	SIM clock	VIHmin=0.7*SI M_VDD	
SIM_RST	O/4 mA	SIM reset	VIHmax= SIM_VDD+0.3 VOLmin=GND VOLmax=0.2V VOHmin= SIM_VDD-0.2 VOHmax= SIM_VDD	
ADC				
PIN NAME	I/O	DESCRIPTION	DC CHARACTERI STICS	COMMENT
ADC0	I	General purpose analog to digital converter.	Input voltage value scope 0V to 2.4V	If unused keep Pin open
TEMP_BAT	I	For measure the battery temperature		If unused keep Pin open

3.3 Power supply

The power supply of SIM3XXDZ is from a single voltage source of $V_{BAT} = 3.4V \dots 4.5V$. In some case, the ripple in a transmitting burst may cause voltage drops when current consumption rise to typical peaks of 2A. So the power supply must be able to provide sufficient current up to 2A..

For the VBAT input, a local bypass capacitor is recommended. A capacitor (about 100 μ F, low ESR) is recommended. Multi-layer ceramic chip (MLCC) capacitors can provide the best combination of low ESR and small size but may not be cost effective. A lower cost choice may be a 100 μ F tantalum capacitor (low ESR) with a small (0.1 μ F to 1 μ F) ceramic in parallel, which is illustrated as figure1. The capacitors should be put as closer as possible to the SIM3XXDZ VBAT pins.

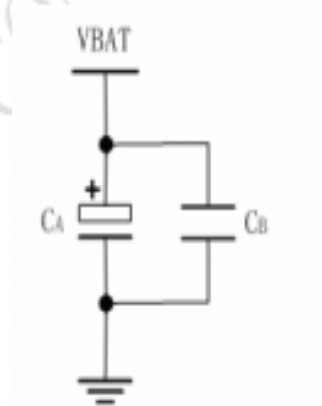


Figure 4: VBAT input

The circuit design of the power supply depends strongly from the power source where this power is drained. The following figure is the reference design of +5V input source power supply. The designed output for the power supply is 4V, thus a linear regulator can be used. If there's a big difference between the input source and the desired output (VBAT), a switching converter power supply will be preferable because of its better efficiency especially with the 2A peak current in burst mode of the module.

The single 3.6V Li-Ion cell battery type can be connected to the power supply of the SIM3XXDZ VBAT directly. But the Ni_Cd or Ni_MH battery types must be used carefully, since their maximum voltage can rise over the absolute maximum voltage for the module and damage it.

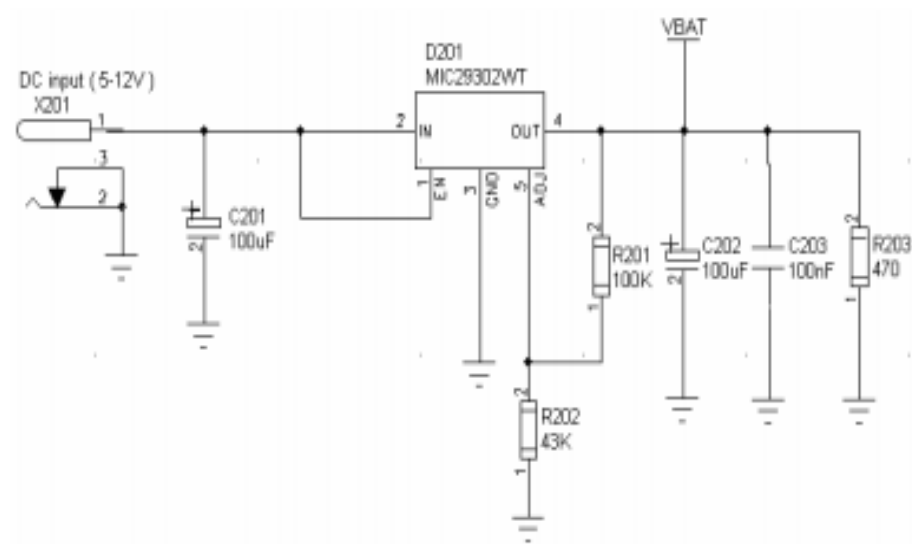


Figure 5: Reference circuit of the source power supply input

The following figure is the VBAT voltage ripple wave at the maximum power transmit phase, the test condition is VBAT=4.0V, VBAT maximum output current =2A, $C_A=100\mu\text{F}$ tantalum capacitor (ESR=0.7 Ω) and $C_B=1\mu\text{F}$.

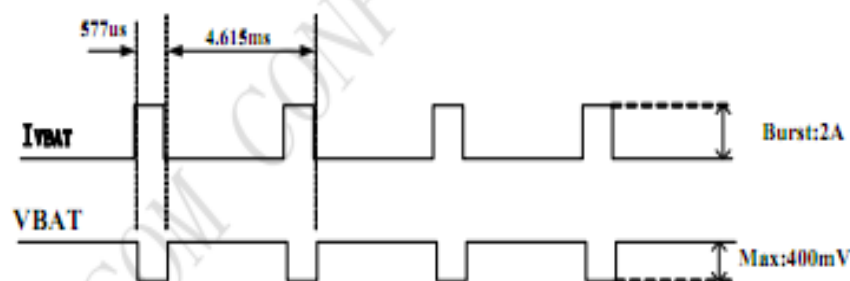


Figure 6: VBAT voltage drop during transmit burst

3.5 Charging interface

SIM3XXDZ has integrated a charging circuit inside the module for Li-Ion batteries charging control, which make it very convenient for applications to manage their battery charging. A common connection is shown in the following figure:

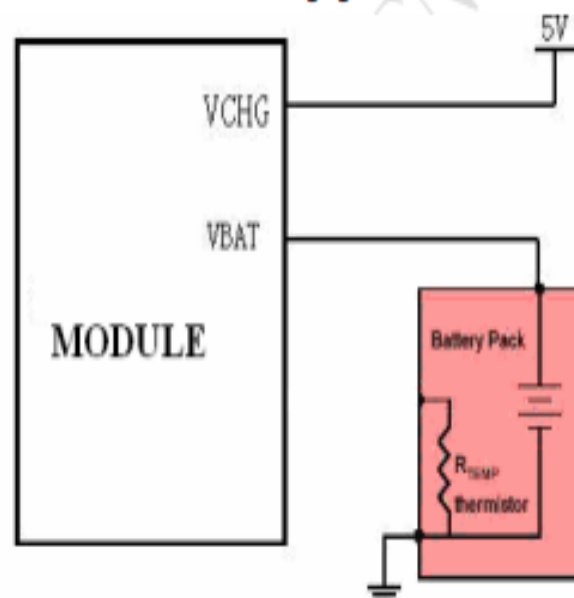


Figure 12: Battery charger and pack

The function of detecting the temperature of battery should be supported by the software in the module. It's a customization function. The R_{TEMP} is a NTC thermistor. We recommend to use NCP15XH103F03RC from MURATA. The impedance of the NTC thermistor is 10Kohm in 25°C. Please refer to the fore figure for the reference circuit.

Anexo C

Código Fuente

del Archivo Principal

```

#include <avr/io.h>
#include "timer640.h"
#include "uart4.h"
#include "rprintf.h"
#include "dronecell.h"
#include "buffer.h"
#include <pololu/orangutan.h>

void FlushReceiveBuffer(void);
unsigned char USART_Receive( void );

int main(void)
{

    int i,j, Band;
    long int temp, temp1, temp2, TimeIzq,TimeDer, Time1m;
    char button, button2, Num[]=" 080240236 ";

    Num[0]=34;
    Num[1]=48;
    Num[10]=34;
    TimeIzq=1000;

    TimeDer=1000;

    Time1m=67;

    DDRB = 0b00001001;
    DDRC = 0b00000000;
    DDRD= 0b01101110;

    do{

        clear();
        lcd_init_printf();

```

```

print("Press B");
lcd_goto_xy(0, 1);
print("to Start");

button = wait_for_button_press(ALL_BUTTONS);

temp2=0;

if(button & BUTTON_A)
{
    do{

        clear();
        print("Calib De");
        lcd_goto_xy(0, 1);
        print("A+ C- Bs");

        while(button_is_pressed(BUTTON_A))
        {
            }

        do{
            button2= wait_for_button_press(ALL_BUTTONS);
            delay_ms_narobo(200);

            if(button2 & BUTTON_A)
            {TimeDer=TimeDer+10;
              temp2=0;}
            else if(button2 & BUTTON_C)
            {TimeDer=TimeDer-10;
              temp2=0;}
            else if(button2 & BUTTON_B)
            {temp=0;
              temp2=1;}

            clear();

```

```

        print("Tiempo D");
        lcd_goto_xy(0, 1);
        print_unsigned_long(TimeDer);

    }while (temp2==0);

    set_motors(200, 200);
    delay_ms_narobo(1000);
    set_motors(200, -100);
    delay_ms_narobo(TimeDer);
    set_motors(0, 0);

    while(button_is_pressed(BUTTON_B))
    {
        button2= wait_for_button_press(ALL_BUTTONS);
        if(button2 & BUTTON_B)
        {
            temp=0;
            temp2=1;
        } else { temp2=0;}
    }

}while(temp2==0);

temp2=0;

do{

    clear();
    lcd_init_printf();
    print("Calib Iz");
    lcd_goto_xy(0, 1);
    print("A+ C- Bs");

    while(button_is_pressed(BUTTON_B))

```

```

}

do{
button2= wait_for_button_press (ALL_BUTTONS);
delay_ms_narobo(200);

    if(button2 & BUTTON_A)
    {TimeIzq=TimeIzq+10;
temp2=0;}
    else if(button2 & BUTTON_C)
    {TimeIzq=TimeIzq-10;
temp2=0;}
    else if(button2 & BUTTON_B)
    { temp=0;
temp2=1;}
        clear();
        print("Tiempo I");
        lcd_goto_xy(0, 1);
        print_unsigned_long(TimeIzq);

    }while(temp2==0) ;

    set_motors(200, 200);
    delay_ms_narobo(1000);
    set_motors(-100, 200);
    delay_ms_narobo(TimeIzq);
    set_motors(0, 0);

while(button_is_pressed(BUTTON_B))
{
button2= wait_for_button_press (ALL_BUTTONS);
if(button2 & BUTTON_B)
    { temp=0;
temp2=1;
    } else { temp2=0;}
}

```

```

    }while(temp2==0);

do{

    clear();
    lcd_init_printf();
    print("Calib 1m");
    lcd_goto_xy(0, 1);
    print("A+ C- Bs");

    while(button_is_pressed(BUTTON_B))
    {

do{
button2= wait_for_button_press(ALL_BUTTONS);
delay_ms_narobo(200);

        if(button2 & BUTTON_A)
        {
            Time1m=Time1m+1;
            temp2=0;
        }
        else if(button2 & BUTTON_C)
        {
            Time1m=Time1m-1;
            temp2=0;
        }
        else if(button2 & BUTTON_B)
        {
            temp=0;
            temp2=1;
        }

        clear();
        print("Tiemp 1m");
    }
}

```

```

        lcd_goto_xy(0, 1);
        print_unsigned_long(Timelm);

    }while(temp2==0) ;

    set_motors(200, 200);

    for(j=0;j<100;j++){
        delay_ms_narobo(Timelm);}

    set_motors(0, 0);

    while(button_is_pressed(BUTTON_B))
    {
        button2= wait_for_button_press(ALL_BUTTONS);
        if(button2 & BUTTON_B)
        {
            temp=0;
            temp2=1;
        } else { temp2=0;}

    }while(temp2==0);

    temp=0;
    while(button_is_pressed(BUTTON_B))
    {

    }

    else if(button & BUTTON_B)
    {temp=1;}
    else if(button & BUTTON_C)
    {temp=0;} else {temp=0;}

    }while(temp==0);

```



```

uart0Init();

uartSetBaudRate(0, 115200);

rprintfInit(uart0SendByte);

init_timer0(TIMER_CLK_64);

clear();
lcd_init_printf();
print("Inicia");
lcd_goto_xy(0, 1);
print("DroneCel");

DroneCell_Init( uart0SendByte , uart0GetByte , FlushReceiveBuffer
);

delay_ms_narobo(2000);
delay_ms_narobo(2000);

clear();
lcd_init_printf();
print("DroneCel");
lcd_goto_xy(0, 1);
print("Iniciado");

int ErrorCode=0;
ErrorCode = DroneCell_PowerOn_Pololu();
    if ( ErrorCode == OK) {

        clear();
        lcd_init_printf();
        print("DroneCel");

```

```

        lcd_goto_xy(0, 1);
        print("Listo");

    }
    else {

        clear();
        lcd_init_printf();
        print("DroneCel");
        lcd_goto_xy(0, 1);
        print("Error 1");

    }

    rprintf("AT+CMGDA=");
    uartSendByte(0,34);
    rprintf("DEL ALL");
    uartSendByte(0,34);
    rprintf("\n");
    delay_ms_narobo(2000);

    ErrorCode = DroneCell_Receive_TextInit();
    if ( ErrorCode != OK) {

        clear();
        lcd_init_printf();
        print("DroneCel");
        lcd_goto_xy(0, 1);
        print("Error 2");

    }
    else {

        clear();
        lcd_init_printf();
        print("Espero");
        lcd_goto_xy(0, 1);
        print("el SMS");
    }

```

```

    }

while(1){

    delay_ms_narobo(2000);
    uartFlushReceiveBuffer(0);
    delay_ms_narobo(3000);

    while( (uart0GetByte())!=-1){}

        clear();
        print("SMS");
        lcd_goto_xy(0, 1);
        print("Recibido");
        delay_ms_narobo(1000);
        clear();
        print("Num. Telf");
        lcd_goto_xy(0, 1);
        print("0");

    j=2;
    for(i=0;i<47;i++)
    {

        temp=uart0GetByte();

        if((i>=12)&&(i<20))
        {
            Num[j]=temp;
            print_character(temp);
            j++;
        }

        delay_ms_narobo(100);
    }

    delay_ms_narobo(2000);

    Band=0;

```

```

temp='A';

while((temp>=48&&temp<=57)
|| (temp>=48&&temp<=57)||temp=='A' ||temp=='a' ||temp=='R' ||temp=='r' ||temp
=='D' ||temp=='d' ||temp=='I' ||temp=='i' )
{

temp=uart0GetByte();
uartSendByte(0,temp);

if(temp=='A' ||temp=='a')

{
clear();
print("Adelante");
temp=uart0GetByte();

if((temp>=48&&temp<=57) || (temp>=48&&temp<=57))
{

temp1=100*(temp-48);
temp=uart0GetByte();

if((temp>=48&&temp<=57)
|| (temp>=48&&temp<=57))
{

temp2=10*(temp-48)+temp1;
temp=uart0GetByte();

if((temp>=48&&temp<=57) || (temp>=48&&temp<=57))
{
temp1=(temp2)+(temp-48);
set_motors(200, 200);

for(j=0;j<temp1;j++){
delay_ms_narobo(Timelm);}}

```

```

        }
    }

    Band=0;
}

if(temp=='R' || temp=='r')

{
    clear();
    print("Atras");
    temp=uart0GetByte();

    if((temp>=48&&temp<=57) || (temp>=48&&temp<=57))
    {
        temp1=100*(temp-48);
        temp=uart0GetByte();

        if((temp>=48&&temp<=57)
|| (temp>=48&&temp<=57))
        {
            temp2=10*(temp-48)+temp1;
            temp=uart0GetByte();

            if((temp>=48&&temp<=57)
|| (temp>=48&&temp<=57))
            {
                temp1=(temp2)+(temp-48);
                set_motors(-200, -200);

                for(j=0;j<temp1;j++){

                    delay_ms_narobo(Time1m);}}
            }
        }
        Band=1;
    }

    if(temp=='D' || temp=='d')

```

```

        {
            clear();
            print("Giro Der");
            if(Band==0){
                set_motors(200, -100);
                delay_ms_narobo(TimeDer);}
            else{set_motors(-200,100);
                delay_ms_narobo(TimeDer);
            }
        }

        if(temp=='I' || temp=='i')

        {
            clear();
            print("Giro Izq");
            if(Band==0){
                set_motors(-100, 200);
                delay_ms_narobo(TimeIzq);}
            else{set_motors(100,-200);
                delay_ms_narobo(TimeIzq);}
        }
    }
    temp=uart0GetByte();
    set_motors(0, 0);

    ErrorCode = DroneCell_ConnectCalling();

    if ( ErrorCode == OK) {
        clear();
        print("Envio SMS");
        delay_ms_narobo(300);

    }
    else {

```

```

        clear();
        print("Error1");
    }

    ErrorCode = DroneCell_Text (Num,"DroneCell");
    if ( ErrorCode == OK) {

        clear();
        print("Envio SMS");
        delay_ms_narobo(300);
    }
    else {
        clear();
        print("Error2");
        delay_ms_narobo(300);
    }

    clear();
    lcd_init_printf();
    print("Espero");
    lcd_goto_xy(0, 1);
    print("Nue. SMS");
}
}

void FlushReceiveBuffer(void)
{
    uartFlushReceiveBuffer(0);
}

unsigned char USART_Receive( void )
{
    while ( !(UCSR0A & (1<<RXC0)) );
    return UDR0;
}

```

REFERENCIAS BIBLIOGRAFICAS

- [1]. Wikipedia: Mando a distancia historia;
http://es.wikipedia.org/wiki/Mando_a_distancia
Fecha de consulta: 20 /Enero/2011.

- [2]. Alarmagsm: Comandos AT;
<http://code.google.com/p/alarmagsm/downloads/list>
Fecha de consulta: 25 /Enero/2011.

- [3]. Narobo: Getting started guide for the DroneCell;
<http://narobo.com/products/DroneCell/DroneCell.html>
Fecha de consulta: 25 /Enero/2011.

- [4]. Pololu: Orangutan SV-168/SV-328 schematic diagram;
<http://www.pololu.com/catalog/product/1227/resources>
Fecha de consulta: 30 /Enero/2011.

- [5]. SIM Com: AT Commands Set;
<http://www.pololu.com/catalog/product/1690/resources>
Fecha de consulta: 30 /Enero/2011.

- [6]. SIM Com: SMS Application Notes;
<http://www.pololu.com/catalog/product/1690/resources>
Fecha de consulta: 30 /Enero/2011.
- [7]. Pololu: Pololu Orangutan SV-xx8 and LV-xx8 User's Guide;
<http://www.pololu.com/catalog/product/1227/resources>
Fecha de consulta: 30 /Enero/2011.
- [8]. Pololu: Pololu AVR C/C++ Library User's Guide;
<http://www.pololu.com/resources/software>
Fecha de consulta: 26 /Enero/2011.
- [9]. Narobo: DroneCell AVR library;
<http://www.pololu.com/catalog/product/1690/resources>
Fecha de consulta: 29 /Enero/2011.
- [10]. Pololu: RP5 Tracked Chassis Yellow;
<http://www.pololu.com/catalog/product/1059>
Fecha de consulta: 29 /Enero/2011.
- [11]. Wikipedia: Qué significa SMS;
<http://en.wikipedia.org/wiki/SMS>
Fecha de consulta: 15 /Enero/2011.

[12]. Atmel: AVR Tools User Guide;

AVR Studio help;

Fecha de consulta: 15 /Enero/2011.

[13]. GNU: Avr libc user manual;

<http://www.nongnu.org/avr-libc/user-manual/index.html>

Fecha de consulta: 15 /Enero/2011.

[14]. Atmel: Data sheet Microcontroller ATMega328P;

http://www.atmel.com/dyn/products/product_docs.asp?category_id=163&family_id=607&subfamily_id=760&part_id=4198

Fecha de consulta: 20 /Enero/2011.