



UNIVERSIDAD NACIONAL DE LOJA

**ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y LOS
RECURSOS NATURALES NO RENOVABLES**

**CARRERA DE INGENIERIA EN ELECTRONICA Y
TELECOMUNICACIONES**

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE
TELEMETRÍA PARA LA AUTOMATIZACIÓN INALÁMBRICA
DEL CONSUMO DE AGUA MENSUAL MEDIANTE LA RED GSM.
CASO PRÁCTICO UNIVERSIDAD NACIONAL DE LOJA**

**TESIS DE GRADO PREVIO A OPTAR POR EL
TÍTULO DE INGENIERO (A) EN
ELECTRONICA Y TELECOMUNICACIONES**

AUTOR: MARIA ESTHER MEDINA SUCUNUTA

DIRECTOR: ING. BENJAMÍN ANDRÉS PUSAY VILLARROEL Mg. Sc.

LOJA-ECUADOR

AÑO 2015

CERTIFICACIÓN

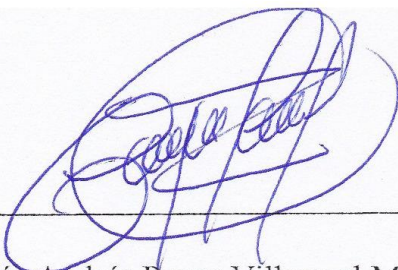
Ing. Benjamín Andrés Pusay Villarroel

DIRECTOR DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación cuyo tema versa en **“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE TELEMETRÍA PARA LA AUTOMATIZACIÓN INÁLAMBRICA DEL CONSUMO DE AGUA MENSUAL MEDIANTE LA RED GSM. CASO PRACTICO UNIVERSIDAD NACIONAL DE LOJA”**, previa a la obtención del título de Ingeniero (a) en Electrónica y Telecomunicaciones, realizado por los señor egresado: **Maria Esther Medina Sucunuta**, la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, 28 de julio de 2015



Ing. Benjamín Andrés Pusay Villarroel Ms. Sc.
DIRECTOR DE TESIS

AUTORÍA

María Esther Medina Sucunuta, declaro ser autor del presente trabajo de tesis y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi tesis en el Repositorio Institucional - Biblioteca Virtual.

Autor: María Esther Medina Sucunuta



Firma:

Cédula: 1104011539

Fecha: 23 de Septiembre del 2015

**CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL
AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O
TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO
COMPLETO.**

Yo **MARIA ESTHER MEDINA SUCUNUTA**, declaro ser autora de la tesis titulada: **DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE TELEMETRÍA PARA LA AUTOMATIZACIÓN INÁLAMBRICA DEL CONSUMO DE AGUA MENSUAL MEDIANTE LA RED GSM. CASO PRACTICO UIVERSIDAD NACIONAL DE LOJA**, como requisito para optar al grado de: **INGENIERO EN ELECTRÓNICA y TELECOMUNICACIONES**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, 22 días del mes de septiembre del dos mil quince.



Firma:

Autor: Maria Esther Medina Sucunuta

Cédula: 1104011539

Dirección: Loja (Dinamarca 36-50 y Pakistan)

Correo Electrónico:maryesther_1782@hotmail.com

Teléfono: 2108000 **Celular:** 0996036142

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Benjamín Andrés Pusay Villarroel Ms. Sc

Tribunal de Grado: Ing. Marco Vinicio Rojas Moncayo, Ms. Sc

Ing. Juan Pablo Cabrera Samaniego Mg. Sc.

Ing. Ángel José Ordóñez Mendieta Ms. Sc.

DEDICATORIA

El presente trabajo de tesis quiero dedicar en primer lugar a Dios por guiarme y darme las fuerzas necesarias para no desmayar durante en este largo camino estudiantil y al final cumplir con la meta anhelada.

A mis padres quien con amor, cariño, fuerza y trabajo me supieron guiar por el buen camino, además me dieron una excelente educación y supieron inculcarme valores que me han ayudado a conseguir todo lo que me he propuesto hasta hoy, especialmente a mi madre quien a pesar de quedarse sola no se derrumbó en ningún momento y supo incentivarlos para que sigamos adelante.

A mis hermanos por estar siempre presentes apoyándome en cada paso de mi vida y con sus consejos ayudarme para cumplir mis objetivos.

A mi hija quien es mi inspiración, motivación, para seguir adelante.

AGRADECIMIENTO

Agradezco a Dios por haberme guiado a lo largo de mi vida darme la sabiduría e inteligencia para poder alcanzar esta meta y así hacer realidad este sueño tan deseado

A mi madre que ha estado a mi lado en todos los momentos de mi vida apoyándome tanto moral, espiritual como económicamente.

A mis hermanos quienes me han incentivado con su ejemplo de superación a continuar con mis estudios y ser una persona de bien para el servicio de la comunidad.

A mis amigos visibles e invisibles que de alguna forma me incentivaron para continuar adelante en mis estudios.

Quiero hacer un agradecimiento especial Marco quien con su amor cariño y comprensión me ha sabido apoyar para poder culminar esta meta tan deseada.

A mis profesores quienes me dieron los conocimientos necesarios para poder realizar este trabajo de tesis, especialmente al Ing. Juan Gabriel Ochoa y Ing. Andy Fabricio Vega León quienes con su ejemplo, consejo y amistad me incentivar a cada día superarme y ser un mejor profesional.

TABLA DE CONTENIDOS

CERTIFICACIÓN	II
AUTORÍA	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO	VI
TABLA DE CONTENIDOS	VII
a. TÍTULO	1
b. RESUMEN.....	2
SUMARY	3
c. INTRODUCCIÓN	4
d. REVISIÓN DE LITERATURA.....	5
d.1. CAPÍTULO I: Fundamentos Teóricos.....	5
d.1.1. Sistema de Telemetría.....	5
d.1.1.1. Componentes De Un Sistema De Telemetría.....	5
d.1.1.1.1. Transductor	6
d.1.1.1.2. Unidad Terminal Remota (RTU)	6
d.1.1.1.3. El Medio de Transmisión (Referido como canal o Línea).....	7
d.1.1.2. Sistemas AMR.....	8
d.1.1.3. Sistemas IMR	9
d.1.2. Tecnologías de Comunicación Inalámbrica	9
d.1.2.1. WIFI	10
d.1.2.2. WIMAX	11
d.1.2.3. Red GSM	12
d.1.2.3.1. Arquitectura de la red GSM	13
d.1.3. Medidores de Agua	16
d.1.3.1. Medidores Inteligentes	16
d.1.4. Sensores	16
d.1.4.1. Tipos de Sensores.....	16
d.1.4.1.1. Atendiendo a su Funcionamiento	17
d.1.4.1.2. Señal de Salida	17
d.1.4.1.3. Modo de Operación	18
d.1.5. Entorno Arduino [26] [27]	18
d.1.5.1. Arduino Uno R3	19
d.1.5.2. Tarjeta Arduino Leonardo	20
d.1.5.3. Arduino Ethernet.....	21
d.1.5.4. Tarjeta Arduino GSM.....	23
d.1.5.5. Descripción de la placa Arduino	24
d.1.5.6. Programación	26
d.2. CAPÍTULO II: Diseño y Construcción del Hardware	27
d.2.1 Diagrama de bloques del sistema de telemetría del proyecto	27
d.2.2. Sistema de adquisición de datos.....	27
d.2.2.1. Sensor efecto Hall.....	27
d.2.2.2. Sensor de caudal de agua FS300a	28
d.2.2.3. Especificaciones Técnicas del sensor	28

d.2.3.	Selección de la tarjeta Arduino para la adquisición de datos	29
d.2.3.1.	Diagrama de Pines del Atmega 328	30
d.2.4.	Selección de la tarjeta <i>shield</i> para la comunicación con el Internet.....	33
d.2.5.	Diagrama de conexión del prototipo	34
d.2.6.	Descripción del diagrama de conexión	34
d.3.	CAPÍTULO III: Diseño y Construcción de Software	35
d.3.1.	Entorno de programación Arduino	35
d.3.1.1.	Primera ejecución del IDE de Arduino y configuración inicial para desarrollar proyectos hardware.....	35
d.3.1.2.	Librerías.....	36
d.3.2.	Diagrama de flujo del funcionamiento del sistema de telemetría	38
d.3.2.1.	Cálculo del flujo de agua	40
d.3.3.	Configuración de la tarjeta <i>shield</i> GSM/GPRS.....	42
d.3.3.1.	Envío de datos hacia la web	43
d.3.4.	Arduino Teclado LCD <i>Shield</i> (SKU: DFR0009)	44
d.3.4.1.	Tabla de estados de presentación en pantalla.....	46
d.3.5.	Descripción del entorno emoncms.org utilizado para el registro de datos [33]	47
e.	MATERIALES Y MÉTODOS	52
e.1.	Materiales	52
e.2.	Métodos	53
e.	RESULTADOS	54
e.	DISCUSIÓN	58
g.1.	ANÁLISIS DEL COSTO DEL PROTOTIPO	58
g.1.1.	Costos del Prototipo Diseñado.....	58
e.	CONCLUSIONES	60
e.	RECOMENDACIONES	61
e.	BIBLIOGRAFÍA.....	62
l.	ANEXOS.....	65
l.1.	Datasheet Arduino GSM Sim900	65
l.2.	Diagrama de conexión Key LCD.....	69
l.3.	Diseño de prototipo en proteus.....	70
l.4.	Código fuente	71
l.4.1.	Código fuente Arduino	71

Tabla de Tablas

Tabla 1.	Características básicas del Arduino uno, fuente: elaboración autora.....	19
Tabla 2	Características de la tarjeta Leonardo, fuente: elaboración autora	21
Tabla 3.	Características técnicas del Arduino Ethernet, fuente: elaboración autora.	22
Tabla 4	Componentes del sensor Fs300a [29].....	29
Tabla 5	Resumen de las principales tarjetas Arduino, fuente: elaboración autora.....	30

Tabla 6 de opciones de interfaz IDE, Fuente: elaboración [28].....	35
Tabla 7. Especificación de los botones de tarjeta LCD, fuente: elaboración autora	45
Tabla 8. Tabla de estados, fuente: elaboración autora	46
Tabla 9. Materiales utilizados en el desarrollo de la tesis; fuente: elaboración autora	52
Tabla 10 Datos de consumo de agua en lpm y m3, fuente: elaboración autora	56
Tabla 11. Costo del sistema, fuente: elaboración autora.....	59

Tabla de Figuras

Figura 1. Diagrama de una red WI-FI, fuente: elaboración autora	10
Figura 2. Arquitectura de la GSM, fuente: elaboración autora	14
Figura 3. Clasificación de los sensores, fuente: elaboración autora.....	17
Figura 4. Tarjeta Arduino UNO R3, fuente de elaboración: [28]	19
Figura 5. Tarjeta Leonardo. Fuente: [28]	20
Figura 6.Arduino Ethernet, fueute de elaboración: [28]	21
Figura 7. Arduino GSM, fuente elaboración: [28]	23
Figura 8. Descripción de la placa Arduino, fuente de elaboración: [28].....	24
Figura 9 Diagrama de bloques de una placa Arduino, fuente: elaboración autora	26
Figura 10. Diagrama de bloques del sistema de telemetría, fuente: elaboración autora	27
Figura 11 componentes del sensor Fs300a, fuente de elaboración: [29].....	28
Figura 12 dimensiones del sensor Fs300a, fuente de elaboración [29].....	29
Figura 13. Pines de Salida del ATmega164P/324P/644P, [28].....	30
Figura 14 Diagrama de conexión del sistema de telemetría, fuente: elaboración autora	34
Figura 15 Diagrama de flujo del funcionamiento general del sistema, fuente: elaboración autora	39
Figura 16 Algoritmo del cálculo de flujo de agua, fuente: Elaboración autora.....	40
Figura 17 Grafica de la constante (pendiente), Fuente: [29].....	41
Figura 18 Diagrama de conexión del sensor, fuente: [29].....	42
Figura 19 Algoritmo de configuración de la tarjeta shield GSM, fuente: elaboración autora	43
Figura 20 Algoritmo de envío de datos hasta la web, fuente: elaboración autora.....	44
Figura 21 Tarjeta LCD Shield, fuente: hoja de datos del fabricante	45
Figura 22: Entradas de emoncms.org, Fuente:[33].....	48
Figura 23 Feed de emoncms.org, Fuente:[33]	49
Figura 24: Visualización de variables en emoncms.org. Fuente: (El Autor).....	50
Figura 25: Interfaz gráfica para monitorear consumo m3 y L, en tiempo real. Fuente: (El Autor).	51
Figura 26: Gráfica para el monitoreo del consumo del agua en metros cúbicos y litros. Fuente: (El Autor).	51
Figura 27. Pruebas de funcionamiento General del sistema en el programa Putty, fuente: elaboración autora	54
Figura 28 Funcionamiento general del sistema en Arduino, fuente: elaboración autora	55
Figura 29 Envío de datos, fuente: elaboración autora	55
Figura 30.comprobación de envío de o de datos, fuente: elaboración autora	56

a. TÍTULO

**DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE
TELEMETRÍA PARA LA AUTOMATIZACIÓN INÁLAMBRICA
DEL CONSUMO DE AGUA MENSUAL MEDIANTE LA RED GSM.
CASO PRACTICO UNL**

b. RESUMEN

La presente tesis se enfoca en la realización del diseño y construcción de un prototipo de telemetría para la adquisición de datos del consumo de agua y posteriormente enviar dichos datos a la plataforma web utilizando la red GSM/GPRS. Básicamente el proyecto se divide en dos partes: el hardware y el Software.

En lo que se refiere a la parte del hardware se hizo un análisis detallado de los diferentes componentes para la adquisición de los datos. Se utilizó un sensor de caudal mediante el cual se va a obtener los valores de consumo de agua en litros/minuto. Posteriormente se analizó cada tarjeta Arduino para determinar cuál se adapta mejor al sistema de telemetría para la adquisición de los datos. Para enviar los datos hasta la web se utilizó la tarjeta *shield* GSM/GPRS puesto que ésta ya es una red establecida y se tendrá un ahorro en costo de instalación. Para visualizar la lectura del consumo de agua que da el sensor se utilizó el Arduino LCD KeyPad *Shield*.

En lo referente al diseño de software se utilizó el lenguaje de programación de la plataforma Arduino, el mismo que cuenta con un lenguaje de código abierto. Para ello se utilizó cuatro algoritmos: adquisición de datos del sensor en el Arduino, configuración de la tarjeta *shield* GSM/GPRS que permitirá comunicarse con el internet, cálculo de flujo agua y su respectivo envío de datos hasta la web y

Para la transmisión de los datos hasta el servidor se utilizó el servidor emoncms.org, es una plataforma “open source” que permite el monitoreo remoto de un dispositivo el cual toma los datos de variables de sensores de temperatura, flujo, energía, caudal humedad, etc. Además se puede tener un historial del consumo diario, mensual o anual, se puede visualizar la gráfica del monitoreo de las variables, el prototipo permite adquisición y registros de los datos de consumo.

SUMMARY

The present thesis is based on the design, development and construction of a telemetric prototype in order to obtain water consumption data and consequently send this information to the virtual platform using GSM/GPRS network. Basically, the project is divided into two parts: hardware and software.

Concerning the software part, a detailed analysis of different components was done in order to obtain data. A torrent sensor was applied through which we will obtain the quantity of gas consumption in liters/ per minutes. After this, each Arduino card was analyzed to determine which would adapt best to the telemetric system to achieve data acquisition. To send the data to the web, it was necessary to use a shield GSM/GPRS card, since this is a stablished network and some saving in the installation work will take place. To visualize the water consumption pattern that the sensor provides, we used the Arduino LCD Keypad shield.

Concerning the software design, programmed language from the Arduino platform was used, which counts with an open language code. To do this, four algorithms were used: acquisition of data from the Arduino sensor, Shield GPS/GPRS card settings which will allow to connect to the internet, water fluid calculus and data sending.

For the transmission of data to the server the server emoncms.org used , it is a platform "open source" that enables remote monitoring of a device which takes data variables of temperature sensors, flow, power , flow moisture , etc. Plus you have a history of daily, monthly or annual consummation you can display the graph of monitoring variables, the prototype allows acquisition and records consumption data .

c. INTRODUCCIÓN

El avance tecnológico está jugando un papel importante dentro de la sociedad, lo que ha permitido mejorar la calidad de vida de las personas. Las telecomunicaciones van de la mano del avance tecnológico permitiendo una información veloz y eficaz, cubriendo todas las formas de comunicación alámbricas o inalámbricas, incluyendo radio, telegrafía, televisión, telefonía, transmisión de datos, acceso a Internet. La telemetría se ha valido de estas tecnologías para la transmisión el control y monitoreo de diferentes procesos de adquisición de datos.

Los sistemas de telemetría permiten obtener información de manera automática y sin necesidad de tener contacto con el proceso que lo genera, la misma que será transmitida a una central para su posterior utilización.

Uno de los campos de la telemetría es la lectura automática de los medidores AMR (Automatic Meter Reading) que permitirán una captura de los datos de manera remota, ágil, confiable y segura, para su posterior transmisión, la misma que puede ser alámbrica o inalámbrica, ello permitirá dar solución a problemas de recolección de información y generación tardía de las facturas. Los sistemas de automatización son utilizados en diferentes ámbitos en la industria: para automatizar y controlar los procesos de producción, sistemas automatizados de seguridad, consumo de agua y energía que se centran en la macro y micromedición.

Considerando las necesidades específicas del sistema de medición, se puede considerar para la transmisión de los datos, el uso de sistemas basados en redes fijas (PSTN: Public Switch Telephony Network o Red de Telefonía Pública Conmutada), redes inalámbricas, comunicaciones vía satélite (para largo alcance) o telefonía celular. Un elemento importante a considerar al momento de decidir qué sistema de telemetría utilizar es la distancia desde el emisor hasta el receptor. Para el caso de sistemas inalámbricos se debe considerar también la posibilidad de contar con “línea de vista” (significa que una antena tiene que “verse”)

d. REVISIÓN DE LITERATURA

d.1. CAPÍTULO I: Fundamentos Teóricos

d.1.1. Sistema de Telemetría

La Telemetría hace referencia al uso de componentes eléctricos y electrónicos para detectar, almacenar datos físicos en un lugar y posteriormente mediante un proceso de transmisión remota enviar dicha información hacia una base de datos o el operador del sistema. La palabra telemetría procede de las palabras griegas tele ("lejos") y metrón ("medida") [1] [2].

Un sistema de telemetría debe ser capaz de medir una magnitud física, generando una señal que pueda ser modificada mediante un proceso de modulación y a su vez los datos obtenidos puedan ser transportados mediante un canal de transmisión. El equipo receptor debe ser apto de decodificar la señal en un formato adecuado para su posterior análisis y almacenamiento.

Además la telemetría es una técnica automatizada ya que mediante la transmisión, control y vigilancia del sistema no se necesita ir a lugar donde se genera la información debido que generalmente se utiliza transmisión inalámbrica pudiendo verificarse en una base de datos mediante un sitio web.

Se puede decir que la telemetría, es un conjunto de servicios y técnicas que relacionan las telecomunicaciones y la informática. La telemetría brinda los medios de comunicación e información, en diferentes ámbitos como en el trabajo o el hogar. Asocia diversos servicios, por ejemplo, telemando, las redes telemáticas como Internet. [3]

d.1.1.1. Componentes De Un Sistema De Telemetría

Un sistema de Telemetría para su correcto funcionamiento está constituido básicamente por los siguientes elementos:

- Transductor
- Unidad Terminal
- El Medio de Transmisión (Referido como canal o Línea)
- El Receptor

d.1.1.1.1. Transductor

Es un dispositivo físico que capta señales de una fuente determinada de energía en la entrada y la transforma en otro tipo de energía a la salida; un transductor puede incluir un sensor para censar la cantidad de líquido que a traviesa en un determinado caudal, como por ejemplo un circuito para convertir la cantidad de líquido a una variable eléctrica y enviarla a un sistemas de telemetría para su posterior análisis. Por todo esto es esencial estar apto para identificar cualquier tipo de energía, siendo importante entender que la forma final de la energía debe ser combatible con los circuitos electrónicos, para ello debe ser capaz de cambiar voltaje, corriente, resistencia, frecuencia, capacitancia o inductancia. [4]

Un transductor en un sistema de telemetría es de vital importancia ya que permite obtener los datos de entrada mediante un sensor y luego transformar en una señal eléctrica que sea legible para los sistemas electrónicos y de esta manera poder obtener los datos para enviarlos mediante un medio de comunicación ya sea alámbrica o inalámbrica para su posterior utilización.

Las principales fuentes de energía que los sensores pueden captar son: movimiento temperatura, luz presión, eléctrica, magnética, caudal. [4]

d.1.1.1.2. Unidad Terminal Remota (RTU)

Es la unidad que se encargan de la adquisición de datos que provienen del transductor así como la ejecución de comandos de operación para la codificación de la información para su posterior transmisión, además puede realizar funciones de procesamiento de datos y control si fuera necesario. Para el caso práctico que se va a realizar la RTU del proyecto será un Arduino. [5]

Generalmente es una unidad basada en un micro controlador. Su principal tarea es la de recoger datos en un área remota y transferirla a una estación central, la distancia que existe entre la RTU y el centro de control puede comprender entre unos cuantos metros hasta varios kilómetros. Cuando la distancia es demasiado grande la manera de transmitir los datos a la estación central es mediante conexión telefónica a través de módems, inalámbricamente vía radio, para el proyecto la RTU debe ser capaz de conectarse mediante la red GSM, cuando la distancia es corta la transmisión de la información se la puede realizar de manera cableada. [6]

d.1.1.1.3. El Medio de Transmisión (Referido como canal o Línea)

Se denomina medio de transmisión a las “vías” por el cual se transportan los datos desde el transmisor hasta el receptor, este medio puede ser de forma alámbrica (guiado) o inalámbrica (no guiado). El medio de transmisión juega un papel importante debido a que de las características del mismo va a depender de algunos aspectos como: velocidad de transmisión máxima, número de equipos conectados, distancia máxima entre el receptor y el transmisor. [7] [8]

- **Medio de transmisión alámbrica o guiado**

Son aquellos medios en los cuales el canal de transmisión es físico, es decir, que para poder enviar los datos se necesita un cable que permita establecer la conexión entre los equipos de comunicación, además estos medios evitan la pérdida, la dispersión de potencia mientras se acerca a su destino. Se dividen en dos grupos: los conductores por cobre y por medios ópticos: [7]

- ✓ **Medios conductores por cobre**

Este tipo de conductores son los más utilizados, están constituidos por dos o más conductores. Dependiendo del tipo de cable y de su utilización pueden ser usados en distintos tipos a continuación detallaremos los más utilizados.

- Coaxial
- UTP
- Par trenzado
- Cable STP

Este tipo de medios se utiliza para radio enlaces terrestres y vía satélite, en redes de área local, telefonía móvil, mandos a distancia, enlaces de datos dedicados, etc.

- ✓ **Medios Ópticos**

A diferencia de los conductores en donde la información se transmite por una tensión eléctrica, en la fibra óptica la información se transmite por pulsos de luz. La fibra óptica está formada por un cristal de silicio por donde se envía un haz de luz alrededor de este núcleo, también tiene un revestimiento que tiene un índice de reflexión menor protegiendo al núcleo provocando el fenómeno de reflexión interna, es decir que cuando un haz de luz

entra por extremo del cable no se disipa hacia el exterior si no que mediante reflexiones sucesivas dentro del núcleo se propaga hacia el otro extremo. [7] [9]

- **Medios de transmisión inalámbrica o no guiada**

La mayor parte de las telecomunicaciones en la actualidad se las realiza de forma inalámbrica debido a que ha tenido un gran crecimiento. La comunicaciones inalámbricas permiten transmitir datos sin la necesidad de que exista un medio físico, se logra establecer la comunicación mediante la generación ondas de radio electromagnéticas o de microondas. [7]

Las ondas electromagnéticas se producen por los campos eléctrico y magnético denominadas ondas de radio, gracias a estas ondas las distancias se han acortado permitiendo la comunicación con lugares remotos agilizando la economía y la política entre los países dando origen a la globalización. [8]

Existen varias ventajas de las comunicaciones inalámbricas que las han destacado por encima de las comunicaciones alámbricas estas son:

- Comunicación con lugares difíciles de acceder con el tendido de cable.
- Lograr la comunicación a cualquier distancia
- Abaratamiento de costos
- Flexibilidad y escalabilidad
- Movilidad, permitiendo el desplazamiento por toda la zona donde exista cobertura.

d.1.1.2. Sistemas AMR

Los sistemas AMR Lectura Automatizada de Medidor (automated meter reading), fue desarrollado en la década de los 80 [10]. Consiste en un sistema que permite la medición automática desde un lugar remoto mediante un sistema de comunicación, la recolección de información puede ser del consumo de agua, luz gas, etc., la transferencia de los datos hacia una base de datos para su facturación. Este tipo sistemas ayudan a reducir los costos operacionales generados por la recolección manual. La lectura puede ser semanal, mensual, trimestral, aún no está definido

Los datos de los medidores pueden ser leídos y consolidados en reportes sumarios para análisis o para ser reenviados a otros usuarios. Además los datos pueden ser vistos

remotamente vía telefonía fija o celular; en forma local con conexiones directas o en redes TCP/IP. La comunicación se la realiza en una sola “Vía” es decir que la información que es envía a la base de datos no puede ser cambiada ni modificada o enviar un lectura por pedido. [11, 10]

d.1.1.3. Sistemas IMR

Sistema de medición inteligente (IMR), son sistemas de varios componentes de hardware y de software, que establece una comunicación bidireccional, manteniendo una comunicación entre el cliente y la empresa de distribución de agua, electricidad, gas, etc. Esto quiere decir que puede enviar y solicitar información a pedido, o efectuar cambios remotos desde el centro de control. Este concepto fue desarrollado en la década de los noventa en Europa, principalmente por el Reino Unido. [11]

Es un sistema avanzado el cual es una solución para las empresas públicas que se encarga de la distribución de los servicios de agua, luz, gas, etc. Esto les ayudara a brindar un mejor servicio a los clientes para que estén más satisfechos, implementando nuevos servicios que permitieran que exista interactividad entre cliente usuario, obteniendo información en tiempo real.

El elemento principal de un sistema de medición inteligente es el medidor inteligente, el mismo que ha aparecido gracias a la evolución de la electrónica y las telecomunicaciones. Dispone de mayores capacidades que los medidores electromecánicos y electrónicos de estado sólido. [12]

d.1.2. Tecnologías de Comunicación Inalámbrica

La principal característica de las comunicaciones inalámbricas es permitir el envío y recepción de información sin la necesidad de conectarse con cables ya que para la generación de la misma se la establece por medio de ondas de radio electromagnética. Para que se pueda establecer dicha comunicación han surgido diferentes tecnologías que ayudan a que la comunicación inalámbrica sea más veloz y eficaz.

A continuación hablaremos de las diferentes tecnologías inalámbricas existentes y se determinará cual se adapta mejor para el desarrollo del presente proyecto.

d.1.2.1. WIFI

La red WI-FI (Wireless Fidelity), conjunto de normas y estándares para comunicaciones inalámbricas basados en las especificaciones de la IEEE 802.11 que trabaja en la capa física y de enlace de datos del modelo OSI, fue creado con la finalidad de conectar redes de área local, trabaja en la frecuencia 2.4 GHz.

Para establecer una red WI-FI se necesita un Access Point (AP) o punto de acceso inalámbrico el cual va conectado a un MODEM el mismo que se conecta con el núcleo de la red. El AP se encarga de transmitir y recibe datos a través de ondas de radio y de los equipos remotos que cuentan con un transceptor (transmisor-receptor) en una tarjeta de acceso. Se comunican con él como se muestra en la figura 1 [13]

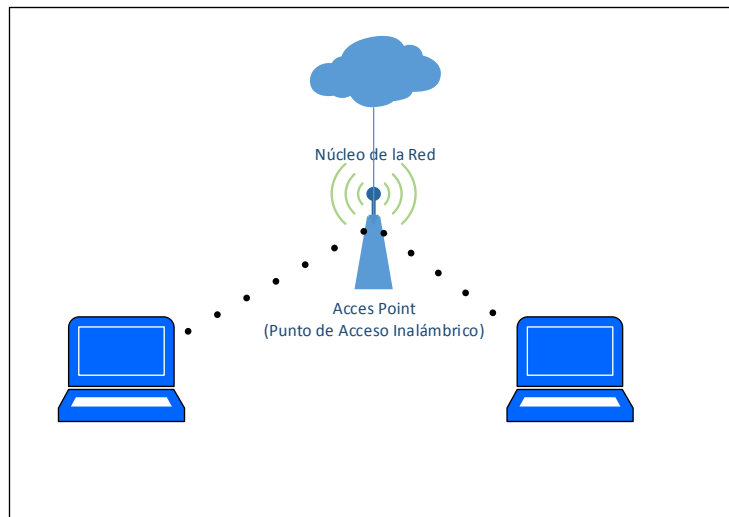


Figura 1. Diagrama de una red WI-FI, fuente: elaboración autora

Características [14] [13]

- Tiene una velocidad de transmisión de hasta 54kbps en un canal de 20 MHz
- Trabaja en la banda de los 2.4 GHz Esta banda de frecuencia no requiere de licencia para su operación.
- Opera con las modulaciones PSK, QPSK Y OFDM.
- Movilidad, la posibilidad de poderse conectar desde cualquier parte donde exista cobertura dentro de la red
- Desplazamiento, poder desplazarse por cualquier parte del área de cobertura sin perder conexión.

- Flexibilidad, en las redes WI-FI se puede colocar un ordenador sin la necesidad de realizar ningún cambio en la configuración
- Escalabilidad, Facilidad de expandir la red después de su instalación inicial.
- Ahorro de instalación y mantenimiento
- Fácil de instalar, el costo de instalación y mantenimiento es más barata que una red cableada
- Tiene un alcance de 30 a 100m

Una de las cosas que preocupa de las redes inalámbricas es la poca seguridad que brinda debido a que son vulnerables y pueden ser jaqueadas e interceptar la información que se transmite.

d.1.2.2. WIMAX

Worldwide Interoperability for Microwave Access (interoperabilidad mundial para acceso por microondas) conocida por su siglas WIMAX nombre con el que se le conoce al estándar 802.16 aprobado en el 2013 en el WIMAX fórum, permite el acceso de banda ancha última milla sin cables como una alternativa al cable, DSL, y algunas otras tecnologías inalámbricas. Admite conectividad entre puntos fijos, nómadas, portátiles y casualmente la conexión móvil de banda ancha sin la necesidad de tener una línea punto a punto [15] [16].

La tecnología WIMAX trabaja en la banda licenciada en el rango desde 2,3 a 3,5 GHz, necesita línea de vista directa (LOS) y el rango 5,8 no licenciada no necesita línea de vista directa NLOS. Diseñada para una red metropolitana con una cobertura de 50Km por celda y tasas de transmisión de 70 Mbps. [17]

Características [15] [16] [17]

- Acceso de banda ancha, integra diferentes servicios como voz, datos, tanto comerciales como residenciales
- Cubre distancias de hasta 50 kilómetros, con antenas muy direccionales y de alta ganancia.
- Velocidades de hasta 70 Mbps, siempre que el espectro este completamente limpio.

- Facilidades para añadir más canales, dependiendo de la regulación de cada país.
- Anchos de banda configurables y no cerrados, sujetos a la relación de espectro
- Brinda Calidad de servicio (QoS), lo que las redes WIFI no ofrecen.
- Creada principalmente para empleo en las ciudades densamente pobladas, además ha sido perfeccionada para trabajar sin necesidad de tener visión directa.
- WIMAX intenta garantizar la interoperabilidad con estándares para redes de área metropolitana inalámbrica o WMAN como HiperMAN (High Performance Radio Metropolitan Area Network), LMDS (Local Multipoint Distribution Service), MMDS (Multichannel Multipoint Distribution Service) que tienen objetivos muy similares a WIMAX
- Los métodos usados de multiplexación para la transmisión son OFDM(Multiplexación por División de Frecuencias Ortogonales) OFDMA (Acceso Múltiple por División de Frecuencias Ortogonales) con 256 y 2048 subportadoras y QAM en la banda de 10 a 66GHz; que permite altas tasas de transmisión.
- La modulación que utiliza es QPSK, 16 QAM y 64 QAM.
- Una característica importante es la independencia de protocolo, esto es que puede transportar IP. Ethernet, ATM y otros.

d.1.2.3. Red GSM

La red GSM Sistema Global para Comunicaciones Móviles estandarizado por el Instituto Europeo para la Normalización en Telecomunicaciones (ETSI) en la década de los 80. GSM ha tenido tres fases de evolución, en la primera fase se produjeron sus especificaciones, en la segunda fase se produjo la inclusión de servicios de fax y de voz, en la fase 2+ en la cual se realizan mejoras sobre la codificación de voz y se implementan servicios de transmisión de datos avanzados, entre ellos GPRS y EDGE. Pensó desde un inicio como una red PLMN ya que realiza todas las funciones normales de telefonía, como la gestión de llamadas, control de tráfico, análisis de numeración, tarifación y estadísticas de llamadas, servicios adicionales como voz, datos, mensajes cortos.

La tecnología GSM establece tres bandas de frecuencia para su implementación: 900, 1800 y 1900, tiene una tasa de transferencia de 9,6 Kbps. Las capacidades de GSM son más amplias se dividen en dos grupos principales: Servicios Básicos y Servicios Suplementarios.

1. Servicios básicos:

- Servicios Portadores.- proporcionan la capacidad de transferencia entre terminales conectados a red GSM, así como con equipos conectados a otras redes, además soporta datos por conmutación de servicios a velocidades de 300, 1200,2400,4800, 9600 bits/s.
- Teleservicios.- tienen la capacidad de comunicación completa entre los usuarios o terminales, de acuerdo a protocolos ya establecidos, la red GSM soporta los siguientes teleservicios: Telefonía, llamadas de emergencia, servicios de mensajes cortos (SMS), fax automático.

2. Servicios Suplementarios.- son servicios agregados que ofrece la comunicación entre usuarios ejemplo: autenticación de usuarios, identificación de llamadas entrante, restricción de llamadas, transferencia de llamadas, llamadas en espera entre otras.

d.1.2.3.1. Arquitectura de la red GSM

Una red GSM está dividido en tres partes: el sistema de conmutación, el sistema de estación base, sistema de operación y mantenimiento (figura 2), cada sistema contiene una serie de unidades funcionales en las cuales se realizan todas las funciones que el sistema GSM es capaz de proporcionar. [18]

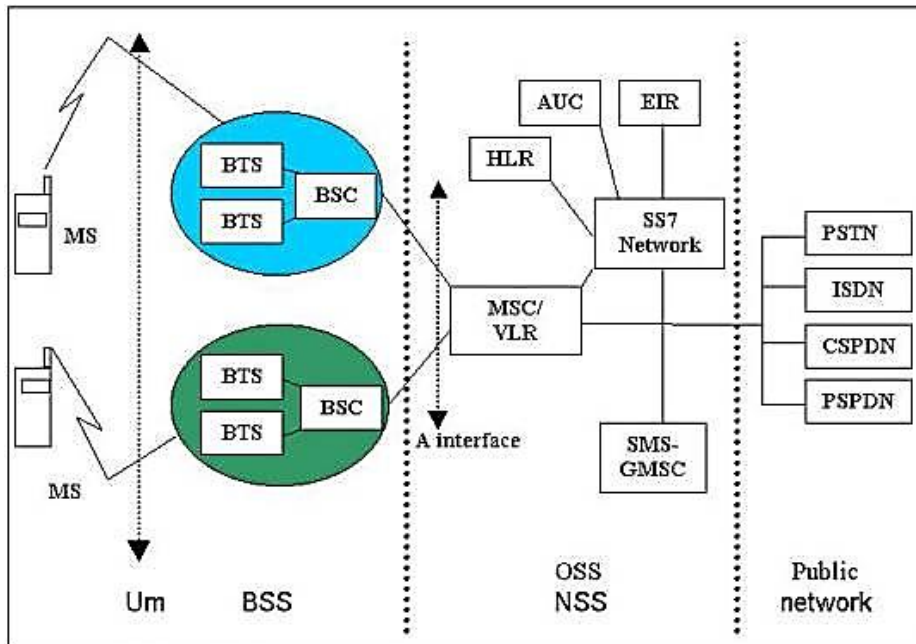


Figura 2. Arquitectura de la GSM, fuente: elaboración autora

d.1.2.3.1.1. Sistema de conmutación (NSS) [19]

Su función es de gestionar las identidades de los usuarios, la ubicación y el establecimiento de comunicación con otros usuarios, se encarga de todas las funciones normales en telefonía, incluye las siguientes unidades funcionales o nodos de la red GSM.

- Central de conmutación de móviles (MSC)
- Registro de posiciones base (HLR)
- Registro de posiciones visitado
- Centro de autenticación (AUC)
- Registro de identificación de estaciones móviles (EIR)

d.1.2.3.1.2. Sistema de Estación Base (BSS)

Constituye las funciones de la capa física, según el modelo OSI, contrala la interfaz de radio, para ello hace uso de un grupo de canales lógicos, se encarga de gestión de recursos para la transmisión vía radio, está compuesto por varias BTS (Estación Base), o también denominadas radio bases las cuales forman una celda o célula las cuales son controladas por una estación controladora de bases (BSC). [20] [18]

d.1.2.3.1.3. Sistema de operación y Mantenimiento

El sistema de operación y mantenimiento centralizado y remoto, proporcionara los medios necesarios para poder llevar a cabo una eficiente gestión de red tanto en la parte de conmutación como en la de la radio. [20]

Las principales funciones que realiza este sistema son: gestión de llamadas, administración de abonados, autenticación de llamada, Conmutación entre BSS del mismo NSS o con otros MCS o redes. Funciones de Gateway con otras redes (PLMN, ISDN, PSTB, etc.), proceso de Handover, confidencialidad de la identidad de usuario. [20]

d.1.2.3.1.4. Mensajes cortos

El servicio de mensajes cortos (SMS), permite el envío de un mensaje de texto desde una estación Móvil (MS) y entre otra entidad (SME), de hasta 160 caracteres de ASCII de 7 bits. El mensaje contiene: Mensaje de texto, Número de emisor y hora de emisión. Los mensajes se pueden enviar o recibir aun cuando se está recibiendo una llamada de voz o datos, el centro SMS es el encargado de enviar, recibir y almacenar los mensajes cortos. Existe tres tipos de mensajes [21] [20] [18]:

Point to Point (SMS /PP): Los mensajes se originan en el SMSC, y van destinados a una estación móvil particular, se realiza en dos fases.

SMS/CB (SMS Cell Broadcast): envió de mensajes a todos los MS. Dentro de una celda los mensajes se reducen a un tamaño de 93 caracteres como máximo y posibilidad de enlazar hasta 15 mensajes.

d.1.2.3.1.5. Comandos AT [20] [22]

Los comandos AT son un conjunto de instrucciones que forman un estándar de comunicación, fue creado en el año 1977, con la finalidad de enviar instrucciones aun terminal modem para poder configurarlo, es utilizado especialmente en la red GSM y radio frecuencia.

Los modem GSM son más específicos en cuanto al funcionamiento, las instrucciones que se pueden utilizar, la configuración, conexión y realización de pruebas con otros modem se lo realiza por medio de un ordenador o micro controlador debido a que no dispone de teclado ni pantalla, los modem GSM disponen de puertos RS 232 o USB.

A pesar de que existen varias listas de comandos AT se recomienda utilizar la lista de comandos dados por el fabricante del equipo en el manual, así se asegura mayor compatibilidad y mejor funcionamiento.

d.1.3. Medidores de Agua

Son instrumentos que utiliza para contabilizar el consumo de agua en instalaciones de agua domiciliarias o industriales, se colocan en tuberías de diámetro pequeño, la giración de este dispositivo provoca descarga de agua en la casa del usuario, que establece una relación entre el número de vueltas que da el dispositivo con el volumen del consumo de líquido consumido, los cuales se registran en diales dotados de diferentes dígitos. [23]

d.1.3.1. Medidores Inteligentes

Un medidor inteligente de gas, agua o electricidad, puede comunicarse de forma bidireccional es decir puede transmitir y recibir información del consumo, El medidor inteligente regula el consumo de agua de la misma manera como un medidor tradicional, pero tiene capacidad de comunicación permitiendo que la información sea leída remotamente y mostrarla en un dispositivo en el interior del domicilio del usuario o sean transmitidos a una base de datos de forma segura. El medidor también puede recibir los datos de forma remota, por ejemplo, para actualizar información sobre las tarifas o cambiar el tipo de consumo a modo de prepago.

d.1.4. Sensores

Son dispositivos que se los utiliza para recibir cierta cantidad de energía y transformar magnitudes físicas o químicas, denominadas variables de instrumentación en magnitudes eléctricas, las variables de instrumentación dependen del tipo de sensor estos pueden ser de temperatura, humedad, de presión, intensidad, lumínica, distancia, torsión, etc.

d.1.4.1. Tipos de Sensores

Los sensores para su mayor comprensión se han clasificado de la siguiente manera como indica la figura3:

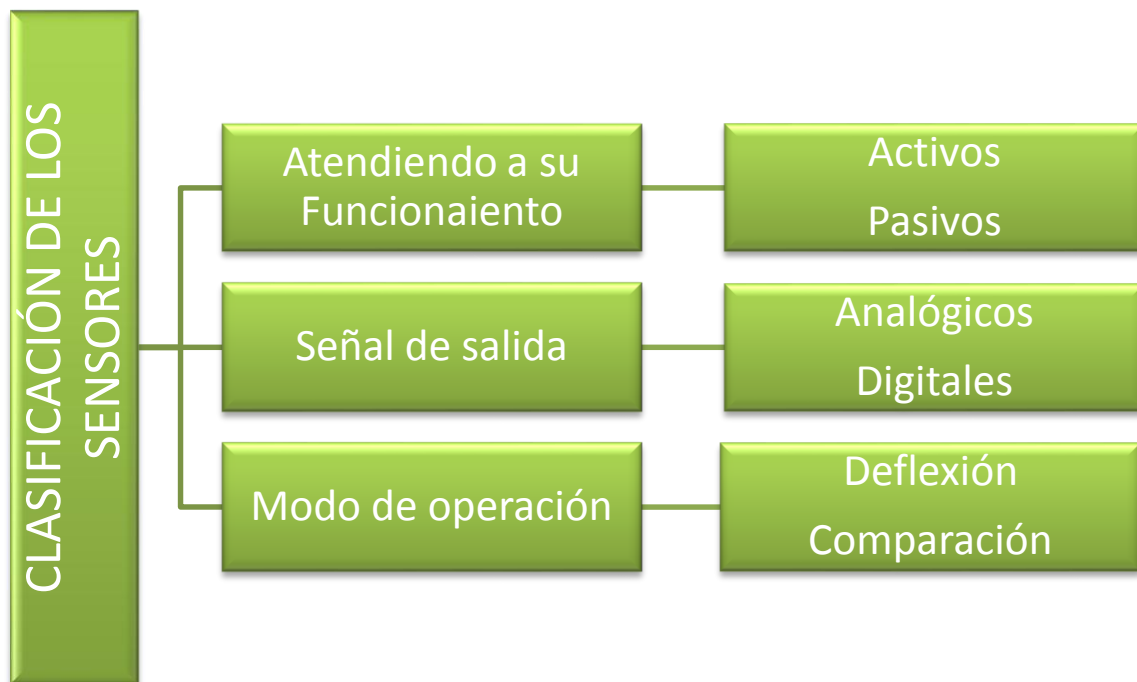


Figura 3. Clasificación de los sensores, fuente: elaboración autora

d.1.4.1.1. Atendiendo a su Funcionamiento

Activos: necesita de una fuente de energía externa, de donde proviene la energía eléctrica para su funcionamiento. Ejemplo de este tipo de sensores son los basados en los efectos piezoeléctrico y termoeléctrico.

Pasivos: No requiere de fuente externa de energía, La magnitud física a medir modifica alguno de los parámetros eléctricos del sensor como por ejemplo la resistencia, la capacidad, etc. Ejemplos de este tipo de sensores, son los basados en las resistencias cuyo valor depende de la temperatura (Termorresistivos) o de la luz (Fotorresistivos). [24]

d.1.4.1.2. Señal de Salida

Analógicos: Sensores que generan señales eléctricas denominadas analógicas que pueden tomar cualquier valor, dentro de unos determinados márgenes y que llevan la información en su amplitud.

Digitales: En general proporciona una señal digital a través de un procesador de comunicaciones que utiliza un protocolo normalizado como por ejemplo RS-232, RS-422, RS-423, RS-485, un bus de campo (Field bus) o una red industrial que combina la red de

área local Ethernet con el protocolo IP para facilitar la comunicación con cualquier sistema de control o informático (Industrial Ethernet) [25]

d.1.4.1.3. Modo de Operación

De Deflexión: La magnitud medida produce otra similar en el sensor pero opuesta y relacionada directamente con ella.

De comparación: Compara la señal a medir con la de un patrón. Las medidas por comparación son más precisas porque se pueden calibrar con un patrón de calidad contrastada. El detector de desequilibrio debe medir alrededor del cero y en ocasiones ha de ser muy sensible. Tienen menor respuesta dinámica.

d.1.5. Entorno Arduino [26] [27]

Arduino es una plataforma de electrónica, de hardware y software libre creado con la finalidad de facilitar el desarrollo de proyectos electrónicos, al ser una plataforma cuenta con una gama de tarjetas muy sencillas que tienen entradas y salidas analógicas y digitales, la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing), puede conectar el mundo físico con el virtual, para poder conectar una tarjeta a un PC solo se necesita de un cable USB, son fáciles de usar. Fue desarrollado por el zaragozano David Cuartielles, Ingeniero Electrónico y docente de la Universidad de Malmo, Suecia y Massimo Banzi, italiano, diseñador y desarrollador Web. El proyecto fue concebido en Italia en el año 2005. Se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Cuenta con varias tarjetas a continuación se detallara las más importantes y así determinar cuál se acopla mejor para el desarrollo del proyecto.

d.1.5.1. Arduino Uno R3

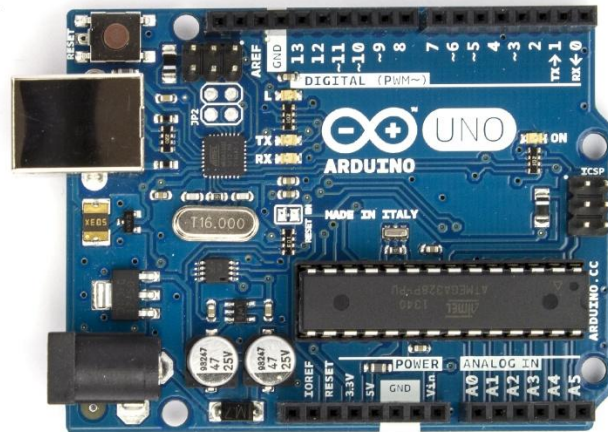


Figura 4. Tarjeta Arduino UNO R3, fuente de elaboración: [28]

Esta es la más reciente versión de Arduino UNO: R3. Como indica la figura 4. Esta tarjeta incorpora nuevas características como el empleo de un ATmega16U2 en lugar del 8U2 usado en versiones anteriores de Arduino. Esto permite tasas de transferencia más altas y brinda más memoria. Permite que el UNO sea reconocido como un teclado, mouse, joystick o cualquier otro periférico USB.

El Uno R3 también incorpora nuevos pines como: SDA (línea de datos), SCL (Línea de reloj) al lado de AREF. Así mismo se presenta un par de nuevos pines cerca del pin de RESET. Uno de ellos es el IOREF que permite que los *shields* se adapten al voltaje empleado por la tarjeta; el otro pin está reservado para futuras versiones. El Arduino UNO R3 se puede emplear con todos los *shields* existentes así como con las versiones más nuevas que ya hacen uso de estos pines. En la tabla 1 se detallan las características más importantes del Arduino UNO R3.

Tabla 1. Características básicas del Arduino uno, fuente: elaboración autora

Microcontroladores	ATmega328
Tensión De Funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital pines I / O	14 (6 salida PWM)
Pines de entrada analógica	6
Corriente DC por Pin I / O	40 mA

Corriente DC de 3.3V Pin	50 Ma
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad De Reloj	16 MHz

d.1.5.2. Tarjeta Arduino Leonardo

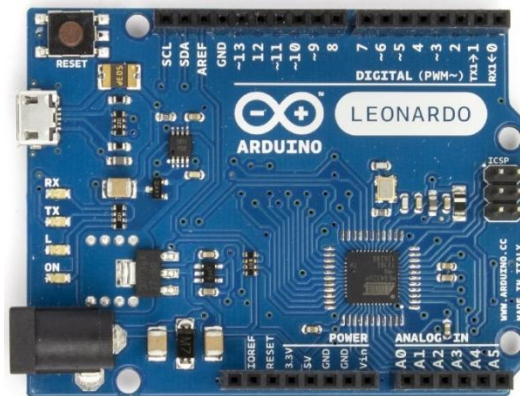


Figura 5. Tarjeta Leonardo. Fuente: [28]

El modelo Arduino Leonardo como se ve en la figura 5, es el nuevo modelo del team de Arduino. Utiliza un microcontrolador ATmega32U4 que permite un diseño mucho más sencillo y económico. Una de las ventajas de este nuevo microcontrolador es que dispone de USB nativo por hardware y por lo tanto no necesita de ninguna conversión serie-USB.

Soporta 12 entradas analógicas y dado que el puerto de comunicación USB es emulado, deja el puerto serial hardware libre para la programación. De esta forma ya no ocurren conflictos de programación mientras tenemos periféricos seriales conectados a la placa.

El Leonardo difiere de todas las otras placas porque tiene una arquitectura de un único procesador solo un microcontrolador en la placa que se comunica con el ordenador directamente por USB. Nuevas librerías en el Arduino IDE para que dicha tarjeta pueda actuar como un USB. Permitiéndole así emular un teclado o un ratón USB, además de un virtual (CDC) de puerto serie / COM. En la tabla 2 se describe las principales características de Leonardo.

Tabla 2 Características de la tarjeta Leonardo, fuente: elaboración autora

Microcontroladores	ATmega32u4
Tensión De Funcionamiento	5V
Voltaje de entrada (recomendado)	7-12V
Voltaje de entrada (límites)	6-20V
Digital pines I / O	20
Canales PWM	7
Canales de entrada analógicos	12
Corriente DC por Pin I / O	40 Ma
Corriente DC de 3.3V Pin	50 Ma
Memoria Flash	32 KB (ATmega32u4) de los cuales 4 KB utilizado por el gestor de arranque
SRAM	2,5 KB (ATmega32u4)
EEPROM	1 KB (ATmega32u4)
Velocidad De Reloj	16 MHz

d.1.5.3. Arduino Ethernet

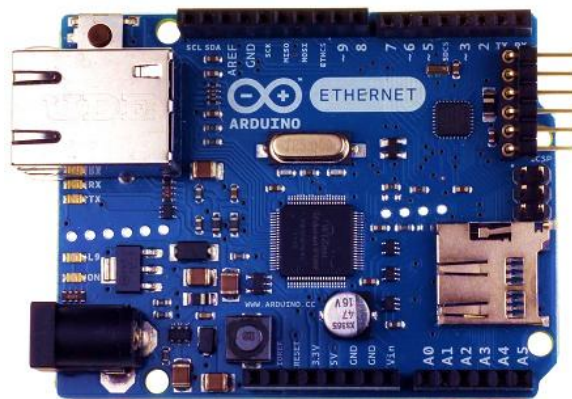


Figura 6.Arduino Ethernet, fuente de elaboración: [28]

La tarjeta Arduino Ethernet (figura 6) es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 pines digitales de entrada / salida, 6 entradas analógicas, un oscilador de cristal de 16 MHz, una ranura RJ45 de conexión, un conector de alimentación, una cabecera ICSP, y un botón de reinicio.

Los pines 10, 11, 12 y 13 están reservados para la conexión con el módulo Ethernet y no debe ser utilizado de otra manera. Esto reduce el número de pines disponibles a 9, con

4 disponibles como salidas PWM. Se puede añadir una alimentación opcional sobre el módulo Ethernet.

La tarjeta Ethernet se diferencia de otras placas en que no tiene un chip integrado controlador de USB a serie, pero tiene una interfaz Wiznet Ethernet. Esta es la misma interfaz que se encuentra en el escudo Ethernet.

Un lector de tarjetas microSD a bordo, que se puede utilizar para almacenar archivos para servir a través de la red, es accesible a través de la Biblioteca SD. Pin 10 está reservado para la interfaz Wiznet, para la tarjeta SD está en el pin 4.

La cabecera de programación en serie de 6 pines es compatible con el de serie USB adaptador y también con los cables USB FTDI o con Sparkfun y tableros Adafruit estilo FTDI básicos de USB a serial de descanso. Cuenta con soporte para rearme automático, permitiendo bocetos que se cargan sin necesidad de pulsar el botón de reinicio en el tablero. Cuando se conecta a un adaptador USB a Serial, el Arduino Ethernet es alimentado desde el adaptador. En la tabla 3 se pueden ver las principales características de la tarjeta *shield* Ethernet.

Tabla 3. Características técnicas del Arduino Ethernet, fuente: elaboración autora.

Microcontroladores	ATmega328
Tensión De Funcionamiento	5V
Plug Voltaje de entrada (recomendado)	7-12V
Tensión de entrada (límites)	6-20V
Voltaje de entrada PoE (límites)	36-57V
Digital pines I / O	14 (4 salida PWM)
Pines de entrada analógica	6
Corriente DC por Pin I / O	40 mA
Corriente DC de 3.3V Pin	50 mA
Memoria Flash	32 KB (ATmega328) de los cuales 0,5 KB utilizado por el gestor de arranque
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidad De Reloj	16 MHz

d.1.5.4. Tarjeta Arduino GSM

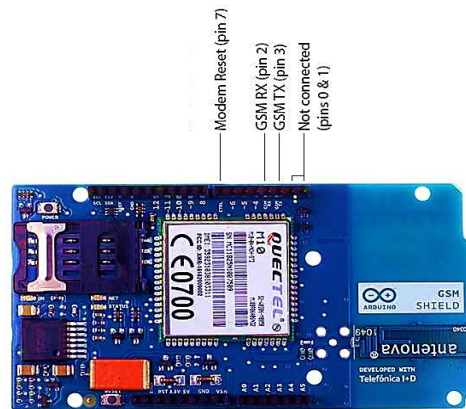


Figura 7. Arduino GSM, fuente elaboración: [28]

La tarjeta Arduino GSM *Shield* permite que una placa Arduino se conecte a Internet, con la cual se pueden realizar/recibir llamadas de voz y enviar / recibir mensajes SMS, utiliza un módem de radio M10 por Quetel. Es posible comunicarse con la placa mediante comandos AT. La biblioteca GSM tiene un gran número de métodos para la comunicación con la tarjeta.

La tarjeta GSM usa los pines digitales 2 y 3 para el software de comunicación serie con el M10. El pin 2 se conecta al pin TX del M10 y el pin 3 a su pin RX. El M10 es un módem cuatribanda GSM / GPRS que funciona en las frecuencias GSM850MHz, GSM900MHz, DCS1800MHz y PCS1900MHz. Es compatible con los protocolos TCP / UDP y HTTP a través de una conexión GPRS. GPRS enlace descendente de datos y la transferencia de enlace ascendente de velocidad máxima es de 85,6 kbps, para interactuar con la red celular, se requiere una tarjeta SIM proporcionada por un operador de red.

Se recomienda que la placa se alimente con una fuente de alimentación externa que puede proporcionar entre 700 mA y 1000 mA. Encender un Arduino y el *shield* GSM desde una conexión USB no es recomendable, como USB no puede proporcionar la corriente requerida para cuando el módem está en uso intensivo.

El escudo apoya AIN1 y AOUT1 como interfaces de audio; un canal de entrada analógica y un canal de salida analógica. La entrada, expuesta en los pines MIC1P / MIC1N, se puede utilizar tanto para entradas de micrófono y de línea. Un micrófono electret se puede

utilizar para esta interfaz. La salida, expuesta como líneas SPK1P / SPK1N, se puede utilizar ya sea con un receptor o el altavoz. A través del módem, es posible realizar llamadas de voz. Con el fin de hablar y escuchar a la otra parte, tendrá que añadir un altavoz y un micrófono.

d.1.5.5. Descripción de la placa Arduino



Figura 8. Descripción de la placa Arduino, fuente de elaboración: [28]

La placa Arduino está formada por las siguientes componentes en la figura 8 se puede ver enumerado cada uno de los componentes.

- 1. Conector USB:** provee la comunicación para la programación y la toma de datos, además tiene una fuente de 5VDC para alimentar la placa pero de bajo potencia ya que no sirve para alimentar motores de gran potencia.
- 2. Regulador de voltaje de 5V:** Su función es transformar el voltaje que ingresa por el pin 3 en un voltaje de 5V que es lo que necesita la placa para su correcto funcionamiento así como para alimentar los circuitos externos.
- 3. Plug de conexión para fuente de alimentación externa:** este conector permite que se conecte una fuente externa la misma que debe suministrar entre 6V y 18V o hasta 20V, se debe tener en cuenta que el terminal del centro quede conectado a positivo ya que algunos adaptadores tienen la opción de intercambiar la polaridad de los cables.
- 4. Puerto de conexiones:** Está formado por 6 pines de conexión con funciones de RESET que permite resetear el microcontrolador al enviarle un cero lógico. El pin

3V3 que dispone de una fuente de 3,3VDC para conectar componentes externos como para trabajar en el protoboard, dos pines GND que permite la salida de cero voltios para componentes externos. Pin Vin está conectado con el positivo del plug 3 que se utiliza para conectar la alimentación de la placa con una fuente externa de entre 6 y 12 VDC.

- 5. Puertos de entradas analógicas:** puertos que sirven como entradas las salidas de los sensores analógicos, dichos pines funcionan solo entradas recibiendo voltajes entre 0 y 5 voltios directos.
- 6. Microcontrolador atmega 328:** Usado por la plataforma Arduino en algunos de los modelos inicialmente en el Arduino uno, esto permite que programar las tarjetas Arduino sean más fáciles.
- 7. Botón reset:** Permite resetear el microcontrolador haciendo que vuelva el programa al modo predeterminado de inicio.
- 8. Modos de programación ICSP:** Son usados para programar para microcontroladores en el protoboard o en PCB sin tener que retirarlos de lugar.
- 9. Led ON:** indicador que permite identificar cuando el Arduino está encendido.
- 10. Leds de recepción y transmisión:** Se encienden cuando la tarjeta se comunica con el PC, el Tx muestra cuando se transmiten los datos y Rx la recepción.
- 11. Puertos de conexiones de pines de entradas o salidas digitales:** la configuración de entradas o salidas debe ser incluida en el programa. Cuando se utiliza la terminal serial se recomienda no utilizar los pines cero (Rx) y uno (Tx). Los pines 3,5 y 6 están señalados por el símbolo ~, lo que indica que permiten su uso como salidas controladas por ancho de pulso PWM.
- 12. Puerto de conexiones 5 entradas o salidas adicionales:** Las salidas 9,10 y 11 permiten control por ancho de pulso; la salida, además hay un una salida AREF que permite ser empleado como referencia para las entradas análogas.
- 13. Led pin 13:** indica el estado en que se encuentra.
- 14. Chip de comunicación:** Permite la conversión de serial a USB.

En el diagrama de bloques de la figura 9, se puede ver la descripción de una placa Arduino.

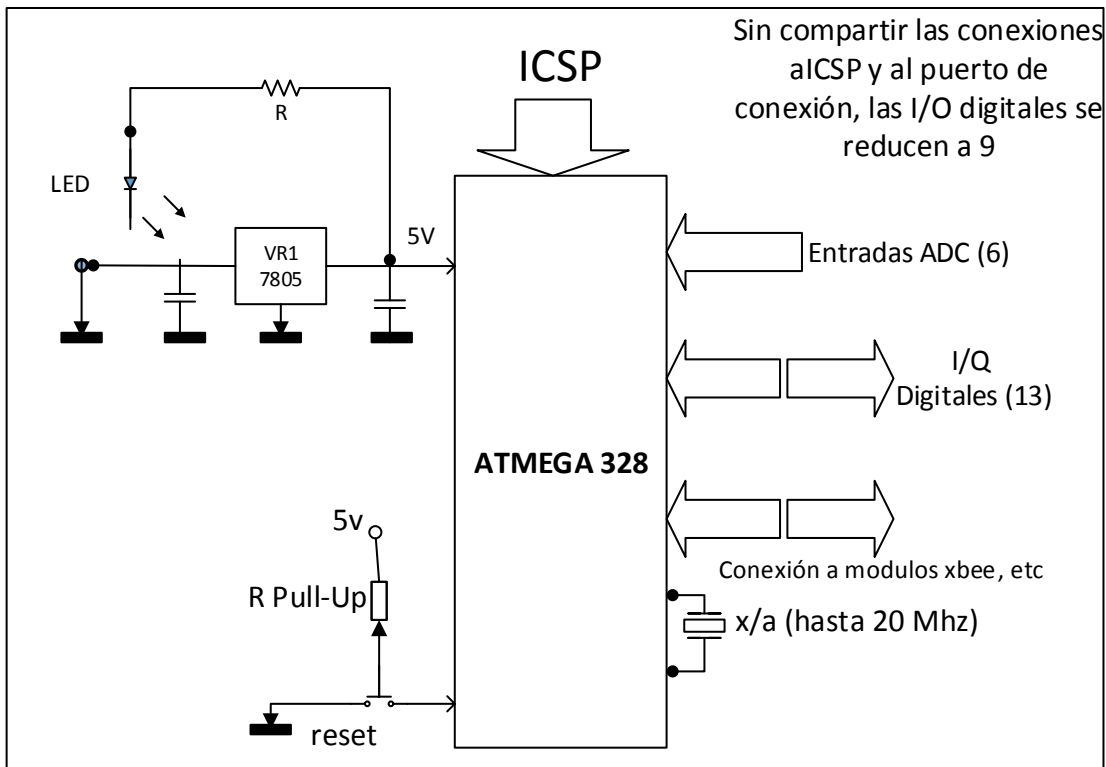


Figura 9 Diagrama de bloques de una placa Arduino, fuente: elaboración autora

d.1.5.6. Programación

La placa Arduino se puede programar a través del IDE (Entorno de Desarrollo Integrado) de Arduino. Primero hay que decirle al IDE nuestro tipo de placa. Para ello, selecciona bien "Arduino Duemilanove w/ ATmega328" o bien Arduino Uno del menú Herramientas > Tarjeta. También hay que decirle al IDE en qué puerto USB hemos conectado la placa. Lo haremos a través del menú Herramientas > Puerto Serial. Aparecen una serie de puertos denominados COMx, donde x es un número. Para saber cuál es, desconecta y conecta el cable USB del ordenador, y el que desaparezca y vuelva a aparecer es nuestro puerto.

d.2. CAPÍTULO II: Diseño y Construcción del Hardware

d.2.1 Diagrama de bloques del sistema de telemetría del proyecto

El sistema de telemetría se lo ha simplificado en el diagrama de bloque como se lo puede ver en la figura 9:

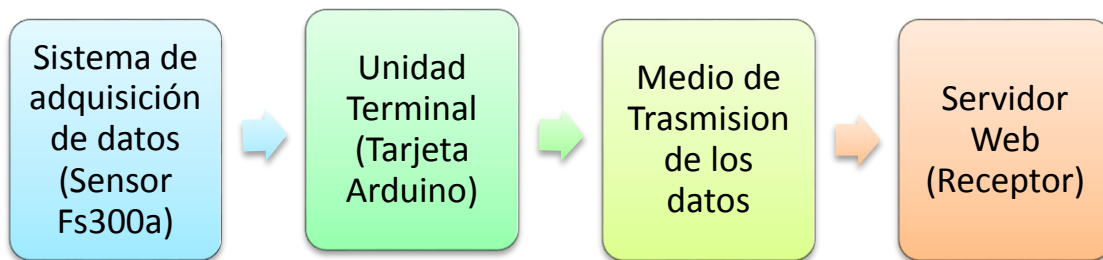


Figura 10. Diagrama de bloques del sistema de telemetría, fuente: elaboración autora

d.2.2. Sistema de adquisición de datos

El sistema de adquisición de datos está basado principalmente en el sensor de flujo de agua FS300a, mediante el cual se puede obtener la señal del flujo de agua permitiendo de esta manera al usuario poder hacer un análisis del consumo de agua.

d.2.2.1. Sensor efecto Hall

El efecto Hall, descubierto por Edwin C. Hall en 1879, se genera por la producción de una caída de voltaje a través de un conductor o semiconductor con corriente, bajo la influencia de un campo magnético externo. Para esto es necesario que la dirección del campo magnético sea perpendicular a la dirección de flujo de la corriente.

El campo magnético transversal ejerce una fuerza desviadora (Fuerza de Lorentz) sobre el conductor o semiconductor. Esta fuerza causa la desviación de los portadores de carga que se mueven a través del material. Como resultado, aparece una diferencia de potencial V_{xy} (denominada voltaje de Hall) entre los extremos del conductor. Este voltaje es proporcional a la intensidad del campo magnético aplicado y su polaridad depende del signo de los portadores de carga.

d.2.2.2. Sensor de caudal de agua FS300a

El sensor de flujo de agua consiste en un cuerpo de plástico como válvula, un rotor de agua, y un sensor de efecto Hall. Cuando el agua fluye a través del rotor, el rotor rueda. Su velocidad cambia con diferente tasa de flujo. El sensor de efecto Hall emite la señal de impulso correspondiente. [29]

d.2.2.3. Especificaciones Técnicas del sensor

- Min. Voltaje de trabajo: DC 4.5V
- Max. Trabajo actual: 15 mA (DC 5V)
- Voltaje de funcionamiento: DC 5V ~ 24V
- Flujo Rango de Tarifas: 1 ~ 60litros/min
- Capacidad de carga: ≤ 10 mA (DC 5V)
- Temperatura de funcionamiento: ≤ 80 °C
- Temperatura del líquido: ≤ 120 °C
- Humedad de funcionamiento: 35% ~ 90% RH
- Presión del agua: ≤ 1.75 MPa
- Temperatura de almacenamiento: -25 ~ + 80 °C
- Humedad de almacenamiento: 25% ~ 95% RH

Dimensiones Mecánicas

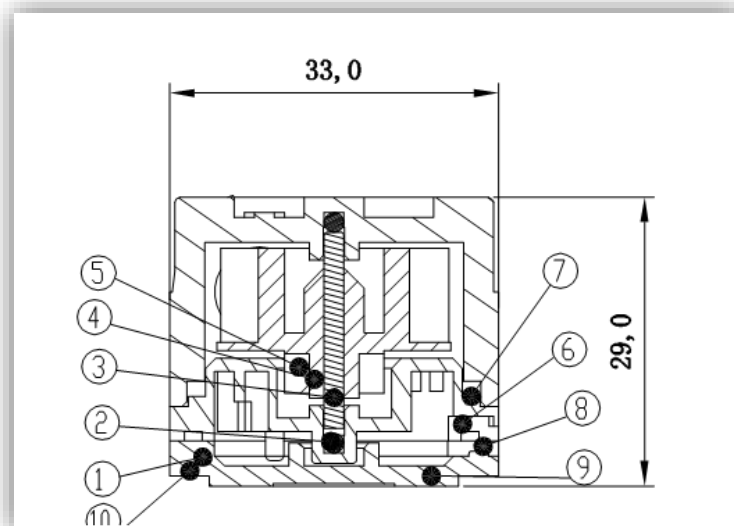


Figura 11 componentes del sensor Fs300a, fuente de elaboración: [29]

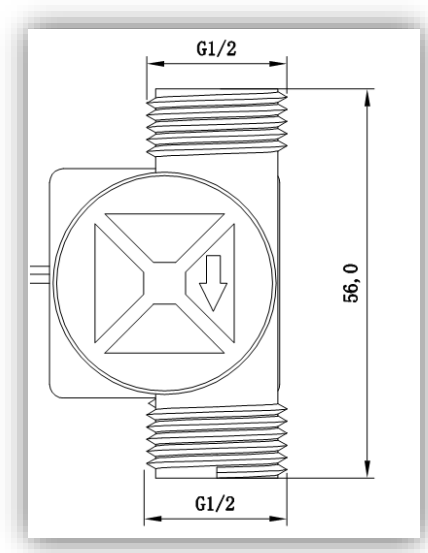


Figura 12 dimensiones del sensor Fs300a, fuente de elaboración [29]

Componentes del sensor

Tabla 4 Componentes del sensor Fs300a [29]

1	Nombre	Cantidad	Material
1	El cuerpo de válvula	1	PA66 33% de fibra de vidrio
2	Recorte de acero inoxidable	1	Acero inoxidable SUS304
3	Eje	1	Acero inoxidable SUS304
4	Impulsadorw2	1	POM
5	Imán de anillo	1	Ferrita
6	Anillo de Medio	1	PA66 33% de fibra de vidrio
7	Junta teórica de sellado	1	Caucho
8	Anillo de sello electrónico	1	Caucho
9	Cubierta	1	PA66 33% de fibra de vidrio
10	Tornillo	1	Acero inoxidable SUS304
11	Cable	1	1007 24 AWG

La figura 11, 12 y la tabla 4 presentan los diferentes componentes que constituyen el sensor

d.2.3. Selección de la tarjeta Arduino para la adquisición de datos

Para la elección de la tarjeta Arduino se ha hecho un análisis detallado de las características de cada tarjeta y determinar cuál se adapta mejor al proyecto, uno de los parámetros a tomar en cuenta son los puertos de entrada y salida del Arduino. La tarjeta Uno R3 dispone de 16 I/O y 6 entradas analógicas, además dispone de una memoria RAM 32kb que es lo suficiente para el proyecto, por eso se tomó en cuenta el Arduino Uno,

haciendo una comparación con el Arduino Mega se determinó que esta tarjeta tiene mayor memoria RAM de 256kb, y por ende mayor capacidad de almacenamiento, además se comparó con otras tarjetas Arduino como la Leonardo que posee una memoria de RAM de 32Kb que es muy similar al Arduino Uno, la diferencia es que el Arduino Mega tiene más entradas y salidas digitales así como salidas, a continuación en la tabla 5 se muestra las características más importantes de las tarjetas Arduino.

Tabla 5 Resumen de las principales tarjetas Arduino, fuente: elaboración autora

Modelo	I/O digitales	Entradas analógicas	Salidas PWR	UART	Memoria
Uno r3	16	6	6	1	32kb
Leonardo	20	12	7	1	32kb
Mega r3	54	16	14	4	256kb
Ethernet					
Mini 05	14	6	8	1	32kb

d.2.3.1. Diagrama de Pines del Atmega 328

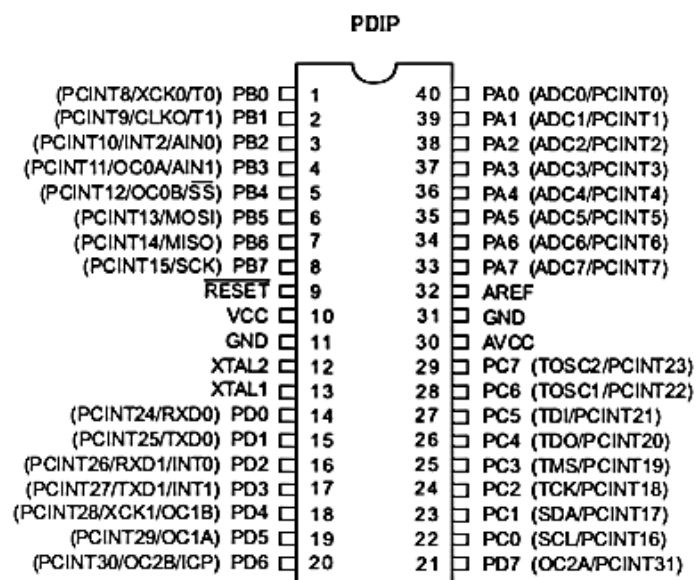


Figura 13. Pines de Salida del ATmega164P/324P/644P, [28]

Descripción de Pines

En la figura 13 se muestra los pines utilizados en el proyecto y que a continuación se describen:

Pin 10 VCC

Alimentación de Voltaje Digital

Pin 11 y 31 GND

Tierra

Pin 40 al 33 puerto A (PA7:PA0)

El puerto A sirve como entradas analógicas para el conversor Análogo Digital.

El puerto A también sirve como un puerto bidireccional de 8 bits con resistencias internas de pull up (seleccionables para cada bit). Los buffers de salida del puerto A tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puerto A están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no este corriendo. El puerto A también sirve para varias funciones especiales del ATmega164P/324P/644P como la Conversión Análoga Digital.

Pin 1 a 8 Port B (PB7:PB0)

El puerto B es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up.

Las salidas de los buffers del puerto B tienen características simétricas controladas con fuentes de alta capacidad. Los pines del puesto B están en tri-estado cuando las condiciones de reset están activadas o cuando el reloj no esté corriendo. El puerto B también sirve para varias funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

Pin 22 al 29 Port C (PC7:PC0)

El puerto C es un puerto bidireccional de 8 bits de E/S con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto C tienen características simétricas controladas con fuentes de alta capacidad.

Los pines del puesto C están en tri-estado cuando las condiciones de reset están activadas siempre y cuando el reloj no este corriendo. El puerto C también sirve para las funciones de Interfaz del JTAG, con funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

Pin 14 al 20 Port D (PD7:PD0)

El Puerto D es un puerto bidireccional de entradas y salidas con resistencias internas de pull up (seleccionadas por cada bit). Las salidas de los buffers del puerto D tienen características simétricas controladas con sumideros de fuentes de alta capacidad.

Los pines del Puerto D están en tri-estado cuando llega una condición de reset activa, siempre y cuando el reloj no esté corriendo.

El puerto D también sirve para varias funciones especiales del ATmega164P/324P/644P como se menciona en las páginas iniciales.

Pin 9 RESET

Entrada del Reset. Un pulso de nivel bajo en este pin por períodos de pulso mínimo genera un reset, siempre y cuando el reloj no esté corriendo.

La longitud del pulso mínimo está especificada en las Características y Sistemas del Data Sheet. Pulsos cortos no son garantizados para generar un reset.

Pin 13 XTAL1

Entrada para el amplificador del oscilador invertido y entrada para el circuito de operación del reloj interno.

Pin 13 XTAL2

Salida del Oscilador amplificador de salida.

Pin 30 AVCC

AVCC es la alimentación de voltaje para el pin del Puerto F y el Conversor Análogo a Digital. Este debe ser conectado externamente a VCC, siempre y cuando el ADC no sea usado. Si el ADC es usado, este deberá ser conectado a VCC a través de un filtro paso bajo.

Pin 32 AREF

Esta es la referencia para el pin de la conversión Análoga a Digital.

d.2.4. Selección de la tarjeta *shield* para la comunicación con el Internet

Para la selección del medio que permita la comunicación con el mundo exterior se realizará un análisis detallado de los medios disponibles para determinar cuál es la mejor opción y se adapta al medio.

Medios Guiados: En primer lugar se analizara la red guiada, la característica principal de esta red es que las ondas electromagnéticas viajan a través de un cable o alambre conductor, pero no se adapta al proyecto debido a que en el proyecto se propone el envío de los datos de manera inalámbrica, además realizar una red cableada sería demasiado costosa y conllevaría mucho tiempo en construir.

Medios Inalámbricos: La principal característica de estos medios es la transmisión de las ondas electromagnéticas utilizando como medio de transmisión el espacio, estos medios ofrece algunas alternativas que podrían ser de gran utilidad para la transmisión de los datos hasta el servidor web, entre ellas tenemos la tecnología WIFI 802.3, ofrece una gran cobertura, fácil de instalar, pero gran costo para su implementación debido a que se necesitaría implementar una red WIFI para la transmisión de los datos por tal motivo quedó descartada; otra alternativa es la tecnología Bluetooth que permite la transmisión de los datos a una distancia de 5m este es un gran inconveniente para el proyecto debido que se necesita enviar la información hasta un servidor web por esto queda descartada. Las tecnologías móviles ofrecen una alternativa para la transmisión y recepción de los datos, una ventaja que ofrecen es que estas redes ya están implementadas y además existen tarjetas auxiliares en Arduino que facilitarían que se pueda realizar esta comunicación, lo que se tendría que hacer es contar con un plan de datos en cualquier operadora celular, esto ahorra tiempo y dinero siendo la mejor opción para nuestro proyecto, esto se lo puede hacer mediante la red GPRS que es una extensión de la red GSM.

d.2.5. Diagrama de conexión del prototipo

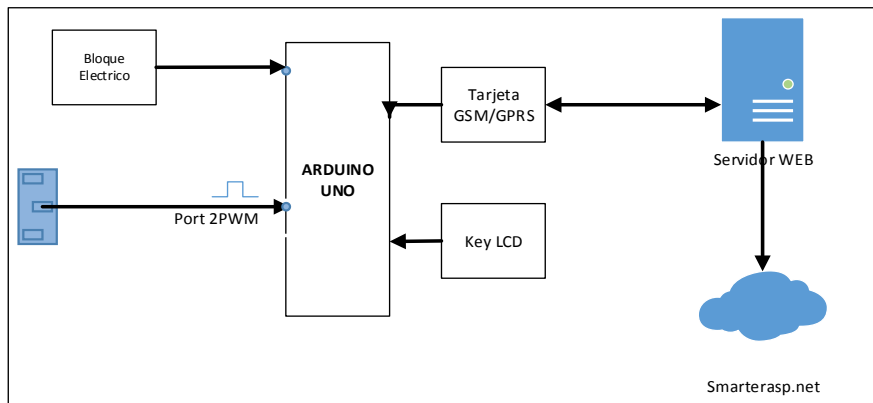


Figura 14 Diagrama de conexión del sistema de telemetría, fuente: elaboración autora

d.2.6. Descripción del diagrama de conexión

Una vez que se ha realizada minuciosamente la selección de los componentes para la construcción del prototipo se procedió a realizar el diseño del circuito en la figura 14 se muestra el diagrama bloques, en primer lugar tenemos el sensor de caudal FS300a el cual permite obtener los datos del medidor de agua como ya se mencionó el sensor tiene efecto hall, la señal de salida del sensor es PWM la cual se conecta al pin 2 de la tarjeta Arduino UNO que es una entrada PWM, una vez que se han obtenido los datos del medidor de la casa del usuario en nuestro caso del medidor de la Universidad Nacional de Loja, se programa la tarjeta Arduino para que reconozca los datos del sensor y poderlos transmitir hasta el servidor web, para la transmisión de la información hasta el servidor se utiliza una tarjeta *shield* GSM/GPRS la cual va conectada en el pin 0 para transmisión Tx y el pin 1 para recepción Rx, la tarjeta Arduino se la debe configurar para poder enviar la información utilizando la red GPRS, posteriormente se conecta la tarjeta *Shield* LCD la cual va a permitir visualizar los datos de medición del sensor así como manejo manual de los datos, esta información será almacenada en el servidor web.

d.3. CAPÍTULO III: Diseño y Construcción de Software

d.3.1. Entorno de programación Arduino

El entorno de programación programa la tarjeta Arduino desde un ambiente gráfico, contiene un editor de texto para escribir código, un área de mensajes, una consola de texto, una barra de herramientas con botones para funciones comunes, y una serie de menús. Se conecta al hardware Arduino para cargar programas y comunicarse con ellos.

d.3.1.1. Primera ejecución del IDE de Arduino y configuración inicial para desarrollar proyectos hardware



Tras instalar el controlador de Arduino, ahora realizaremos una primera ejecución para configurar el IDE de desarrollo, se pulsa el botón derecho del ratón sobre el ejecutable "arduino.exe" de la carpeta descomprimida anteriormente y seleccionaremos "Ejecutar como administrador"





En el IDE de desarrollo de Arduino, en primer lugar seleccionaremos el tipo de dispositivo (Board). Para ello pulsaremos en el menú "Tools" - "Board" y seleccionaremos "Arduino Uno" (o el que hayamos adquirido):

Seleccionaremos también el puerto serie asignado al controlador de Arduino (en nuestro caso COM3), para ello accederemos al menú "Tools" - "Serial Port" - "COM3": en la tabla 6 se puede ver las opciones de interfaz.

Opciones de la interfaz

Tabla 6 de opciones de interfaz IDE, Fuente: elaboración [28]

	Verificar Cheques su código de errores.
	Verificar <i>Cheques su código de errores.</i>
	Subir Compila el código y lo sube a la placa Arduino E / S. Ver la posibilidad de subir los detalles a continuación.

	<p>Nota: Si está utilizando un programador externo, puede mantener pulsada la tecla "shift" en su ordenador cuando se utiliza este icono. El texto cambiará a "Subir usando programador"</p>
	<p>Nuevo</p> <p>Crea un nuevo boceto.</p>
	<p>Abrir</p> <p>Presenta un menú con todos los bocetos en su cuaderno de dibujo. Al hacer clic en uno se abrirá dentro de la ventana actual.</p> <p>Nota: debido a un error en Java, este menú no se desplaza; si usted necesita para abrir un boceto final de la lista, utilice el archivo Sketchbook menú en lugar.</p>
	<p>Guardar</p> <p>Guarda el boceto.</p>
	<p>Serial Monitor</p> <p>Abre el monitor serie.</p>

Comandos adicionales se encuentran dentro de los cinco menús: Archivo, Editar, Sketch, Herramientas, Ayuda.

d.3.1.2. Librerías

El entorno Arduino se puede extender a través del uso de las librerías, las mismas proporcionan funcionalidad adicional para uso en sketches, por ejemplo, trabajar con el hardware o la manipulación de los datos. Para utilizar una biblioteca en un sketches, Un número de bibliotecas viene instalada con el IDE, pero también se puede descargar o crear uno sus propias librerías, se puede importar una biblioteca de un archivo zip y utilizarlo en un sketches abierta.

Para utilizar una biblioteca en un sketches, selecciónelo de Sketch> Import Library.

d.3.1.3. Librerías Estándar [28]

EEPROM - leer y escribir en el almacenamiento "permanente"

Ethernet - para conectarse a Internet a través de la Ethernet *Shield* Arduino

Firmata - para comunicarse con aplicaciones en el equipo utilizando un protocolo de serie estándar.

GSM - para la conexión a una red / GRPS GSM con el escudo GSM.

LiquidCrystal - para el control de pantallas de cristal líquido (LCD)

SD - para la lectura y escritura de tarjetas SD

Servo - para el control de servomotores

SPI - para comunicarse con los dispositivos que utilizan la Interfaz Periférico Serial (SPI) Autobús

Software Serial - para la comunicación de serie en cualquier pines digitales. Versión 1.0 y posterior de Arduino incorporan Mikal Hart 's NewSoftSerial biblioteca como Software Serial.

Paso a paso - para el control de motores paso a paso

TFT - para dibujar texto, imágenes, y las formas en la pantalla TFT de Arduino

WiFi - para conectar a Internet utilizando el Arduino WiFi escudo

Alambre - Dos Interfaz alambre (TWI / I2C) para enviar y recibir datos a través de una red de dispositivos o sensores.

d.3.2. Diagrama de flujo del funcionamiento del sistema de telemetría

A continuación se describe de forma general el algoritmo para el funcionamiento del sistema de telemetría como se puede ver en la figura 15.

El algoritmo inicia con la instalación de las librerías necesarias para el desarrollo del programa, a continuación se declararon los puertos de E/S (entrada y salida) de la tarjeta Arduino UNO, posteriormente se utiliza el comando AT-HTTPxx para establecer la conexión con la red GSM/GPRS utilizando la tarjeta *shield* GSM/GPRS una vez establecida la conexión de datos móviles se leen la lectura de los datos que da el sensor de flujo de agua por último se envían los datos al servidor.

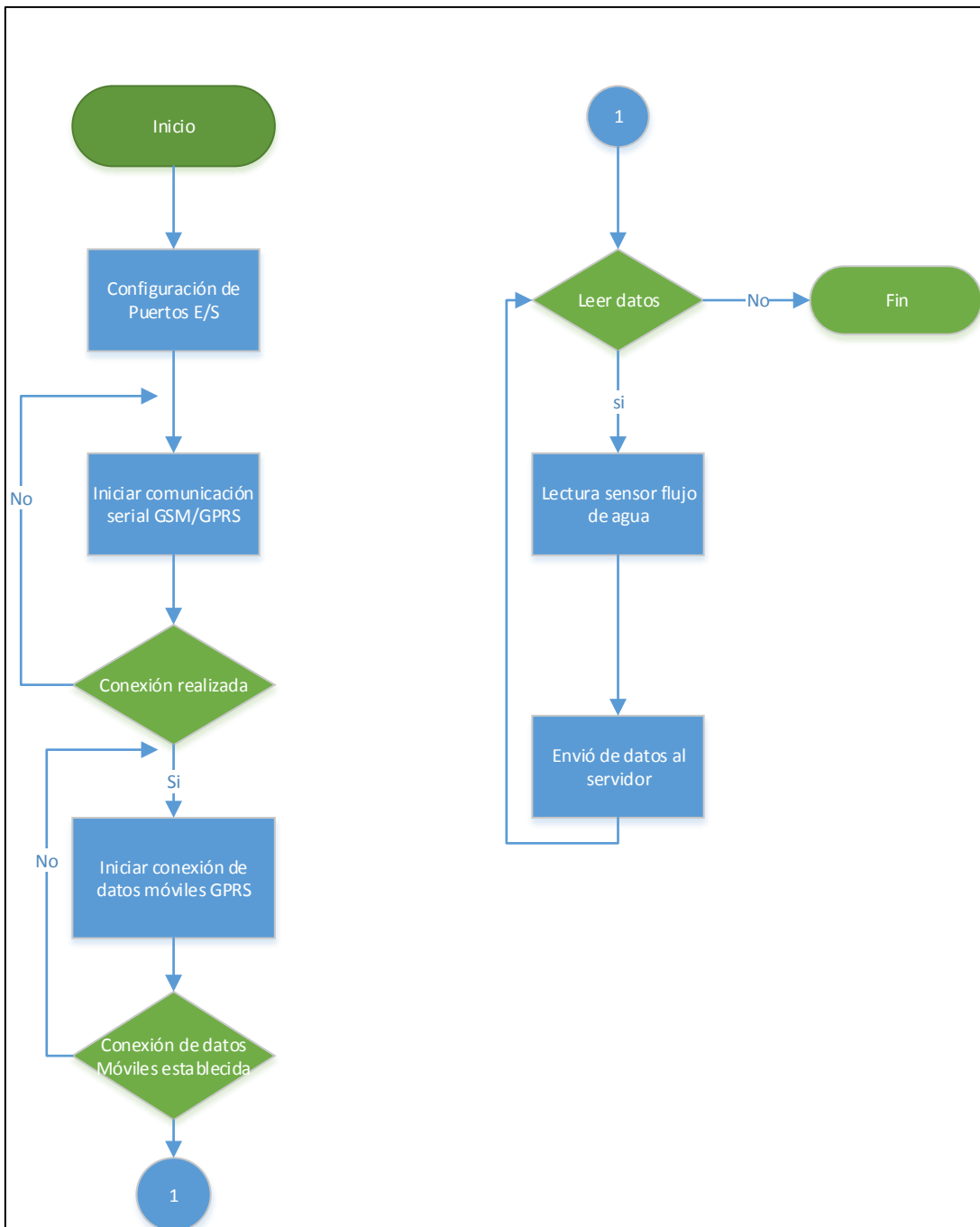


Figura 15 Diagrama de flujo del funcionamiento general del sistema, fuente: elaboración autora

d.3.2.1. Cálculo del flujo de agua

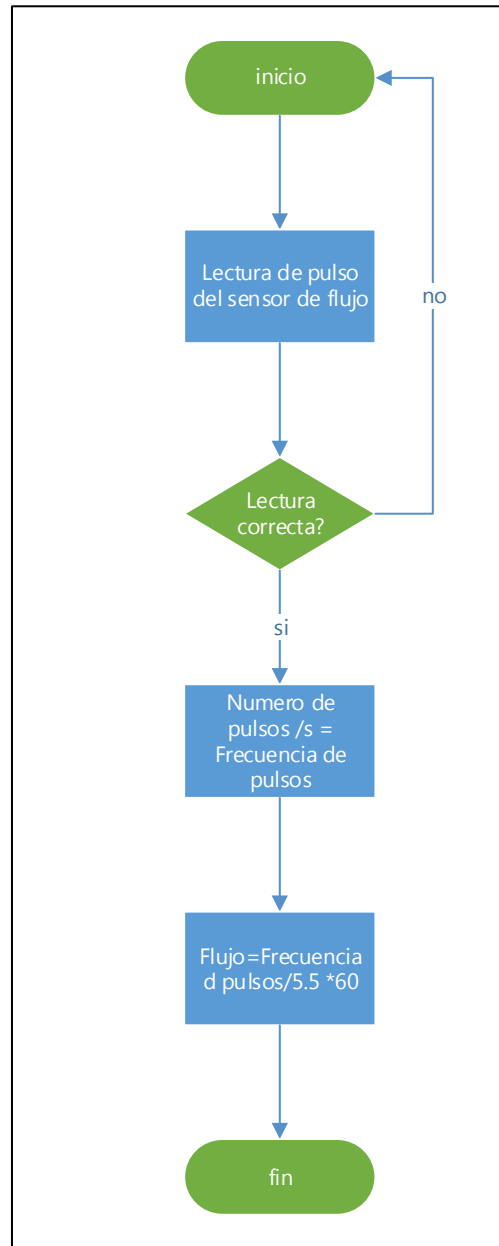


Figura 16 Algoritmo del cálculo de flujo de agua, fuente: Elaboración autora

En la figura 16 se muestra la estructura de funcionamiento del sensor de agua fs300a g3/4, el cual es digital y consta de un núcleo magnético, un rotor y un sensor efecto-hall. Cuando el flujo de agua pasa a través del rotor este activa el núcleo magnético que excita al sensor efecto – hall, mientras más volumen de agua pasa por el tubo, mayor es la velocidad a la cual gira el rotor, por lo tanto se produce señales de pulsos eléctricos a la salida del sensor de agua efecto-hall, dichos pulsos sirven para hallar la frecuencia con la que gira el

rotor y posteriormente se puede hallar el flujo por unidad de tiempo expresada en Litros/minuto.

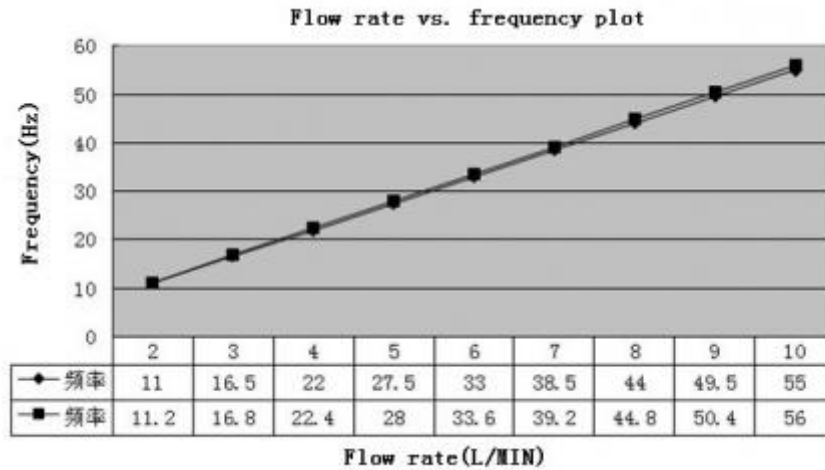


Figura 17 Grafica de la constante (pendiente), Fuente: [29]

Fórmula:

$$\text{Frecuencia de Pulso} = \text{Flujo} * 5.5 \text{ (Según la referencia del fabricante)}$$

La constante 5.5 proviene de la gráfica (es la pendiente), la fórmula es de la forma $y=mx$; como se quiere calcular el flujo, se despeja la fórmula:

$$\frac{\text{Frecuencia de Pulso}}{5.5} = \text{Flujo} \quad (1)$$

Al utilizar esta fórmula, se tiene el flujo en Litros/minuto, con la Frecuencia de Pulso en Hz.

Si se quiere el flujo en Litros/Hora, simplemente se aplica la conversión, teniendo en cuenta que 60 minutos son 1 hora:

$$\text{Flujo} \left[\frac{L}{\text{Hora}} \right] = \frac{\text{Frecuencia de Pulso}}{5.5} * 60 \quad (2)$$

Para obtener la Frecuencia de Pulso en Hz, si yo cuento el número de pulsos (provienen del pin SIG) en 1 segundo, entonces yo habría leído Pulsos/segundo, que es frecuencia $\text{Hz}=1/\text{s}$, entonces con ese valor, se puede aplicar la fórmula arriba explicada y obtener Litros/minutos, o mililitros/hora.

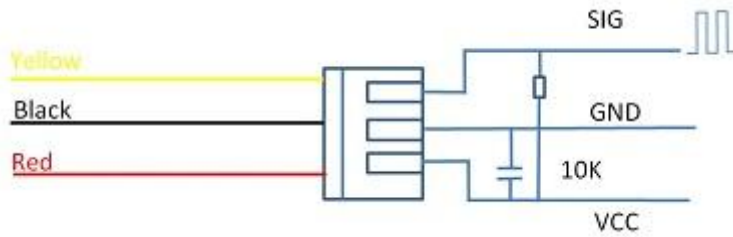


Figura 18 Diagrama de conexión del sensor, fuente: [29]

d.3.3. Configuración de la tarjeta *shield* GSM/GPRS

Para la configuración de la tarjeta *shield* GSM/GPRS se establece la conexión serial con la tarjeta Arduino UNO, como nuestro objetivo es poder transmitir los datos hasta la web con la ayuda del comando `AT+CGATT=1` se enlaza al servicio a la red GPRS, posteriormente se configura los parámetros de GPRS `AT+SAPBR=3,1,"CONTYPE","GPRS"`, si los parámetros son correctos se ingresan las credenciales del APN `AT+SAPBR=3,1,"APN","internet.claro.com.ec"` para el proyecto se utilizó `www.internet.claro.com`, si todo es correcto se abre la conexión con la portadora, cuando se conecta se consulta la dirección IP, la tarjeta Arduino está lista para empezar enviar los datos hacia la web. Como se muestra en la figura 18.

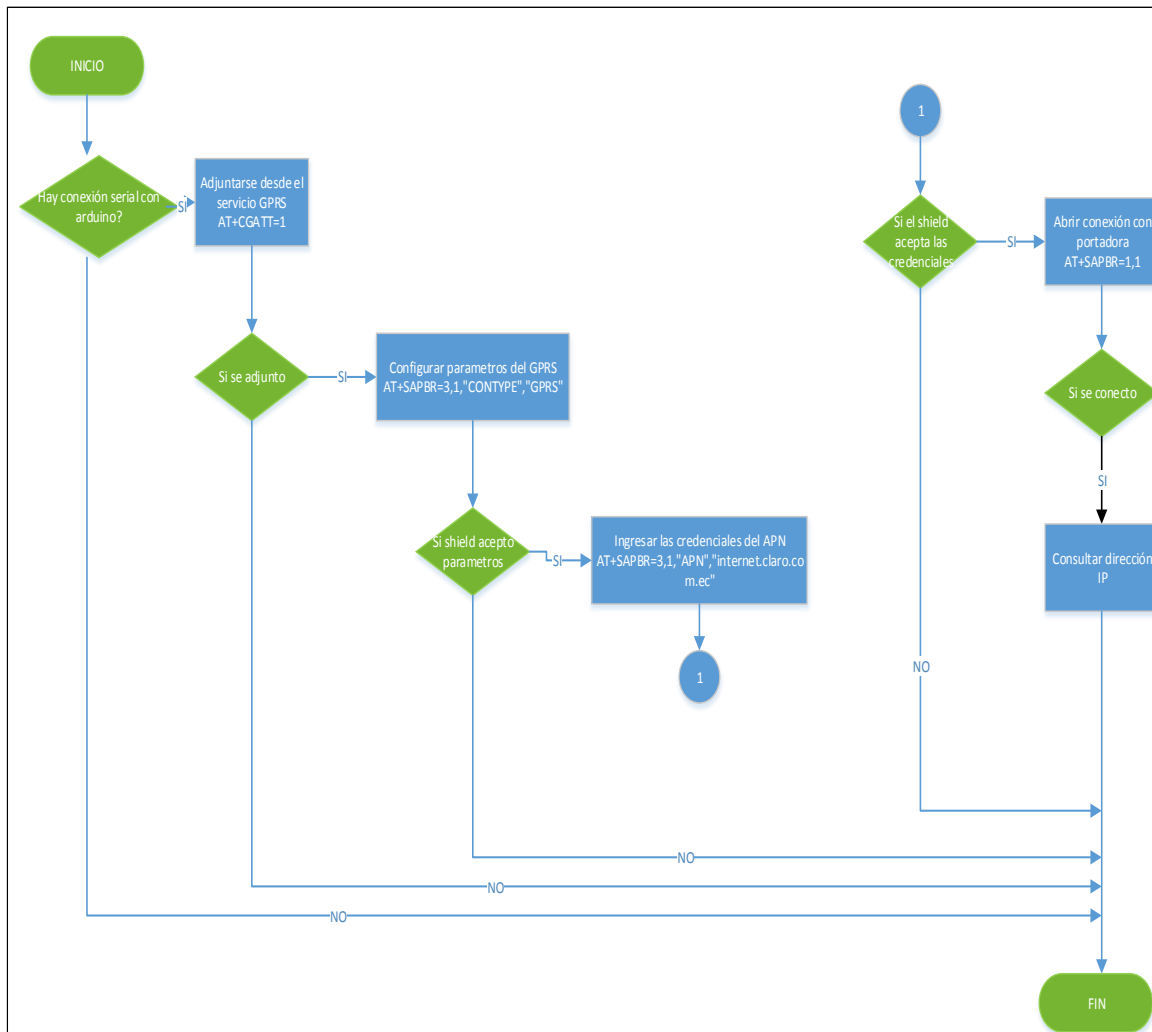


Figura 19 Algoritmo de configuración de la tarjeta shield GSM, fuente: elaboración autora

d.3.3.1. Envío de datos hacia la web

Una vez configurada la tarjeta *shield* GSM SIM 900 para enviar los datos por la red GPRS se inicia el servicio HTTP (At+HTTPIPINIT), posteriormente se establece los parámetros HTTP, como por ejemplo la URL u otros parámetros como la redirección automática por defecto en off, en nuestro caso se arma la cadena de la trama HTTP: "AT+HTTTPARA={"URL"}, {"http://www.pgprs.somee.com/ServicioGPRS.svc/Data?usuario=" + usuario + "&clave=" + clave + "&valor=" + valorSensor + "\""; //armar URL, luego se realiza el método especificado. En nuestro este caso se trata de un HTTP GET, después lee los datos del servidor por ultimo termina el servicio. En la tabla 7 se puede ver las funciones de las teclas del Key LCD.

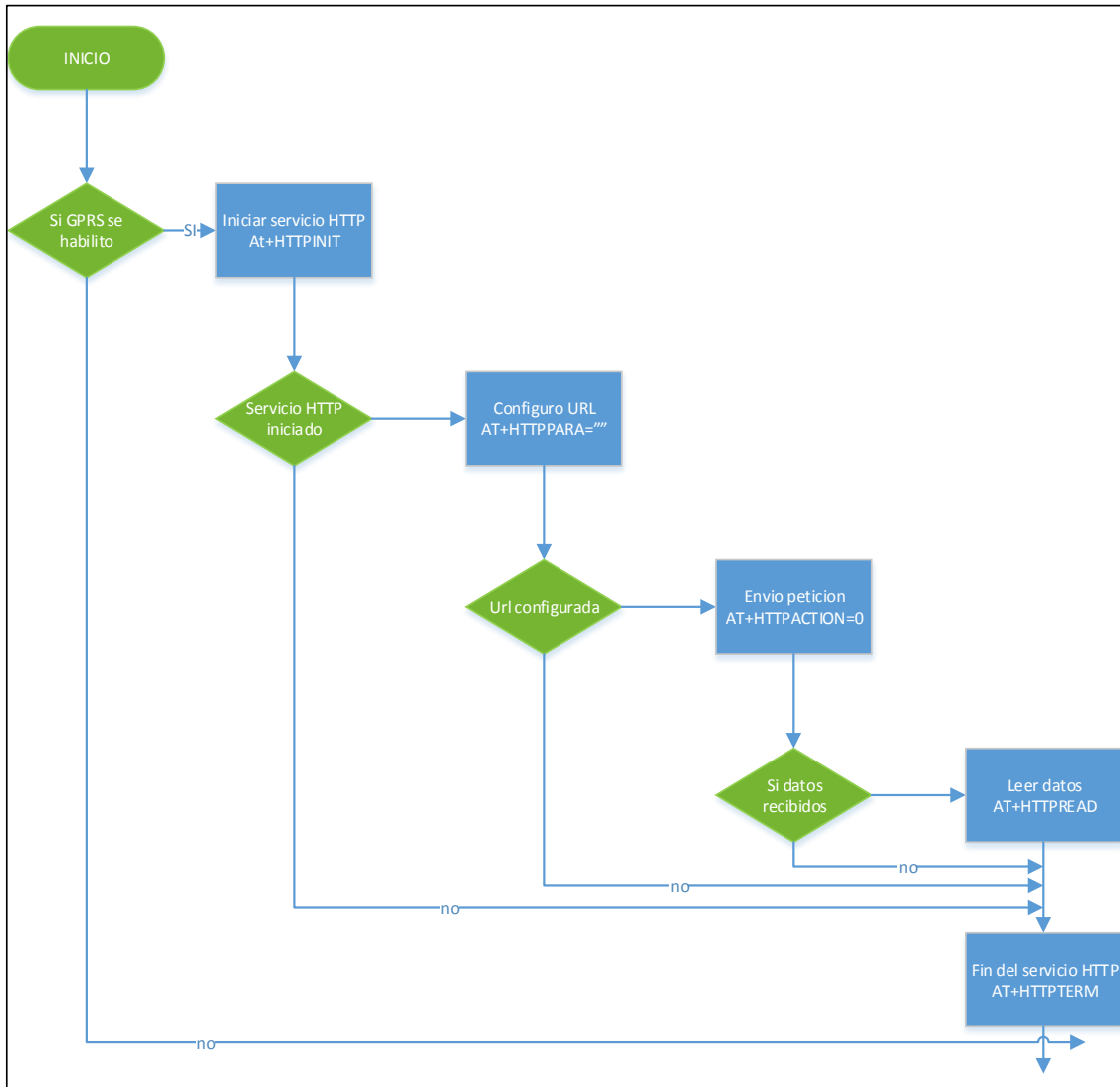


Figura 20 Algoritmo de envio de datos hasta la web, fuente: elaboración autora

d.3.4. Arduino Teclado LCD Shield (SKU: DFR0009)

La tarjeta *Teclado LCD* se desarrolla para tableros compatibles Arduino, para proporcionar una interfaz fácil de usar que permite a los usuarios pasar por el menú, hacer selecciones etc. Consta de un 16x2 caracteres blanco azul de retroiluminación LCD. El teclado se compone de 5 teclas - seleccione, arriba, derecha, abajo e izquierda. Para guardar los pines IO digitales, la interfaz de teclado utiliza sólo un canal ADC. El valor de la clave se lee a través de un divisor de tensión 5 etapas.

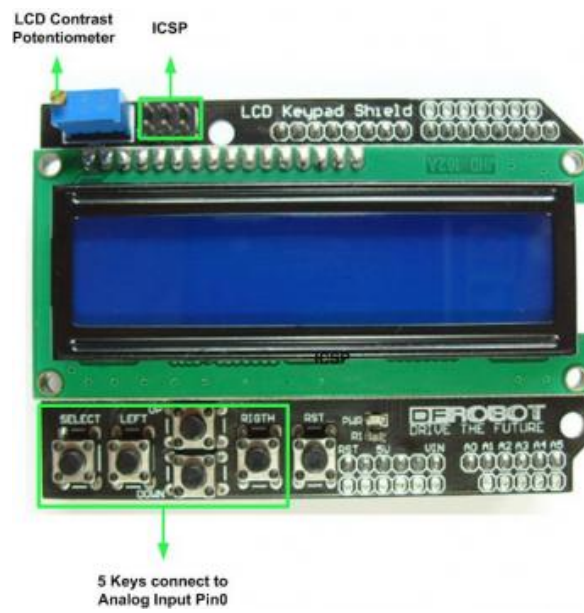


Figura 21 Tarjeta LCD Shield, fuente: hoja de datos del fabricante

Tabla 7. Especificación de los botones de tarjeta LCD, fuente: elaboración autora

Select	Inicializa la comunicación GPRS
Left	Envía manualmente
Up	Subir tiempo con que envía los datos al servidor web
Down	Baja el tiempo con envía los datos al servidor web
Rigth	Cambia L/M, L/S, L/H
RST	Resetea el sistema Arduino

d.3.4.1. Tabla de estados de presentación en pantalla

En la siguiente tabla 8 se muestra detalladamente los diferentes datos que aparecerán en la pantalla LCD.

Tabla 8. Tabla de estados, fuente: elaboración autora

GSM	GPRS	TIEMPO ENVIO	VALOR ENVIO	DATO ENVIADO	TOTAL DE LITROS CONSUMIDOS
E: 1 > comunicación serial con el <i>shield</i> establecida. 0 > error al establecer comunicación con el <i>shield</i> .	0 > Error adjuntando GPRS. 1 > Error intentando configurar el GPRS. 2 > Error configurando los parámetros de acceso a la RED. 3 > Error intentando iniciar conexión en la RED. 4 > Configuración correcta del GPRS.	H > Tiempo de envió en horas. M > Tiempo de envió en minutos. S > Tiempo de envió en segundos.	Valor del tiempo en el que se va a enviar los datos. Depende de tiempo de envío.	0 > Error al iniciar comando HTTP. 1 > Error al crear URL 2 > Error al enviar último dato. 3 > Envío exitoso de último dato.	Valor total de los litros consumidos en un segundo lpm/6

d.3.5. Descripción del entorno emoncms.org utilizado para el registro de datos

[33]

Para el envío de datos desde Arduino UNO R3 hacia el entorno web se ha establecido una conexión con el servidor web emoncms.org, dicha plataforma es una herramienta gratuita eficaz facilitando el trabajo en este tipo de proyectos ya que permite procesamiento, administración y visualización de datos especialmente de energía, además permite transmitir datos de sensores de volumen (agua, gas, velocidad de fluidos, etc). Ese entorno web está es “open source” lo que hace que sea mucho más eficiente puesto que se puede hacer modificaciones para personalizar de acuerdo a las necesidades de cada usuario.

Para poder acceder se debe registrar una vez validada la cuenta se le asigna un apikey de escritura que permitirá vincular Arduino con emoncms para el registro de las entradas y envío de los datos, además un apikey de lectura que permitirá compartir las entradas, alimentación con usuarios externos, un ejemplo de envío se encuentra en la página de emoncms.org en el inciso clave api.

Para la visualización de los datos emoncms presenta un menú que facilita la configuración de las entradas y registro de los datos para lo cual ésta formado cuatro módulos principales que son:

Inputs.- Es donde se registran las variables de los sensores que se monitorean de manera remota, como algo adicional que ofrece es el permite realizar una calibración adicional para mejorar el procesamiento de los datos y adquirir los mismos con la mayor precisión posible.

Para configurar los parámetros de las entradas se lo realiza mediante tres procesos fundamentales, en el que se le designa un nombre al o los procesos que se estén ejecutando, además se especifica el tiempo para el registro de los datos, se debe tomar en cuenta que el tiempo sea mayor o igual al dado para el proceso de transmisión de los datos, por último se realiza la asignación de un proceso como realizar una suma, resta o también colocar el valor máximo o mínimo de los datos que se transmiten, y que como administrador de la cuenta se puede configurar. En la figura 22 se pueden observar los parámetros que se deben configurar.

Node	Key	Name	Process list	last updated	value
0	Litros		log	19 hours ago	0.03
0	Consumo		log	19 hours ago	0.00

Figura 22: Entradas de emoncms.org, Fuente:[33]

- ✓ **Node.-** Hace referencia al módulo que envía los datos al servidor (pueden contener distintas variables).
- ✓ **Name.-** El sistema no sabe qué tipo de datos está recibiendo entonces se debe proporcionar un nombre que se le asigna a la entrada: por ejemplo flujo, Total litros, consumo, etc, se debe tener en cuenta que el nombre sea el mismo que se asigna en la cadena de envío desde Arduino, si no se asignado el nombre automáticamente lo detecta cuando envía el primer dato.
- ✓ **Process list.-** Muestra el tipo de procesamiento que se le haya asignado a cada variable de entrada.
- ✓ **last updated.-** indica el si el nodo está activo o inactivo,

Feeds.- Es donde se gestiona el almacenamiento de los datos de entrada, en este módulo se tiene un ID que es el identificador de un proceso determinado, el cual será utilizado después para crear la interfaz gráfica de la variable sensada y se encarga de pasarlos a los siguientes parámetros, se puede ver la actualización de cada variable de acuerdo al tiempo configurado para su actualización en el servidor, Siempre dentro de este parámetro, se puede visualizar la actividad de envío de datos, tiene algunos campos, el más importante de estos es el ID que corresponde al identificador del proceso, existe otro campo que es “Public” este se configura dependiendo si se desea que los datos sean públicos para que puedan ser vistos por cualquier personas, o simplemente privados que limitarán el acceso a una cuenta segura. La figura 23 podemos observar en detalle las opciones que presta el servidor dentro del parámetro Feed.

Id	Name	Tag	Datatype	Engine	Public	Size	Updated	Value	
90462	node:0:Litros	Node:0	REALTIME	PHPFIWA	🔒	0.0kb	6 mins ago	32.5	✎ 🗑️ 👁️ 🔇
90463	node:0:Consumo	Node:0	REALTIME	PHPFIWA	🔒	0.0kb	6 mins ago	0.03	✎ 🗑️ 👁️ 🔇

Refresh feed size 🔄

Figura 23 Feed de emoncms.org, Fuente:[33]

Vis.- En este módulo se puede visualizar cada una de las variables transmitidas, donde se configura para poder visualizar de manera gráfica el envío de datos de cada variable, cuando se visualiza la gráfica deseada, también se tiene un código html de la misma, del que se puede hacer uso para colocarla en una página web y poder presentarla en tiempo real. El proceso para la visualización de una gráfica se describe a continuación:

- ✓ **Select visualisation.-** En la pestaña que se encuentra a la izquierda de la ventana escogemos el parámetro en el cual se está transmitiendo las variables, en este caso es en tiempo real.
- ✓ **Set options.-** En la pestaña que se encuentra frente a FeddID, escogemos la variable a la que queremos visualizar.
- ✓ **View.-** Al dar click en el icono view nos muestra la gráfica de la variable que llamamos o que deseamos monitorear.
- ✓ **Full screen.-** Nos permite mostrar la gráfica en pantalla completa.

En la parte inferior nos muestra un código html que se describió anteriormente del mismo que se puede hacer uso para publicarla en una página web. En la figura 24 se puede ver lo referente acerca de cada uno de los parámetros.

Visualisations

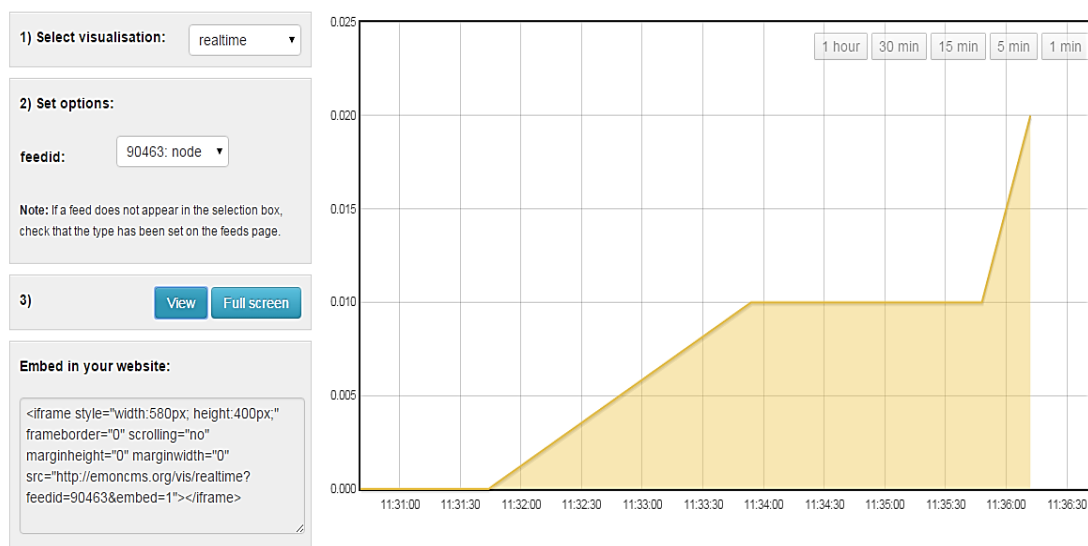


Figura 24: Visualización de variables en emoncms.org. Fuente: (El Autor).

Dashboard.- El Dashboard es la ventana en donde se diseñará la interfaz gráfica, la misma que se presenta para la visualización de las variables, existen tres tipos de configuraciones para la presentación de esta interfaz, cada uno de estos con características distintas, las mismas que describiremos a continuación:

- ✓ **Main page.-** Esta configuración de página nos permite mostrar la aplicación diseñada, únicamente con escribir siguiendo el formato de la siguiente dirección electrónica “emoncms.org/Usuario”.
- ✓ **Page Published.-** Estas páginas son de libre acceso para que sea visitada por cualquier persona, para observar los datos presentados sin opción a editar los mismos.
- ✓ **Page Private.-** Para acceder a estas páginas debe autenticar su cuenta en el servidor emoncms.org.

Para el desarrollo de la interfaz hacemos uso de la barra de herramientas que nos presta en esta ventana dentro del cual tenemos los siguientes:

- ✓ **Text.-** Esto nos permite escribir un texto en el editor de dibujo, existen varios tipos: título centrado, subtítulo, párrafo, con distintos tamaños de letra.
- ✓ **Widgets.-** Permite realizar una animación en tiempo real de los cambios de las entradas, dentro de los cuales tenemos los dial, jguage, feed value entre otros.
- ✓ **Visualisations.-** Las visualizaciones se pueden configurar de varias maneras, dependiendo de los requerimientos gráficos se los puede presentar en tiempo real,

gráficos del valor de los datos, en gráficos de estadísticas diarias, mensuales y anuales, histogramas, multigráficas, entre otros.

- ✓ **Configure.-** Este campo nos permite configurar nuestros widgets, seleccionando el widget que se requiera configurar y luego se establecen algunos parámetros, como unidades de acuerdo a las variables que se estén midiendo.

Se pueden ver los datos de consumo en metros cúbicos así como la gráfica de consumo esta puede ser en intervalos cada 10s, 1,5,10,15,30 minutos o 1 hora, además se puede visualizar el consumo en litros totales además si se desea saber el consumo total también se puede visualizar, el servidor guarda un registro de todos los datos se puede ver incluso los datos de todo un año de consumo en la figura 25 se puede ver los datos de consumo en metros cúbicos y de los litros consumidos.

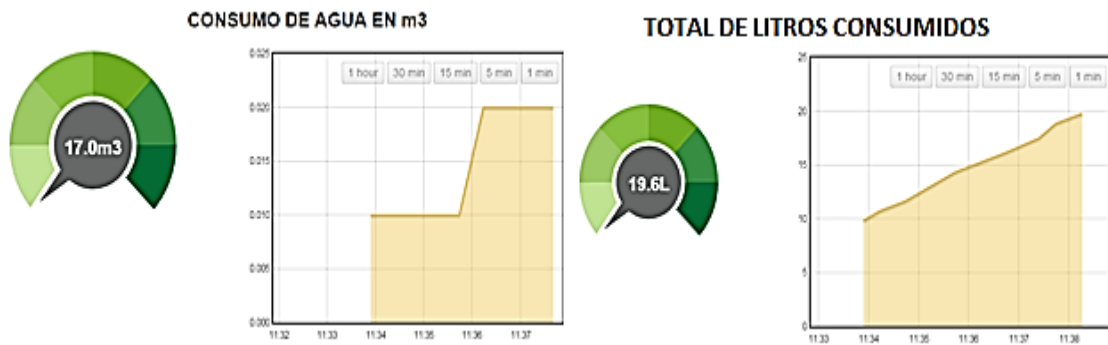


Figura 25: Interfaz gráfica para monitorear consumo m³ y L, en tiempo real. Fuente: (El Autor).

También se puede graficar las variables de consumo en metros cúbicos y los litros totales en el lapso de días, semanas, meses o años, donde se visualizan las variaciones producidas en estos determinados intervalos de tiempo. La interfaz con estas dos variables se muestra en la siguiente figura 26.

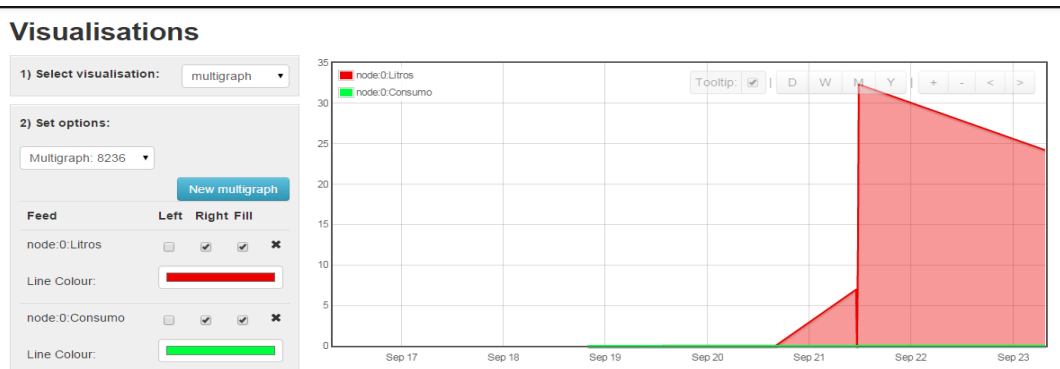


Figura 26: Gráfica para el monitoreo del consumo del agua en metros cúbicos y litros. Fuente: (El Autor).

e. MATERIALES Y MÉTODOS

e.1. Materiales

Para el desarrollo del prototipo, los materiales que se utilizaron son: revistas, artículos científicos, libros, páginas web y manuales que ayudaron a recopilar la información necesaria para la elaboración del proyecto de tesis; materiales electrónicos para la construcción del prototipo, tarjetas Arduino, software para la adquisición de los datos, un equipo informático para redacción de la información y la recopilación de la misma.

A continuación en la tabla Nro.9 hago referencia de los materiales utilizados en el presente proyecto.

Tabla 9. Materiales utilizados en el desarrollo de la tesis; fuente: elaboración autora

Material	Cantidad
Tarjeta Arduino UNO R3	1
Shield GSM Sim9000	1
Shield Key LCD	1
Resistencias	Varias
Caja	1
Conectores y cables	Varios
Datashet de Arduino	Varios
Manuales de Programación	Varios
Tarjeta SIM claro	1
Recargas de Datos	1
Computador	1
Manguera (para pruebas)	1
Hojas	Varias
Páginas Web	Varias
Artículos Científicos	Varios
Manuales	Varios
Libros	Varios

e.2. Métodos

Para el desarrollo del proyecto se hizo uso del método analítico, mediante el cual se realizó un análisis detallado de las características, técnicas de los sistemas de telemetría, mediante el cual se determinó que tecnología es la más apropiada y se adapte a las necesidades del medio.

El primer paso que se realizó fue estructurar el marco teórico para lo cual se hizo una recopilación bibliográfica, mediante el uso de libros, páginas web, catálogos, manuales, los mismos que ayudaron a la recopilación de la información.

Posteriormente se procedió a la realización del diseño y construcción del prototipo de telemetría, para lo cual se realizó un análisis técnico de cada uno de los componentes del sistema de telemetría, así como del servidor web, además se hicieron las pruebas pertinentes para comprobar el correcto funcionamiento del mismo.

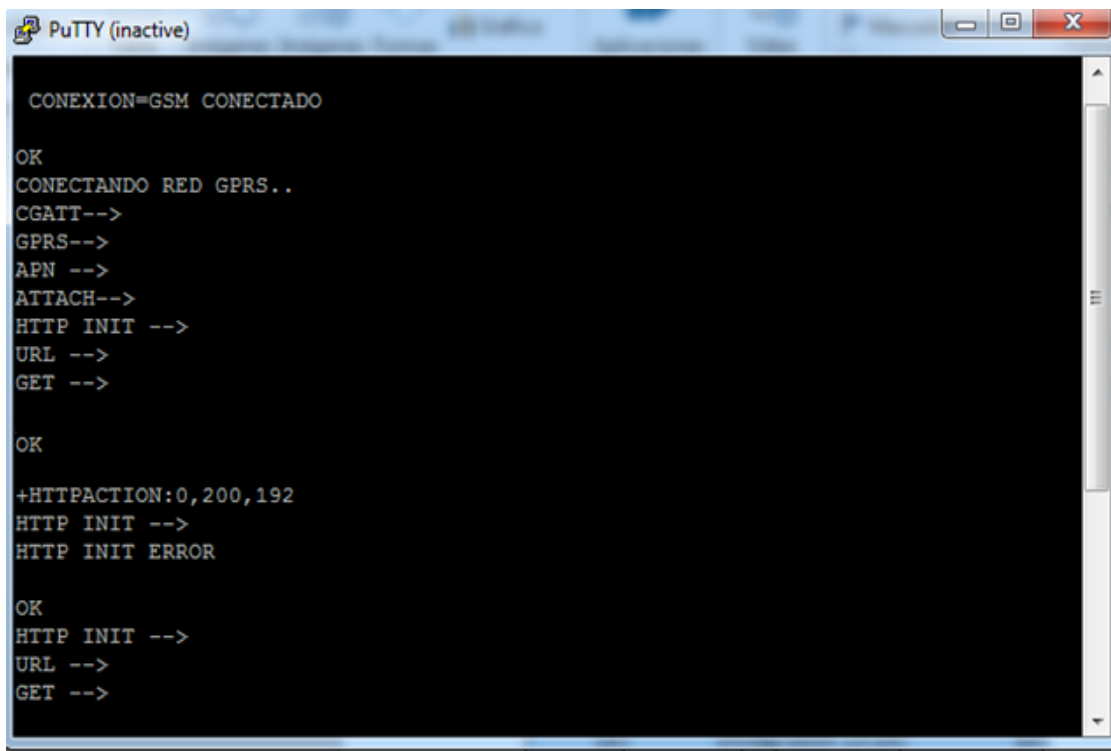
Por último se utilizó el método inductivo deductivo mediante el cual se determinaron las conclusiones y recomendaciones del funcionamiento del sistema.

e. RESULTADOS

Para hacer el análisis del comportamiento del prototipo se realizaron las pruebas necesarias esto es: conectar el dispositivo a la red de agua potable; además se realizó pruebas del funcionamiento general del sistema; conexión con la red GSM y GPRS de manera que el envío de los datos sea exitoso; medición del flujo de agua utilizada en una casa.

f.1. Funcionamiento general del sistema

Para verificar el funcionamiento general del sistema se utilizó la consola del programa Arduino “monitor serie” además del programa PuTTY como se muestra en la figura 22.



```
CONEXION=GSM CONECTADO
OK
CONECTANDO RED GPRS..
CGATT-->
GPRS-->
APN -->
ATTACH-->
HTTP INIT -->
URL -->
GET -->
OK
+HTTPACTION:0,200,192
HTTP INIT -->
HTTP INIT ERROR
OK
HTTP INIT -->
URL -->
GET -->
```

Figura 27. Pruebas de funcionamiento General del sistema en el programa Putty, fuente: elaboración autora

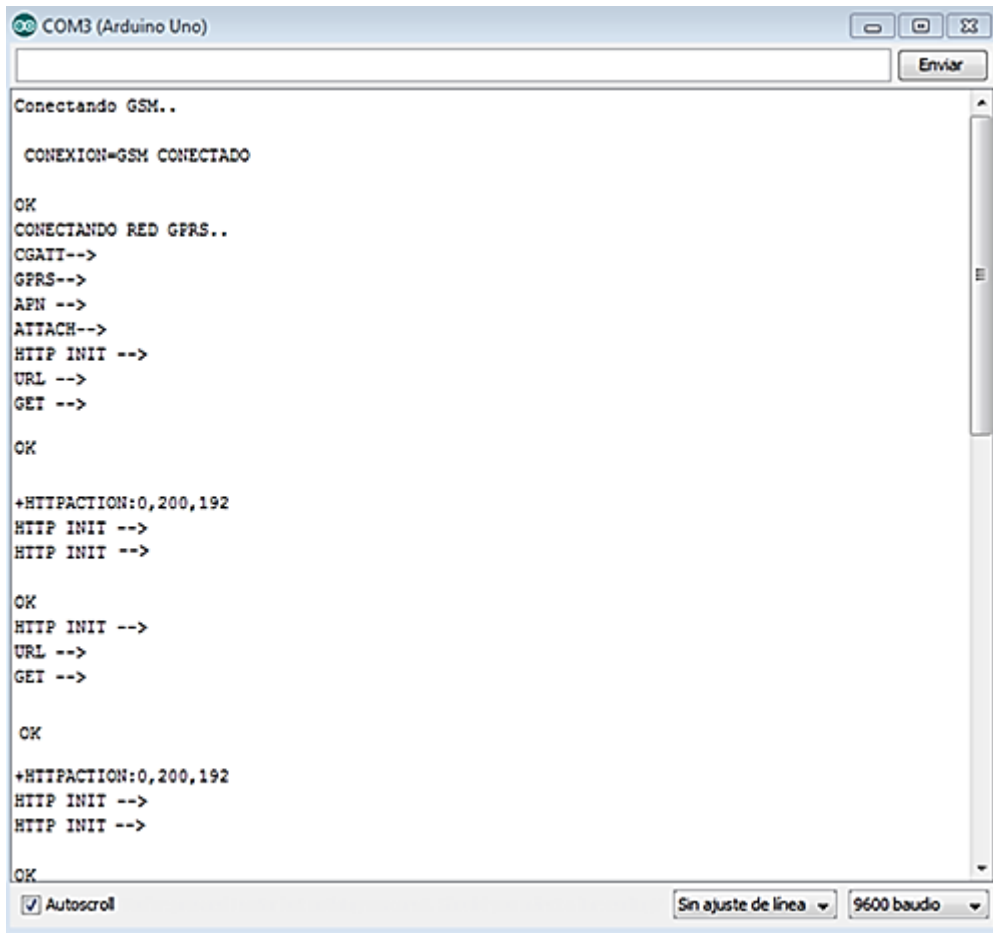


Figura 28 Funcionamiento general del sistema en Arduino, fuente: elaboración autora

e.2. Verificación de envío de datos



Figura 29 Envío de datos, fuente: elaboración autora

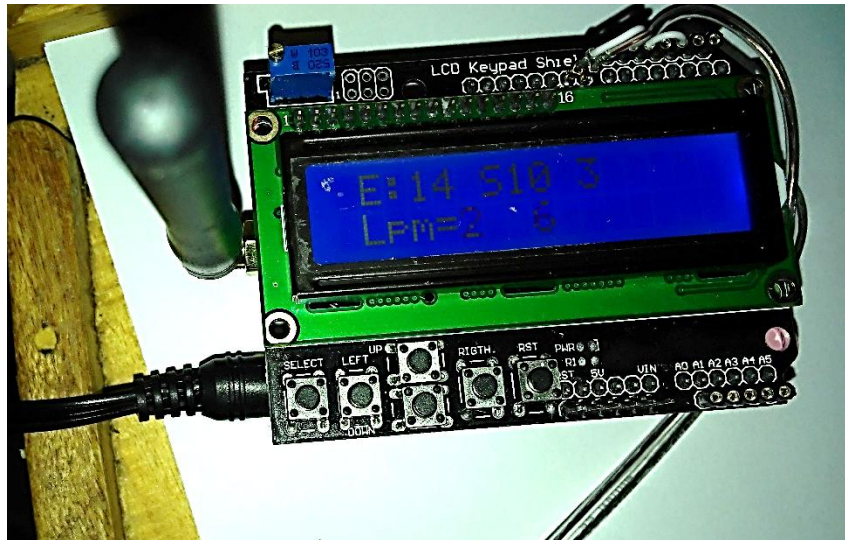


Figura 30.comprobación de envío de o de datos, fuente: elaboración autora

Una vez instalado el prototipo, funciona automáticamente estando en la capacidad de monitorear el consumo de agua de una vivienda; para verificar que los datos se han enviado correctamente en la pantalla LCD en primera fila columna 9 aparece un número, si es 2 indica que existe error al enviar el último dato, si es 3 el envío fue exitoso, el prototipo se puede manipular manualmente con la ayuda de la tarjeta *shield* LCD, por ejemplo: *select* inicializa la comunicación GPRS, *Left* envío manual de los datos, esto se detalla en la tabla 7,8, permitiendo ver el valor total de los litros consumidos en metros cúbicos.

e.3. Recepción de datos en el sitio Web

Para poder consultar los datos que ha enviado el sensor, entramos al sitio web del servidor:(<http://emoncms.org/serviciogprs>), mediante el cual se puede ver los parámetros del consumo así como los litros consumidos, además se puede exportar la tabla con los valores de consumo y se verifican los valores de consumo de agua.

Tabla 10 Datos de consumo de agua en lpm y m3, fuente: elaboración autora

Nº	Litros	Nº	Litros Totales
1442783050	0.03	1442783190	0.00
1442783070	0.03	1442783240	0.00
1442783090	0.03	1442783270	0.00
1442783120	0.03	1442783290	0.00

1442783140	0.03	1442783310	0.00
1442783170	0.03	1442783340	0.00
1442783190	0.03	1442783360	0.00
1442783240	0.03	1442783380	0.00
1442783270	0.03	1442783400	0.00
1442783290	0.03	1442852000	0.01
1442783310	0.03	1442852050	0.01
1442783340	0.03	1442852100	0.01
1442783360	0.53	1442852120	0.01
1442783380	5.80	1442852970	0.01
1442783400	5.81	1442852990	0.01
1442783430	7.55	1442853020	0.01
1442783450	7.55	1442853060	0.01
1442783470	9.85	1442853070	0.01
1442783500	10.75	1442853100	0.01
1442783520	10.75	1442853230	0.02
1442783540	11.65	1442853250	0.02
1442783560	13.45	1442853280	0.02
1442783610	14.35	1442853320	0.02
1442783630	15.25	1442853340	0.02
1442783660	16.15	1442853370	0.02
1442783680	17.50	1442853400	0.02
1442783730	18.85	1442853440	0.03
1442783750	19.75	1442853460	0.03
1442783770	31.75	1442853490	0.03
1442783790	32.55	1442853790	0.03

e. DISCUSIÓN

En temas relacionados al envío de datos hacia la sitios web, se lo realiza mediante la tecnología Ethernet, en este caso se han enviado para la trasmisión de datos utilizando la tecnología GSM/GPRS, los resultados obtenidos durante las pruebas realizadas fueron satisfactorios.

La red GSM/GPRS se encuentra ya implementada por las operadoras móviles, es bastante estable y se halla actualmente disponible en la mayor parte de lugares, esto permite que el usuario pueda tener conocimiento de la calidad de servicio que tiene.

Al existir el envío automático de los datos hasta el servidor web, ya no sería necesario la recolección manual por lo que no existirá errores en la cantidad de líquido consumido por ende el usuario estará satisfecho.

El prototipo está instalado en la Universidad Nacional de Loja, en el área de la Energía, de las Industrias y Recursos Naturales no Renovables, el mismos que dará un aporte académico a los estudiantes de la Carrera de Ingeniería en Electrónica y Telecomunicaciones.

Los sistemas de telemetría basados en Open Source como Arduino, raspberry pi, son de fácil programación, por cuanto este sistema es de código abierto y existe buen soporte bibliográfico, permitiendo además dejar a criterio de cada programador utilizar las librerías que se encuentran incrustadas o puede crear sus propias librerías.

g.1. ANÁLISIS DEL COSTO DEL PROTOTIPO

A continuación se detalla el costo del prototipo, el cual está basado en los precios actuales de los componentes y elementos utilizados para el desarrollo del mismo.

g.1.1. Costos del Prototipo Diseñado

En la tabla 11 se describen los componentes y elementos utilizados con sus respectivos costos, los cuales son parte del hardware del sistema de telemetría.

Tabla 11. Costo del sistema, fuente: elaboración autora

Descripción	Cantidad	Valor Unitario	Valor Total
Sensor F300a	1	19	19
Arduino Uno	1	25	25
Arduino <i>Shield</i> GSM	1	60	60
Arduino LCD KeyPad <i>Shield</i>	1	25	25
Cables y conectores	4m	10	10
Fuentes para el prototipo	2	10	20
Caja para el prototipo	1	30	30
TOTAL			189

Cabe mencionar que los costos que están tomados en cuenta son solo de la construcción del hardware, además se tuvieron costos adicionales para la parte del software que no se detallaron en esta lista como: las horas de programación, el equipo de cómputo, equipo de medición como el multímetro, la mano de obra debido a que el prototipo fue construido de manera experimental y no comercial.

g.1.2 Ventajas del prototipo.

La principal ventaja del prototipo telemetría diseñado e implementado es la capacidad para medir el consumo de agua en tiempo real y en unidades de volumen que son comprensibles para el usuario, esto permite:

- Que el usuario pueda hacer un plan de ahorro del consumo de agua.
- Es muy fácil de manipular pudiendo el usuario manipular el prototipo como cambiando el tiempo en el que se envían los datos hasta el servidor, así como las variables de medición.
- El usuario tiene un historial de consumo diario, mensual, lo que le permite saber qué mes ha tenido más consumo.
- Al enviar los datos del consumo de agua de forma remota no abra la necesidad de la recolección manual.

e. CONCLUSIONES

- El sistema de telemetría del consumo de agua potable diseñado cumple satisfactoriamente con las principales características de un medidor digital, haciendo posible la medición del consumo de agua de manera eficiente.
- Se determinó que el sensor efecto hall cumple perfectamente con los requerimientos técnicos necesarios para poder realizar mediciones de flujo electrónicas debido a que combina la mecánica con la teoría electromagnética, con una precisión del 3%.
- Al prototipo se le sometió algunas pruebas para el envío de los datos en diferentes herramientas web, que permiten el registro de datos en la nube, tales como: softzone.es, smarterasp, emoncms de las cuales se eligió la plataforma web “emoncms” por ser la idónea para el monitoreo del consumo de agua.
- El sistema de telemetría utiliza la red de telefonía móvil, el mismo que está sujeto a la eficiencia, eficacia y la cobertura que tenga la red. Además, para poder enviar los datos es necesario contar con un plan de datos en cualquiera de las operadoras de telefonía celular.
- La tecnología utilizada para el prototipo es de hardware y software libre por lo que no se requiere de ningún tipo de licencia o permiso para la implementación, además no está sujeta a librerías pre – establecidas, permitiendo el desarrollo de librerías de acuerdo a la necesidad de quien lo implemente.

e. RECOMENDACIONES

- Es necesario que el programador tenga conocimientos básicos de programación para que la configuración del sistema no se le haga tan compleja.
- Es importante el uso de equipos y herramientas, adecuadas para la construcción del dispositivo, Particularmente se recomienda realizar la medición de variación de voltaje del sensor para poder adecuar la señal de salida del sensor.
- Se recomienda tener una dirección IP pública, para poder visualizar la información en cualquier parte del mundo.
- Se recomienda montar un servidor propio, teniendo así la administración total para la gestión y el registro de datos en la nube.
- Se aconseja tener un lugar adecuado para la ubicación del prototipo de tal forma que no le afecte las inclemencias del tiempo.
- EL presente proyecto puede ser utilizado por las empresas proveedoras de agua potable para monitoreo del buen funcionamiento de la red.

e. BIBLIOGRAFÍA

- [1] Á. H. CÁRDENAS VALENCIA y A. F. ECHEVERRY GIRALDO, *PROTOTIPO DE UN SISTEMA DE TELEMETRÍA Y CONTROL PARA SEGURIDAD EN VEHÍCULOS, SOPORTADO EN REDES MÓVILES*, PEREIRA, 2010.
- [2] J. E. BRICEÑO MÁRQUEZ, *TRANSMISIÓN DE DATOS*, MÉRIDA: Taller de Publicaciones de la Facultad de Ingeniería, ULA, 2005.
- [3] I. E. I. ortega, *Diseño y construcción de un sistema de telecontrol y telemetría*, Mexico D.F, 2006.
- [4] E. HARPER, *EL ABC DE LA INSTRUMENTACION EN EL CONTROL DE PROCESOS INDUSTRIALES*, MEXICO: LIMUSA, 2012.
- [5] A. R. Penin, *Sistema SCADA*, España: Marcombo, ediciones tecnicas , 2007.
- [6] M. B. S. P. Jean Herrera, «Diseño e implementación de un sistema scada inalámbrico mediante la tecnología zigbee y arduino,» *Prospect*, vol. 12, nº 2, pp. 65-72, 2014.
- [7] E. H. Pérez, *Tecnologías y redes de transmisión de datos*, Mexico: Limusa S.A, 2003.
- [8] U. D. Black, *Redes de transmisión de datos y proceso distribuido*, España: Graficas Lormo, 1983.
- [9] J. P. B. F. A. C. H. Pablo Gil Vazquez, *Redes y Transmisión de Datos*, España: Universidad de Alicante, 2010.
- [10] L. A. L. Reyes, *Análisis de alternativas de desarrollo de sistema de medida avanzada bajo el concepto de redes inteligentes en Chilectra S.A*, Santiago de Chile, 2008.
- [11] M. V. C. Gutierrez, *Estudio para la implementación del sistema de infraestructura de medición avanzada (AMI) en la empresa eléctrica regional centro sur C.A*, Cuenca, 2011.
- [12] S. J. C. A. C. D. N. H. A. Dias Ricardo, «ANÁLISIS COMPARATIVO DE SISTEMAS DE MEDICIÓN INTELIGENTES EN EL CONTEXTO DE LAS REDES INTELIGENTES,» ciudad de la Plata, 2013.
- [13] J. C. P. L. G. César Viloría Núñez, «Análisis comparativo de tecnologías inalámbricas para una solución de servicios de telemedicina,» *Ingeniería y Desarrollo*, nº 25, pp. 204-209, 2009.
- [14] C. J. Antonio, *WI-FI: lo que se necesita conocer*, Madrid: RC libros, 2010.
- [15] R. d. C. A. y. Gestión, Jorge Ghe. Voinea, Almeria: Fired, 2011.

- [16] J. M. Huidrobo, Radiocomunicaciones viajando a través de las ondas, España: Creaciones copyright, S.L, 2011.
- [17] G. A. C. J. Escobar Z Freddy A, *Diseño de una red inalámbrica de datos para las unidades fijas y móviles de la policía nacional en el distrito metropolitano de Quito usando la tecnología WIMAX*, Quito, 2011.
- [18] M. d. C. España, Telecomunicaciones Móviles, Madrid: Ediciones Diaz de Santos, Servicios avanzados de telecomunicaciones.
- [19] Á. P. d. I. Cruz, «Evolución de los sistemas móviles,» *Sistematicas y Telematicas*, pp. 16,17, 2004.
- [20] N. F. Esteban, *Redes Celulares GSM, GPRS*, Ciudad del Rosario, 2008.
- [21] A. R. Fernandez, *Despliegue de un sistema de telefonía móvil GSM/GPRS en las comarcas de Tarragona*, Valencia, 2007.
- [22] J. L. T. S. Angel Danilo Cornejo Ortega, *Diseño, construcción e implementación de un sistema de telemetría utilizando GSM; para el monitoreo de los parámetros de temperatura, presión de aceite, velocidad del giro del motor y velocidad del desplazamiento de un vehículo chevrolet optra 2008*, Cuenca, 2010.
- [23] M. V. Méndez, Tuberías a presión en los sistemas de abastecimiento de agua, 2007: Universidad Católica Andrés Bello, Caracas.
- [24] A. S. Ruiz, F. A. Ros y J. C. Rico, *Guía Práctica de Sensores*, España: Creaciones Copyright, S.L, 2010.
- [25] C. P. Ruiz, Escritor, *Tipos de Sensores*. [Performance]. Dpto de Ingeniería Cartográfica, 2003.
- [26] O. T. Artero, *Curso práctico de formación*, Madrid: RC libros, 2013.
- [27] I. IVREA, «Arduino,» Instituto IVREA, enero 2005. [En línea]. Available: <http://www.arduino.cc/en/Main/Products>. [Último acceso: 15 abril 2015].
- [28] Arduino, «Arduino Genuino,» 26 marzo 2008. [En línea]. Available: www.arduino.cc. [Último acceso: 25 marzo 2015].
- [29] s. T. Limited, «seed Grow Diference,» 2008. [En línea]. Available: www.seeedstudio.com. [Último acceso: 15 abril 2015].
- [30] S. I. d. C. Personal, David Muñoz Rodríguez, Mexico: Marcombo, 2002.

- [31] A. M. R. Enrique Mandado Perez, Escritor, *Sensores y Acondicionadores*. [Performance]. E.T.S. de Ingenieros de Telecomunicacion de Vigo, 2010.
- [32] R. P. R. M. Murdocca, «Universidad Nacional de San Luis,» 2009. [En línea]. Available: <http://www0.unsl.edu.ar/>. [Último acceso: 5 junio 2015].
- [33] emoncms.org. [En línea] <http://emoncms.org/site/home>.

1. ANEXOS

1.1. Datasheet Arduino GSM Sim900



G50EFC56E3197F

GSM/GPRS Shield

Datasheet



Tinysine Electronics @ 2014
Version 1.0

INTRODUCTION

The GPRS/GSM Shield provides you a way to use the GSM cell phone network to receive data from a remote location. The shield allows you to achieve this via any of the three methods:

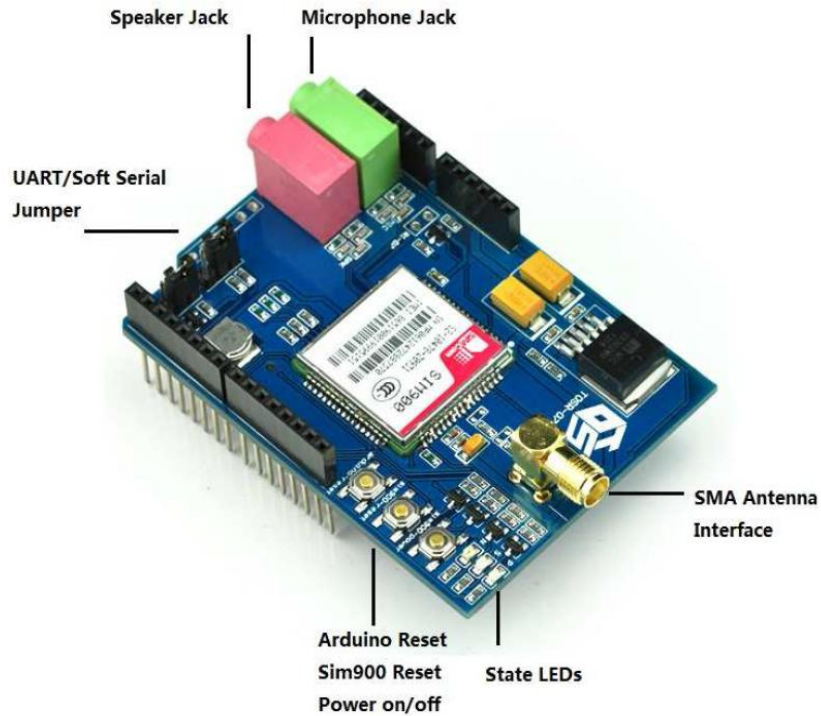
- Short Message Service
- Audio
- GPRS Service

The GPRS Shield is compatible with all boards which have the same form factor (and pinout) as a standard Arduino Board. The GPRS Shield is configured and controlled via its UART using simple AT commands. Based on the SIM900 module from SIMCOM, the GPRS Shield is like a cell phone. Besides the communications features, the GPRS Shield has 12 GPIOs, 2 PWMs and an ADC.

SPECIFICATIONS

- Quad-Band 850 / 900/ 1800 / 1900 MHz - would work on GSM networks in all countries across the world.
- GPRS multi-slot class 10/8
- GPRS mobile station class B
- Compliant to GSM phase 2/2+
- Class 4 (2W@850/900MHz)
- Class 1 (1W@1800/1900MHz)
- Control via commands (GSM 07.07, 07.05 and SIMCOM enhanced AT Commands)
- Short message service
- Free serial port selection
- RTC supported with Super Cap
- Power on/off and reset function supported by Arduino interface

Overview



The indicator LEDs

The GSM Shield has three indicator LEDs for the GSM Shield power, SIM900 power and net status.

GSM Shield power(P):

This LED is used to indicate the power status of the GSM Shield. If the external power supply is connected to the arduino board, then the GSM Shield will get power, this LED will light on.

SIM900 power(S):

This LED is used to indicate the power status of the SIM900. After the SIM900 is power on, the status LED will light on.

Net Status(N):

This LED is used to indicate the net status. The LED will blink slowly or quickly according to different states.

Status	Description
Off	SIM900 is not running
64ms On/800ms Off	SIM900 not registered the network
64ms On/3000ms Off	SIM900 registered to the network
64ms On/300ms Off	GPRS communication is established

The buttons

There are 3 buttons on the GSM shield board.

SIM900 Power Button:

Stack the GSM shield on your arduino main board. The GSM shield will running automatically when you power up your arduino main board. If you want turn on/off the SIM900 module manually. Just press the the sim900 power button one second.

SIM900 Reset Button

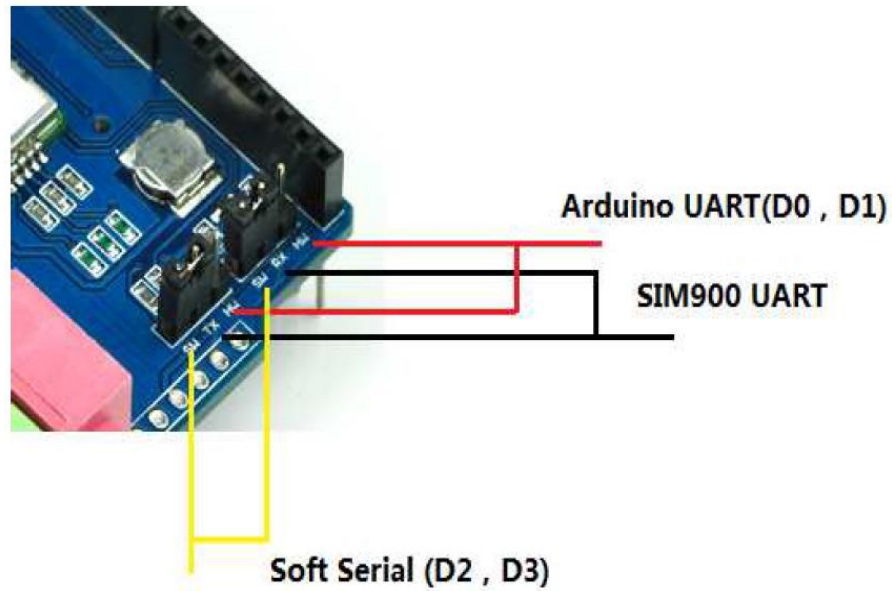
Reset the SIM900 module.

Arduino Reset Button:

Reset the Arduino main board. Same function as the Reset Button which on the arduino main board.

Note: Our test sketch will control D8 pin to turn on the SIM900 when the arduino board is powered up.

Setting Jumper

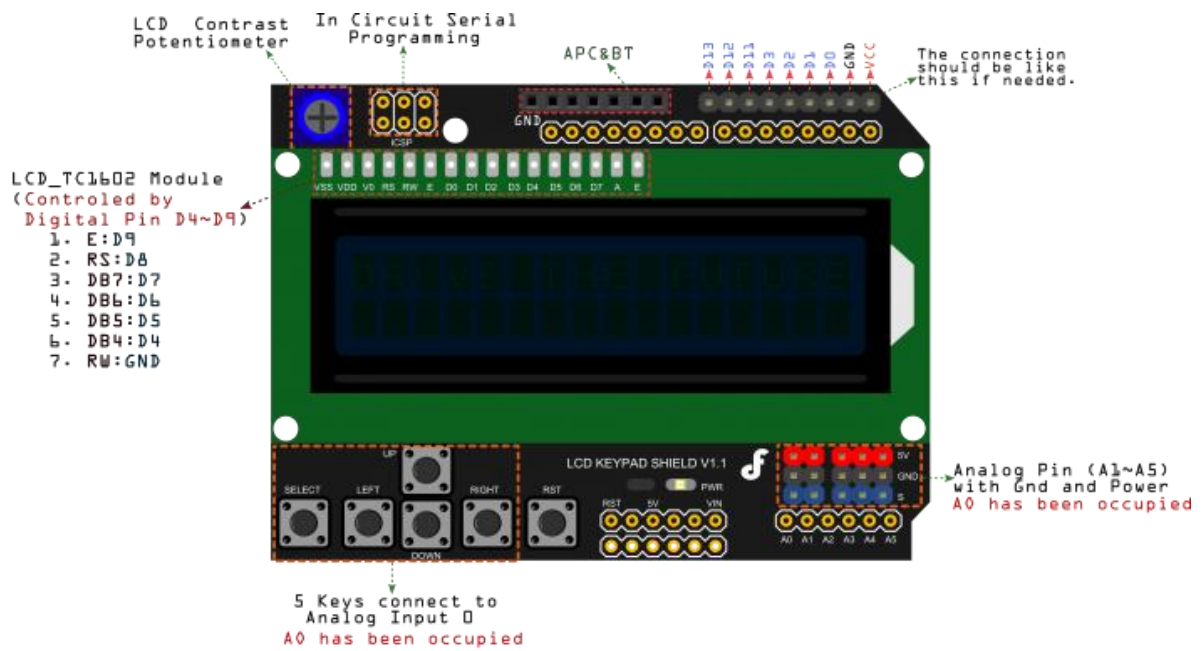


If you want use UART send AT commands. Set the jumper to HW position. If you use soft serial. Set to SW position.

Our test sketch use soft serial. So we keep it on SW position normally.

If you are using MEGA2560. Just need to connect the middle pins of RX and TX jumpers to RX1 TX1 on Arduino Mega 2560. And change the code setting.

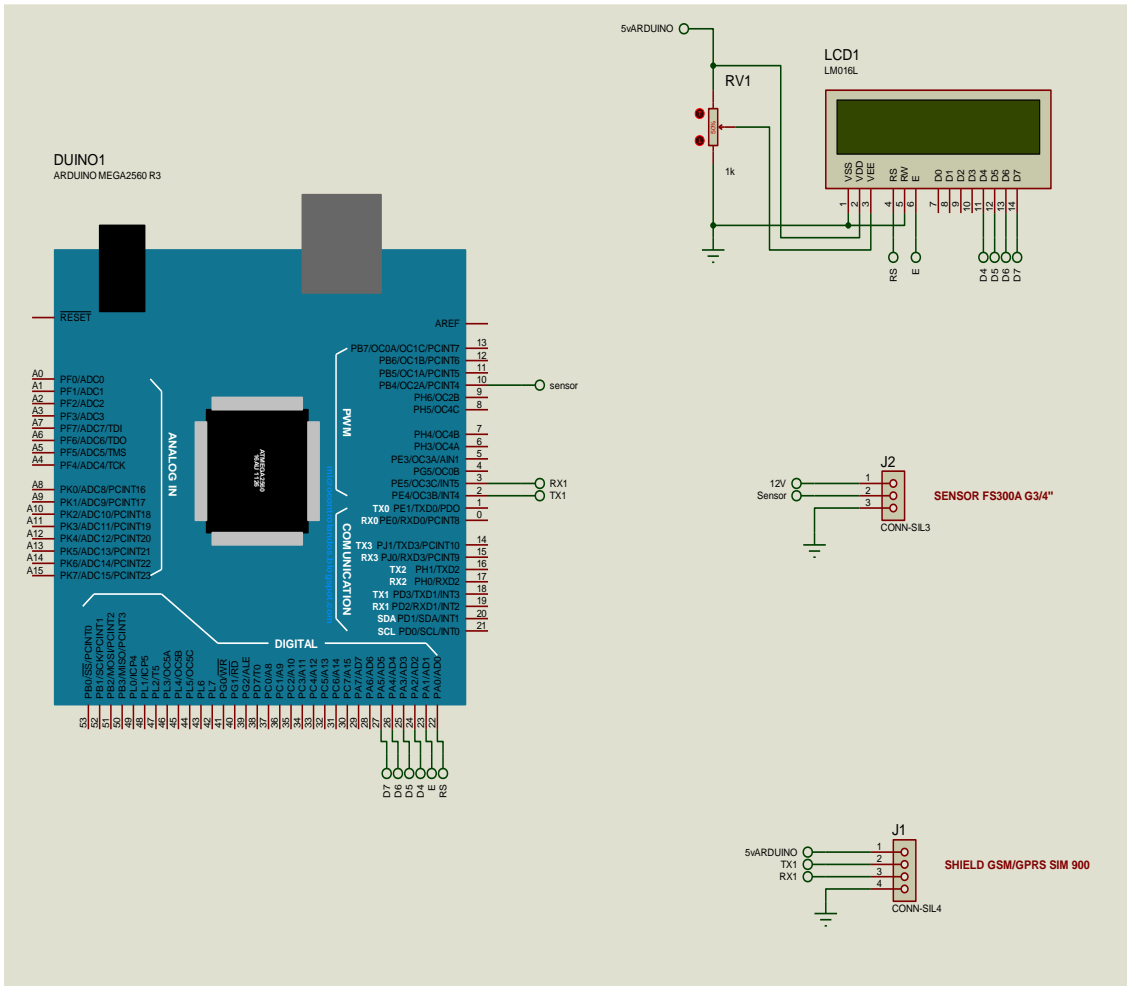
1.2. Diagrama de conexión Key LCD



Instruction for D4 To D10 and Analog Pin 0

Pin	Function	Instruction
Digital 4(D4)	D4~D7 are used as DB4~DB7	Four high order bidirectional tristate data bus pins. Used for data transfer and receive between the MPU and the LCD.
Digital 5(D5)		
Digital 6(D6)		
Digital 7(D7)		
Digital 8(D8)	RS	Choose Data or Signal Display
Digital 9(D9)	Enable	Starts data read/write
Digital 10(D10)	LCD Backlight Control	
Analog 0(A0)	Button select	Select, up, right, down and left

1.3 Diseño de prototipo en proteus



1.4. Código fuente

1.4.1. Código fuente Arduino


```

//String request = "";          //variable para armar la solicitud GET
unsigned long currentMillis = 0;  //valor actual del timer
unsigned long previousMillis = 0; //valor anterior del timer
int interval = 1000;             //intervalo de lectura de sensor cada segundo
unsigned long tiempoTranscurrido = 0; //en segundos
unsigned long tiempoActualizacion = 10; //en segundos

//-----

//DECLARACION DE VARIABLES PARA CALCULOS DE AGUA M3/H

//-----

int lpm;          //variable para almacenar valor en litros / segundo o litros por
segundo

String valorSensor = "0"; //variable para almacenar el valor del sensor

float Calc ;

float totalLitros; //variable para acumular los litros consumidos
float consumo;    //variable que acumula el consumo total de Agua

//-----

//DECLARACION DE VARIABLES PARA CONEXION GSM/GPRS

//-----

String apn = "internet.claro.com.ec"; //variable para el apn de operadora celular

String apnConexion = ""; //variable para establecer

boolean conexionIniciada = false; //variable de estado para conexion entre arduino
y el shield GSM/GPRS.

boolean gprsIniciado = false; //variable de estado de conexion del GPRS.

//-----

//DECLARACION DE VARIABLES PARA ENVIAR LA TRAMA AL SERVIDOR

//-----

String request = ""; //variable para armar la solicitud http

//-----

//DECLARACION DE VARIABLES PARA BOTONES EN EL PANEL LCD

//-----

```

```

/*
-----
Medidor de Agua en Metros Cubicos
-----

Programa que calcula el consumo de agua en un determinado
periodo tiempo y muestra el consumo total en metros cubicos
*/

//-----
// CARGA DE LIBRERIAS
//-----

#include<SoftwareSerial.h> // Librería para la comunicación serial por software.
#include<LiquidCrystal.h> // Librería para el LCD 16X2.
#include "SIM900.h" // Libreria para Shield SIM900
#include "inetGSM.h"

//-----
//DECLARACION DE VARIABLES
//-----

LiquidCrystal lcd(12, 13, 4, 5, 6, 7); // Declaramos los pines 4,5,6,7,12,13 para mostrar los
datos en el LCD.

char pinSensor = 2;

volatile int contadorPulso;

char gsmSt = '0';

char gprsSt = '0';

char letraTiempoSt = 'S';

int valorTiempoSt = 10;

char datoSt = '0';

int i = 0;

char inSerial[100];

```

```

int lcd_key  = 0;           //almacenar el boton pulsado
int adc_key_in = 0;        //almacenar el valor del boton pulsado
//-----
// DEFINE LOS BOTONES DEL TECLADO DEL LCD
//-----
#define btnRIGHT 0
#define btnUP 1
#define btnDOWN 2
#define btnLEFT 3
#define btnSELECT 4
#define btnNONE 5
//-----
//Funcion principal
//-----
void setup() {             // Se ejecuta cada vez que el Arduino se inicia
  pinMode(pinSensor, INPUT); //declaracion de entrada
  attachInterrupt(0, interrupcionPulso, RISING); //en el flanco de subida
  lcd.begin(16, 2);       // inicializar LCD
  lcd.setCursor(2, 0);
  lcd.print("INICIANDO!!!");
  delay(1000);
  lcd.clear();
  Serial.begin(9600);     //Inicio de conexión serial hacia el PC
  ConectarGPRS();
  delay(1000);
  delay(1000);
  ConectarRED();
}

```

```

/-----
//Funcion ciclicla
//-----
void loop() {          // Esta funcion se mantiene ejecutando

// cuando este energizado el Arduino

//Leer bytes de entrada en Hardware Serial ,
serialhwread();      //y escribirlos en el nuevo Software Serial.
serialswread();      //Leer bytes en el nuevo Software Serial.

////////////////////*****TIMER *****////////////////////////////////////
currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
  previousMillis = currentMillis;
  //se lee pulsos/segundo
  lpm = (contadorPulso / 5.50); //(Pulse frequency) / 5.5Q, = flujo en litros/minuto
  valorSensor = String(lpm);
  actualizarPantalla();

  //*****PARA IMPRIMIR TOTAL DE LITROS*****MODIFICAR SI ES
  NECESARIO

  //Si leo en Litros / segundo, entonces seria decir que se consumo tantos litros en
  //ese segundo, entoncs, voy acumulando ese valor

  Calc = lpm / 60.00;
  totalLitros = totalLitros + Calc;
  consumo = totalLitros / 1000;
  lcd.setCursor(1, 1);
  lcd.print("L:");
  lcd.setCursor(3, 1);
  lcd.print(totalLitros);
  lcd.setCursor (8, 1);
  lcd.print("m3=");
  lcd.print (consumo);
}
}

```

```

if (tiempoTranscurrido >= tiempoActualizacion) {
    EnviarDatos();
    tiempoTranscurrido = 0;
}
tiempoTranscurrido++;          //actualizar el tiempo transcurrido
//reiniciar el contador
contadorPulso = 0;
}

////////////////////////////////////**boton pulsado **////////////////////////////////////

lcd_key = LeerBotonesLCD();    // Leer el boton pulsado
switch (lcd_key) {            // diferenciar pulsador
    case btnRIGHT: {         // Si boton pulsado es "RIGHT" elegir tipo de envio:
segundos/minutos /horas
        delay(200);
        switch (letraTiempoSt ) {
            case 'S':
                letraTiempoSt = 'M';
                tiempoActualizacion = tiempoActualizacion * 60;
                break;
            case 'M':
                letraTiempoSt = 'H';
                tiempoActualizacion = tiempoActualizacion * 60;
                break;
            case 'H':
                letraTiempoSt = 'S';
                tiempoActualizacion = tiempoActualizacion / 3600;
                break;
        }
        break;
    }
}
}

```

```

    }

    case btnUP: {                                //Si boton pulsado es "UP" subir tiempo de envio de
datos
        delay(200);

        if (valorTiempoSt < 59 && letraTiempoSt == 'S' ) {

            valorTiempoSt++;

            tiempoActualizacion = valorTiempoSt;
        }

        else if (valorTiempoSt < 59 && letraTiempoSt == 'M') {

            valorTiempoSt++;

            tiempoActualizacion = valorTiempoSt * 60;
        }

        else if (valorTiempoSt < 24 && letraTiempoSt == 'H' ) {

            valorTiempoSt++;

            tiempoActualizacion = valorTiempoSt * 3600;
        }

        //else

        actualizarPantalla();

        break;
    }

    case btnDOWN: {

        // Si boton pulsado es "DOWN"      Bjar tiempo de envio de datos

        delay(200);

        if (valorTiempoSt > 10 && letraTiempoSt == 'S' ) {

            valorTiempoSt--;

            tiempoActualizacion = valorTiempoSt;
        }
    }

```

```

else if (valorTiempoSt > 1 && letraTiempoSt == 'M') {
    valorTiempoSt--;
    tiempoActualizacion = valorTiempoSt * 60;
}
else if (valorTiempoSt > 1 && letraTiempoSt == 'H' ) {
    valorTiempoSt--;
    tiempoActualizacion = valorTiempoSt * 3600;
}
actualizarPantalla();
break;
}
case btnLEFT: {
    // Si boton pulsado es "LEFT"      Enviar datos al servicio web
    delay(200);
    EnviarDatos();
    break;
}
case btnSELECT: {
    // Si boton pulsado es "SELECT"    //OPERACIONES DE CONEXION A LA RED GPRS
    delay(200);
    ConectarRED();
    break;
}
case btnNONE: {
    // Si ningun pulsador
    break;
}
}

```

```

//-----
//Funcion de la interrupcion
//-----
void interrupcionPulso () {
    contadorPulso++; //Incrementar el contador
}
//-----
//Funcion para conectar a la red de la Operadora
//-----
void ConectarGPRS() {
    Serial.println("Conectando GSM..");
    lcd.setCursor(0, 0);
    lcd.print("Conectando GSM..");
    //Inicio de conexión serial hacia el SHIELD, pines:
    //Para aplicaciones con HTTP usar 4800 bps o inferiores.

    if (gsm.begin(2400)) {
        Serial.println("\n CONEXION=GSM CONECTADO");
        gsmSt = '1';
        conexionIniciada = true;
    }
    else {
        Serial.println("\n CONEXION=GSM ERROR");
        gsmSt = '0';
    }
    lcd.clear();
}

```



```

//-----
//Funcion para conectar a la red de la Operadora en modo de trasmision de datos
//-----
void ConectarRED() {
  if (conexionIniciada) { //SI SE HA REALIZADO UNA CONEXION SERIAL CON EL SHIELD GPRS
    Serial.println("CONECTANDO RED GPRS..");
    Serial.println("CGATT-->");
    gprsSt = '0';
    actualizarPantalla();
    if (gsm.SendATCmdWaitResp ("AT+CGATT=1", 1000, 200, "OK", 3) == 1 ) { //adjuntar
el gprs
      Serial.println("GPRS-->");
      gprsSt = '1';
      actualizarPantalla();
      if (gsm.SendATCmdWaitResp ("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\"", 1000, 200,
"OK", 3) == 1 ) { //configurar parametros del gprs
        apnConexion = "AT+SAPBR=3,1,\"APN\", \"\" + apn + "\""; //ES UNA CADENA DE
CARACTERE
        Serial.println("APN -->");
        gprsSt = '2';
        actualizarPantalla();
        if (gsm.SendATCmdWaitResp (apnConexion.c_str(), 3000, 200, "OK", 3) == 1 ) {
//configurar parametros del apn
          Serial.println("ATTACH-->");
          gprsSt = '3';
          actualizarPantalla();
        if (gsm.SendATCmdWaitResp ("AT+SAPBR=1,1", 10000, 200, "OK", 3) == 1 ) { //abrir una
conexion
          gprsIniciado = true;
          gprsSt = '4';
          actualizarPantalla();
        }
      }
    }
  }
}

```

```

//-----
//Funcion para conectar a la red de la Operadora en modo de transmision de datos
//-----
void ConectarRED() {
  if (conexionIniciada) { //SI SE HA REALIZADO UNA CONEXION SERIAL CON EL SHIELD GPRS
    Serial.println("CONECTANDO RED GPRS..");
    Serial.println("CGATT-->");
    gprsSt = '0';
    actualizarPantalla();

    if (gsm.SendATCmdWaitResp ("AT+CGATT=1", 1000, 200, "OK", 3) == 1 ) { //adjuntar
el gprs
      Serial.println("GPRS-->");
      gprsSt = '1';
      actualizarPantalla();

      if (gsm.SendATCmdWaitResp ("AT+SAPBR=3,1,\"CONTYPE\", \"GPRS\"", 1000, 200,
"OK", 3) == 1 ) { //configurar parametros del gprs
        apnConexion = "AT+SAPBR=3,1,\"APN\", \"\" + apn + "\""; //ES UNA CADENA DE
CARACTERE
        Serial.println("APN -->");
        gprsSt = '2';
        actualizarPantalla();

        if (gsm.SendATCmdWaitResp (apnConexion.c_str(), 3000, 200, "OK", 3) == 1 ) {
//configurar parametros del apn
          Serial.println("ATTACH-->");
          gprsSt = '3';
          actualizarPantalla();

          if (gsm.SendATCmdWaitResp ("AT+SAPBR=1,1", 10000, 200, "OK", 3) == 1 ) { //abrir una
conexion
            gprsIniciado = true;
            gprsSt = '4';

```

```

else
    Serial.println("CONNECTION ERROR");
}
else
    Serial.println("APN ERROR");
}
else
    Serial.println("GPRS PARAMETER ERROR");
}
else
    Serial.println("GPRS ATTACH ERROR");
}
}
void actualizarPantalla() {
    lcd.clear();
    lcd.setCursor(1, 0);
    lcd.print("E:");
    lcd.setCursor(3, 0);
    lcd.print(gsmSt );
    lcd.setCursor(4, 0);
    lcd.print( gprsSt);
    lcd.setCursor(6, 0);
    lcd.print(letraTiempoSt);
    lcd.setCursor(7, 0);
    lcd.print( String(valorTiempoSt));
    lcd.setCursor(10, 0);
    lcd.print( datoSt );
}

```

```

//-----
//Funcion serial con el PC
//-----
void serialhwread() {
  i = 0;
  if (Serial.available() > 0) {
    while (Serial.available() > 0) {
      inSerial[i] = (Serial.read());
      delay(10);
      i++;
    }
    inSerial[i] = '\0';
    Serial.println(inSerial);
    gsm.SimpleWriteLn(inSerial);
    inSerial[0] = '\0';
  }
  Serial.flush();
}
//-----
//Funcion serial con el shield GSM/PRS
//-----
void serialswread() {
  gsm.SimpleRead();
}
//-----
//Funcion para enviar los datos al servidor
//-----

```

```

    actualizarPantalla();

Serial.println("HTTP INIT-->");

    datoSt = '0';

actualizarPantalla();

    if (gsm.SendATCmdWaitResp ("AT+HTTPIPINIT", 1000, 200, "OK", 3) == 1) {
//Inicializa servicio HTTP

        if (gsm.SendATCmdWaitResp ("AT+HTTTPARA=\\"CID\\",1", 1000, 200, "OK", 3) == 1) {
// establece parametros para la seccion HTTPS

            request =
"AT+HTTTPARA=\\"URL\\",\\"emoncms.org/api/post?apikey=1e7743b555b2f14ab1bd4586f7
32c111&json={"";

            request += "Litros:";

            request += totalLitros;

            request += ",Consumo:";

            request += consumo;

            request += "}\\"r\n";

            Serial.println("Trama: " + request);

            Serial.println("URL -->");

            datoSt = '1';

            actualizarPantalla();

            if (gsm.SendATCmdWaitResp (request.c_str(), 2000, 100, "OK", 5) == 1) {
//Enviar datos al servidor

                Serial.println("POST -->");

                datoSt = '2';

                actualizarPantalla();

                if (gsm.SendATCmdWaitResp ("AT+HTTTPACTION=1", 2000, 200, "OK", 2) == 1) {
//se establece inicio de la sesi3n mediante metodo POST=1 GET=0

                    request = "";

```

```

    } else {
        //error al iniciar metodo de trasmision
        Serial.println("SESION ERROR");
        gsm.SimpleWriteln("AT+HTTPTERM"); //Termina el servicio HTTP
    }
} else {
    //error al enviar los Datos(Error Data)
    Serial.println("DATA POST ERROR");
    gsm.SimpleWriteln("AT+HTTPTERM"); //Termina el servicio HTTP
}
} else {
    //error al establecer paramatros HTTPS
    Serial.println("PARAMETROS HTTP ERROR");
    gsm.SimpleWriteln("AT+HTTPTERM"); //Termina el servicio HTTP
}
} else {
    //error al inicializar servicio HTTP(Error HTTP)
    Serial.println("HTTP INIT ERROR");
    gsm.SimpleWriteln("AT+HTTPTERM"); //Termina el servicio HTTP
}
} //Fin inicio GPRS
}
//-----
//Funcion para leer botones del teclado
//-----
int LeerBotonesLCD() { // metodo para leer los botones
    adc_key_in = analogRead(0); // leer el valor del canal analogo 0 ---- resolucio de 0 a
    1023...
}

```

```
// valores de botones centrados en : 0, 144, 329, 504, 741
// se suma 50 para comprobar si esta cerca
// regresar numero de boton
if (adc_key_in > 1000) return btnNONE;
if (adc_key_in < 50) return btnRIGHT;
if (adc_key_in < 250) return btnUP;
if (adc_key_in < 450) return btnDOWN;
if (adc_key_in < 650) return btnLEFT;
if (adc_key_in < 850) return btnSELECT;
return btnNONE; // si no concuerda con ningun boton
}
```

Loja, 17 de septiembre del 2015

Lic. Luisa Marillac Quezada Cabrera.

DOCENTE DE INGLES GRADUADA EN LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Haber realizado la traducción del resumen del idioma ingles de la tesis denominada: **“DISEÑO Y CONSTRUCCIÓN DE UN PROTOTIPO DE TELEMETRÍA PARA LA AUTOMATIZACIÓN INALÁMBRICA DEL CONSUMO DE AGUA MENSUAL MEDIANTE LA RED GSM. CASO PRACTICO UNIVERSIDAD NACIONAL DE LOJA”** correspondiente a la señorita egresada **MARIA ESTHER MEDINA SUCUNUTA.**

Es todo cuanto puedo certificar en honor a la verdad y autorizo a la interesada darle el uso que crea conveniente.

Atentamente,



Lic. Luisa Marillac Quezada Cabrera
Docente