



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

**ESCUELA DE INGENIERÍA EN ELECTRÓNICA,
TELECOMUNICACIONES Y REDES**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE
ADQUISICIÓN DE SEÑALES BIOMÉTRICAS MEDIANTE
MENSAJES SMS”**

TESIS DE GRADO

Previa la obtención del título de

INGENIERO EN ELECTRÓNICA Y COMPUTACIÓN

Autor:

WASHINGTON PATRICIO HIDALGO GUACHO

RIOBAMBA – ECUADOR

2015

Agradezco a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO, y a todos los docentes que me brindaron sus conocimientos y como personas su amistad sincera.

A mi tutor el Ing. Franklín Moreno por brindarme su apoyo, comprensión y conocimientos.

A todos mis profes no sólo de la carrera sino de toda la vida, mil gracias porque de alguna manera forman parte de lo que ahora soy.

Son muchas las personas especiales a quienes me gustaría agradecer su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos y en el corazón. Sin importar en dónde estén o si alguna vez llegan a leer estas palabras quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.

A las personas más importantes de mi vida:

Con todo mi cariño y mi amor para las personas que hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, a ustedes por siempre mi corazón y mi agradecimiento Aunque hoy no los tenga a mi lado.

Papá y mamá

Firma de Responsabilidad

| NOMBRE | FIRMA | FECHA |
|---|--------------|--------------|
| Ing. Nicolay Samaniego DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA | ----- | ----- |
| Ing. Franklin Moreno DIRECTOR DE LA ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES | ----- | ----- |
| Ing. Franklin Moreno DIRECTOR DE TESIS | ----- | ----- |
| Ing. José Guerra MIEMBRO DEL TRIBUNAL | ----- | ----- |
| DIRECTOR CENTRO DOCUMENTACIÓN | ----- | ----- |
| NOTA DE LA TESIS | ----- | |

Firma de Autoría

“Yo, WASHINGTON PATRICIO HIDALGO GUACHO, soy responsable de las ideas, doctrinas y resultados expuestos en esta tesis, y el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

Patricio Hidalgo.
AUTOR

Índice de Abreviaturas

| | |
|----------------|---|
| ALU | Unidad Aritmético Lógica. |
| A/D. | Analógico/Digital. |
| ASK | (Amplitude Shift Keying)Modulación por desplazamiento de Amplitud |
| CI. | Circuito Integrado. |
| D/A. | Digital/Analógico |
| E/S | Entrada/Salida |
| EEPROM | Memoria de solo lectura programable y borrrable eléctricamente. |
| FAMOS | Floating Gate Avalanche-Injection Metal Oxide Semiconductor |
| FLASH | Memoria no volátil de bajo consumo. |
| FSK | (Frequency Shift Keying) Modulación por desplazamiento de Frecuencia |
| GPR | Registros de propósito General. |
| HS | Cristal de alta velocidad. |
| HSEROUT | Salida serial de hardware asíncrono. |
| INTOSC | Oscilador Interno. |
| I/O. | Input/Output. |
| Led | Diodo emisor de luz |
| LP | Cristal para baja potencia y bajo consumo de corriente. |
| MCLR | Master Clear (Reset). |
| MHz | Megahercios. |
| Mbps | Mega Bits por Segundo. |
| MHz. | Megahercios |
| mA | miliamperios |
| ms | milisegundos |
| mV | mili voltios |
| MPSK | (Multiple Phase Shift Keying)Modulación por desplazamiento de Fase Multiple |

| | |
|-------------------|--|
| OSCI/CLKIN | Entrada del oscilador (cristal). Entrada de oscilador externo. |
| PIC | Peripheral Interface Controller. |
| SFR | Registros de Función Específica. |
| USART | Transmisor/Receptor Asíncrono Universal. |
| Vcc | Voltios de Corriente continúa |
| VCA | Voltios de corriente Alterna |

INDICE

CAPÍTULO I

| | | |
|-------|-----------------------------|------|
| 1.2 | MARCO REFERENCIAL ----- | 16 - |
| 1.3 | ANTECEDENTES ----- | 16 - |
| 1.4 | JUSTIFICACIÓN ----- | 17 - |
| 1.5 | OBJETIVOS ----- | 18 - |
| 1.5.1 | OBJETIVO GENERAL ----- | 18 - |
| 1.5.2 | OBJETIVOS ESPECIFICOS ----- | 18 - |
| 1.6 | HIPÓTESIS ----- | 18 - |

CAPÍTULO II

| | | |
|--------|------------------------------------|------|
| 2.1 | FUNDAMENTO TEÓRICO ----- | 19 - |
| 2.2 | ¿Qué es Arduino? ----- | 19 - |
| 2.3 | Introducción ----- | 19 - |
| 2.4 | Tipos De Arduino ----- | 20 - |
| 2.4.1 | ¿Por Qué Arduino? ----- | 25 - |
| 2.5 | Selección De Arduino ----- | 27 - |
| 2.6 | Alimentación ----- | 30 - |
| 2.7 | Entrada y Salida ----- | 31 - |
| 2.8 | Comunicación ----- | 32 - |
| 2.8.1 | ¿Qué es el puerto de serie? ----- | 33 - |
| 2.9 | Arduino y el puerto de serie ----- | 35 - |
| 2.10 | Programación ----- | 36 - |
| 2.11 | Conecte la tarjeta ----- | 37 - |
| 2.11.1 | Instale los controladores ----- | 37 - |
| 2.12 | Selecciona tu tablero ----- | 38 - |
| 2.13 | Selección de un Arduino Uno ----- | 39 - |

| | | |
|--------|---|------|
| 2.14 | Seleccione su puerto serie ----- | 39 - |
| 2.15 | Sube el programa ----- | 39 - |
| 2.16 | Reseteo Automático (Software) ----- | 40 - |
| 2.17 | Protección de Sobrecarga del USB ----- | 41 - |
| 2.18 | Características Físicas ----- | 41 - |
| 2.19 | SENSORES ----- | 41 - |
| 2.19.1 | Introducción ----- | 41 - |
| 2.20 | Tipo de Sensores ----- | 41 - |
| 2.20.1 | Termistor ----- | 41 - |
| 2.20.2 | RTD (RESISTANCE TEMPERATURE DETECTOR) ----- | 42 - |
| 2.20.3 | TERMOPAR ----- | 42 - |
| 2.20.4 | Termocuplas: ----- | 43 - |
| 2.21 | Tipos de sensores de temperatura para Arduino UNO ----- | 45 - |
| 2.22 | LM35 ----- | 45 - |
| 2.23 | Características ----- | 46 - |
| 2.24 | DHT11 y DHT22 ----- | 47 - |
| 2.25 | DS18B20 ----- | 48 - |
| 2.26 | Criterios para la Selección de Sensores ----- | 49 - |
| 2.26.1 | Sensibilidad ----- | 49 - |
| 2.26.2 | Linealidad ----- | 49 - |
| 2.26.3 | Rango ----- | 49 - |
| 2.26.4 | Tiempo de Respuesta ----- | 49 - |
| 2.26.5 | Precisión ----- | 49 - |
| 2.26.6 | Repetitividad ----- | 49 - |
| 2.26.7 | Resolución ----- | 49 - |
| 2.27 | Selección Del Sensor De Temperatura (LM35) ----- | 50 - |
| 2.28 | Interfaz LCD para Arduino Uno. ----- | 52 - |
| 2.29 | Módulo LCD JHD162A 16 × 2 ----- | 52 - |
| 2.30 | Diagrama de circuitos. ----- | 54 - |

| | | |
|-------------------------|---|------|
| 2.31 | LOS SISTEMAS DE COMUNICACIÓN----- | 55 - |
| 2.32 | TRANSMISIÓN DE ONDAS DE RADIO ----- | 55 - |
| 2.33 | COMPONENTES DE UN SISTEMA DE COMUNICACIÓN----- | 57 - |
| 2.33.1 | Emisor ----- | 58 - |
| 2.33.2 | RECEPTOR ----- | 58 - |
| 2.33.3 | CANAL ----- | 58 - |
| 2.34 | COMUNICACIÓN INALÁMBRICA----- | 58 - |
| 2.35 | Aspecto histórico y generalidades ----- | 59 - |
| 2.36 | Campos de Utilización----- | 60 - |
| 2.37 | Algunos problemas asociados con la tecnología inalámbrica ----- | 60 - |
| 2.38 | Equipo inalámbrico ----- | 60 - |
| 2.39 | COMUNICACIÓN MOVIL ----- | 60 - |
| 2.40 | Telefonía celular----- | 61 - |
| 2.41 | Celda o Célula ----- | 61 - |
| 2.42 | Clúster o Racimo ----- | 62 - |
| 2.43 | Cobertura Celular ----- | 63 - |
| 2.44 | Capacidad Celular ----- | 63 - |
| 2.45 | Señalización----- | 64 - |
| 2.46 | Handover o Transferencia----- | 65 - |
| 2.47 | Uso del Espectro en Sistemas Celulares ----- | 65 - |
| 2.48 | Roaming o Itinerancia ----- | 66 - |
| 2.49 | Subdivisión de Celdas ----- | 66 - |
| 2.50 | SISTEMA GSM ----- | 66 - |
| 2.51 | Arquitectura de una Red GSM ----- | 67 - |
| 2.51.2 | Los Subsistemas de soporte y Operación----- | 70 - |
| 2.51.3 | Mensajes de Texto ----- | 70 - |
| CAPÍTULO III | | |
| 3.1 | COMANDOS AT ----- | 72 - |

| | | |
|--------|---|------|
| 3.2 | Introducción ----- | 72 - |
| 3.3 | Sintaxis del Comando AT ----- | 73 - |
| 3.4 | Comandos de modem----- | 73 - |
| 3.4.1 | Comandos generales ----- | 74 - |
| 3.4.2 | Comandos del servicio de red ----- | 74 - |
| 3.4.3 | Comandos de seguridad ----- | 74 - |
| 3.4.4 | Comandos para la agenda de teléfonos----- | 74 - |
| 3.4.5 | Comandos para SMS ----- | 75 - |
| 3.4.6 | Petición:----- | 75 - |
| 3.4.7 | Respuesta correcta:----- | 75 - |
| 3.4.8 | Respuesta incorrecta: ----- | 76 - |
| 3.5 | Herramientas para el desarrollo de aplicaciones----- | 76 - |
| 3.6 | Ensamblador ----- | 76 - |
| 3.7 | Compilador ----- | 77 - |
| 3.8 | Simulador ----- | 77 - |
| 3.9 | Programador ----- | 77 - |
| 3.10 | Paquetes IDE ----- | 78 - |
| 3.11 | ENTORNO DE DESARROLLO EN ARDUINO----- | 78 - |
| 3.11.1 | UN EDITOR DE TEXTO----- | 78 - |
| 3.11.2 | UN ÁREA DE MENSAJES ----- | 78 - |
| 3.11.3 | UNA CONSOLA DE TEXTO----- | 78 - |
| 3.11.4 | UNA BARRA DE HERRAMIENTAS----- | 79 - |
| 3.12 | SHIELDS GPRS/GSM----- | 83 - |
| 3.13 | MÓDULO GPRS QUADBAND PARA ARDUINO: ----- | 84 - |
| 3.14 | MÓDULO GPRS+GPS QUADBAND PARA ARDUINO/RASPBERRY PI (SIM908) ----- | 85 - |
| 3.15 | MÓDULO 3G/GPRS+GPS PARA ARDUINO/RASPBERRY PI ----- | 87 - |
| 3.16 | MÓDULO GPRS/GSM QUADBAND PARA ARDUINO (SIM900)----- | 88 - |

CAPÍTULO IV

| | | |
|-----|-----------------------------|------|
| 4.1 | DISEÑO Y CONSTRUCCIÓN ----- | 90 - |
|-----|-----------------------------|------|

| | | |
|-------|---|----|
| 4.2 | ESQUEMA GENERAL DE FUNCIONAMIENTO | 90 |
| 4.3 | Sensores | 91 |
| 4.4 | Acondicionamiento de señal | 91 |
| 4.4.1 | Acondicionamiento del Sensor de Temperatura | 92 |
| 4.4.2 | Acondicionamiento de la señal analógica del sensor LM35 | 93 |
| 4.5 | Procesamiento | 95 |
| 4.6 | Pantalla (LCD) | 97 |
| 4.7 | Modem | 97 |
| 4.8 | ALGORITMO DE ENVÍO DE UN MENSAJE DE TEXTO | 99 |

CAPÍTULO V

| | | |
|-------|---|-----|
| 5.1 | IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES | 100 |
| 5.2 | IMPLEMENTACIÓN | 100 |
| 5.3 | Análisis estadístico del sistema implementado | 100 |
| 5.3.1 | HIPÓTESIS NULA H_0 | 102 |
| 5.3.2 | HIPÓTESIS ALTERNA H_1 | 102 |
| 5.3.3 | Hipótesis: | 102 |
| 5.3.4 | Donde: | 103 |
| 5.4 | Análisis de Temperatura | 105 |
| 5.5 | Prueba de Envío de un SMS | 107 |
| 5.6 | ENTORNO DE TRABAJO | 108 |
| 5.7 | ANÁLISIS ECONÓMICO | 110 |
| 5.8 | Costo de Materiales | 110 |
| 5.9 | Presupuesto General | 111 |
| 5.10 | CONCLUSIONES | 112 |
| 5.11 | RECOMENDACIONES | 113 |
| 5.12 | RESUMEN | 114 |
| 5.13 | ABSTRACT | 115 |
| 5.14 | GLOSARIO | 116 |

| | |
|---------------------|------------|
| BIBLIOGRAFÍA | 119 |
| ANEXOS | 121 |

ÍNDICE DE FIGURAS

| | |
|--|--------|
| Ilustración 1 Arduino UNO..... | - 27 - |
| Ilustración 2 Disposición de Arduino UNO..... | - 29 - |
| Ilustración 3 comunicación serial de Arduino UNO..... | - 32 - |
| Ilustración 4 Cable serial Para Arduino UNO..... | - 33 - |
| Ilustración 5 Comunicación serie Y Paralelo de Arduino UNO | - 34 - |
| Ilustración 6 Cable de Programación de Arduino UNO | - 36 - |
| Ilustración 7 Panel Principal Software Arduino..... | - 38 - |
| Ilustración 8 Panel de compilación Arduino..... | - 39 - |
| Ilustración 9 Tipos de Termistor | - 42 - |
| Ilustración 10 Termopar..... | - 43 - |
| Ilustración 11 Termocuplas..... | - 43 - |
| Ilustración 12 Esquema de conexión..... | - 44 - |
| Ilustración 13 Tipos de sensores Arduino | - 45 - |
| Ilustración 14LM 35..... | - 45 - |
| Ilustración 15 DHT11 Y DHT22 | - 47 - |
| Ilustración 16 DS18B20..... | - 48 - |
| Ilustración 17 SENSORES DE TEMPERATURA..... | - 50 - |
| Ilustración 18 Sensor de Temperatura Básico..... | - 51 - |

| | |
|---|--------|
| Ilustración 19 Diagrama Esquemático | - 51 - |
| Ilustración 20 LCD 16 X 2..... | - 52 - |
| Ilustración 21 Diagrama de configuración de LCD | - 54 - |
| Ilustración 22 Reflexión de las ondas en la ionosfera..... | - 55 - |
| Ilustración 23 Sistema de comunicación..... | - 58 - |
| Ilustración 24 Celdas en un sistema de comunicaciones móviles. | - 61 - |
| Ilustración 25 Clúster de 7 celdas. | - 63 - |
| Ilustración 26 Operation and Su upport Subsystem..... | - 70 - |
| Ilustración 27 Entorno de desarrollo | - 79 - |
| Ilustración 28 Monitor Serial. | - 82 - |
| Ilustración 29 Valores baudrate | - 82 - |
| Ilustración 30 Diseño Modular..... | - 83 - |
| Ilustración 31 Placa HILO SAGEM. | - 84 - |
| Ilustración 32 Puertos Serial y Arduino | - 85 - |
| Ilustración 33 Shield GPRS para Arduino | - 85 - |
| Ilustración 34 Pines y Puertos Shield GPRS..... | - 86 - |
| Ilustración 35 Aspecto del Shield. | - 87 - |
| Ilustración 36 Diagrama de puertos y conexiones del módulo 3G/GPRS+GPS..... | - 88 - |
| Ilustración 37 Módulo GPRS para Arduino SIM900..... | - 89 - |
| Ilustración 38 Diagrama General del Sistema..... | - 91 - |

| | |
|---|---------|
| Ilustración 39 Voltaje..... | - 93 - |
| Ilustración 40 VOUT | - 95 - |
| Ilustración 41 PANTALLA (LCD)..... | - 97 - |
| Ilustración 42 DISEÑO MPODULAR..... | - 98 - |
| Ilustración 43 Datos recolectados por el Sistema Implementado | - 101 - |
| Ilustración 44 Datos recolectados por el Sistema Profesional | - 101 - |
| Ilustración 45 Área bajo la curva de una normal estándar | - 104 - |
| Ilustración 46 Temperatura (Sistema Profesional y Sistema Implementado) | - 107 - |
| Ilustración 47 Funcionamiento del módulo Biométrico..... | - 108 - |
| Ilustración 48 Visualización de la temperatura censada. | - 109 - |
| Ilustración 49 Mensaje recibido en el celular..... | - 109 - |

ÍNDICE DE TABLAS

| | |
|---|---------|
| Tabla 1 TIPOS DE ARDUINO..... | - 20 - |
| Tabla 2 Características de las capas de la tierra | - 56 - |
| Tabla 3 Espectro de frecuencias..... | - 65 - |
| Tabla 4 voltaje del sensor LM35 con su temperatura asociada..... | - 93 - |
| Tabla 5 ANÁLISIS DE TEMPERATURA..... | - 106 - |
| Tabla 6 Materiales empleados en el ensamblaje del medidor de temperatura y sus costos ... | - 110 - |
| Tabla 7 Presupuesto General..... | - 111 - |

INTRODUCCIÓN

Desde la invención del circuito integrado, el desarrollo constante de la electrónica digital ha dado lugar a dispositivos cada vez más complejos. Entre ellos los microprocesadores y los microcontroladores. Estos se utilizan en circuitos electrónicos comerciales desde hace unos años de forma masiva, debido a que permiten reducir el tamaño y el precio de los equipos. En los últimos años se ha facilitado enormemente el trabajo con los microcontroladores al bajar los precios, aumentar las prestaciones y simplificar los montajes, de manera que en muchas ocasiones vale la pena utilizarlos en aplicaciones donde antes se utilizaba lógica discreta.

Mediante un microcontrolador PIC, podemos realizar diferentes máquinas electrónicas que incluso puedan sustituir el trabajo humano, La solución a la problemática planteada se puede dividir en “hardware” y en “software”.

El presente trabajo de investigación se encuentra dividido en cinco capítulos para el análisis del DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA DE ADQUISICIÓN DE SEÑALES BIOMÉTRICAS, los que se detallan a continuación:

Capítulo I.- En este capítulo se especifica los antecedentes, la debida justificación y los objetivos de la realización de esta tesis.

Capítulo II y III.- En estos capítulos teóricamente se establecen los tipos de sensores, actuadores, microcontroladores, baterías y demás conceptos de los elementos que se utilizan posteriormente.

Capítulo IV.- Este capítulo contiene los pasos a considerar para el diseño y construcción del sensor de señales biométricas.

Capítulo V.- En este capítulo se detallará el Análisis respectivo así como los Resultados obtenidos en la implementación de la tesis.

Esta estructura permitirá arribar a conclusiones y proponer recomendaciones fundamentadas en el proceso de investigación realizado.

CAPÍTULO I

1.2 MARCO REFERENCIAL

1.3 ANTECEDENTES

El sector médico ha encontrado en la industria electrónica un refugio en el que ha depositado sus esperanzas para perfeccionar sus técnicas de soporte tanto de carácter preventivo como intra-operatorio, dispensando asistencia incluso en sitios tan remotos en el que el acceso a la medicina moderna parecía casi imposible de suceder.

A la hora de hablar de avances, todo de una manera u otra tiene que ver con la tecnología, dentro del campo de la medicina eso no es una excepción, el desarrollo de la electrónica en la ciencia de la medicina, ha permitido grandes cambios que han logrado el bienestar de miles y miles de personas.

El monitoreo constante y personalizado de pacientes en estado crítico, es de vital importancia ya que de esto podría depender la vida misma de una persona.

El SMS en los últimos se ha convertido en un excelente medio de comunicación accesible por la mayoría de la población al tener que disponer, únicamente de un teléfono móvil.

Vista la potencia de este nuevo canal que es la mensajería corta, el objetivo principal de la presente tesis es el de integrarlo en gestión de sistema de seguridad electrónico.

1.4 JUSTIFICACIÓN

La sociedad actual demanda cada día mas, una mayor y mejor asistencia medica, la cual puede ser enriquecida en gran parte del adecuado desarrollo de la biomedicina.

Sabiendo que en nuestro país la atención medica deja mucho que desear en cuanto a la eficiencia en la adquisición de nuestros signos vitales, ya que son realizados manualmente. Se puede optimizar este proceso valiéndonos de dispositivos electrónicos.

El presente proyecto pretende ser una contribución a la medicina, aportando con datos de interés del paciente, al galeno para una asistencia inmediata.

Valiéndonos de sensores pretendemos recolectar señales, las cuales al ser procesadas por un programa serán convertidas en datos los mismos que serán transmitidos en un SMS por medio de un teléfono celular. Esta información será recibida por el médico tratante quien podrá interpretar dicha información.

1.5 OBJETIVOS

1.5.1 OBJETIVO GENERAL

- ✓ Diseñar e implementar un sistema de adquisición de señales biométricas (Temperatura corporal) mediante mensajes SMS.

1.5.2 OBJETIVOS ESPECIFICOS

- ✓ Realizar un estudio de la electrónica médica y señales biométricas.
- ✓ Realizar el estudio, investigación y programación de celulares para servicios SMS.
- ✓ Diseñar los circuitos de acondicionamiento de señales biométricas.
- ✓ Implementar el circuito de control necesario, utilizando plataforma Arduino
- ✓ Implementar el software necesario para la comunicación SMS entre celulares.

1.6 HIPÓTESIS

La electro medicina puede convertirse en una alternativa para realizar toma de signos vitales, utilizando la tecnología móvil como plataforma complementaria.

CAPÍTULO II

1.7 FUNDAMENTO TEÓRICO

1.8 ¿QUÉ ES ARDUINO?

1.9 INTRODUCCIÓN

Es una plataforma electrónica de código abierto basada en software y hardware muy fáciles de usar. Está pensado para diseñadores, como hobby y para cualquiera interesado en crear entornos interactivos.

Los proyectos que se realiza pueden ser autónomos o se pueden comunicar con software en un ordenador.

Arduino puede sentir el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el `_Arduino Programming Language_` (basado en Wiring¹) y el `_Arduino Development Environment_` (basado en Processing²). Los proyectos de Arduino pueden ser autonomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo con Flash, Processing, MaxMSP, etc.).

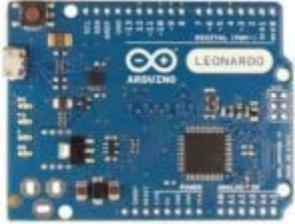
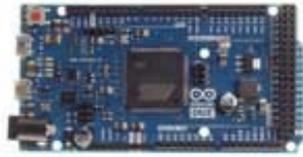
Las placas se pueden ensamblar a mano o encargarlas preensambladas. Los diseños de referencia del hardware (archivos CAD) están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Arduino recibió una mención honorífica en la sección Digital Communities del Ars Electronica

Prix en 2006

1.10 TIPOS DE ARDUINO

Tabla 1 TIPOS DE ARDUINO

| Modelo | Características |
|--|--|
| <p data-bbox="363 640 517 667">Arduino UNO</p>  | <ul style="list-style-type: none"> • Microcontrolador: ATmega328 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) • Pines de entradas análogas: 6 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader • SRAM: 2 KB (ATmega328) • EEPROM: 1 KB (ATmega328) • Velocidad de reloj: 16 MHz |
| <p data-bbox="363 1079 564 1106">Arduino Leonardo</p>  | <ul style="list-style-type: none"> • Microcontrolador: ATmega32u4 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 20 • Canales PWM: 7 • Pines de entradas análogas: 12 • Corriente DC por cada pin I/O: 40 mA • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader • SRAM: 2 KB (ATmega32u4) • EEPROM: 1 KB (ATmega32u4) • Velocidad de reloj: 16 MHz |
| <p data-bbox="363 1525 507 1552">Arduino Due</p>  | <ul style="list-style-type: none"> • Microcontrolador: AT91SAM3X8E • Voltaje de funcionamiento: 3.3 V • Pines I/O digitales: 54 (de los cuales 12 proveen salida PWM) • Pines de entradas análogas: 12 • Corriente DC total en todos los pines I/O: 130 mA • Corriente DC en el pin de 5 V: 800 mA • Corriente DC en el pin de 3.3 V: 800 mA • Memoria Flash: 512 KB disponibles para las aplicaciones de usuario. • SRAM: 96 KB (dos bancos: 64KB Y 32 KB) • Velocidad de reloj: 84 MHz |

Arduino Yún



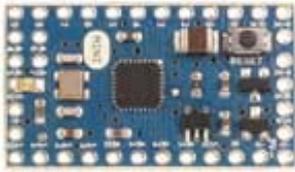
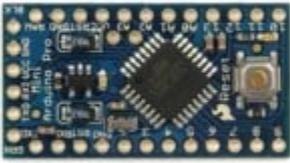
- **Microcontrolador AVR**
Arduino: ATmega32u4
- **Voltaje de funcionamiento:** 5 V
- **Pines I/O digitales:** 20
- **Canales PWM:** 7
- **Pines de entradas análogas:**12
- **Corriente DC por cada pin I/O:** 40 mA
- **Corriente DC en el pin de 3.3 V:** 50 mA
- **Memoria Flash:** 32 KB (de los cuales 4 KB son utilizados por el bootloader)
- **SRAM:** 2.5 KB
- **EEPROM:** 1 KB
- **Velocidad de reloj:** 16 MHz
- **Procesador Linux:** Atheros AR9331
- **Arquitectura:** MIPS @400MHz
- **Ethernet:** IEEE 802.3 10/100Mbit/s
- **WiFi:** IEEE 802.11b/g/n
- **USB Tipo A:** 2.0
- **Lector de tarjeta:** sólo Micro-SD
- **RAM:** 64 MB DDR2
- **Memoria Flash:**16 MB

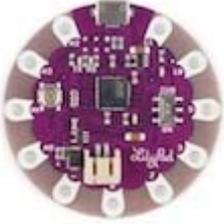
Arduino Robot

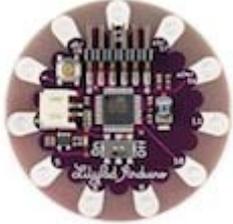
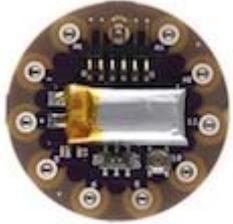
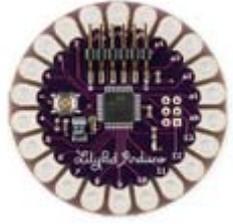


- **Microcontrolador:** ATmega32u4
- **Voltaje de funcionamiento:** 5 V
- **Pines I/O digitales:** 5
- **Canales PWM:** 6
- **Canales de entradas análogas:**4 (de los pines digitales I/O)
- **Canales (multiplexados) de entradas análogas:** 8
- **Corriente DC por cada pin I/O:** 40 mA
- **Memoria Flash:** 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader
- **SRAM:** 2 KB (ATmega32u4)
- **EEPROM (interno):** 1 KB (ATmega32u4)
- **EEPROM (externo):** 512 KB (I2C)
- **Velocidad de reloj:** 16 MHz
- **Teclado:** 5 teclas
- **Perilla:** Potenciómetro conectado a un pin análogo
- **LCD a color:** Comunicación SPI
- **Lector de tarjetas SD:** Para tarjetas formateadas FAT16
- **Altavoz:** 8 Ohms
- **Compás digital:** Proporciona la desviación desde el norte geográfico en grados
- **Áreas de prototipado:** 4

| | |
|---|--|
| <p>Arduino Esplora</p>  The image shows the Arduino Esplora board, a blue PCB with various sensors and components. It features a joystick, push buttons, a potentiometer, and several sensors like a microphone and temperature sensor. | <ul style="list-style-type: none">• Microcontrolador: ATmega32u4• Voltaje de funcionamiento: 5 V• Memoria Flash: 32 KB de los cuales 4 KB son utilizados por el bootloader• SRAM: 2.5 KB• EEPROM: 1 KB• Velocidad de reloj: 16 MHz• 4 Push bottons• Joystick análoga con un push botton central• Potenciómetro lineal• Micrófono• Fotorresistor• Sensor de temperatura• Acelerómetro de 3 ejes• Buzzer• Led RGB• Conector para LCD |
| <p>Arduino Mega ADK</p>  The image shows the Arduino Mega ADK board, a blue PCB with a USB Type-A port, a USB Type-B port, and a large number of pins. It is designed for use with the Arduino Motor Shield. | <ul style="list-style-type: none">• Microcontrolador: ATmega2560• Voltaje de funcionamiento: 5 V• Pines I/O digitales: 54 (de los cuales 15 proveen salida PWM)• Pines de entradas análogas:16• Corriente DC por cada pin I/O: 40 mA• Corriente DC en el pin de 3.3 V: 50 mA• Memoria Flash: 256 KB de los cuales 8 KB son utilizados por el bootloader• SRAM: 8 KB• EEPROM: 4 KB• Velocidad de reloj: 16 MHz |
| <p>Arduino Ethernet</p>  The image shows the Arduino Ethernet board, a blue PCB with a USB Type-A port, an Ethernet port, and a MicroSD card slot. It is designed for use with the Arduino Motor Shield. | <ul style="list-style-type: none">• Microcontrolador: ATmega328• Voltaje de funcionamiento: 5 V• Pines I/O digitales: 14 (de los cuales 4 proveen salida PWM)• Pines de entradas análogas: 6• Corriente DC por cada pin I/O: 40 mA• Corriente DC en el pin de 3.3 V: 50 mA• Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader• SRAM: 2 KB (ATmega328)• EEPROM: 1 KB (ATmega328)• Velocidad de reloj: 16 MHz• Controlador embebido Ethernet W5100 TCP/IP• Tarjeta MicroSD, con adaptadores activos de voltaje |
| <p>Arduino Mega 2560</p> | <ul style="list-style-type: none">• Microcontrolador: ATmega2560• Voltaje de funcionamiento: 5 V• Pines I/O digitales: 54 (de los cuales 15 proveen salida PWM)• Pines de entradas análogas:16• Corriente DC por cada pin I/O: 40 mA |

| | |
|--|---|
|  | <ul style="list-style-type: none"> • Corriente DC en el pin de 3.3 V: 50 mA • Memoria Flash: 256 KB de los cuales 8 KB son utilizados por el bootloader • SRAM: 8 KB (ATmega328) • EEPROM: 4 KB (ATmega328) • Velocidad del reloj: 16 MHz |
| <p>Arduino</p>  <p>Mini</p> | <ul style="list-style-type: none"> • Microcontrolador: ATmega328 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) • Pines de entradas análogas: 8 • Corriente DC por cada pin I/O: 40 mA • Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader • SRAM: 2 KB • EEPROM: 1 KB • Velocidad de reloj: 16 MHz |
| <p>Arduino Nano</p>  | <ul style="list-style-type: none"> • Microcontrolador: ATmega168 • Voltaje de funcionamiento: 5 V • Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) • Pines de entradas análogas: 8 • Corriente DC por cada pin I/O: 40 mA • Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader • SRAM: 1 KB • EEPROM: 512 bytes • Velocidad de reloj: 16 MHz |
| <p>Arduino Pro Mini</p>  | <ul style="list-style-type: none"> • Microcontrolador: ATmega168 • Voltaje de funcionamiento: 3.3 V • Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) • Pines de entradas análogas: 8 • Corriente DC por cada pin I/O: 40 mA • Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader • SRAM: 1 KB • EEPROM: 512 bytes • Velocidad de reloj: 8 MHz |

| | |
|---|---|
| <p>Arduino Pro</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega168• Voltaje de funcionamiento: 3.3 V• Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM)• Pines de entradas analógicas: 6• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader• SRAM: 1 KB• EEPROM: 512 bytes• Velocidad de reloj: 8 MHz |
| <p>Arduino Micro</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega32u4• Voltaje de funcionamiento: 5 V• Pines I/O digitales: 20• Canales PWM: 7• Pines de entradas analógicas: 12• Corriente DC por cada pin I/O: 40 mA• Corriente DC en el pin de 3.3 V: 50 mA• Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader• SRAM: 2.5 KB (ATmega32u4)• EEPROM: 1 KB (ATmega32u4)• Velocidad de reloj: 16 MHz |
| <p>Arduino Fio</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega328P• Voltaje de funcionamiento: 3.3 V• Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM)• Pines de entradas analógicas: 8• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader• SRAM: 2 KB• EEPROM: 1 KB• Velocidad de reloj: 8 MHz |
| <p>LilyPad Arduino USB</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega32u4• Voltaje de funcionamiento: 3.3 V• Pines I/O digitales: 9• Canales PWM: 4• Pines de entradas analógicas: 4• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 32 KB de los cuales 4 KB son utilizados por el bootloader• SRAM: 2.5 KB• EEPROM: 1 KB• Velocidad de reloj: 8 MHz |

| | |
|--|--|
| <p>LilyPad Arduino Simple</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega328• Voltaje de funcionamiento:2.7-5.5 V• Pines I/O digitales: 9 (de los cuales 5 proveen salida PWM)• Pines de entradas análogas: 4• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader• SRAM: 2 KB• EEPROM: 1 KB• Velocidad de reloj: 8 MHz |
| <p>LilyPad Arduino SimpleSnap</p>  | <ul style="list-style-type: none">• Microcontrolador: ATmega328• Voltaje de funcionamiento:2.7-5.5 V• Pines I/O digitales: 9 (de los cuales 5 proveen salida PWM)• Pines de entradas análogas: 4• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader• SRAM: 2 KB• EEPROM: 1 KB• Velocidad de reloj: 8 MHz |
| <p>LilyPad Arduino</p>  | <ul style="list-style-type: none">• Microcontrolador:ATmega168V• Voltaje de funcionamiento:2.7-5.5 V• Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM)• Pines de entradas análogas: 6• Corriente DC por cada pin I/O: 40 mA• Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader• SRAM: 1 KB• EEPROM: 512 bytes• Velocidad de reloj: 8 MHz |

Fuente: Internet

1.10.1 ¿Por Qué Arduino?

Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación física. Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, y muchas otras ofertas de funcionalidad similar. Todas estas herramientas toman los desordenados detalles de la programación de microcontrolador y la encierran en un paquete fácil de usar. Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas:

- **Barato:** Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino pre ensamblados cuestan menos de 50\$.

- **Multiplataforma:** El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.

- **Entorno de programación simple:** El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.

- **Código abierto y software extensible:** El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar, puedes añadir código AVR-C directamente en tus programas Arduino si quieres.

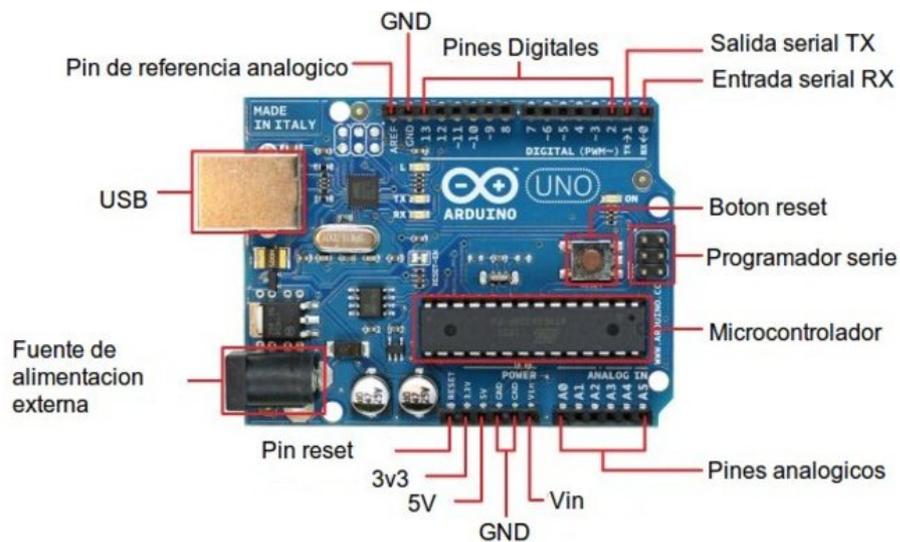
- **Código abierto y hardware extensible:** El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente

inexpertos pueden construir la versión de la placa del módulo para entender como funciona y ahorrar dinero.

1.11 SELECCIÓN DE ARDUINO

Antes de pasar a la programación, es necesario describir sobre una serie de **datos importantes que condicionan la elección** de la placa Arduino según el uso que le vayamos a dar.

Ilustración 1 Arduino UNO



Fuente: Manual de configuración de Arduino Uno

- Lo principal que debemos saber es que tipo de proyectos vamos a implementar. Con esto nos da una idea de la **cantidad de pines** analógicos y digitales (normales y de tipo PWM o modulados por ancho de pulso para simular una salida analógica) que necesitamos para nuestro trabajo. Este primer escrutinio nos permite descartar algunas placas más simples que no tengan suficientes pines o, al contrario, descartar las de mayor número de ellos para reducir los costes puesto que con menos pines nos conformamos.

- También podemos deducir **el tamaño de código** que vamos a generar para nuestros sketches. Un programa muy largo, con muchas constantes y variables demandará una cantidad mayor de memoria flash para su almacenamiento, por lo que se debe elegir una placa adecuada para no quedarnos cortos.
- La **RAM** será la encargada de cargar los datos para su inmediato procesamiento, pero no es uno de los mayores escollos, puesto que esto solo afectaría a la velocidad de procesamiento. La RAM va ligada al microcontrolador, puesto que ambos afectan a la agilidad de procesamiento de Arduino.
- En los Arduino's oficiales podemos diferenciar entre dos tipos fundamentales **demicrocontroladores**, los de 8 y 32 bits basados en ATmega AVR y los SMART basados en ARM de 32 bits y con un rendimiento superior, ambos creados por la compañía Atmel. En principio no debes guiarte por tu deseo de tener un chip de 32 bits, puesto que para la mayoría de proyectos que implementamos uno de 8 bits basta.
- Por último, en cuanto al **voltaje**, no importan demasiado a nivel electrónico, excepto en algunos casos, para tener en cuenta la cantidad de tensión que la placa puede manejar para montar nuestros circuitos. Esto no supone mayor problema, puesto que una placa de Arduino podría trabajar incluso con tensiones de 220v en alterna con el uso por ejemplo de relés. Pero cuando queremos prescindir de una fuente de alimentación externa, hay que tener en cuenta que este es el voltaje que se puede manejar. Y entre otras cosas marcar el límite para no destruir la placa con sobretensiones no soportadas. Pero no confundas el voltaje al que trabaja el microcontrolador y al que funcionan los periféricos de la placa.
- De todas formas, **la placa más vendida y que es la más aconsejable** para la mayoría de proyectos, sobre todo si estás empezando, es la Arduino UNO. Es suficiente para la mayoría de proyectos, tiene un buen precio y dispone de unos parámetros equilibrados.

Tomando en cuenta todos estos criterios de selección hemos elegido a Arduino UNO para el diseño e implementación de nuestro proyecto

Ilustración 2 Disposición de Arduino UNO



Fuente: Manual de Disposición de Arduino UNO

El Arduino Uno es una placa electrónica basada en el ATmega328. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden utilizarse para salidas PWM), 6 entradas analógicas, un 16 MHz resonador cerámico, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo

lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o el poder con un adaptador de CA o la batería a CC para empezar.

El Uno es diferente de todas las placas anteriores en que no utiliza el chip controlador de USB a serial FTDI. En lugar de ello, cuenta con la Atmega16U2 (Atmega8U2 hasta la versión R2) programado como convertidor USB a serie.

de la junta Uno tiene una resistencia tirando de la línea 8U2 HWB a tierra, por lo que es más fácil de poner en modo DFU de la Junta tiene las siguientes características nuevas:

pinout: SDA añadido y pines SCL que están cerca al pin AREF y otros dos nuevos pasadores colocados cerca del pin RESET, la instrucción IOREF que permiten a los escudos para adaptarse

a la tensión suministrado desde la pizarra. En el futuro, escudos serán compatibles tanto con el tablero que utiliza el AVR, que funciona con 5V y con el Arduino Debido que funciona con 3.3V. El segundo es un pin no está conectado, que se reserva para usos futuros.

- Circuito de rearme fuerte.
- ATmega 16U2 sustituir el 8U2.

"UNO" en italiano y se nombra para conmemorar el próximo lanzamiento de Arduino 1.0. El Uno y la versión 1.0 serán las versiones de referencia de Arduino, moviéndose hacia adelante. El Uno es el último de una serie de placas Arduino USB y el modelo de referencia para la plataforma Arduino.

1.12 ALIMENTACIÓN

El Arduino UNO puede ser alimentado a través de la conexión USB o con un suministro de energía externo. La fuente de energía se selecciona mediante el jumper PWR_SEL: para alimentar a la placa desde la conexión USB, colocarlo en los dos pines más cercanos al conector USB, para un suministro de energía externo, en los dos pines más cercanos al conector de alimentación externa.

La alimentación externa (no USB) puede venir o desde un adaptador AC-a-DC (wall-wart) o desde una batería. El adaptador puede ser conectado mediante un enchufe centro-positivo en el conector de alimentación de la placa. Los cables de la batería pueden insertarse en las cabeceras de los pines Gnd y Vin del conector POWER. Un regulador de bajo abandono proporciona eficiencia energética mejorada.

La placa puede operar con un suministro externo de 6 a 20 voltios. Si es suministrada con menos de 7 V, sin embargo, el pin de 5 V puede suministrar menos de cinco voltios y la placa podría ser inestable. Si usa más de 12 V, el regulador de tensión puede sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

- **VIN.** La entrada de tensión a la placa Arduino cuando está usando una fuente de alimentación externa (al contrario de los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Puedes suministrar tensión a través de este pin, o, si suministra tensión a través del conector de alimentación, acceder a él a través de este pin.
- **5V.** El suministro regulado de energía usado para alimentar al microcontrolador y otros componentes de la placa. Este puede venir o desde VIN a través de un regulador en la placa, o ser suministrado por USB u otro suministro regulado de 5 V.
- **3V3.** Un suministro de 3.3 V generado por el chip FTDI de la placa. La corriente máxima es de 50 mA.
- **GND.** Pines de Tierra.

1.13 ENTRADA Y SALIDA

Cada uno de los 14 pines digitales del Diecimila puede ser usado como entrada o salida, usando funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia interna (desconectada por defecto) de 20-50 KOhms. Además, algunos pines tienen funciones especiales:

- **Serial:** 0 (Rx) y 1 (Tx). Usados para recibir (Rx) y transmitir (Tx) datos TTL en serie. Estos pines están conectados a los pines correspondientes del chip FTDI USB-a-TTL Serie.
- **Interruptores externos:** 2 y 3. Estos pines pueden ser configurados para disparar un interruptor en un valor bajo, un margen creciente o decreciente, o un cambio de valor. Mirar la función `attachInterrupt()`.
- **PWM:** 3, 5, 6, 9, 10 y 11. Proporcionan salida PWM de 8 bits con la función `analogWrite()`.
- **SPI:** 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan comunicación SPI, la cual, aunque proporcionada por el hardware subyacente, no está actualmente incluida en el lenguaje Arduino.

- **LED:** 13. Hay un LED empotrado conectado al pin digital 13. Cuando el pin está a valor HIGH, el LED está encendido, cuando el pin está a LOW, está apagado.

El Diecimila tiene 6 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (por ejemplo 1024 valores diferentes). Por defecto miden 5 voltios desde tierra, aunque es posible cambiar el valor más alto de su rango usando el pin ARF y algún código de bajo nivel. Además, algunos pines tienen funcionalidad especializada:

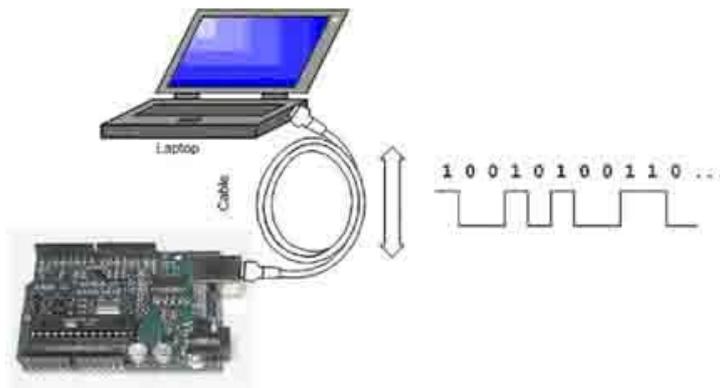
- **I²C:** 4 (SDA) y 5 (SCL). Soportan comunicación I²C (TWI) usando la librería Wire.

Hay otro par de pines en la placa:

- **AREF.** Voltaje de referencia para las entradas analógicas. Usado con analogReference().
- **Reset.** Pone esta línea a LOW para resetear el microcontrolador. Típicamente usada para añadir un botón de reset a dispositivos que bloquean a la placa principal.

1.14 COMUNICACIÓN

Ilustración 3 comunicación serial de Arduino UNO



Fuente: Manual de Configuración de Arduino UNO

El Arduino tiene un número de infraestructuras para comunicarse con un ordenador, otro Arduino, u otros microcontroladores. El software Arduino incluye un monitor serie que permite a datos de texto simple ser enviados a y desde la placa Arduino.

Ilustración 4 Cable serial Para Arduino UNO



Fuente: Manual de configuración de Arduino UNO

Los puertos serie **son la forma principal de comunicar una placa Arduino con un ordenador.** Gracias al puerto de serie podemos, por ejemplo, mover el ratón o simular la escritura de un usuario en el teclado, enviar correos con alertas, controlar un robot realizando los cálculos en el ordenador, encender o apagar un dispositivo desde una página Web a través de Internet, o desde una aplicación móvil a través de Bluetooth.

Existen un sin fin de posibilidades en las que se requiere el empleo del puerto serie. Por tanto **el puerto serie es un componente fundamental** de una gran cantidad de proyectos de Arduino, y es uno de los elementos básicos que debemos aprender para poder sacar todo el potencial de Arduino.

En esta entrada aprenderemos el funcionamiento básico de los puertos serie en Arduino. Al final de la entrada se adjuntan varios códigos de ejemplo, pero antes conviene explicar brevemente algo de teoría sobre qué es un puerto de serie, y algunos términos que necesitaremos para entender correctamente el funcionamiento del puerto serie.

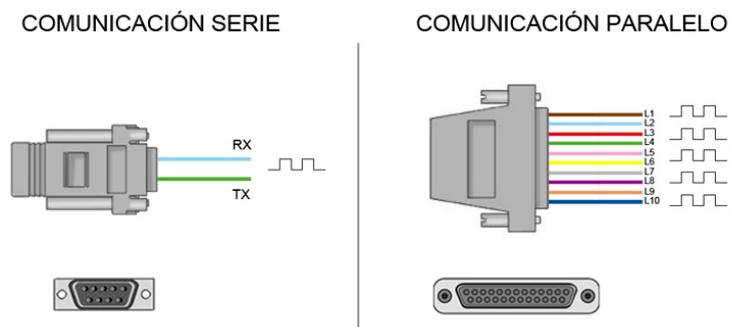
1.14.1 ¿Qué es el puerto de serie?

Un puerto es el nombre genérico con que denominamos a los interfaces, físicos o virtuales, que **permiten la comunicación entre dos ordenadores o dispositivos.**

Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión). No obstante, pueden existir otros conductores para referencia de tensión, sincronismo de reloj, etc.

Por el contrario, un puerto paralelo **envía la información mediante múltiples canales de forma simultánea**. Para ello necesita un número superior de conductores de comunicación, que varían en función del tipo de puerto. Igualmente existe la posibilidad de conductores adicionales además de los de comunicación.

Ilustración 5 Comunicación serie Y Paralelo de Arduino UNO



Fuente: disposiciones del puerto en seria internet

Históricamente ambos tipos de puertos han convivido en los ordenadores, empleándose los puertos paralelos en aquellas aplicaciones que requerían la transmisión de mayores volúmenes de datos. Sin embargo, a medida que los procesadores se hicieron más rápidos **los puertos de serie fueron desplazando progresivamente a los puertos paralelos** en la mayoría de aplicaciones.

Un ordenador convencional dispone de varios puertos de serie. Los más conocidos son el popular USB (universal serial port) y el ya casi olvidado RS-232 (el de los antiguos ratones). Sin embargo, dentro del ámbito de la informática y automatización **existen una gran cantidad adicional de**

tipos de puertos serie, como por ejemplo el RS-485, I2C, SPI, Serial Ata, Pcie Express, Ethernet o FireWire, entre otros.

En ocasiones se **referira a los puertos de serie como UART**. La UART (universally asynchronous receiver/transmitter) es una unidad que incorporan ciertos procesadores, encargada de realiza la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada.

Por otro lado, **también el término TTL**(transistor-transistor logic). Esto significa que la comunicación se realiza mediante variaciones en la señal entre 0V y Vcc (donde Vcc suele ser 3.3V o 5V). Por el contrario, otros sistemas de transmisión emplean variaciones de voltaje de -Vcc a +Vcc (por ejemplo, los puertos RS-232 típicamente varían entre -13V a 13V).

1.15 ARDUINO Y EL PUERTO DE SERIE

Prácticamente todas **las placas Arduino disponen al menos de una unidad UART**. Las placas Arduino UNO y Mini Pro disponen de una unidad UART que operan a nivel TTL 0V / 5V, por lo que son directamente compatibles con la conexión USB. Por su parte, Arduino Mega y Arduino Due disponen de 4 unidades UART TTL 0V / 5V.

Los puertos serie están físicamente unidos a distintos pines de la placa Arduino.

Lógicamente, mientras usamos los puertos de serie no podemos usar como entradas o salidas digitales los pines asociados con el puerto de serie en uso.

En Arduino UNO y Mini Pro los pines empleados son 0 (RX) y 1 (TX). En el caso de Arduino Mega y Arduino Due el puerto de serie 1 está conectado a los pines 0 (RX) y 1 (TX), el puerto de serie 1 a los pines 19 (RX) y 18 (TX) el puerto de serie 2 a los pines 17 (RX) y 16 (TX), y el puerto serie 3 a los pines 15 (RX) y 14 (TX).

Muchos modelos de placas Arduino disponen de un conector USB o Micro USB conectado a uno de los puertos de serie, lo que simplifica el proceso de conexión con un ordenador. Sin embargo algunas placas, como por ejemplo la Mini Pro, prescinden de este conector por lo que la única forma de conectarse a las mismas es directamente a través de los pines correspondientes.

1.16 PROGRAMACIÓN

El Arduino Diecimila puede ser programado con el software Arduino9.

El ATmega168 del Arduino Diecimila viene con un bootloader 10 pregrabado que te permite subirle nuevo código sin usar un programador hardware externo. Se comunica usando el protocolo original STK500 como detallamos a continuación.

También se necesita un cable USB estándar (Un enchufe a enchufe B): el tipo que se conectaría a una impresora USB, por ejemplo. (Para el Arduino Nano, usted necesitará un cable de A a mini-B en su lugar.)

Ilustración 6 Cable de Programación de Arduino UNO



Fuente: Manual de Programación de Arduino UNO

1.17 CONECTE LA TARJETA

El Arduino Uno, Mega, Duemilanove y Arduino Nano dibujar automáticamente de alimentación de la conexión USB al ordenador o una fuente de alimentación externa. Si usted está utilizando un Diecimila Arduino, usted necesita para asegurarse de que la tarjeta está configurada para extraer energía de la conexión USB. La fuente de alimentación se selecciona con un puente, una pequeña pieza de plástico que se coloca en dos de los tres pasadores entre los conectores USB y de alimentación. Compruebe que está en los dos pines más cercanos al puerto USB.

Conecte la placa Arduino al ordenador mediante el cable USB. El LED verde de alimentación (etiquetado PWR) debe continuar.

1.17.1 Instale los controladores

Instalación de los controladores para el Arduino Uno o Arduino Mega 2560 con Windows 7, Vista o XP:

- Conecte su tablero y espere a que Windows comience su proceso de instalación del controlador. Después de unos momentos, el proceso va a fracasar, a pesar de sus mejores esfuerzos
- Haga clic en el menú Inicio y abra el Panel de Control.
- Mientras que en el Panel de control, vaya a Sistema y seguridad. A continuación, haga clic en Sistema. Una vez que la ventana del sistema es, abra el Administrador de dispositivos.
- Busque en Puertos (COM & LPT). Usted debe ver a un puerto abierto llamado "Arduino UNO (COMxx) ". Si no hay una sección LPT COM y, mire en "Otros dispositivos" para "Dispositivo desconocido".
- Haga clic derecho sobre el "UNO Arduino (COMxx) "puerto y elegir la opción " Actualizar software de controlador ".
- A continuación, seleccione la opción "Buscar software de controlador".

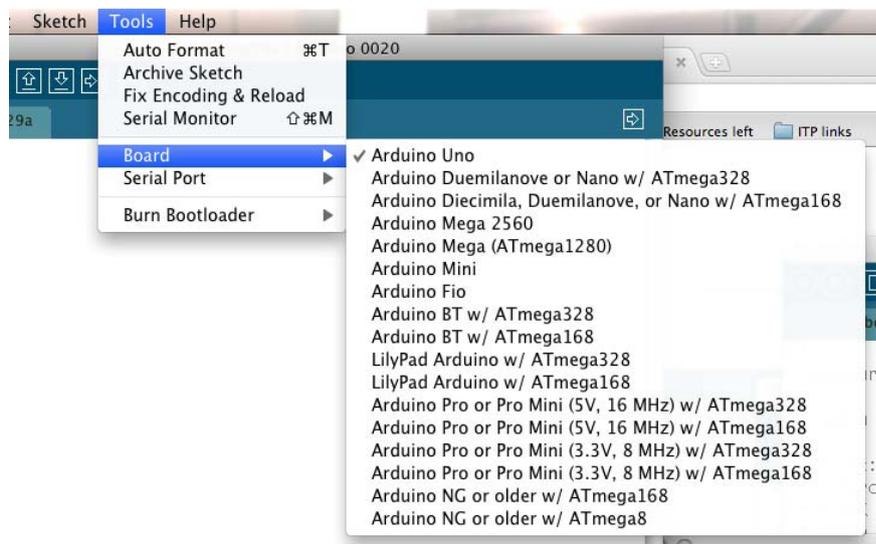
- Por último, busque y seleccione el archivo controlador llamado "arduino.inf" , ubicado en la carpeta "Drivers" de la descarga del software de Arduino (no el "Controladores USB FTDI" sub-directorio). Si está utilizando una versión antigua del IDE (1.0.3 o anterior), seleccione el archivo del controlador Uno llamado "Arduino UNO.inf"
- Windows terminar la instalación del controlador de allí.

Usted puede comprobar que los controladores se han instalado abriendo el Manager de dispositivos de Windows (en la pestaña Hardware del panel de control del sistema). Busque un "USB Serial Port" en la sección de Puertos; esa es la placa Arduino.

1.18 SELECCIONA TU TABLERO

Usted tendrá que seleccionar la entrada en la Herramientas> Junta menú que corresponde a su Arduino.

Ilustración 7 Panel Principal Software Arduino



Fuente: Software Arduino Captura de pantallas

1.19 SELECCIÓN DE UN ARDUINO UNO

Para las placas Arduino Duemilanove con un ATmega328 (compruebe el texto en el chip de la placa), seleccione Arduino Duemilanove o Nano w / ATmega328 . Anteriormente, las placas Arduino vinieron con un ATmega168 ; para aquellos, seleccione Arduino Diecimila, Duemilanove, o Nano w / ATmega168 .

1.20 SELECCIONE SU PUERTO SERIE

Seleccione el dispositivo de serie de la placa Arduino desde el menú Tools | Serial Port. Esto es probable que sea COM3 o superior (COM1 y COM2 son generalmente reservados para puertos serie de hardware). Para averiguarlo, puede desconectar la placa Arduino y vuelva a abrir el menú la entrada que desaparece debe ser la placa Arduino. Vuelva a conectar el tablero y seleccionar ese puerto serie.

1.21 SUBE EL PROGRAMA

Ahora, simplemente haga clic en el botón "Subir" en el medio ambiente. Espere unos segundos - debería ver los led RX y TX en el parpadeo bordo. Si la carga se realiza correctamente, el mensaje "Done subir." aparecerá en la barra de estado. (*Nota:* Si usted tiene un Mini Arduino, NG, o el otro tablero, tendrás que presentar físicamente el botón de reinicio en la junta inmediatamente antes de pulsar el botón de subida.)

Ilustración 8 Panel de compilación Arduino



Fuente: Software Arduino Captura de pantallas

Unos segundos después de que finalice la carga, debería ver el pin 13 (L) LED en el tablero comenzará a parpadear (en naranja). Si es así, ¡felicitaciones! Te has vuelto Arduino arriba y en ejecución.

1.22 RESETEO AUTOMÁTICO (SOFTWARE)

En lugar de requerir una pulsación física del botón de reset antes de una subida, el Arduino Diecimila está diseñado de forma que permite ser reseteado por software en ejecución en una computadora conectada. Una de las líneas de control de _ujo de hardware (DTR) del FT232RL está conectada a la línea de reset del ATmega168 a través de un condensador de 100 nF. Cuando esta línea toma el valor LOW, la línea reset se mantiene el tiempo suficiente para resetear el chip.

La versión 0009 del software Arduino usa esta capacidad para permitirte cargar código simplemente presionando el botón upload en el entorno Arduino. Esto significa que el bootloader puede tener un tiempo de espera más corto, mientras la bajada del DTR puede ser coordinada correctamente con el comienzo de la subida.

Esta configuración tiene otras repercusiones. Cuando el Diecimila está conectado a un ordenador que ejecuta o Mac OS X o Linux, se resetea cada vez que se hace una conexión a él por software (a través de USB). Durante el siguiente medio segundo aproximadamente, el bootloader se ejecutará en el Diecimila. Mientras esté programado para ignorar datos malformados (por ejemplo, cualquiera excepto una subida de código nuevo), interceptará los primeros bytes de datos enviados a la placa después de abrir la conexión. Si una rutina que se ejecuta en la placa recibe una configuración una vez u otros datos cuando empieza, asegurarse de que el software con el que se comunica espera un segundo después de abrir la conexión y antes de enviar estos datos.

1.23 PROTECCIÓN DE SOBRECARGA DEL USB

El Arduino UNO tiene un fusible reseteable que protege tus puertos USB del ordenador de cortes y sobrecargas. Aunque la mayoría de los ordenadores proporcionan su propia protección interna, el fusible proporciona una capa de protección extra. Si más de 500 mA se aplican al puerto USB, el fusible automáticamente romperá la conexión hasta que el corte o la sobrecarga sean eliminados.

1.24 CARACTERÍSTICAS FÍSICAS

La máxima longitud y anchura del Arduino UNO PCB son 2.7 y 2.1 pulgadas respectivamente, con el conector USB y el conector de alimentación que se extienden más allá de las primeras dimensiones. Tres agujeros de tornillo permiten a la placa atornillarse a una superficie o caja.

1.25 SENSORES

1.25.1 Introducción

Los sensores de temperatura son dispositivos que transforman los cambios de temperatura en cambios en señales eléctricas que son procesados por equipo eléctrico o electrónico.

Hay tres tipos de sensores de temperatura, los termistores, los RTD y los termopares.

El sensor de temperatura, típicamente suele estar formado por el elemento sensor, de cualquiera de los tipos anteriores, la vaina que lo envuelve y que está rellena de un material muy conductor de la temperatura, para que los cambios se transmitan rápidamente al elemento sensor y del cable al que se conectarán el equipo electrónico.

1.26 TIPO DE SENSORES

1.26.1 Termistor

El termistor está basado en que el comportamiento de la resistencia de los semiconductores es variable en función de la temperatura.

Existen los termistores tipo NTC y los termistores tipo PTC. En los primeros, al aumentar la temperatura, disminuye la resistencia. En los PTC, al aumentar la temperatura, aumenta la resistencia.

El principal problema de los termistores es que no son lineales según la temperatura por lo que es necesario aplicar fórmulas complejas para determinar la temperatura según la corriente que circula y son complicados de calibrar.

Ilustración 9 Tipos de Termistor



Fuente: Tipos de Termistor Internet

1.26.2 RTD (RESISTANCE TEMPERATURE DETECTOR)

Un RTD es un sensor de temperatura basado en la variación de la resistencia de un conductor con la temperatura.

Los metales empleados normalmente como RTD son platino, cobre, níquel y molibdeno.

De entre los anteriores, los sensores de platino son los más comunes por tener mejor linealidad, más rapidez y mayor margen de temperatura.

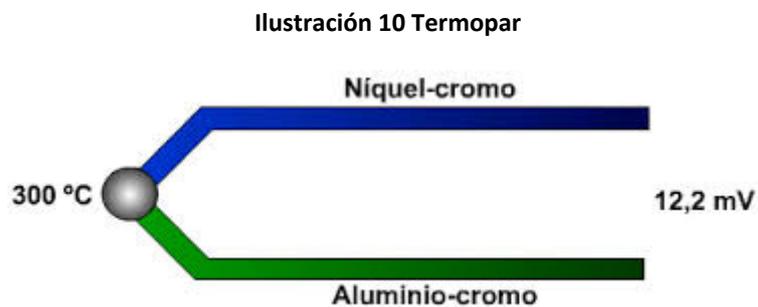
1.26.3 TERMOPAR

El termopar, también llamado termocupla y que recibe este nombre por estar formado por dos metales, es un instrumento de medida cuyo principio de funcionamiento es el efecto termoeléctrico.

Un material termoeléctrico permite transformar directamente el calor en electricidad, o bien generar frío cuando se le aplica una corriente eléctrica.

El termopar genera una tensión que está en función de la temperatura que se está aplicando al sensor. Midiendo con un voltímetro la tensión generada, conoceremos la temperatura.

Los termopares tienen un amplio rango de medida, son económicos y están muy extendidos en la industria. El principal inconveniente estriba en su precisión, que es pequeña en comparación con sensores de temperatura RTD o termistores.



1.26.4 Termocuplas:

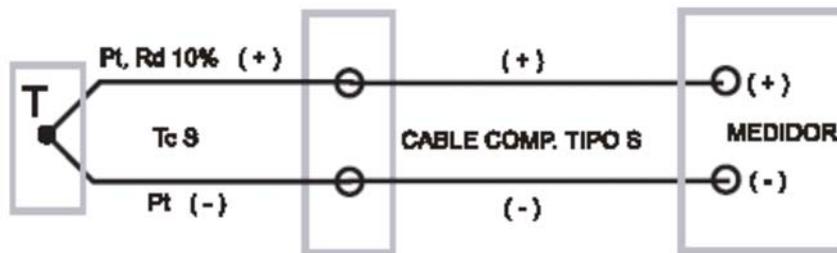
Las termocuplas son los sensores de temperatura eléctricos más utilizados en la industria. Una termocupla se hace con dos alambres de distinto material unidos en un extremo, al aplicar temperatura en la unión de los metales se genera un voltaje muy pequeño, del orden de los mili volts el cual aumenta con la temperatura. Este sería un esquema de ejemplo de una termocupla cualquiera.



Estos dispositivos suelen ir encapsulados en vainas, para protegerlos de las condiciones extremas en ocasiones del proceso industrial que tratan de ayudar a controlar, por ejemplo suele utilizarse acero inoxidable para la vaina, de manera que en un extremo está la unión y en el otro el terminal eléctrico de los cables, protegido adentro de una caja redonda de aluminio (cabezal). Además según la distancia a los aparatos encargados de tratar la pequeña señal eléctrica de estos transductores, también deben utilizarse cables compensados para transportar esta señal sin que la modifique o la modifique de una manera fácilmente reconocible y reversible para los dispositivos de tratamiento de la señal. También se da el caso de que los materiales empleados en la termocupla como el platino puro, hagan inviable económicamente extender la longitud de los terminales de medición de la termocupla

Esquema de conexión de cable compensado y termocupla:

Ilustración 12 Esquema de conexión



Fuente: Patricio Hidalgo

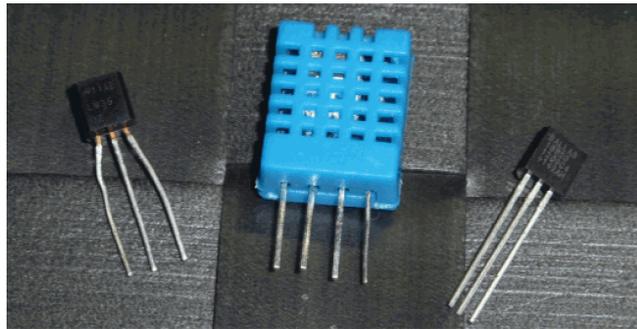
Los cables compensados tienen una polaridad de conexión (+) y (-) que al conectarse con la termocupla se debe respetar. Es importantísimo que estos dos cables compensados sean para el tipo de termocupla que se está usando y además estén conectados con la polaridad correcta (+) con (+) y (-) con (-). De otra forma será imposible obtener una medición sin error.

Las termocuplas podrían clasificarse atendiendo a varios criterios como material del que están construidas, su tolerancia o desviación, etc. Durante varios años ha habido diferentes organismos

de estandarización de nacionalidades diferentes intentando normalizar la gran variedad de este tipo de sensores e incluso unificar sus criterios de normalización.

1.27 TIPOS DE SENSORES DE TEMPERATURA PARA ARDUINO UNO

Ilustración 13 Tipos de sensores Arduino

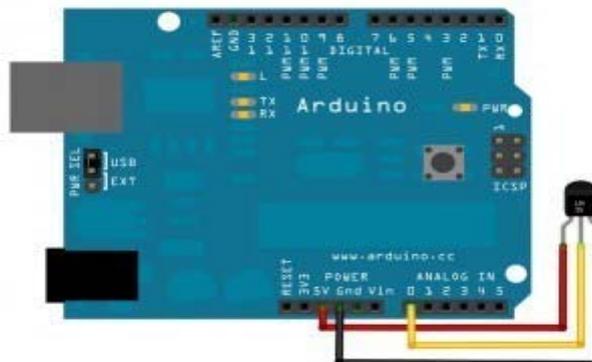


Fuente: Tipos de sensores Arduino Internet

Disponemos de gran cantidad de sensores para medir la temperatura. Algunos de ellos son analógicos y otros digitales, cada uno con sus pros y sus contras. También los hay de diversos precios, a continuación veremos 4 de ellos.

1.28 LM35

Ilustración 14LM 35



Fuente: Manual de conexión LM 35

Este sensor suele encontrarse con encapsulado TO-92, es **analógico** y también el más económico de todos pudiendo encontrarlo por menos de 1€ También es el más sencillo de usar, solo es necesario realizar una lectura desde el pin analógico y convertir, mediante unas cuentas, el valor leído a la temperatura.

Tiene un rango de medición de -55 a 150°C, con una precisión de 0,25°C a temperatura ambiente o de 0,75°C entre 55 y 150°C. El sensor es bastante sensible y nos permite obtener la rápidamente. Al ser un sensor analógico el cable que une el sensor y el Arduino puede influenciar en las lecturas.

1.29 CARACTERÍSTICAS

- Calibrado directamente en grados Celsius (Centígrados)
- Factor de escala lineal de +10 mV / °C
- 0,5°C de precisión a +25 °C
- Rango de trabajo: -55 °C a +150 °C
- Apropiado para aplicaciones remotas
- Bajo coste
- Funciona con alimentaciones entre 4V y 30V
- Menos de 60 µA de consumo
- Bajo auto-calentamiento (0,08 °C en aire estático)
- Baja impedancia de salida, 0,1W para cargas de 1mA

Typical Applications

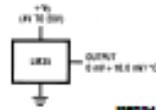


FIGURE 1. Basic Centigrade Temperature Sensor (-2°C to +150°C)

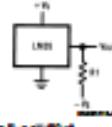
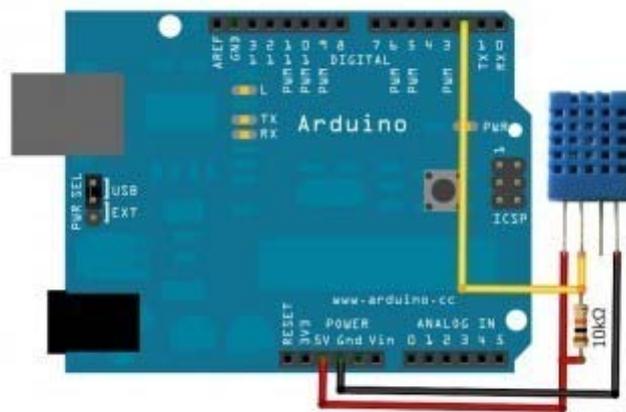


FIGURE 2. Full-Range Centigrade Temperature Sensor

1.30 DHT11 Y DHT22

Ilustración 15 DHT11 Y DHT22



Fuente: Manual de configuración ArduinoDHT11 Y DHT22

Estos dos sensores nos proporcionan la temperatura y la **humedad** relativa. Usan un encapsulado de 4 pines, aunque solo usan 3, y emplean para su funcionamiento un pin **digital**. El precio del DHT11 ronda el euro, en el caso del DHT22, mucho más sensible, sale por unos 4€ Su uso se complica un poco mas respecto el anterior, sin embargo haciendo uso de lalibrería adjunta se facilita todo mucho.

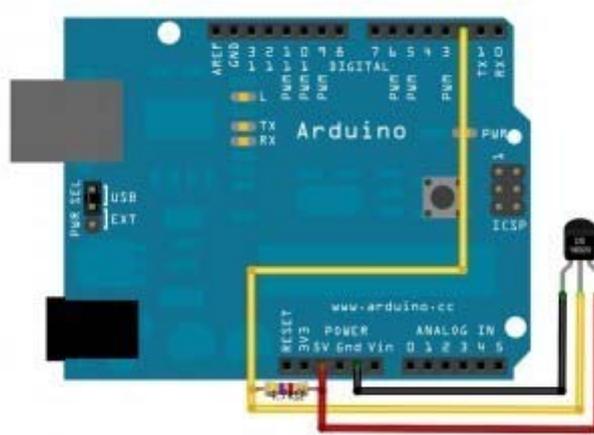
El DHT11 tiene un rango de medición de 0 a 50°C y 20 a 90%RH con una tolerancia del 5% y una precisión de 1.

En el caso del DHT22 su rango de medición es de -40 a 125°C y de 0 a 100%RH con una tolerancia de 0,2°C con al temperatura y de 2%RH (con un máximo de 5%) con la humedad, la precisión es de 0,1 para ambos valores.

Ambos sensores son bastantes lentos, refrescando cada 2 segundos aproximadamente.

1.31 DS18B20

Ilustración 16 DS18B20



Fuente: Manual de configuración de Arduino Internet

Este otro sensor, también con encapsulado TO-92, usa un pin **digital** y su precio ronda el euro. A diferencia del resto este nos permite conectar varios de ellos usando el mismo pin ya que se identifican del resto con un identificador único de 64 bits. Para su funcionamiento se hace uso de un bus 1-wire y es necesario varias librerías, la primera de ellas la *OneWire* y la segunda la *DallasTemperature*.

Tiene un rango de medición de -55 a 125°C con una precisión de 0,5°C cuando se encuentra entre un rango de -10 a 85°C

El sensor en sí funciona muy bien, personalmente pudiendo elegir me quedo con este en vez del LM35 ya que este es digital, permite conectar mas sensores usando el mismo pin, y es mas preciso. En contra tiene que es necesario usar unas librerías para su funcionamiento, aunque eso también facilita su uso pues no necesita operaciones matemáticas.

1.32 CRITERIOS PARA LA SELECCIÓN DE SENSORES

1.32.1 Sensibilidad

- Se define como el cociente entre la tasa de cambio de valores de salida para el cambio los valores de entrada.

1.32.2 Linealidad

- Es la medida de la constancia de la tasa de salida con respecto a la entrada.

1.32.3 Rango

- Es la medida de la diferencia entre el mínimo y el máximo valor medido.

1.32.4 Tiempo de Respuesta

- Puede ser un tiempo fijo o depender de cuánto varíe la magnitud a medir. Depende de la capacidad del sistema para seguir las variaciones de la magnitud de entrada.

1.32.5 Precisión

- Es el error de medida máximo esperado.

1.32.6 Repetitividad

- Error esperado al repetir varias veces la misma medida.

1.32.7 Resolución

- Mínima variación de la magnitud de entrada que puede apreciarse a la salida.

1.33 SELECCIÓN DEL SENSOR DE TEMPERATURA (LM35)

Para realizar nuestro proyecto de tesis, además de una pantalla (LCD3310/5110), necesitamos un microcontrolador (Arduino) y lo más importante, un sensor que sea capaz de convertir en tensión, los cambios que se registren en la temperatura ambiente. Encontraremos disponibles, distintos sensores (sensores pasivos; diodo 1N4148, PTC, NTC y sensores activos; LM35, DS18B20, LM385B-2.5, etc.). Según el sensor que elijamos, deberemos conocer sus características y adaptar los parámetros a ese sensor, de él dependen, el margen de temperaturas a registrar y la exactitud de los resultados.

Ilustración 17 SENSORES DE TEMPERATURA



NTC LM35. DS18B20.

Fuente: SENSORES DE TEMPERATURA INTERNET

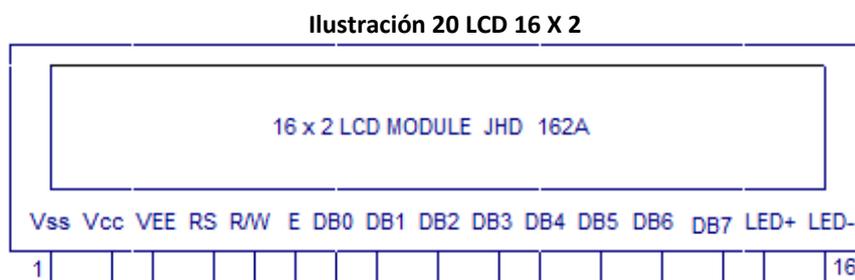
En esta ocasión utilizaremos un sensor analógico pasivo, en otras palabras una NTC que es una resistencia que depende de la temperatura. Según las notas del fabricante, el LM35 es un sensor de temperatura en centígrados de precisión, son circuitos integrados de precisión, sensores de temperatura, cuya tensión de salida, es linealmente proporcional a la temperatura Celsius (centígrados). El LM35 por lo tanto tiene una ventaja sobre los sensores de temperatura lineales calibradas en ° Kelvin. El LM35 no requiere ninguna calibración externa o el recorte para proporcionar precisiones típicas de $\pm 1/4$ °C a temperatura ambiente y $\pm 3/4$ °C durante un gama completa -55-150 temperatura °C.

1.34 INTERFAZ LCD PARA ARDUINO UNO.

Módulos LCD forman una parte muy importante en muchos diseños de sistemas embebidos basados en Arduino. Así que el conocimiento sobre la interconexión LCD para Arduino es muy esencial en el diseño de sistemas embebidos. Este artículo es sobre la conexión de un LCD de 16×2 a Arduino. JHD162A es el módulo LCD utilizado aquí. JHD162A es un módulo LCD 16×2 basado en el controlador de Hitachi HD44780. El JHD162A tiene 16 pines y puede ser operado en el modo de 4 bits o el modo de 8 bits. Aquí estamos utilizando el módulo LCD en modo de 4 bits. En primer lugar, te voy a mostrar cómo mostrar mensajes de texto sin formato en el módulo LCD utilizando Arduino y luego algunos proyectos útiles usando LCD y Arduino. Antes de entrar en los detalles del proyecto, vamos a echar un vistazo al módulo JHD162A LCD.

1.35 MÓDULO LCD JHD162A 16×2

El JHD162A tiene 16 pines y puede ser operado en el modo de 4 bits o el modo de 8 bits. Aquí estamos utilizando el módulo LCD en modo de 4 bits. Antes de entrar en los detalles del proyecto, vamos a echar un vistazo a la esquemática JHD162A LCD module. The de un módulo JHD162A LCD se da a continuación.



Fuente: LCD 16 X 2 Internet

El nombre y las funciones de cada pin del módulo LCD JHD162A es la siguiente.

Pin1 (Vss) : conector de tierra del módulo LCD.

Pin2 (Vcc) : + 5V se le da a este pin

Pin3 (VEE) : pasador de ajuste de contraste. Esto se realiza mediante la conexión de los extremos de un potenciómetro 10K a + 5V y tierra y luego conectar el perno deslizante a la clavija de VEE. El voltaje en el pin de VEE define el contraste. El ajuste normal es de entre 0.4 y 0.9V.

Pin4 (RS) : Regístrese seleccionar pin. The JHD162A tiene dos registros saber comando de registro y registro de datos. Lógico alto en RS pin selecciona el registro de datos y la lógica BAJA en el pin RS seleccionará registro de comandos. Si hacemos las RS PIN de alto y poner un dato de las líneas de datos (DB0 a DB7) Se reconocerá como datos. Si hacemos las RS pin BAJA y poner un dato de las líneas de datos, entonces se tomará como un comando.

Pin5 (R / W) : Leer modos / escritura. Este pin se utiliza para seleccionar entre los modos de leer y escribir. Lógico alto en este pin se activa el modo de leer y lógico bajo en este pin se activa el modo de escritura.

PIN6 (E) : Este pin es para habilitar el módulo LCD. UN ALTO a la señal BAJO en este pin permitirá al módulo.

Pin7 (DB0) para Pin14 (DB7) : Estos son pines de datos. Los comandos y los datos se ponen en estos pines.

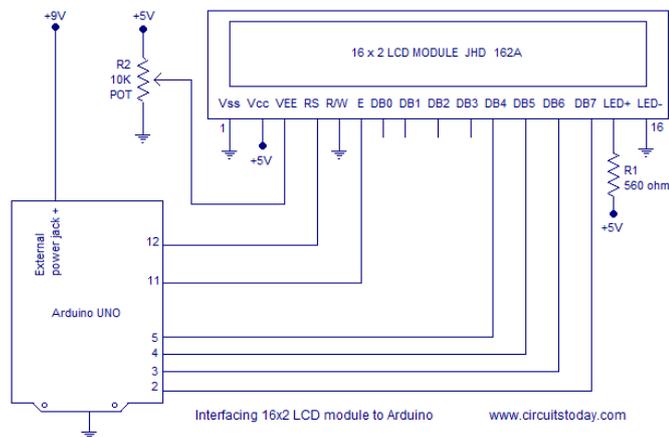
PIN15 (LED +) : ánodo de la luz de fondo LED. Cuando se opera en 5V, una resistencia de 560 ohm debe estar conectado en serie a esta clavija. En los proyectos de Arduino basado el LED luz de fondo puede ser alimentado desde la fuente de 3.3V de la placa Arduino.

Pin16 (LED-) : Cátodo de la luz de fondo LED.

Para saber más sobre el módulo LCD JHD162A y la interconexión, lea este artículo: [Interfaz de 16 × 2 LCD y 8051](#) . El diagrama de circuito de interfaz LCD para Arduino para la visualización de un mensaje de texto se muestra a continuación.

1.36 DIAGRAMA DE CIRCUITOS.

Ilustración 21 Diagrama de configuración de LCD



Fuente: ANGULO, J.M^a

RS pin del módulo LCD está conectado al pin digital 12 de la Arduino. R / W pin de la LCD está conectado a tierra. Habilitar pin del módulo LCD está conectado al pin digital 11 de la Arduino. En este proyecto, el módulo LCD y arduino están interconectados en el modo de 4 bits. Eso significa que sólo cuatro de las líneas de entrada digitales (DB4 a DB7 del LCD se utilizan). Este método es muy simple, requiere menos conexiones y casi se puede utilizar todo el potencial del módulo LCD. Líneas digitales DB4, DB5, DB6 y DB7 están interconectados a los pines digitales 5, 4, 3 y 2 de la Arduino. El potenciómetro de 10K se utiliza para ajustar el

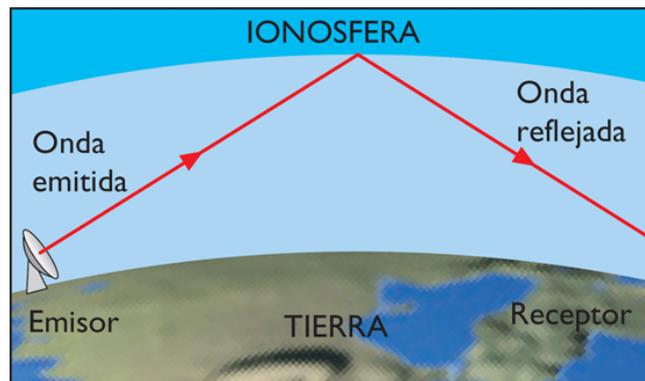
contraste de la pantalla. 560 ohm resistencia R1 limita la corriente a través de la luz de fondo LED. El Arduino puede ser alimentado a través de la toma de alimentación externa proporcionada en el tablero. + 5V requerido en algunas otras partes del circuito se pueden tomar de la fuente de 5 V de la placa Arduino. El Arduino puede también alimenta desde el PC a través del puerto USB. El programa completo para la interfaz LCD para Arduino se muestra a continuación.

1.37 LOS SISTEMAS DE COMUNICACIÓN

El conjunto de elementos que intervienen en el proceso de intercambio de información forma un sistema de comunicación como se observa en la figura II.3.

1.38 TRANSMISIÓN DE ONDAS DE RADIO

Ilustración 22 Reflexión de las ondas en la ionosfera



Fuente: ANGULO, J.M

Las ondas electromagnéticas se propagan en línea recta. En consecuencia, si pretendemos enviar una señal de radio a larga distancia, dado que la Tierra tiene una superficie redondeada, la señal se alejará de la superficie de la Tierra y se perderá en el espacio. Sin embargo, las ondas de radio tienen la propiedad de reflejarse en las capas altas de la atmósfera, en concreto en la ionosfera.

La primera vez que se realizó una transmisión de radio a larga distancia fue en 1901. Entonces se desconocía la existencia de la ionosfera. Fue Marconi quien dispuso un transmisor y un receptor

a ambos lados del Atlántico, entre Cornualles, en Inglaterra, y Terranova, en Canadá. Tras el éxito del experimento, Oliver Heavyside.

Kennelly descubrieron la existencia de la ionosfera y sus propiedades como reflectante de señales de cierta banda de frecuencias.

La ionosfera es la capa de la atmósfera situada entre los 90 y los 400 km de altura. Presenta la particularidad de que en ella los átomos se ionizan y liberan electrones por efecto de la luz solar. Según la concentración de iones, la ionosfera se puede dividir en varias capas, que se comportan de diferente forma ante la reflexión de las ondas. En cierto modo, al existir una nube electrónica en la ionosfera, esta se comporta como una pantalla para las señales eléctricas. No obstante, dependiendo de dicha concentración, existirá mayor o menor «blindaje» frente a las señales.

Tabla 2 Características de las capas de la tierra

| Capa de Ionosfera | Características de la capa | Rango frecuencias que refleja |
|--------------------------|--|--------------------------------------|
| Capa D | Capa apenas ionizada, únicamente refleja las frecuencias más bajas, debilitando las señales de frecuencias medias. | 0-3 MHz |
| Capas E-F | Son dos capas de características similares, con igual comportamiento frente a las señales. Están situadas entre los 120 y 150 km, y reflejan las señales de frecuencia media. La reflexión de la señal depende de las variaciones en la concentración iónica | 3-8 MHz |

| | | |
|----------------|---|----------|
| | entre el día, debido a la incidencia de la luz solar, y la noche, cuando decrece la concentración. | |
| Capa F' | Es la capa más importante por la gran concentración de iones, cubriendo un espesor de hasta los 500 km. | 8-15 MHz |

Fuente: ANGULO, J.M

Las ondas reflejadas en la ionosfera que vuelven a la Tierra pueden ser de nuevo emitidas hacia el espacio y sufrir una segunda reflexión en la ionosfera. De hecho, este proceso se puede repetir sucesivas veces, de manera que las ondas podrán salvar grandes distancias gracias a las continuas reflexiones. Incluso, si se emite una señal con potencia y frecuencia adecuada, es posible que las ondas circunden la Tierra.

Por otra parte, no todas las frecuencias rebotan en la ionosfera. Las señales con frecuencias superiores a 15 MHz «escapan» a la reflexión de la atmósfera. Este es el rango de las señales de alta frecuencia (HF), de muy alta frecuencia (VHF), de ultra-alta frecuencia (UHF), así como de frecuencias superiores como las microondas. Este tipo de señales solo puede utilizarse para comunicaciones a corta distancia mediante estaciones terrestres repetidoras. Para salvar mayores distancias sería necesario recurrir al uso de satélites de comunicaciones.

1.39 COMPONENTES DE UN SISTEMA DE COMUNICACIÓN

En todo sistema de comunicación podemos distinguir los siguientes componentes:

1.39.1 Emisor

es el elemento que transmite la información.

1.39.2 RECEPTOR

es el elemento que recibe la información.

1.39.3 CANAL

es el medio a través del cual tiene lugar el trasvase de información entre el emisor y el receptor.

Ilustración 23 Sistema de comunicación



Fuente: Patricio Hidalgo

El emisor y el receptor pueden encontrarse a unos pocos metros de distancia o bien estar alejados cientos e incluso miles de kilómetros, como ocurre en los sistemas de telecomunicaciones vía satélite o en los vuelos espaciales.

1.40 COMUNICACIÓN INALÁMBRICA

La comunicación inalámbrica (inglés *wireless*, sin cables) es aquella en la que extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, PDA, teléfonos móviles, etc.

1.41 ASPECTO HISTÓRICO Y GENERALIDADES

La comunicación inalámbrica, que se realiza a través de ondas de radiofrecuencia, facilita la operación en lugares donde la computadora no se encuentra en una ubicación fija (almacenes, oficinas de varios pisos, etc.); pero se trata de una tecnología sometida a investigación que en el futuro será utilizada de forma general.

Cabe también mencionar actualmente que las redes cableadas presentan ventaja en cuanto a transmisión de datos sobre las inalámbricas. Mientras que las cableadas proporcionan velocidades de hasta 1 Gbps (Red Gigabit), las inalámbricas alcanzan sólo hasta 108 Mbps.

Se puede realizar una “mezcla” entre inalámbricas y alámbricas, de manera que pueden funcionar de la siguiente manera: que el sistema cableado sea la parte principal y la inalámbrica sea la que le proporcione movilidad al equipo y al operador para desplazarse con facilidad en distintos campos (almacén u oficina).

Un ejemplo de redes a larga distancia son las Redes públicas de Conmutación por Radio. Estas redes no tienen problemas en pérdida de señal, debido a que su arquitectura está diseñada para soportar paquetes de datos en vez de comunicaciones por voz.

Actualmente, las transmisiones inalámbricas constituyen una eficaz herramienta que permite la transferencia de voz, datos y vídeo sin la necesidad de cableado. Esta transferencia de información es lograda a través de la emisión de ondas de radio teniendo dos ventajas: movilidad y flexibilidad del sistema en general.

En general, la tecnología inalámbrica utiliza ondas de radiofrecuencia de baja potencia y una banda específica, de uso libre para transmitir, entre dispositivos.

1.42 CAMPOS DE UTILIZACIÓN

La tendencia a la movilidad y la difusión hacen que cada vez sean más utilizados los sistemas inalámbricos, y el objetivo es ir evitando los cables en todo tipo de comunicación, no solo en el campo informático sino en televisión, telefonía, seguridad, etc.

1.43 ALGUNOS PROBLEMAS ASOCIADOS CON LA TECNOLOGÍA INALÁMBRICA

Los hornos de microondas utilizan radiaciones en el espectro de 2,45 Ghz. Es por ello que las redes y teléfonos inalámbricos que utilizan el espectro de 2,4 Ghz. pueden verse afectados por la proximidad de este tipo de hornos, que pueden producir interferencias en las comunicaciones. Otras veces, este tipo de interferencias provienen de una fuente que no es accidental. Mediante el uso de un perturbador o inhibidor de señal se puede dificultar e incluso imposibilitar las comunicaciones en un determinado rango de frecuencias.

1.44 EQUIPO INALÁMBRICO

Son los equipos de punto de acceso que normalmente vienen con antena omni 2 Dbi, muchas veces desmontables, en las cuales se puede hacer enlaces por encima de los 500 metros y además se pueden interconectar entre sí. No debe haber obstáculos para que la señal sea excelente, ya que esto interfiere en la señal y puede haber problemas en la conexión.

1.45 COMUNICACIÓN MOVIL

En las comunicaciones móviles, en las que emisor o receptor están en movimiento, la movilidad de los extremos de la comunicación, excluye casi por completo la utilización de cables para alcanzar dichos extremos, por tanto utiliza básicamente la comunicación vía radio.

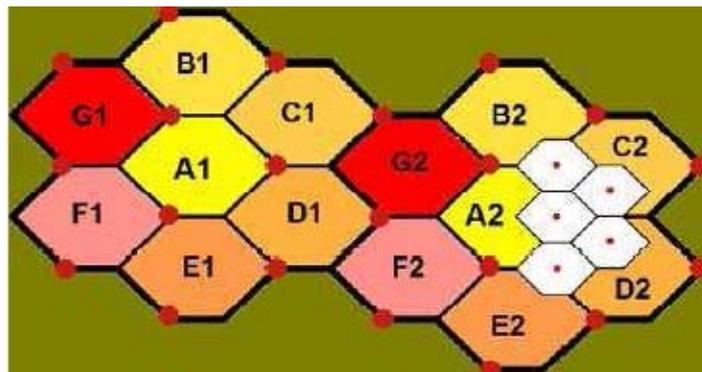
Esta se convierte en una de las mayores ventajas de la comunicación vía radio: la movilidad de los extremos de la conexión, otras bondades de las redes inalámbricas son el ancho de banda que

proporcionan, el rápido despliegue que conllevan al no tener que llevar a cabo obra civil, se puede efectuar un sinnúmero de actividades potenciadas por el gran auge de nuevos dispositivos suficientemente pequeños para ser transportados de un lugar a otro y que pueden ser utilizados durante su transporte, con capacidades de almacenamiento y procesamiento de datos como lo es un teléfono celular.

1.46 TELEFONÍA CELULAR

El concepto celular permite que un sistema de comunicaciones móviles pueda cubrir un área determinada con una densidad de usuarios variable, normalmente creciente, sin requerir más espectro radioeléctrico que el inicialmente asignado.

Ilustración 24 Celdas en un sistema de comunicaciones móviles.



Fuente: STALLING, W

El nombre de telefonía celular viene de la idea de dividir una zona geográfica, a la que se desea dar servicio, en áreas pequeñas llamadas células o celdas. El concepto celular se puede resumir en dos aspectos claves: re-uso de frecuencias y división de celdas, la Figura III.1 ilustra el concepto celular.

1.47 CELDA O CÉLULA

La celda o célula es una unidad básica de cobertura en que se divide un sistema celular, una celda se define como el área que cubre un transmisor o una colección de transmisores. El tamaño de las celdas está determinado por la potencia del transmisor y restricciones naturales y/o artificiales

impuestas por el sector a cubrir. La forma de las celdas puede ser cualquiera, pero se elige la forma hexagonal para una mejor descripción del sistema, se eligió la forma de hexágono porque proporciona la transmisión más efectiva aproximada a un patrón circular, mientras elimina espacios presentes entre los círculos adyacentes, las celdas dentro del área de cobertura se las identifica por un número llamado CGI.

El tipo de celdas a utilizar en un sistema celular depende de la demanda del servicio celular en la población y de parámetros físicos propios del área a la que se dará servicio, así se tienen diferentes tipos de celdas:

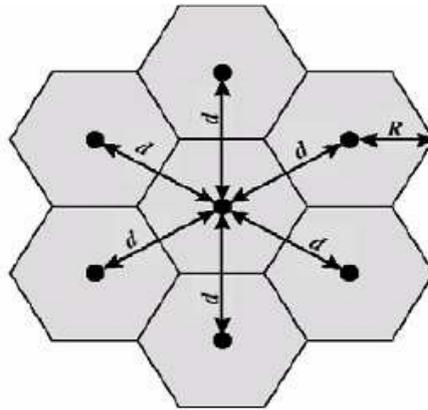
- Macro-celdas.-Las macro-celdas son celdas que permiten brindar servicio a áreas geográficas grandes, remotas y escasamente pobladas.
- Micro-celdas.-Las micro-celdas se usan para cubrir áreas densamente pobladas, o dividir celdas existentes en celdas más pequeñas.
- Celdas selectivas.-Las celdas selectivas son diseñadas para cubrir zonas con una cobertura menor a los 360 grados.
- Celdas de paraguas.-La celda de paraguas cubre varias micro-celdas, se la usa con el propósito de disminuir el número de handovers que se producen en estaciones móviles que cambian rápidamente de micro-celdas y disminuir el trabajo de la red. El nivel de potencia dentro de la celda de paraguas se incrementa en comparación con el nivel de potencia usado en las micro-celdas.

1.48 CLÚSTER O RACIMO

El clúster está formado por un conjunto de celdas, las cuales agrupan la totalidad de frecuencias disponibles para la red celular, es decir ningún canal puede ser re-usado dentro del clúster, el uso de las frecuencias en los clúster se realiza usando patrones de re-uso de frecuencia. Debe encontrarse un equilibrio en el número de celdas del clúster para evitar la interferencia que podría ocurrir entre los clúster vecinos, los clúster típicos agrupan 4, 7, 12 o 21 celdas.

El clúster con menor número de celdas posee el mayor número de canales por celda, en la Figura III.2 se muestra un ejemplo de un clúster de 7 celdas.

Ilustración 25 Clúster de 7 celdas.



Fuente: STALLING, W

1.49 COBERTURA CELULAR

Cobertura es la zona en la cual la estación móvil puede comunicarse con las estaciones transceptoras y viceversa, la cobertura se planifica tomando en cuenta las condiciones de transmisión en las que se puede encontrar la estación móvil, las cuales son determinadas por las características particulares del proyecto radioeléctrico. Para implementar un sitio de cobertura se analizan requerimientos tales como: área de servicio o comercialización, tipo de servicio, población y el crecimiento del área proyectada, únicamente puede hablarse de cobertura en sentido estadístico, esto implica que, las áreas que se representan teóricamente cubiertas, lo están en un determinado porcentaje de tiempo.

1.50 CAPACIDAD CELULAR

La capacidad celular es el tráfico total que puede soportar la red celular, la capacidad del sistema es función del número de canales utilizados, ancho de banda disponible, tamaño de las celdas y configuración de los clúster, este último parámetro está ligado a la relación de interferencia co-canal que el sistema sea capaz de soportar.

- ✓ Las bandas más importantes con aplicaciones inalámbricas, del rango de frecuencias que abarcan las ondas de radio, son la VLF (comunicaciones en navegación y submarinos), LF (radio AM de onda larga), MF (radio AM de onda media), HF (radio AM de onda corta), VHF (radio FM y TV), UHF (TV).

- ✓ Mediante las microondas terrestres, existen diferentes aplicaciones basadas en protocolos como Bluetooth o ZigBee para interconectar ordenadores portátiles, teléfonos u otros aparatos. También se utilizan las microondas para comunicaciones con radares (detección de velocidad u otras características de objetos remotos) y para la televisión digital terrestre.

- ✓ Las microondas por satélite se usan para la difusión de televisión por satélite, transmisión telefónica a larga distancia y en redes privadas, por ejemplo.

- ✓ Los infrarrojos tienen aplicaciones como la comunicación a corta distancia de los ordenadores con sus periféricos. También se utilizan para mandos a distancia, ya que así no interfieren con otras señales electromagnéticas, por ejemplo la señal de televisión. Uno de los estándares más usados en estas comunicaciones es el IrDA (*Infrared Data Association*). Otros usos que tienen los infrarrojos son técnicas como la tomografía, la cual permite determinar la temperatura de objetos a distancia.

1.51 SEÑALIZACIÓN

Señalización es toda comunicación dedicada a gestionar los recursos del sistema para permitir la comunicación, al hablar de comunicaciones celulares, se tiene la señalización asociada a la transmisión de radio y la relativa a la propia estructura de red, sólo se diferencian por el tipo de entidades a las que ponen en comunicación.

Funcionalmente se puede distinguir los siguientes tipos de señalización:

- Señalización destinada a la gestión de los recursos de radio.
- Señalización destinada a la gestión de la movilidad.
- Señalización destinada al establecimiento de la comunicación.

1.52 HANDOVER O TRANSFERENCIA

Es el proceso de pasar la comunicación de una estación móvil que ocupa un canal de radio a otro, con el propósito de no dejar caer una llamada y asegurar una relación señal a ruido adecuada durante todo el lapso de la llamada, cuando un usuario cruza el límite entre celdas adyacentes se realiza el proceso de Handover sin interrumpir la llamada o alertar al usuario.

1.53 USO DEL ESPECTRO EN SISTEMAS CELULARES

Un sistema de comunicaciones de radio se basa en la transmisión y recepción de señales transportadas por frecuencias portadoras, el plan de frecuencias organiza el uso de estas frecuencias portadoras, para cada operadora. Los sistemas celulares en Ecuador trabajan en la banda de 800 MHz para Movistar, 850 MHz para Porta y 1900Mhz para la operadora estatal Alegro (Tabla III.I).

Tabla 3 Espectro de frecuencias

| Operadora | Frecuencia (MHz) |
|------------------|-------------------------|
| Movistar | 800 |
| Porta | 850 |
| Alegro | 1900 |

Fuente: : STALLING, W

1.54 ROAMING O ITINERANCIA

Roaming es la capacidad que se ofrece a una estación móvil para poder registrarse en cualquier VLR de la misma u otra red celular. Se produce cuando un suscriptor móvil hace una llamada desde una celda a la que usualmente no pertenece, o también se produce cuando un operador al no proveer cobertura en un área de servicio procede a un convenio con otra operadora para que el usuario pueda registrarse en su VLR, denominándole usuario Roaming. El Roaming se da como consecuencia de acuerdos mutuos entre operadores de otras redes y a la vez de acuerdos entre el usuario y el

operador de la red, se pueden tener casos de Roaming nacional o internacional, el Roaming no puede ocurrir dentro de sistemas técnicamente incompatibles.

1.55 SUBDIVISIÓN DE CELDAS

Si el tráfico llega a acceder la capacidad del número de canales de la celda esta se subdivide de manera que la celda original contenga varias celdas de menor tamaño que cumplan con la norma de re-uso, la subdivisión de celdas permite al sistema ajustarse al crecimiento de la densidad de tráfico demandada sin incrementar el espectro que se utiliza, no obstante, la subdivisión de celdas implica también la degradación de la señal.

1.56 SISTEMA GSM

El sistema GSM es un sistema de telefonía celular para la transmisión digital de voz y datos con una gran calidad que se ha extendiendo por todo el mundo. Para poder utilizarlo, debe disponer de un teléfono celular compatible, abonarse al servicio GSM, utilizar un cable GSM y emplear un software de comunicaciones compatible con GSM.

1.57 ARQUITECTURA DE UNA RED GSM

Todas las redes GSM se pueden dividir en cuatro partes fundamentales y bien diferenciadas **La Estación Móvil o Mobile Station (MS)**.-Consta a su vez de dos elementos básicos que debemos conocer, por un lado el terminal o equipo móvil y por otro lado el SIM, con respecto a los terminales poco tenemos que decir ya que los hay para todos los gustos, lo que si tenemos que comentar es que la diferencia entre unos y otros radica fundamentalmente en la potencia que tienen que va desde los 20 vatios (generalmente instalados en vehículos) hasta los 2 vatios de nuestros terminales. El SIM es una pequeña tarjeta inteligente que sirve para identificar las características de nuestro terminal, esta tarjeta se inserta en el interior del móvil y permite al usuario acceder a todos los servicios que haya disponibles por su operador, sin la tarjeta SIM el terminal no nos sirve de nada porque no podemos hacer uso de la red. El SIM está protegido por un número de cuatro dígitos que recibe el nombre de PIN.

La mayor ventaja de las tarjetas SIM es que proporcionan movilidad al usuario ya que puede cambiar de terminal y llevarse consigo el SIM aunque todos sabemos que esto en la práctica en muchas ocasiones no resulta tan sencillo. Una vez que se introduce el PIN en el terminal, el terminal va a ponerse a buscar redes GSM que estén disponibles y va a tratar de validarse en ellas, una vez que la red (generalmente la que tenemos contratada), ha validado nuestro terminal el teléfono queda registrado en la célula que lo ha validado.

La Estación Base o Base Station Subsystem (BSS).-Sirve para conectar a las estaciones móviles con los NSS, además de ser los encargados de la transmisión y recepción. Como los MS también constan de dos elementos diferenciados: La Base Transceiver Station (BTS) o Base Station y la Base Station Controller (BSC).

La BTS consta de transceivers y antenas usadas en cada célula de la red y que suelen estar situadas en el centro de la célula, generalmente su potencia de transmisión determinan el tamaño de la célula. Los BSC se utilizan como controladores de los BTS y tienen como funciones principales las de estar al cargo de los handovers, los frequency hopping y los controles de las frecuencias de radio de los BTS.

1.57.1.1.1 Subsistema de Conmutación y Red o Network and Switching Subsystem (NSS).

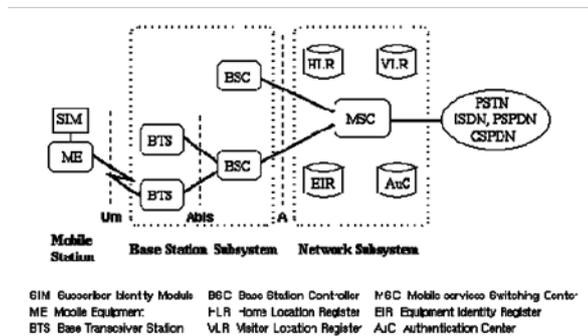
Este sistema se encarga de administrar las comunicaciones que se realizan entre los diferentes usuarios de la red, para poder hacer este trabajo la NSS se divide en siete sistemas diferentes, cada uno con una misión dentro de la red:

- ✓ Mobile Services Switching Center (MSC): Es el componente central del NSS y se encarga de realizar las labores de conmutación dentro de la red, así como de proporcionar conexión con otras redes.
- ✓ Gateway Mobile Services Switching Center (GMSC): Un gateway es un dispositivo traductor (puede ser software o hardware que se encarga de interconectar dos redes haciendo que los protocolos de comunicaciones que existen en ambas redes se entiendan. Bien, la misión del GMSC es esta misma, servir de mediador entre las redes de telefonía fijas y la red GSM .
- ✓ Home Location Register (HLR): El HLR es una base de datos que contiene información sobre los usuarios conectados a un determinado MSC. Entre la información que almacena el HLR tenemos fundamentalmente la localización del usuario y los servicios a los que tiene acceso. El HLR funciona en unión con el VLR que vemos a continuación.
- ✓ Visitor Location Register (VLR): contiene toda la información sobre un usuario necesaria para que dicho usuario acceda a los servicios de red. Forma parte del HLR con quien comparte funcionalidad.
- ✓ Authentication Center (AuC): Proporciona los parámetros necesarios para la autenticación de usuarios dentro de la red; también se encarga de soportar funciones de encriptación.
- ✓ Equipment Identity Register (EIR): También se utiliza para proporcionar más seguridad en las redes GSM pero a nivel de equipos válidos. La EIR contiene una base de datos con todos los terminales que son válidos para ser usados en la red. Esta base de datos contiene los International Mobile Equipment Identity o IMEI de cada terminal, de manera que si un determinado móvil trata de hacer uso de la red y su IMEI no se encuentra localizado en la base de datos del EIR no puede hacer uso de la red.
- ✓ GSM Interworking Unit (GIWU): sirve como interfaz de comunicación entre diferentes redes para comunicación de datos.

1.57.2 Los Subsistemas de soporte y Operación

Operation and Support Subsystem (OSS) los OSS se conectan a diferentes NSS y BSC para controlar y monitorizar toda la red GSM. La tendencia actual en estos sistemas es que, dado que el número de BSS se está incrementando se pretende delegar funciones que actualmente se encargan de hacerlas el subsistema OSS a los BTS de modo que se reduzcan los costes de mantenimiento del sistema.

Ilustración 26 Operation and Support Subsystem



Fuente: STALLING, W

1.57.3 Mensajes de Texto

La tecnología GSM (Global System for Mobile Communications) de radio celular digital, sobre la que opera SMS, supuso grandes mejoras respecto a los sistemas analógicos que la precedieron; consiguiendo un mejor rendimiento, calidad y seguridad, aprovechar mejor el espectro radioeléctrico, y tener amplias coberturas nacionales e internacionales a través del roaming. Este sistema alcanza una velocidad máxima de transferencia de datos de 9,6 Kbps, lo cual limita el tipo de servicios que se pueden ofrecer. SMS consiste en el intercambio de mensajes únicamente de texto sin formato (es decir, no se pueden seleccionar diferentes fuentes, tamaños y estilos) entre teléfonos móviles garantizando

al usuario el correcto envío de la información, aunque no posibilita el envío de mensajes en tiempo real.

SMS permite típicamente un máximo de 160 caracteres por mensaje (70 de utilizar otro alfabeto distinto del latino, como el chino o el árabe). Existen ya varios terminales en el mercado que permiten enviar varios mensajes concatenados, pudiendo escribir textos más largos de hasta 480 caracteres, no obstante esta funcionalidad únicamente es implementada en el terminal; es decir, la persona que envía un mensaje concatenado en realidad está enviando varios SMS.

CAPÍTULO III

1.58 COMANDOS AT

1.59 INTRODUCCIÓN

Los comandos AT los cuales se pueden utilizar con Nokia GSM (incluyendo también DCS1800 y PCS1900) y productos Wideband de CDMA (WCDMA) disponibles después del otoño 2001, por ejemplo, los productos siguientes: Nokia 8310, Nokia 6310, Nokia 7650, Nokia 8910, y Nokia D211. Una corta descripción, la sintaxis, los posibles valores a setear, y las respuestas a los comandos AT se presentan aquí. Tomado del Set de Comandos para Nokia GSM Versión 1.0 lo más representativo buscando con esto ayudar a futuras tesis en un campo como este poco explorado.

Algunos de los comandos AT no son soportados por todos los productos Nokia o por todos los operadores. Dar un comando que no sea soportado por el producto causa una respuesta de error. Algunos productos de Nokia no necesariamente soportan todos los parámetros del comando y al usarlos no todos causan una respuesta de error.

Los computadores utilizan los comandos AT para comunicarse con los módems. La mayoría de las aplicaciones de comunicaciones, sin embargo, tienen un interfaz que utilice ocultos comandos AT. Los comandos AT pueden ser publicados vía una aplicación de comunicaciones. Cuando el software en el producto de Nokia ha recibido el comando AT, responde con un mensaje que se visualice en la pantalla del dispositivo usado, que puede también ser el teléfono móvil.

1.60 SINTAXIS DEL COMANDO AT

El prefijo "AT" o "at" debe ser agregado al principio de cada línea de comando.

Varios comandos AT se pueden pulsar en la misma línea, y en tales casos el prefijo

"AT " o "at " se necesita solamente una vez, al principio de la línea de comando.

La marca < n > usada en la sintaxis del comando es el valor que se fija adentro como parte del comando. Si el valor es opcional, es incluido en los corchetes. Cuando un valor se setea con un comando AT, el seteo es válido hasta que lo vuelva a cambiar o reinicie el equipo.

1.61 COMANDOS DE MODEM

Para ver los valores predeterminados, amplíe la parte derecha de la ventana.

- La arquitectura del procesador sigue el modelo Harvard permitiendo a la CPU acceder simultáneamente a las dos memorias.
- La segmentación permite al procesador realizar al mismo tiempo la ejecución de una instrucción y la búsqueda del código de la siguiente. De esta forma se puede ejecutar cada instrucción en un ciclo.
- Todas las instrucciones de los microcontroladores de la gama baja tienen una longitud de 12 bits. Las de la gama media tienen 14 bits y más las de la gama alta. Esta característica es muy

ventajosa en la optimización de la memoria de instrucciones y facilita enormemente la construcción de ensambladores y compiladores.

- Todas las instrucciones son ortogonales es decir cualquier instrucción puede manejar cualquier elemento de la arquitectura como fuente o como destino.
- Arquitectura basada en un “banco de registros”, esto significa que todos los objetos del sistema (puertas de E/S, temporizadores, posiciones de memoria, están implementados físicamente como registros.

1.61.1 Comandos generales

- a) AT+CGMI: Identificación del fabricante
- b) AT+CGSN: Obtener número de serie
- c) AT+CIMI: Obtener el IMSI.
- d) AT+CPAS: Leer estado del modem

1.61.2 Comandos del servicio de red

- a) AT+CSQ: Obtener calidad de la señal
- b) AT+COPS: Selección de un operador
- c) AT+CREG: Registrarse en una red
- d) AT+WOPN: Leer nombre del operador

1.61.3 Comandos de seguridad

- a) AT+CPIN: Introducir el PIN
- b) AT+CPINC: Obtener el número de reintentos que quedan
- c) AT+CPWD: Cambiar password

1.61.4 Comandos para la agenda de teléfonos

- a) AT+CPBR: Leer todas las entradas

- b) AT+CPBF: Encontrar una entrada
- c) AT+CPBW: Almacenar una entrada
- d) AT+CPBS: Buscar una entrada

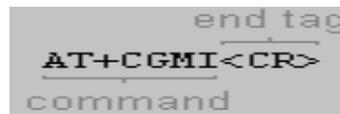
1.61.5 Comandos para SMS

- a) AT+CPMS: Seleccionar lugar de almacenamiento de los SMS
- b) AT+CMGF: Seleccionar formato de los mensajes SMS
Modo = 0 indica formato de mensajes en modo PDU Modo = 1
indica formato de mensajes en modo TEXTO
- c) AT+CMGR: Leer un mensaje SMS almacenado
- d) AT+CMGL: Listar los mensajes almacenados
- e) AT+CMGS: Enviar mensaje SMS
- f) AT+CMGW: Almacenar mensaje en memoria
- g) AT+CMSS: Enviar mensaje almacenado
- h) AT+CSCA: Establecer el Centro de mensajes a usar
- i) AT+ WMSC: Modificar el estado de un mensaje

El envío de comandos AT requiere la siguiente estructura:

1.61.6 Petición:

Imagen 26.1: comandos AT empleados



Fuente: manual sim900

<CR> ... Carriage return

1.61.7 Respuesta correcta:

Imagen 26.2: Respuesta correcta Iperterminal Windows

```
response
  <CR><LF>Nokia Mobile Phones<CR><LF>
  <CR><LF>OK<CR><LF>
start sequence end sequence
```

Fuentes: Iperterminal de Windows

<CR> ... Carriage return

<LF> ... Line feed

1.61.8 Respuesta incorrecta:

Imagen 26.3: Captura de pantalla iperterminal de Windows

```
<CR><LF>ERROR<CR><LF>
start sequence end sequence
```

Fuente Iperterminal de Windows

1.62 HERRAMIENTAS PARA EL DESARROLLO DE APLICACIONES

Para el desarrollo de las diferentes funciones que realiza el microcontrolador existen diferentes programas que facilitan el desarrollo de las mismas, las principales herramientas de ayuda al desarrollo de sistemas basados en microcontroladores son:

1.63 ENSAMBLADOR

La programación en lenguaje ensamblador permite desarrollar programas muy eficientes, ya que otorga al programador el dominio absoluto del sistema, aunque este lenguaje por su larga forma de programación ha sido reemplazo por lenguajes de alto nivel que permiten realizar la programación de forma más rápida y eficiente. Al ser el primer tipo de lenguaje que se utilizó

sigue siendo usado para la enseñanza y aprendizaje del manejo de microcontroladores, ya que el programador decide qué tipo de lenguaje utilizar para el desarrollo de su aplicación.

1.64 COMPILADOR

La programación en un lenguaje de alto nivel (como el C) permite disminuir el tiempo de desarrollo de una determinada aplicación. No obstante, si no se programa con cuidado, el código resultante puede ser mucho más ineficiente que el programado en ensamblador.

Es por ello que se utiliza un determinado compilador que depende del lenguaje de alto nivel con el que estemos programando, el cual nos ayuda a determinar si las sentencias de dicho código están correctamente escritas o caso contrario procede a señalar el error.

1.65 SIMULADOR

Son capaces de ejecutar en un PC programas realizados para el microcontrolador. Los simuladores permiten tener un control absoluto sobre la ejecución de un programa, siendo ideales para la depuración de los mismos. Su gran inconveniente es que es difícil simular la entrada y salida de datos del microcontrolador. Tampoco cuentan con los posibles ruidos en las entradas, pero, al menos, permiten el paso físico de la implementación de un modo más seguro y menos costoso, puesto que ahorraremos en grabaciones de chips para la prueba.

1.66 PROGRAMADOR

Es un dispositivo físico o hardware que es conectado a un PC con el cual se permite grabar en el microcontrolador el programa desarrollado, el diseño del mismo depende del software que se utilice para su manejo.

Existen diferentes tipos de programadores los mismos que tienen conexión a la PC mediante el puerto paralelo, el puerto serial o el puerto USB necesarios para la comunicación con el microcontrolador.

1.67 PAQUETES IDE

Actualmente existen paquetes de software denominados "Entornos de Desarrollo Integrado", IDE, que suelen funcionar bajo Windows y que incluyen editores de texto para el ensamblador o el compilador, permiten la simulación del programa y también pueden integrar el control de emuladores y programadores de dispositivos.

1.68 ENTORNO DE DESARROLLO EN ARDUINO

El entorno de desarrollo en Arduino (IDE) es el encargado de la gestión de la conexión entre el PC y el hardware de Arduino con el fin de establecer una comunicación entre ellos por medio de la carga de programas. Como podemos ver en la figura, el IDE de Arduino se compone de:

1.68.1 UN EDITOR DE TEXTO

Donde escribir el código del programa.

1.68.2 UN ÁREA DE MENSAJES

A través del cual el usuario tendrá constancia en todo momento de los procesos que se encuentren en ejecución, errores en código, problemas de comunicación, etc.

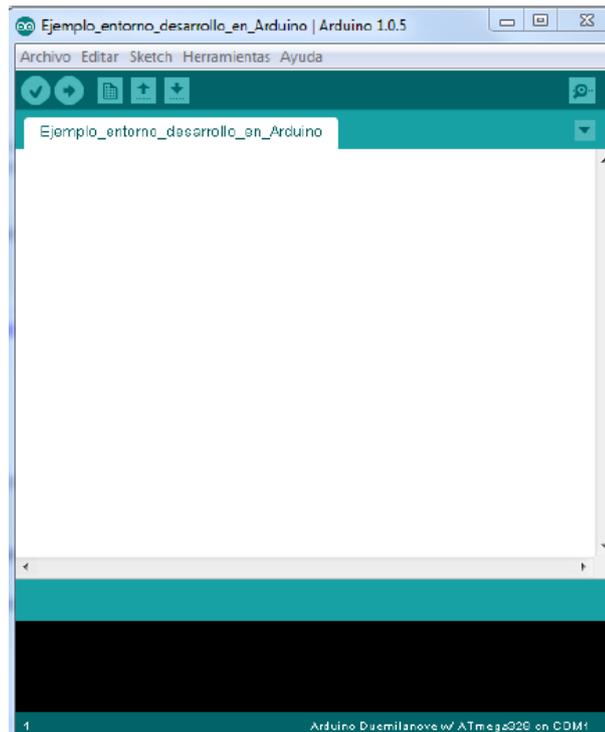
1.68.3 UNA CONSOLA DE TEXTO

Mediante la que podremos comunicarnos con el hardware Arduino y viceversa.

1.68.4 UNA BARRA DE HERRAMIENTAS

Donde podremos acceder a una serie de menús y a los botones con acceso directo a las principales funcionalidades de Arduino.

Ilustración 27 Entorno de desarrollo



Fuente: Software Arduino.

A través de la IDE de Arduino, podemos escribir el código del programa software y crear lo que se conoce por "sketch" (programa). ¿Por qué lo llamamos sketch y no programa? Pues porque el IDE de Arduino viene de Processing, y en este lenguaje de programación enfocado al mundo gráfico, cada código es considerado un boceto, en inglés "sketch".

El sketch permite la comunicación con la placa Arduino. Estos programas son escritos en el editor de texto, el cual admite las posibilidades de cortar, pegar, buscar y reemplazar texto.

En el área de mensajes se muestra, tanto la información mientras se cargan los programas, como los posibles errores que tengamos a la hora de compilar, ya sea por problemas en el código del

sketch, por fallo en la detección de nuestro Arduino en el puerto USB, o por cualquier otro problema que sea detectado.

La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones.

Desde la barra de herramientas tenemos acceso directo a las principales funcionalidades que ofrece el IDE de Arduino, como por ejemplo: verificar el proceso de carga, crear un nuevo sketch, abrir un sketch ya existente, guardar los programas, abrir el Monitor Serial, etc.

A continuación pasamos a describir la utilidad de cada uno de los iconos que aparecen en la pantalla principal del entorno de desarrollo de Arduino:



“Verificar” .- Esta funcionalidad se encarga de verificar el código del sketch en busca de posibles errores. A través del área de mensajes se le notificará al usuario el resultado de dicha verificación. En el caso de que se detecten errores en el código, éstos se detallarán junto con el número de línea en la que han sido detectados. Sólo cuando la comprobación resulta libre de errores podremos proceder a la carga del código en nuestra placa Arduino.



“Cargar” .- Permite compilar el código del sketch y lo carga en Arduino. Cuando la carga a terminado se informa al usuario a través del área de mensajes, y podremos proceder a la apertura del monitor serial.



“Nuevo” .- Para la creación de un nuevo sketch. Abre una nueva hoja de texto donde escribiremos el código correspondiente al sketch.



“Abrir” .- Permite abrir un sketch ya existente que ha sido previamente guardado.

También puedes abrir cualquiera de los sketches que trae instalados por defecto el IDE de Arduino.

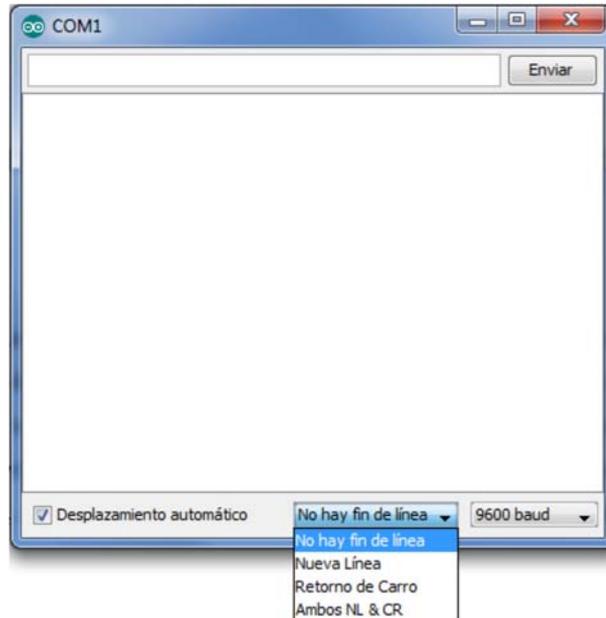


“Guardar” .- Esta funcionalidad nos permite almacenar el sketch que estemos desarrollando en ese momento. Te permite elegir la ruta en la que quieres guardarlo, y te crea automáticamente una carpeta con el mismo nombre que le des al sketch, guardando éste dentro de la misma.



“Monitor Serial” .- Al pinchar sobre este icono, el entorno de desarrollo de Arduino abre una nueva ventana a través de la cual podemos ver la comunicación establecida por el puerto serie entre la placa Arduino y el PC durante la ejecución del programa. Contiene una barra de escritura mediante la que podemos comunicarnos con Arduino a través de su puerto serie, por ejemplo, para seleccionar distintas opciones que contemple un posible menú creado por el usuario dentro de un código, o para enviar directamente comandos AT a una shield GPRS/GSM que tengamos montada sobre el Arduino. También contempla la opción de seleccionar el envío de algunos caracteres junto con el texto que introduzcamos en la barra de entrada del mismo, como el carácter de nueva línea, retorno de carro, o los dos. En la figura 3.17 podemos ver la pantalla correspondiente al Monitor Serial y la pestaña desplegable en la que podemos seleccionar las distintas opciones referentes a los caracteres de fin de línea.

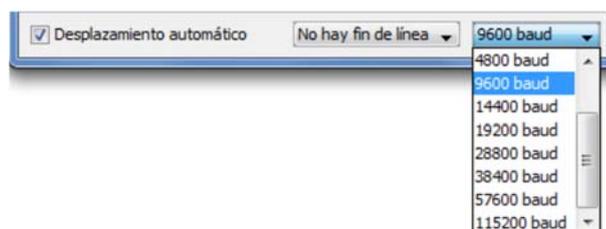
Ilustración 28 Monitor Serial.



Fuente: Administrador de dispositivos de Windows .

Dentro del Monitor Serial disponemos de otra pestaña para establecer la tasa de baudios (Baudrate), que marca el número de unidades de señal transmitidas por segundo. Este valor ha de estar sincronizado con el baudrate en el que esté trabajando el Arduino, el cual puede ser establecido en el código del sketch mediante el comando `Serial.begin("valor del baudrate")`, o de no ser así, se establecerá un valor por defecto. Si Monitor Serial y Arduino no están sincronizados con la misma tasa de baudios, la información que aparezca en la ventana será completamente ilegible. En la figura 3.18 aparece desplegada la pestaña para la selección de los distintos valores de baudrate disponibles.

Ilustración 29 Valores baudrate



Fuente: Software Arduino

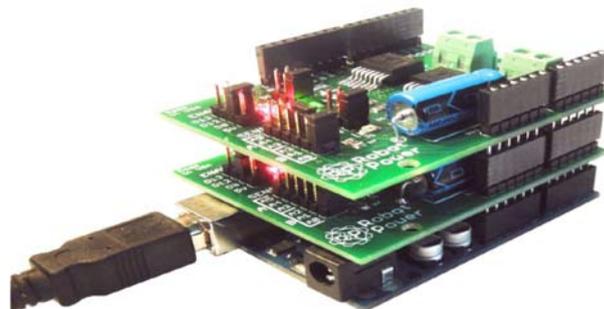
Dentro de los menús, cabe mencionar la existencia de librerías, que pueden proporcionar funcionalidades adicionales para la utilización en sketches, por ejemplo para trabajar con hardware o manipular datos. Para utilizar una librería dentro de un sketch, debemos declararla previamente. Para ello nos iremos al menú “sketch” , y seleccionaremos la opción importar librerías. Dentro buscaremos la librería que sea de nuestro interés y la importaremos al sketch, insertando una sentencia de tipo `#include` al comienzo del mismo. Se debe tener en cuenta que al cargar un código que incluya librerías, éstas también se vuelcan en la placa junto con el resto del sketch, incrementando la ocupación del programa y reduciendo el espacio disponible en la memoria de Arduino.

1.69 SHIELDS GPRS/GSM

Si queremos ampliar las funcionalidades de nuestra plataforma Arduino, siempre podemos recurrir a una gran variedad de shields compatibles prácticamente con cualquiera de sus modelos. De este modo, podemos dotar al dispositivo de funciones adicionales dedicadas específicamente a ofrecer algún tipo de servicio concreto.

Un shield es un módulo de expansión en forma de placa impresa que se puede conectar a la parte superior de la placa Arduino para ampliar sus capacidades, permitiendo además ser apiladas unas encima de otras manteniendo un diseño modular, tal como podemos ver en la Figura 3.19.

Ilustración 30 Diseño Modular



Fuente: Manual de modulación

En nuestro caso, buscamos una shield que nos permita utilizar los sistemas de comunicaciones móviles para poder interactuar a distancia con nuestra plataforma.

Navegando por internet podemos encontrar varias shields que han sido diseñadas específicamente para ofrecer servicios a través de los sistemas GSM, GPRS, 3G, o una combinación de los mismos. Además, son perfectamente compatibles con nuestra placa Arduino. [13]

1.70 MÓDULO GPRS QUADBAND PARA ARDUINO:

Esta placa integra un módulo HILO SAGEM que nos permite ofrecer los servicios de un módem GPRS a través de nuestro Arduino. Con esta shield, podemos enviar SMS o hacer llamadas perdidas a otros dispositivos móviles. Es necesaria la conexión de una antena externa para poder establecer las comunicaciones. El precio de esta shield ronda los 106 USD.

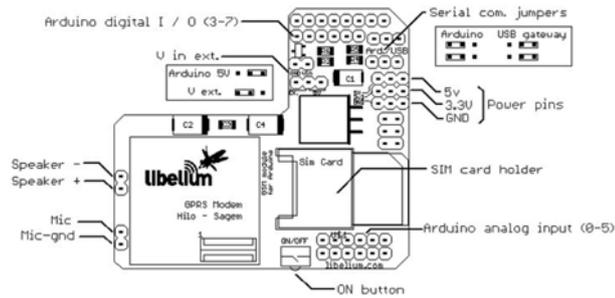
Ilustración 31 Placa HILO SAGEM.



Fuente: Placa HILO SAGEM Internet

En el esquema reflejado en la figura 3.21 aparece indicada la función de cada uno de los puertos que componen la placa:

Ilustración 32 Puertos Serial y Arduino



Fuente:ertos Serial y Arduino

En principio se trata de un módulo que se adapta perfectamente a lo que estábamos buscando, pero más adelante veremos que existen placas más económicas que ofrecen exactamente las mismas funcionalidades, por lo que finalmente quedó descartada.

1.71 MÓDULO GPRS+GPS QUADBAND PARA ARDUINO/RASPBERRY PI (SIM908)

Este es el último modelo de la shield GPRS para Arduino. Gracias a que cuenta con un módulo SIM908 integrado en la propia placa, ofrece la posibilidad de utilizar la tecnología GPS para posicionamiento en tiempo real, resultando muy útil para aquellas aplicaciones en las que necesitamos conocer la ubicación de nuestro dispositivo. En la figura 3.22 se adjunta una imagen de dicha shield

Ilustración 33 Shield GPRS para Arduino



Fuente: Shield GPRS

1.72 MÓDULO 3G/GPRS+GPS PARA ARDUINO/RASPBERRY PI

Este es el modelo más completo entre todas las shields GPRS disponibles. A parte del sistema GPRS, gracias a su módulo SIM5218, integra también servicios 3G y tecnología GPS.

Su precio es bastante elevado, unos 180€, pero es cierto que admite más funcionalidades en comparación con el resto de shields que hemos visto hasta ahora, incluso permite la conexión de una cámara para la toma de imágenes. En la figura 3.24 podemos ver el aspecto que presenta esta shield.

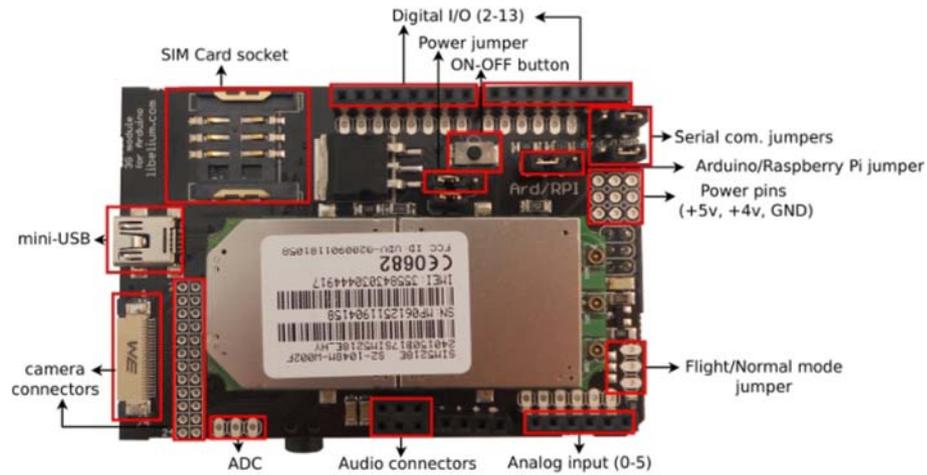
Ilustración 35 Aspecto del Shield.



Fuente: Shield Raspberry.PI

A continuación se detalla la estructura de ésta placa:

Ilustración 36 Diagrama de puertos y conexiones del módulo 3G/GPRS+GPS.



Fuente: módulo 3G/GPRS+GPS.

A pesar de que se trata de la placa con mayores prestaciones de entre todas las que hemos podido encontrar, no tiene ningún sentido decantarnos por ella, pues aparte de que su precio está muy por encima del presupuesto inicial fijado para el proyecto (no más de 100 USD), incluye tecnologías de las cuales no vamos a hacer ningún uso (GPS y 3G). Desestimamos por tanto, la opción de adquirir esta shield.

1.73 MÓDULO GPRS/GSM QUADBAND PARA ARDUINO (SIM900)

Esta shield puede convertir nuestra placa Arduino en un plataforma capaz de ofrecer conectividad GPRS/GSM. Integra un módulo SIM900 que nos permite establecer llamadas con otros dispositivos móviles, enviar SMS, incluso la comunicación de datos a través de los protocolos TCP, UDP, HTTP o FTP. Su precio resulta muy interesante, ronda los 95 usd es por esta razón y sus prestaciones que elegimos esta Shell para nuestro proyecto de tesis. En la figura 3.26 se adjunta una imagen de dicho módulo:

Ilustración 37 Módulo GPRS para Arduino SIM900



Fuente: Modulo SIM900

CAPÍTULO IV

1.74 DISEÑO Y CONSTRUCCIÓN

En este capítulo se da a conocer el conjunto de especificaciones que integran el sistema, permitiendo su implementación.

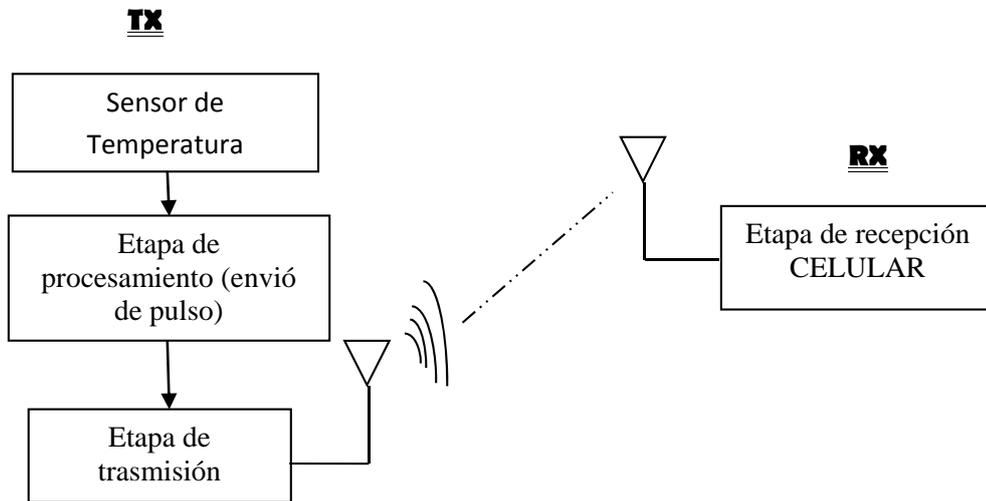
En este capítulo se presenta el diseño de hardware como software para la implementación del sistema de adquisición de señales biométricas mediante mensajes sms, Este sistema es autónomo el mismo que procesara y envía la información biométrica (temperatura, corporal de una persona), cuando esta exceda de sus parámetros normales que esta entre los 36 y 38°C hacia el celular del médico tratante, el sistema estará cada minuto tomando un dato del sensor.

El sistema recolecta toda la cantidad posible de datos para informarle de una mejor manera al medico tratante en un mensaje de texto de 160 caracteres como máximo, y lo que se incluye en un mensaje de texto son 4 datos de cada paciente (Numero de piso, Numero de Habitación, numero de Cama, y código de paciente).

1.75 ESQUEMA GENERAL DE FUNCIONAMIENTO

El diseño del sistema de adquisición de señales biométricas empezó con la formulación de una idea central y de ahí la conformación de un diagrama de bloques que sintetiza el funcionamiento y desarrollo del Hardware.

Ilustración 38 Diagrama General del Sistema



Fuente: Patricio Hidalgo

1.76 SENSORES

En esta etapa se integra el sensor de temperatura

1.77 ACONDICIONAMIENTO DE SEÑAL

No se puede hablar del sensor, como componente electrónico básico, sin ver cómo se pueden adaptar a un sistema de adquisición y control. Por lo que se tendrá que analizar la adaptación de este sensor para el sistema de recolección de información biométrica y sobre todo tomar en cuenta que el comportamiento de un sensor es no lineal o en su mayoría de sensores, estos adaptadores o acondicionadores de señal pueden ser los amplificadores operacionales de instrumentación en sus diferentes estructuras de montaje, resistencias en paralelo, divisores de voltaje, etc. Una vez que obtenemos una curva lineal, lo capturamos con el microcontrolador pasando por el ADC, convirtiendo esta señal de analógica a digital.

1.77.1 Acondicionamiento del Sensor de Temperatura

La siguiente fórmula convierte el valor obtenido del pin análogo al que se encuentra conectado nuestro LM35 a grados Celsius:

$$C = (500.0 * \text{analogRead}(\text{PIN_SENSOR}) * 100.0) / 1023$$

```
float centi()
{
  int dato;
  float c;
  dato=analogRead(A0);
  c=(500.0*dato)/1023;
  return(c);
}
```

Fuente: Código Fuente Proyecto

Si deseamos convertir este valor a la escala Fahrenheit debemos multiplicar el valor obtenido en C por 1.8

$$F = C * 1.8$$

Fuente: Código Fuente Proyecto

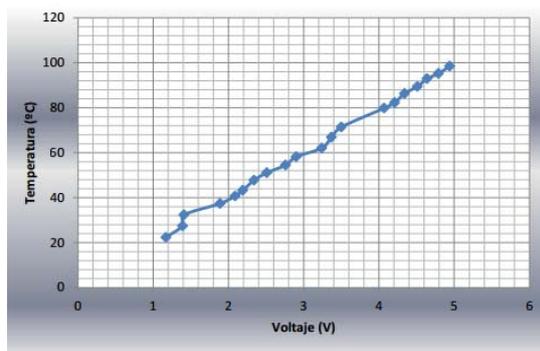
El negativo va a nuestro pin GND de Arduino, el positivo a 5v y el pin OUT se conecta a una de las entradas análogas (en el sistema utilizamos la entrada A0).

En el Sensor LM35 debemos saber que de izquierda a derecha sus pines corresponden a 5v IN, salida análoga y GND.

1.77.2 Acondicionamiento de la señal analógica del sensor LM35

Se obtiene la ecuación que describe el comportamiento del LM35 en función de la temperatura, para ello utilice la tabla de valores del programa y realice la grafica de los valores obtenidos.

Ilustración 39 Voltaje



Fuente: Patricio Hidalgo

Tabla 4 voltaje del sensor LM35 con su temperatura asociada

| Voltaje (V) | Temperatura °C |
|-------------|----------------|
| 1,17 | 22,34 |
| 1,39 | 27,45 |
| 1,41 | 32,37 |
| 1,52 | 34,89 |
| 1,57 | 36,15 |
| 1,63 | 37,14 |
| 1,89 | 37,4 |
| 1,94 | 38,662 |
| 2,09 | 40,74 |
| 2,19 | 43,2 |
| 2,34 | 47,72 |
| 2,51 | 51,07 |

Fuente: Patricio Hidalgo

La gráfica de entrada/salida ilustra el comportamiento lineal del sensor, tal como la hoja de datos predice. Aproximemos la gráfica a la de una ecuación lineal determinada por dos puntos de la recta

La anterior es la ecuación de comportamiento del sensor, donde x es el voltaje en V y Y es la

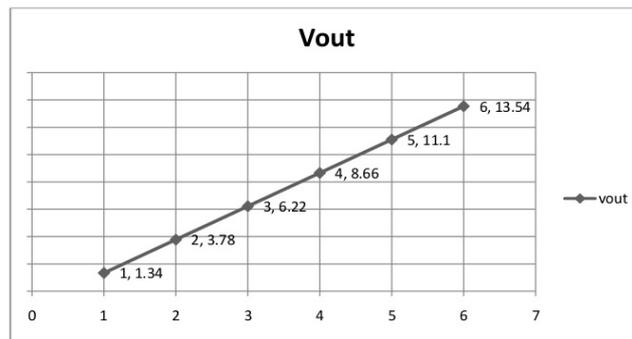
$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{98.44 - 22.34}{4.94 - 1.17} = 20.186$$
$$(y_2 - y_1) = m(x_2 - x_1)$$
$$(y_2 - 86.31) = 20.186(x_2 - 4.34)$$
$$y = 20.186x - 1.297$$

temperatura. La ecuación donde la temperatura es la entrada y el voltaje la salida, es:

$$X = 0.04954Y + 0.064252$$

Se Obtiene el voltaje entregado por el LM35 en los instantes de las temperaturas máximas registradas en sus tablas. Que observo en la respuesta de alta impedancia respecto al de baja impedancia. Es casi el mismo el voltaje, hay de diferencia de 5 a 10 mV, pero el tiempo de respuesta es un poco diferente el muestreo se ve afectado por la lectura de sistemas de altas impedancias.

Ilustración 40 VOUT



Fuente: Patricio Hidalgo

1.78 PROCESAMIENTO

Esta es la parte más importante, este es el cerebro de todo el sistema aquí es donde se encuentra el modulo Arduino UNO (ver Figura IV.10). Parte medular del sistema, siendo el encargado de controlar los periféricos de entrada y salida como LCD y pulsadores, establecer la comunicación

con el sensor LM35 y cuando exceda la temperatura establecida como normal envíe Instrucciones de comandos AT al modem Sim900 para enviar un mensaje de texto.

Este es el encargado de recolectar todos los datos, convertir de datos analógicos a digitales del sensor de temperatura, como se puede apreciar las instrucciones en Arduino para el proceso de conversión de dato analógico a digital.

```
float centi()  
{  
  int dato;  
  float c;  
  dato=analogRead(A0);  
  c=(500.0*dato)/1023;  
  return(c);  
}
```

Fuente: Segmento de código fuente proyecto

Instrucciones para monitorear la temperatura corporal en el rango normal de 36 a 38 grados centígrados e imprimirlos en el LCD

```
void loop()
{
  float Centigrados=centi();
  lcd.setCursor(5,1);
  lcd.print(Centigrados);
  if(Centigrados >= 21 && Centigrados<38){
    lcd.setCursor(10,1);
    lcd.print("NORMAL");
  }else{
    lcd.print("  ");
  }
}
```

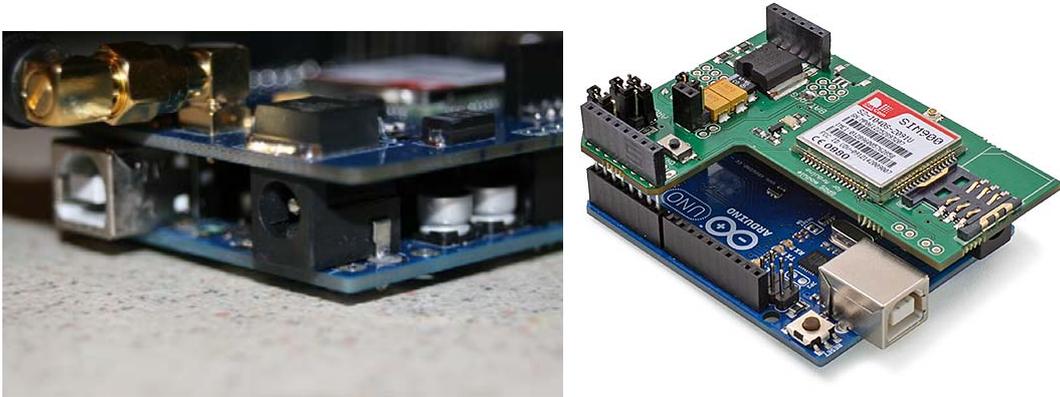
Fuente: Segmento de código fuente proyecto

Instrucciones para encender el modem sim900 mediante software

```
void SIM900power ( )
{
  digitalWrite ( 9 , HIGH ) ;
  delay ( 1000 ) ;
  digitalWrite ( 9 , LOW ) ;
  delay ( 2000 ) ;
}
```

Fuente: Segmento de código fuente proyecto

Ilustración 42 DISEÑO MPODULAR



Fuente: Patricio Hidalgo

Instrucciones para envío de mensajes SMS mediante el modem sim900 hacia en celular del médico tratante establecido en las siguientes líneas de código.

```
void SendTextMessage()
{
  mySerial.print("AT+CMGF=1\r");
  delay(100);
  mySerial.println("AT + CMGS = \"+593984557824\"");
  delay(100);
  mySerial.println("TEMPERATURA MUY ALTA ");
  delay(100);
  mySerial.println((char)26);
  mySerial.println((char)30);
  mySerial.println((char)34);
  mySerial.println((char)38);
  delay(100);
  mySerial.println();
}
```

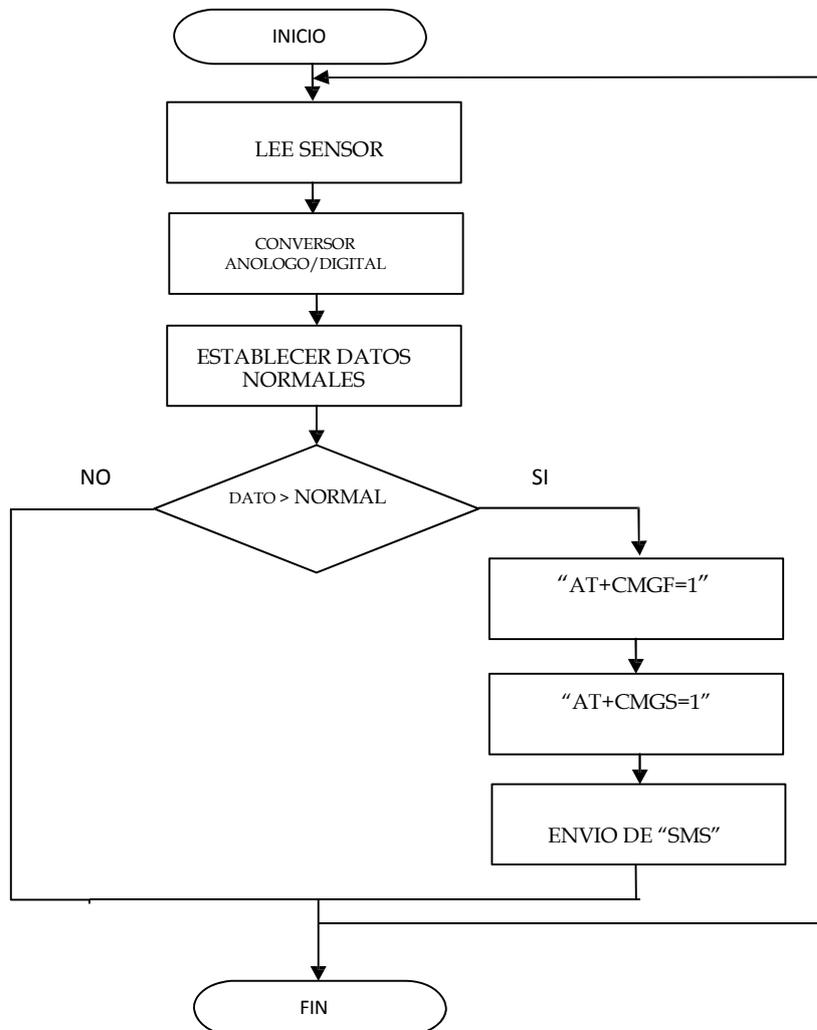
Fuente: Segmento de código fuente proyecto

1.81 ALGORITMO DE ENVÍO DE UN MENSAJE DE TEXTO

Este es el algoritmo que nos indica de cómo se está enviando un mensaje de texto como nos indica el diagrama de flujo utilizando instrucciones de comunicación serial para el envío de comandos AT hacia el teléfono celular, el sistema está constantemente obteniendo los datos del sensor, convirtiéndolos de datos

analógicos a digitales, estos datos son almacenados en variables esperando hasta que se cumpla el tiempo para enviar un mensaje de texto hacia el celular del médico tratante.

Diagrama de flujo de envío de mensaje de texto



Fuente: Patricio Hidalgo

CAPÍTULO V

1.82 IMPLEMENTACIÓN Y RESULTADOS EXPERIMENTALES

1.83 IMPLEMENTACIÓN

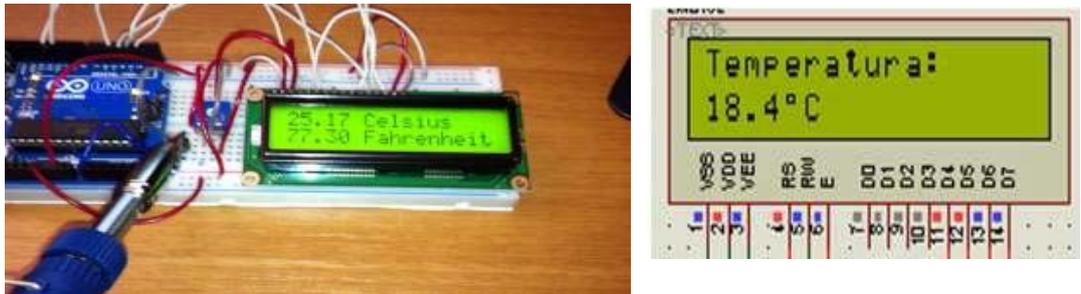
En el presente capítulo se pretende realizar un análisis técnico de los datos recolectados y expuestos anteriormente. Para eso, primeramente, se explicara la metodología de investigación utilizada así como los instrumentos que permitan validar la hipótesis. Existen diferentes instrumentos de medida que pueden referenciar el rendimiento del equipo implementado, cada uno de ellos basados en distintos parámetros, determinando así las potencialidades de la misma. Cada uno de los parámetros a medir mantiene ciertos valores límites recomendados por los organismos especializados en el campo de la medicina para garantizar una exitosa, segura y confiable lectura de datos. A partir de estos valores puede existir una cierta tolerancia alrededor de los mismos, lo cual se revisara a través del análisis respectivo.

1.84 ANÁLISIS ESTADÍSTICO DEL SISTEMA IMPLEMENTADO

En los ensayos aleatorios y en los estudios observacionales de comparación, entre el sistema implementado (ver Figura V.8) y el sistema de uso profesional para la recolección de datos biométricos como es la temperatura corporal de una persona Digital Thermometer

Temperature Sensor LCD Display TPM - 10 (ver Figura V.9), con la muestra tomada de cada sistema se quiere establecer que los datos que recolecta el sistema implementado se asemejan a los del sistema profesional Digital Thermometer Temperature

Ilustración 43 Datos recolectados por el Sistema Implementado



Fuente: Patricio Hidalgo

Ilustración 44 Datos recolectados por el Sistema Profesional



Fuente: Sistema Profesional

Para esto se ha empleado un análisis estadístico siguiendo un procedimiento objetivo por medio del cual se puede aceptar o rechazar un conjunto de datos como confirmatorios de una hipótesis, conocido un nivel de confianza y el riesgo que se corre al tomar tal decisión.

Una Hipótesis es una aseveración de una población con el propósito de poner a prueba, para verificar si la afirmación es razonable se usan los datos o se los rechaza. Al intentar alcanzar una decisión si el sistema implementado es eficiente o no con respecto al sistema

profesional Digital Thermometer Temperature, es útil hacer hipótesis sobre los datos muestrales recolectados, tales hipótesis que pueden ser o no ciertas son hipótesis nula y alterna.

1.84.1 HIPÓTESIS NULA H_0

Si deseamos comprobar que tanto se aproxima el sistema implementado al sistema profesional, formulamos la hipótesis de que no hay diferencia entre ellos (o sea que cualquier diferencia observada se debe simplemente a fluctuaciones en el muestreo de la misma población). Tales hipótesis se suelen llamar hipótesis nula y se denotan por H_0 , esta hipótesis es aquella que nos dice que no existen diferencias significativas entre los dos grupos de datos.

1.84.2 HIPÓTESIS ALTERNA H_1

afirmación que se aceptará si los datos muestrales proporcionan evidencia de que la hipótesis nula es falsa.

Se debe recordarse esta comprobación de hipótesis sólo puede aplicarse en el caso de que se haya procedido a un muestreo aleatorio de la población, lo que nos permite garantizar que dicha muestra es representativa de la referida población, al plantearnos estas dos hipótesis queremos demostrar que las medias de los sensores de cada sistema son iguales.

1.84.3 Hipótesis:

$$H_0 : \mu_1 = \mu_2$$

$$H_1 : \mu_1 \neq \mu_2$$

1.84.4 Donde:

H_0 = Hipótesis nula

H_1 = Hipótesis alternativa.

μ_1 = media de la población 1

μ_2 = media de la población 2

El procedimiento para probar $H_0 : \mu_1 = \mu_2$ es calcular la estadística de prueba Z para muestras mayores a 30 para dos medias mediante la siguiente fórmula:

$$Z = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{\sigma^2_1}{n_1} + \frac{\sigma^2_2}{n_2}}} \quad \text{Ec. 15}$$

Donde:

\bar{X}_1 = media de la muestra 1 (sistema profesional)

\bar{X}_2 = media de la muestra 2 (sistema implementado)

σ^2_1 = varianza de la población 1 (sistema profesional)

σ^2_2 = varianza de la población 2 (sistema implementado)

n_1 = tamaño de la muestra 1 (sistema profesional)

n_2 = tamaño de la muestra 2 (sistema implementado)

La hipótesis nula H_0 se rechaza si:

$$Z > Z_{\alpha/2} \quad \text{ó} \quad Z < -Z_{\alpha/2}$$

Donde:

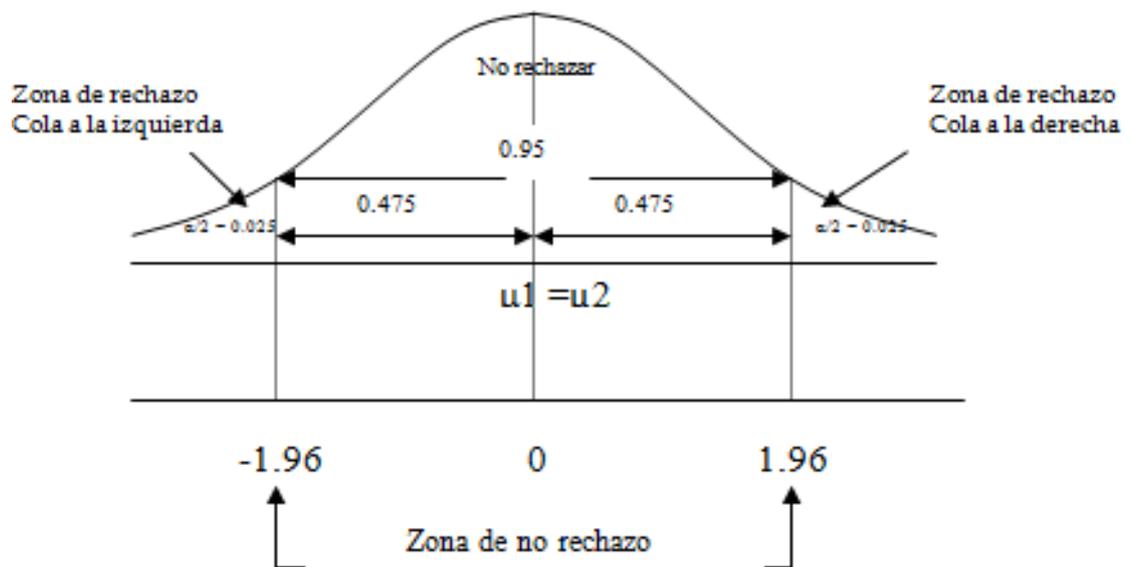
Z = Valor calculado estadístico prueba Z

$Z_{\alpha/2}$ = Valor obtenido de la tabla de distribución normal estándar Z (ver ANEXO).

α = Nivel de significación.

Es frecuente un nivel de significación de 0,05 ó 0,01. Si por ejemplo se escoge el nivel de significación 0,05 (ó 5%), es decir tenemos un 95% de confianza de que hemos adoptado la decisión correcta esto significa que tal hipótesis tiene una probabilidad 0,05 de ser falsa (ver Figura.V.10).

Ilustración 45 Área bajo la curva de una normal estándar



Existe un 95% de probabilidad de que los resultados muestrales puedan caer entre ± 1.96 si la hipótesis nula es verdadera

Fuente: CANAVOS, G

$$\alpha = 0,05$$

$$1 - \frac{\alpha}{2} = \text{zona de aceptación para } Z_{\alpha/2}$$

$1 - 0.025 = 0.975$ buscando el valor en la tabla de Distribución Normal Estándar Z correspondiente a 0.975 encontramos $Z_{\alpha/2} = 1.96$.

Se ha tomado muestras de datos para cada variable medida como temperatura, humedad, velocidad y dirección del viento, tanto con el sistema profesional como con el sistema implementado, aclarando que primero se ha tomado un dato del sistema profesional y luego en un rango de unos treinta segundos después con el sistema implementado, siendo esta diferencia de tiempo necesaria ya que se usaron los mismos sensores para cada sistema.

1.85 ANÁLISIS DE TEMPERATURA

Se ha recolectado una población de 33 datos para el análisis de temperatura, cada dato recolectado en función del tiempo para el sistema implementado y el sistema profesional, con una observación para cada sistema en grados centígrados (Tabla V.I).

Utilizando el criterio de $Z < -Z_{\alpha/2}$ decisión para rechazar la hipótesis nula H_0 , nos damos cuenta de que $-0,31$ es mayor que $-1,96$ por lo cual no rechazamos H_0 . No existe suficiente evidencia estadística para pensar que las medias son diferentes, es decir se dice que el sistema implementado mide temperatura estadísticamente igual al sistema profesional Digital Thermometer Temperature Sensor LCD Display TPM **Tabla V.I.** Toma de muestras para la temperatura

Tabla 5 ANÁLISIS DE TEMPERATURA

| Tiempo | N de muestras | Sistema profesional | Sistema implementado |
|--------|---------------|---------------------|----------------------|
| 8:00 | 1 | 18,4 | 18,3 |
| 8:05 | 2 | 18,2 | 18,4 |
| 8:10 | 3 | 18 | 18,1 |
| 8:15 | 4 | 18,6 | 18,8 |
| 8:20 | 5 | 18,8 | 18,7 |
| 8:25 | 6 | 18 | 18,2 |
| 8:30 | 7 | 18,7 | 18,4 |
| 8:35 | 8 | 18,3 | 18,4 |
| 8:40 | 9 | 18,4 | 18,5 |
| 8:45 | 10 | 18,9 | 19,1 |
| 8:50 | 11 | 18,4 | 18,6 |
| 8:55 | 12 | 18,5 | 18,3 |
| 9:00 | 13 | 18,7 | 18,5 |
| 9:05 | 14 | 18,6 | 18,7 |
| 9:10 | 15 | 18,9 | 19 |
| 9:15 | 16 | 18,6 | 18,8 |
| 9:20 | 17 | 19 | 19,1 |
| 9:25 | 18 | 19,2 | 19,3 |
| 9:30 | 19 | 19,1 | 19,3 |
| 9:35 | 20 | 19,5 | 19,6 |
| 9:40 | 21 | 19,7 | 19,8 |
| 9:45 | 22 | 19,6 | 19,8 |
| 9:50 | 23 | 19,4 | 19,6 |
| 9:55 | 24 | 19,9 | 19,8 |
| 10:00 | 25 | 20,2 | 20,3 |
| 10:05 | 26 | 20,3 | 20,5 |
| 10:10 | 27 | 20,7 | 20,5 |
| 10:15 | 28 | 20,4 | 20,6 |
| 10:20 | 29 | 20,1 | 20,2 |
| 10:25 | 30 | 21 | 21,3 |
| 10:30 | 31 | 21,5 | 21,7 |
| 10:35 | 32 | 21,8 | 21,9 |
| 10:40 | 33 | 22,1 | 22,2 |

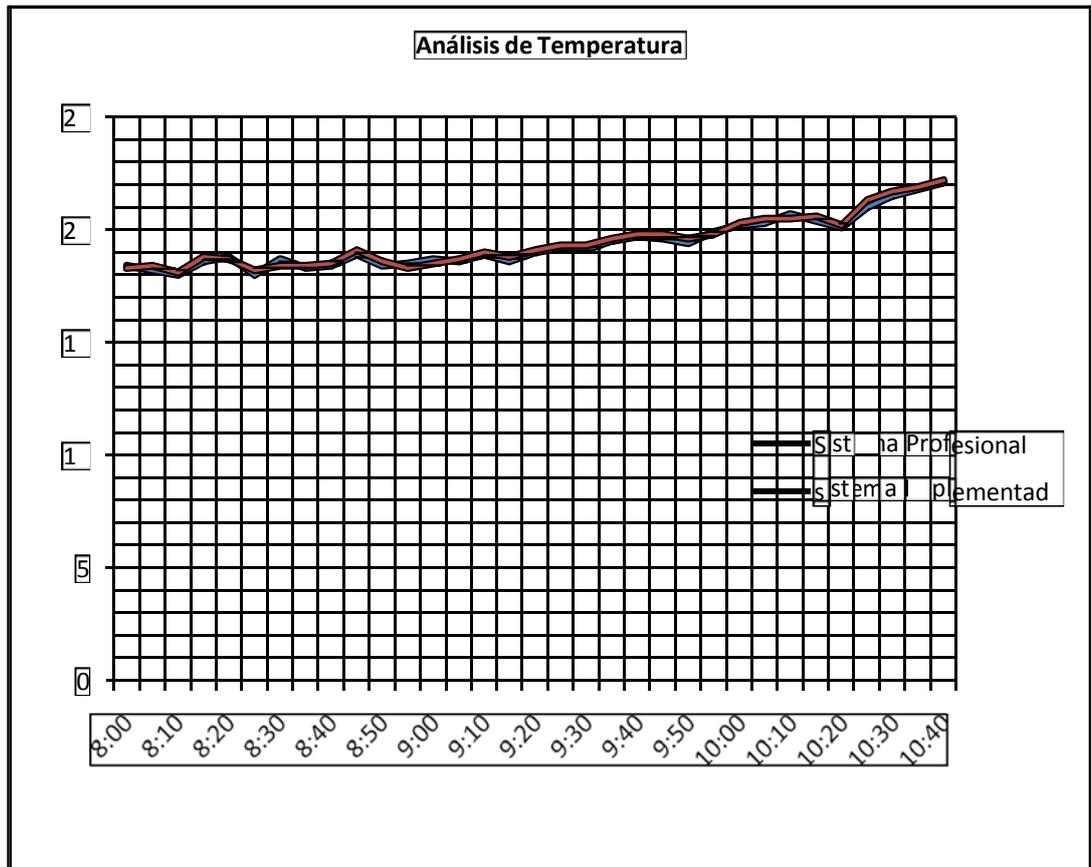
Fuente: Patricio Hidalgo

De la Ec.15 encontramos Z

$$Z = - 0,31$$

En la Figura V.11 se ve la misma tendencia de curvas del sistema implementado con respecto al sistema profesional.

Ilustración 46 Temperatura (Sistema Profesional y Sistema Implementado)



Fuente: Patricio Hidalgo

1.86 PRUEBA DE ENVÍO DE UN SMS

Esta es la etapa el sistema se encarga de procesar los datos que recolecta el sensor y además enviar un SMS al cumplirse la condición de exceso de temperatura corporal, una primera prueba fue enviar un SMS cada 12 minutos, el sistema recolecta cuatro datos (temperatura, Numero de piso, Numero de cama y código de paciente) cada minuto multiplicado por los 12 minutos se completa la capacidad de un mensaje de texto, a lo máximo que se puede llegar es a los

156 caracteres dependiendo de cada dato recolectado de los 160 caracteres de un mensaje de texto.

En esta prueba se determinó que siempre va haber un mínimo retraso para el envío de un mensaje de texto por el mismo tiempo que consume el Modem GSM Sim900 en el procesamiento de sus instrucciones, por lo que se observó un retraso de 1 a 3 segundos aproximadamente en comparación con el envío de mensajes de un teléfono celular Motorola XT919.

1.87 ENTORNO DE TRABAJO

Se obtuvo un dispositivo de temperatura que mide 14 cm de largo, 5 cm de alto y 7,5 cm de ancho, el cual puede interactuar con el usuario a través del módulo de censado.

En el momento de arrancar, el sensor biomédico espera las mediciones corporales para transmitir su mensaje, la recepta a través del módulo de censado, este es procesado y lo transmite mediante comunicación serial al módulo GSM sim900, el cual está dispuesto exclusivamente para la comparación de comandos recibidos por el módulo Arduino.

Ilustración 47 Funcionamiento del módulo Biométrico



Fuente: Patricio Hidalgo Modulo Biométrico

Lectura visualizada en el componente LCD al momento de encender el módulo biomédico.

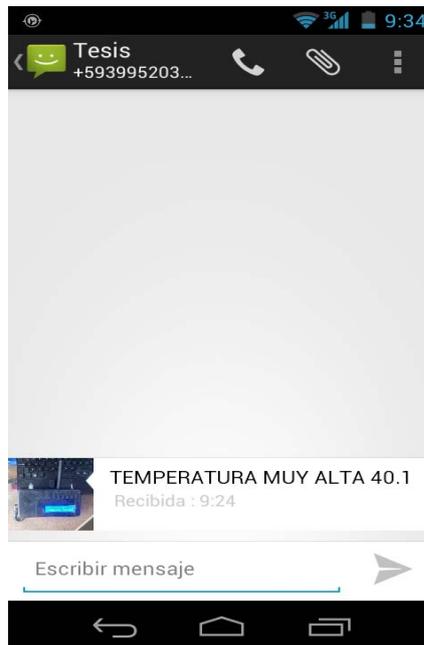
Ilustración 48 Visualización de la temperatura censada.



Fuente: Patricio Hidalgo la temperatura censada.

Mensaje visualizado en el receptor cuando ya se ha cumplido las condiciones en el sistema ver figura 49

Ilustración 49 Mensaje recibido en el celular.



Fuente: Mensaje recibido en el celular.

1.88 ANÁLISIS ECONÓMICO

Todo proyecto o actividad que se realice conlleva un gasto económico. Es por ello que a continuación se detalla cada uno de los elementos utilizados con sus respectivos precios.

1.89 COSTO DE MATERIALES

En la tabla II.II se detalla todo lo que ha sido utilizado para que nuestro sensor biométrico pueda funcionar.

Tabla 6 Materiales empleados en el ensamblaje del medidor de temperatura y sus costos

| PARTE MECÁNICA | | | |
|-----------------------|---|--------------------|-----------------|
| Cant. | DESCRIPCIÓN | V. Unitario | V. Total |
| 1 | Carcasa | 5,00 | 5,00 |
| 1 | Sensor de temperatura | 3,00 | 3,00 |
| HARDWARE | | | |
| 1 | Arduino | 45,00 | 45,00 |
| 2 | Baterías | 24,00 | 58,00 |
| 1 | Modulo SIM GSM | 110,00 | 110,00 |
| | Resistencias | | 2,00 |
| | Capacitores | | 2,00 |
| 1 | Cable Usb-Serial | 15,00 | 15,00 |
| VARIOS | | | |
| 1 | Vaquelita de Fibra de Vidrio | | 9,00 |
| 2 | Brocas - topes –tornillos- cables - ácido | | 22,00 |
| | Molex – sócalos | | 18,00 |
| TOTAL : | | | 281,00 |

Fuente: Patricio Hidalgo

1.90 PRESUPUESTO GENERAL

Para el desarrollo de este proyecto se necesitó de una inversión en tiempo y dinero. Para lo cual he realizado un resumen que se muestra en la tabla V.3.

Tabla 7 Presupuesto General

| TIPO DE RECURSO | TIPO | DESCRIPCIÓN | \$ MONTO |
|------------------------|---------------------------------|---|-----------------|
| Recursos humanos | Personal directo | Autor: Washington Patricio Hidalgo | 200,00 |
| | Personal indirecto | Director y Miembros de la Tesis. | |
| Recursos materiales | Computadora como equipo remoto. | Computadora básica, Corel Duo 1,8 Ghz, DD 160 Gb, Biostar, 1Gb Ram. | 600,00 |
| Recursos tecnológicos | Hardware | Elaboración del circuito impreso Transmisor-Receptor | 150,00 |
| TOTAL | | | 950,00 |

Fuente: Patricio Hidalgo

1.91 CONCLUSIONES

1. El sistema de adquisición de señales biométricas puede captar la temperatura de dos formas distintas como son axial y bucal casi sin mayor problema con un margen de error mínimo.
2. El sistema censa la temperatura corporal únicamente con el contacto directo a la persona a censar.
3. La distancia de transmisión del sistema como está basado en la tecnología GSM no tiene dificultad para el envío de mensajes al receptor.
4. El sistema de alimentación del sistema se basa en 2 baterías recargables para mayor estabilidad del sistema con un máximo de duración de 12 horas como máximo en trabajo continuo, o dicha duración dependerá de los tiempos de descanso que se le asignen al sistema.
5. El programa Arduino es muy práctico en lo que tiene que ver con la programación del módulo Arduino uno, puesto que es de fácil manejo y tiene una interfaz muy potente. De ahí nuestro interés por utilizarlo.

1.92 RECOMENDACIONES

1. Se debe tener cuidado en las conexiones del sistema de alimentación de las tarjetas electrónicas ya que una mala polarización de los mismos pueden dañar los elementos del sistema.
2. El proceso de construcción de este tipo de proyectos debe realizarse a nivel modular de esta manera optimizaremos tiempo en cuanto a ubicación de daños se refiere.
3. Para establecer una comunicación entre el software de aplicación implementado y el módulo sim900 por medio de comandos AT se debe instalar el controlador respectivo para comunicación serial.
4. Se recomienda tener un saldo activo permanente en el módulo sim 900, para poder enviar los datos recolectados.
5. Para un funcionamiento óptimo de este sistema se recomienda una alimentación constante, ya que por cada corte de energía el sistema se reinicia y empieza a funcionar desde cero perdiendo todos los datos.

RESUMEN

Se diseñó e implementó un sistema de adquisición de la señal biométrica “temperatura corporal de una persona”, utilizando tecnología móvil (teléfono) - para brindar la posibilidad de transmitir inalámbricamente hacia el teléfono del médico tratante.

Se emplearon técnicas de investigación, métodos lógicos y sistémicos para modelar el prototipo, “software” de programación como Arduino 1.5.6 y Lenguaje C Basic, sensor LM35 que se encarga de monitorear la temperatura corporal el módulo Arduino Diecimila, dispositivos electrónicos, LCD para visualizar los mensajes, módulo de transmisión y recepción GSM SIM900 para la comunicación inalámbrica.

Este dispositivo fue diseñado con el módulo Arduino Diecimila como cerebro principal, encargado de recolectar los datos transmitidos por el sensor lm35 convirtiéndolos de datos analógicos a digitales mediante técnicas de programación en Arduino, que al sobrepasar el rango establecido como temperatura corporal normal-el módulo SIM900 programado por comandos AT (que son instrucciones codificadas que proporcionan la comunicación de los módems)-envíe un mensaje de texto al teléfono del médico tratante con los datos de la temperatura.

Se realizaron pruebas de funcionamiento con el dispositivo y un medidor de temperatura corporal electrónico el cual dio un rango de precisión de +/- 2 grados centígrados, y en el momento de la transmisión un rango de tiempo de 1 minuto en el envío y recepción del mensaje, obteniendo una efectividad del 95% en la realización de las pruebas, pudiendo ser usado en cualquier lugar del campo de la medicina, llegando a cumplir con el objetivo propuesto.

ABSTRACT

Acquisition system of biometric signal “body temperature of a person” –using mobile technology (cell phone) was designed – to provide the ability to transmit wirelessly to the cell phone of the treating physician.

Research techniques, logical and systemic methods were used to model the prototype, programming software as Arduino1.5.6 and C Basic language, LM35 Sensor which is responsible for monitoring body temperature, Arduino diecimila module, electronic devices, LCD display messages, GSM SIM900 transmission and reception module for the wireless communication.

This device was designed with the Arduino diecimila module as the main brain, responsible for collecting the data transmitted by the sensor lm35 converting analog data to digital using programing techniques on Arduino, that exceed the established range as normal body temperature-the SIM900 module scheduled to AT commands (which are coded instructions that provide communication modems)-will send a text message to the cell phone of the treating physician with the temperature data.

Test runs were performed using the device and an electronic meter body temperature which gave a range of accuracy of +/- 2 degrees C, and when the transmission range of 1 minute in the sending and receiving of message, obtaining and effectiveness of 95% in the performance of tests, may be used anywhere in the field of medicine, coming to meet the proposed objective.

GLOSARIO

CA.- Es la simbología de corriente alterna, es aquella en que la que la intensidad cambia de dirección periódicamente en un conductor. Como consecuencia del cambio periódico de polaridad de la tensión aplicada en los extremos de dicho conductor.

CD.- Es la simbología de corriente directa, La corriente directa (CD) o corriente continua (CC) es aquella cuyas cargas eléctricas o electrones fluyen siempre en el mismo sentido en un circuito eléctrico cerrado, moviéndose del polo negativo hacia el polo positivo de una fuente de fuerza electromotriz, tal como ocurre en las baterías, las dinamos o en cualquier otra fuente generadora de ese tipo de corriente eléctrica.

Aplicación.- Programa que realiza una serie de funciones y con el cual trabajamos en el ordenador.

Arquitectura.- Término que se refiere al tipo de estructura hardware de la máquina y que también se aplica a la clasificación de los microprocesadores o el tipo de ranuras de expansión.

Asíncrono.- Término utilizado para especificar la ejecución de distintos procesos de forma independiente unos de los otros respecto al ^{tiempo}

Bit (binary digit).- Unidad básica de información representada por ceros y unos que se van sucediendo para conformar los distintos significados.

Chip.- Circuito integrado y encapsulado.

Frecuencia.- Número de veces que se repite un proceso periódico por unidad de tiempo.

Hardware.- Componentes materiales propios del computador, sean mecánicos eléctricos o electrónicos.

Interfaz.- Es el punto, el área, o la superficie a lo largo de la cual dos cosas de naturaleza distinta convergen. Por extensión, se denomina interfaz a cualquier medio que permita la interconexión de dos procesos diferenciados con un único propósito común.

Microcontrolador.- Es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de una computadora: CPU, Memoria y Unidades de E/S, es decir, se trata de un computador completo en un solo circuito integrado.

Módulo.- Son procedimientos que nos permiten la realización de una tarea deseada.

Ohm.- Unidad de medición de la resistencia eléctrica, representada por la letra griega.

Programa.- Conjunto de órdenes, palabras y símbolos producto de un análisis efectuado, creado expresamente para ordenar de modo racional los pasos que una computadora ha de dar para realizar un trabajo.

Resistencia.- Es la medida de cuanto se opone un circuito al paso de la corriente eléctrica a través de él.

Señal.- Variación de una corriente eléctrica u otra magnitud que se utiliza para transmitir información.

Software.- Elemento intangible (programas), sin el cual la computadora no podría funcionar.

Usart.- Es uno de los dos periféricos contenidos en el PIC que le permiten realizar comunicación en serie.

Voltios.- Unidad de medición de la diferencia de potencial o tensión eléctrica.

BIBLIOGRAFÍA

LIBROS

ANGULO, J.M^a; EUGENIO, M. y ANGULO, I. Microcontroladores Pic: la solución en un chip. España, Madrid: Paraninfo, 1997. p.129-144

ANGULO, J.M^a.; y ANGULO, I. Microcontroladores Pic: diseño y Aplicaciones. México, Distrito Federal: McGraw Hill, 1997. p.43-53

BLAKE, R. Sistemas Electrónicos de Comunicaciones. 2da. ed. Inglaterra, leeds: Thomson Learning, 2004. p.83-88

CANAVOS, G. Probabilidad y Estadística Aplicaciones y Métodos. 1ra.ed. México, Distrito Federal: McGraw Hill, 1988. p.255-257

CORRALES, S. Electrónica Práctica con Microcontroladores PIC. Ecuador, Quito, Gráfica, 2006. pp. 62-97.

DOGAN, I. Programación de Microcontroladores Pic. Estados Unidos, Nueva York: Interamericana, 2007. pp. 150-200.

BIBLIOGRAFÍA DE INTERNET

COMUNICACIÓN MÓVIL

<http://www.edicionsupc.es>

2013/10/14

<http://es.wikipedia.org>

2011/09/10

<http://www.gsmSpain.com>

2014/08/03

<http://pdf.rincondelvago.com>

2013/04/21

COMANDOS AT

<http://www2.dis.ulpgc.es>

2014/06/29

<http://todopic.mforos.com>

2011/07/12

<http://nds1.nokia.com>

2014/07/21

<http://www.todopic.com>

2013/08/01

ARDUINO

<http://www.arduino.cc/es/>

2013/07/21

<http://arduino.cc/es/Main/Software>

2014/05/23

<http://arduino.cc/es/Tutorial/Blink?from=Tutorial.BlinkingLED>

2014/06/16

<http://blogcmt.com/2012/10/25/mapa-de-la-cobertura-3g-en-espana/>

2014/06/18

ANEXOS

Anexo A. Código Fuente En ARDUINO

```
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <String.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
SoftwareSerial mySerial(7,8);
```

```
float centi()
```

```
{
  int dato;
  float c;
  dato=analogRead(A0);
  c=(500.0*dato)/1023;
  return(c);
}
```

```
void setup()
```

```
{
  mySerial.begin(19200);
  Serial.begin(19200);
  SIM900power();
  delay (1000);

  lcd.begin(16,2);
  lcd.print("TEMPER CORPORAL");

}
```

```
void loop()
```

```
{
  float Centigrados=centi();
  lcd.setCursor(5,1);
  lcd.print(Centigrados);
  if(Centigrados >= 21 && Centigrados<38){
    lcd.setCursor(10,1);
    lcd.print("NORMAL");
  }else{
    lcd.print(" ");
  }

  if(Centigrados >=38){
    lcd.setCursor(10,1);
    lcd.print("ANORMAL");
  }
}
```

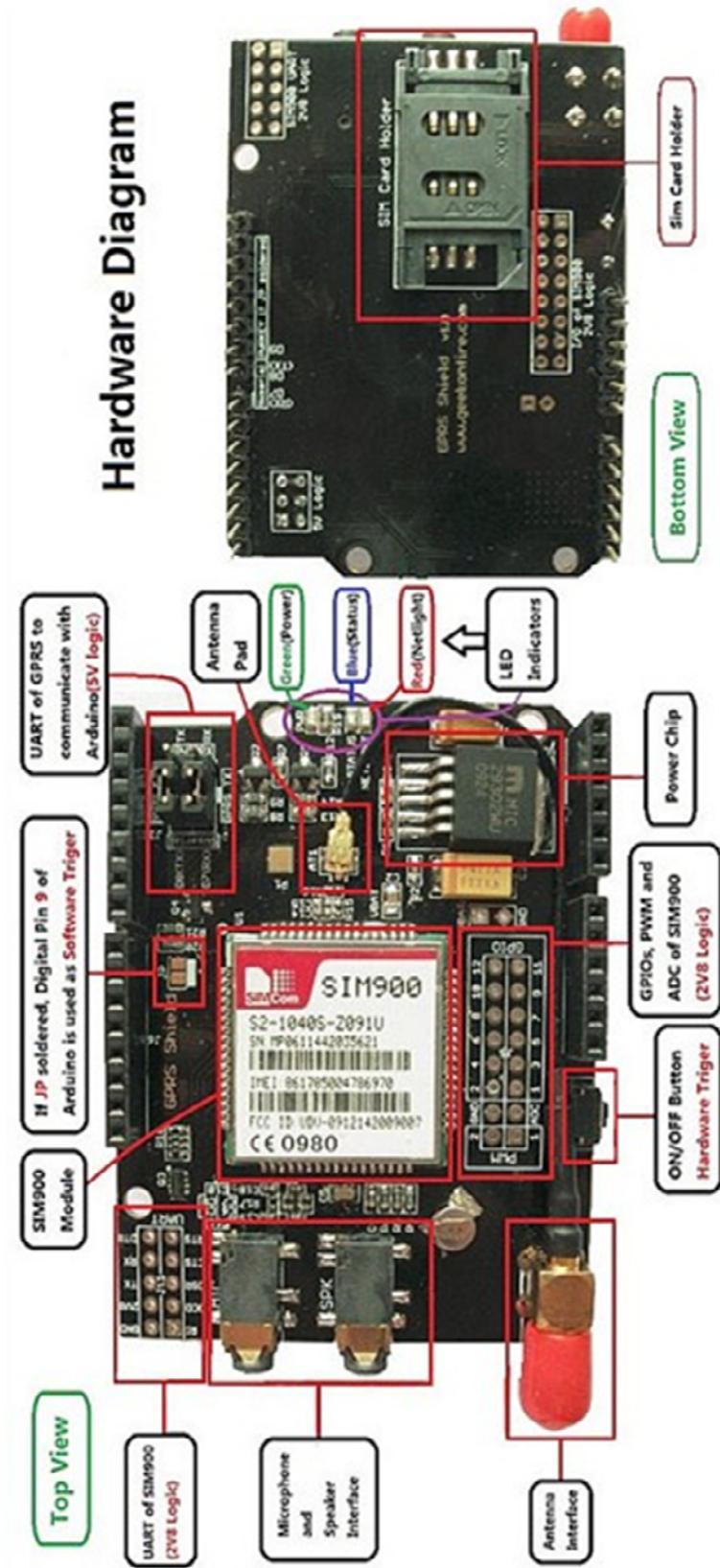
```
    SendTextMessage();
    lcd.print("    ");
}

delay(200);

}
void SendTextMessage()
{
    mySerial.println("AT+CMGF=1\r");
    delay(100);
    mySerial.println("AT + CMGS = \"+593984557824\");
    delay(100);
    mySerial.println("TEMPERATURA MUY ALTA ");
    delay(100);
    mySerial.println((char)26);
    delay(100);
    mySerial.println();
}

void SIM900power ( )
{
    digitalWrite ( 9 , HIGH ) ;
    delay ( 1000 ) ;
    digitalWrite ( 9 , LOW ) ;
    delay ( 2000 ) ;
}
```

Anexo B. GSM SHIELD SIM900



Anexo C. COMANDOS AT SIM900



SIM900 AT Command Manual

2 AT Commands According to V.25TER

These AT Commands are designed according to the ITU-T (International Telecommunication Union, Telecommunication sector) V.25ter document.

2.1 Overview of AT Commands According to V.25TER

| Command | Description |
|----------|---|
| A/ | RE-ISSUES THE LAST COMMAND GIVEN |
| ATA | ANSWER AN INCOMING CALL |
| ATD | MOBILE ORIGINATED CALL TO DIAL A NUMBER |
| ATD<N> | ORIGINATE CALL TO PHONE NUMBER IN CURRENT MEMORY |
| ATD<STR> | ORIGINATE CALL TO PHONE NUMBER IN MEMORY WHICH CORRESPONDS TO FIELD <STR> |
| ATDL | REDIAL LAST TELEPHONE NUMBER USED |
| ATE | SET COMMAND ECHO MODE |
| ATH | DISCONNECT EXISTING CONNECTION |
| ATI | DISPLAY PRODUCT IDENTIFICATION INFORMATION |
| ATL | SET MONITOR SPEAKER LOUDNESS |
| AIM | SET MONITOR SPEAKER MODE |
| +++ | SWITCH FROM DATA MODE OR PPP ONLINE MODE TO COMMAND MODE |
| ATO | SWITCH FROM COMMAND MODE TO DATA MODE |
| ATP | SELECT PULSE DIALLING |
| ATQ | SET RESULT CODE PRESENTATION MODE |
| ATS0 | SET NUMBER OF RINGS BEFORE AUTOMATICALLY ANSWERING THE CALL |
| ATS3 | SET COMMAND LINE TERMINATION CHARACTER |
| ATS4 | SET RESPONSE FORMATTING CHARACTER |
| ATS5 | SET COMMAND LINE EDITING CHARACTER |
| ATS6 | PAUSE BEFORE BLIND DIALLING |
| ATS7 | SET NUMBER OF SECONDS TO WAIT FOR CONNECTION COMPLETION |
| ATS8 | SET NUMBER OF SECONDS TO WAIT FOR COMMA DIAL MODIFIER ENCOUNTERED IN DIAL STRING OF D COMMAND |
| ATS10 | SET DISCONNECT DELAY AFTER INDICATING THE ABSENCE OF DATA CARRIER |
| ATT | SELECT TONE DIALING |
| ATV | TA RESPONSE FORMAT |
| ATX | SET CONNECT RESULT CODE FORMAT AND MONITOR CALL |

| | |
|----------|--|
| | PROGRESS |
| ATZ | RESET DEFAULT CONFIGURATION |
| AT&C | SET DCD FUNCTION MODE |
| AT&D | SET DTR FUNCTION MODE |
| AT&F | FACTORY DEFINED CONFIGURATION |
| AT&V | DISPLAY CURRENT CONFIGURATION |
| AT&W | STORE ACTIVE PROFILE |
| AT+GCAP | REQUEST COMPLETE TA CAPABILITIES LIST |
| AT+GMI | REQUEST MANUFACTURER IDENTIFICATION |
| AT+GMM | REQUEST TA MODEL IDENTIFICATION |
| AT+GMR | REQUEST TA REVISION IDENTIFICATION OF SOFTWARE RELEASE |
| AT+GOI | REQUEST GLOBAL OBJECT IDENTIFICATION |
| AT+GSN | REQUEST TA SERIAL NUMBER IDENTIFICATION (IMEI) |
| AT+ICF | SET TE-TA CONTROL CHARACTER FRAMING |
| AT+IFC | SET TE-TA LOCAL DATA FLOW CONTROL |
| AT+IPR | SET TE-TA FIXED LOCAL RATE |
| AT+HVOIC | DISCONNECT VOICE CALL ONLY |

2.2 Detailed Description of AT Commands According to V.25TER

2.2.1 A/ Re-issues the Last Command Given

A/ Re-issues the Last Command Given

| | |
|-------------------------|--|
| Execution Command A/ | Response Re-issues the previous Command |
| Reference V.25ter | Note |

2.2.2 ATA ANSWER AN INCOMING CALL

ATA ANSWER AN INCOMING CALL

Anexo D. ESPECTRO RADIOELECTRICO

