



UNIVERSIDAD NACIONAL DE LOJA

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

**CARRERA DE INGENIERÍA EN ELECTRÓNICA Y
TELECOMUNICACIONES**

**“DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN
SISTEMA DOMÓTICO PARA LA MEDICIÓN DEL CONSUMO
DE AGUA POTABLE A TRAVÉS DE INTERNET Y CORREO
ELECTRÓNICO”**

TESIS DE GRADO PREVIO A LA OBTENCIÓN
DEL TÍTULO DE INGENIERO EN ELECTRÓNICA
Y TELECOMUNICACIONES.

AUTOR: EDISON FABRICIO CHUQUIMARCA SARANGO

DIRECTOR: ING. MARCELO VALDIVIEZO CONDOLO

**1859
LOJA-ECUADOR**

2014

CERTIFICACIÓN

Señor Ingeniero
Marcelo Fernando Valdiviezo Condolo
DIRECTOR DEL TRABAJO DE TESIS

CERTIFICA:

Haber dirigido, asesorado, revisado y corregido el presente trabajo de tesis de grado, en su proceso de investigación, cuyo tema versa: **“DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DOMÓTICO PARA LA MEDICIÓN DEL CONSUMO DE AGUA POTABLE A TRAVÉS DE INTERNET Y CORREO ELECTRÓNICO”**, previo a la obtención del título de **Ingeniero en Electrónica y Telecomunicaciones**, realizado por el señor egresado: Edison Fabricio Chuquimarca Sarango, la misma que cumple con la reglamentación y políticas de investigación, por lo que autorizo su presentación y posterior sustentación y defensa.

Loja, Julio del 2014.



Ing. Marcelo Fernando Valdiviezo Condolo
DIRECTOR DEL TRABAJO DE TESIS

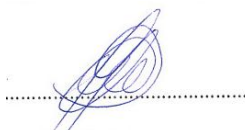
AUTORÍA

Yo **Edison Fabricio Chuquimarca Sarango**, 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: Edison Fabricio Chuquimarca Sarango

Firma:



Cédula: 1104140221

Fecha: 27 de Agosto de 2014

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

Yo **Edison Fabricio Chuquimarca Sarango** declaro ser autor de la tesis titulada **DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DOMÓTICO PARA LA MEDICIÓN DEL CONSUMO DE AGUA POTABLE A TRAVÉS DE INTERNET Y CORREO ELECTRÓNICO**, como requisito para obtener el 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, 27 días del mes de Agosto del dos mil catorce.

Firma:



Autor: Edison Fabricio Chuquimarca Sarango

Cedula: 1104140221

Dirección: Barrio Sauces Norte, Loja

Correo Electrónico: edisonfabricioch@gmail.com

Teléfono: 072558150

Celular: 0993796858

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Marcelo Fernando Valdiviezo Condolo

Tribunal de Grado: Ing. Juan Manuel Galindo Vera, Mg. Sc.

Ing. Rodolfo Pabel Merino Vivanco

Ing. Juan Pablo Cabrera Samaniego, Mg. Sc.

DEDICATORIA

Dedico esta tesis a DIOS quien inspiro mi espíritu para la conclusión de esta carrera universitaria. A mis padres quienes me dieron la vida, educación, apoyo y consejos. A mis compañeros de estudio, a mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma.

AGRADECIMIENTO

Empiezo agradeciendo a Dios por bendecirme, para llegar hasta donde he llegado. A mis padres y hermana que me acompañaron en esta aventura que significó obtener el título de ingeniería y que, de forma incondicional, entendieron mis ausencias y mis malos momentos. A la UNIVERSIDAD NACIONAL DE LOJA por darme la oportunidad de estudiar y ser un profesional. También me gustaría agradecer a mis profesores durante toda mi carrera profesional porque todos han aportado con un grano de arena a este título.

ÍNDICE

CERTIFICACIÓN	1
AUTORÍA	ii
DEDICATORIA	iv
AGRADECIMIENTO.....	v
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE TABLAS	xi
TEMA	xii
RESUMEN.....	xiii
SUMMARY	xiv
OBJETIVOS	xv
INTRODUCCIÓN	xvi
CAPITULO 1	1
1. FUNDAMENTOS TEÓRICOS.....	1
1.1 DOMÓTICA	1
1.2 INTERNET OF THINGS	3
1.2.1 APLICACIONES INTERNET OF THINGS: MONITOREO REMOTO DE SENSORES	6
1.3 MEDIDORES CONVENCIONALES DE AGUA Y SENSORES DE FLUJO.....	7
1.3.1 CAUDAL.....	7
1.3.2 MEDIDORES DE AGUA POTABLE	8
1.3.2.1 TIPOS DE MEDIDORES CONVENCIONALES	9
1.3.2.1.1 MEDIDORES DE DESPLAZAMIENTO O VOLUMÉTRICOS.....	9
1.3.2.1.2 DE TURBINA Y VELOCIDAD.....	12
1.3.3 MÉTODO DE LECTURA DE LOS MEDIDORES CONVENCIONALES.....	14
1.3.4 TIPOS DE MEDIDORES DE AGUA ELECTRÓNICOS	17
1.3.4.1 MEDIDOR ELECTROMAGNÉTICO	17
1.3.4.2 SENSOR DE FLUJO DE AGUA ELECTRÓNICO DE EFECTO HALL.....	19
1.3.4.3 MEDIDORES ULTRASÓNICOS UTILIZANDO EL EFECTO DOPPLER	20
1.4 SISTEMAS EMBEBIDOS	21
1.4.1 HARDWARE	21

1.4.2	SOFTWARE	22
1.4.3	RASPBERRY PI	23
1.4.4	FUENTE DE VOLTAJE Y ALMACENAMIENTO DE LA RASPBERRY PI	24
1.4.5	LECTURA DE DATOS DESDE LA GPIO	25
1.4.6	ARM VERSUS X86.....	25
1.4.7	ENTORNOS DE PROGRAMACIÓN	26
CAPITULO II		28
2.	DISEÑO Y CONSTRUCCIÓN DEL HARDWARE.....	28
2.1	DESCRIPCIÓN GENERAL DEL SISTEMA	28
2.2	ETAPA DE SENSORES	30
2.2.3	SENSOR DE FLUJO	30
2.2.4	SENSOR AM2301.....	32
2.2.5	PANTALLA LCD.....	34
2.2.6	INTERFAZ INALÁMBRICA WI-FI	35
CAPITULO III		38
3.	DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE.....	38
3.1	INSTALACIÓN SISTEMA OPERATIVO	38
3.2	CONEXIÓN REMOTA USANDO EL PROTOCOLO SSH	40
3.3	AÑADIR CONEXIÓN WIFI.....	42
3.4	INSTALACIÓN WIRINGPI.....	44
3.5	LECTURA DE VALORES DEL SENSOR AM2301.....	45
3.6	LECTURA DE VALORES DEL SENSOR DE EFECTO HALL	47
3.7	ENVIÓ DE DATOS A LA PLATAFORMA XIVELY	49
3.8	CREACIÓN DE REGISTROS DIARIOS Y MENSUALES	53
3.9	AÑADIR SERVICIO DE SMTP: GMAIL	54
3.10	CONEXIÓN CON GOOGLE DRIVE	56
3.12	TWITTERBOT	57
3.13	PANTALLA LCD.....	58
CAPITULO IV		60
4.	PRUEBAS Y RESULTADOS.	60

4.1	ENVIÓ DE DATOS A XIVELY.....	65
4.2	ENVIÓ DE CORREOS ELECTRÓNICOS	66
4.3	REGISTRO DE VALORES DIARIOS EN GOOGLE DRIVE	67
4.4	ENVIÓ DE NOTIFICACIONES A WHATSAPP	68
CAPITULO V		69
5.	ANÁLISIS DE COSTOS	69
5.1	ANÁLISIS DE COSTOS DEL DISPOSITIVO IMPLEMENTADO.....	69
5.2	COMPARACIÓN TÉCNICA DEL SISTEMA DE MEDICIÓN IMPLEMENTADO CON MEDIDORES COMERCIALES.....	71
CAPITULO VI		75
6.	CONCLUSIONES Y RECOMENDACIONES	75
6.1	CONCLUSIONES	75
6.2	RECOMENDACIONES	76
REFERENCIAS BIBLIOGRÁFICAS.....		78
ANEXOS.....		82

ÍNDICE DE FIGURAS

Figura 1: Tipos de Arquitecturas en Sistemas Domóticos.....	2
Figura 2: Diagrama general de comunicación de la plataforma Internet of Things.....	5
Figura 3: Dispositivos que se comunican por una red mundial	5
Figura 4: Diagrama de clasificación principal de los medidores convencionales.	9
Figura 5: Funcionamiento de un contador de pistón oscilante	10
Figura 6: Aspecto interior del contador de disco oscilante.....	11
Figura 7: Interior de un medidor de turbina a chorro único.....	12
Figura 8: Interior de un medidor de turbina a chorro múltiple.	13
Figura 9: Indicador numérico de un sensor convencional	15
Figura 10: Indicadores de agujas en medidor convencional.....	15
Figura 11: Indicador de aguja para decenas de litro	16
Figura 12: Indicador de aguja para litros.....	16
Figura 13: Indicador de aguja para decenas de litros.	16
Figura 14: Indicador de aguja para cientos de litros.	16
Figura 15: Clasificación de los medidores de agua electrónicos.....	17
Figura 16: Esquema del funcionamiento de un medidor magnético de flujo.....	18
Figura 17: Interior del tubo por donde circula el fluido en un medidor electromagnético.....	18
Figura 18: Sensor de Flujo de la marca SEEDSTUDIO.....	20
Figura 19: Raspberry Pi Modelo B.	23
Figura 20: Diagrama de bloques del prototipo de medición de agua potable electrónico.....	28
Figura 21: Sensor de flujo de efecto Hall instalado a una lado del medidor convencional	31
Figura 22: Diagrama de conexión de la resistencia Pull-Up	32
Figura 23: Protocolo de comunicación del AM2301.....	33
Figura 24: Protocolo de inicio de comunicación del AM2301.....	33
Figura 25: Valores 0L y 1L del sensor AM2301.....	34
Figura 26: Pantalla LCD manufacturada por la empresa ADAFRUIT Inc	34
Figura 27: USB Wi-Fi de la marca EDIMAX.	35
Figura 28: Circuito de conexión de sensores y Pantalla LCD.....	35
Figura 29: Montaje del prototipo de medición de agua potable.....	36
Figura 30: Interior de la caja de acrílico donde se encuentran los componentes del sistema	37
Figura 31: Tarjeta Kingston y Tarjeta Transcend.....	39
Figura 32: Utilidad Win32 Disk Imager.....	39
Figura 33: Ingreso por SSH a la Raspberry Pi desde terminal Linux.....	41
Figura 34: Menú de configuración general de la Raspberry Pi.	42
Figura 35: Comando 'lsusb' nos permite ver los dispositivos conectados a la interfaz de USB	43
Figura 36: Fichero "interfaces"	43
Figura 37: Fichero "wpa_supplicant".....	44
Figura 38: Diagrama de flujo del algoritmo para obtener la temperatura y humedad relativa	46

Figura 39: Diagrama de flujo del algoritmo para medir cantidad flujo.....	47
Figura 40: Fichero que iniciara el programa del contador del flujo.	48
Figura 41: Salida del terminal de comandos cuando la configuración para Xively esta lista.....	51
Figura 42: Diagrama de flujo del algoritmo para enviar datos hacia XIVELY..	51
Figura 43: Datos almacenados en Xively.....	52
Figura 44: Fichero de configuraciones de SSMTTP.	55
Figura 45: Mensajes generados automáticamente en Twitter.....	57
Figura 46: Interfaz Gráfica de Usuario creado en QT Creator.....	59
Figura 47: Pantalla LCD en funcionamiento.	59
Figura 48: Tarjeta Arduino UNO R3.....	60
Figura 49: Comprobación del sensor de flujo junto con la tarjeta Arduino UNO.	61
Figura 50: Valores obtenidos por el prototipo en seis mediciones diferentes..	62
Figura 51: Comparación de las medidas de consumo diario entre el medidor convencional y el prototipo implementado.....	63
Figura 52: Error porcentual del prototipo calculado en cada medición..	64
Figura 53: Historial diario del consumo de agua potable en Xively.	65
Figura 54: Historial semanal del consumo de agua potable en Xively.....	66
Figura 55: Modelo de correos electrónicos recibidos por los usuarios.	66
Figura 56: Registro del consumo diario de agua potable en un documento en Google Docs.	67
Figura 57: Estadísticas del consumo mensual realizadas en Google Docs.....	68
Figura 58: Notificaciones por medio de mensajes usando la plataforma Whatsapp	68
Figura 59: Medidor Electrónico de ABB GROUP.	72
Figura 60: Sistema de comunicación entre sensores y servidores dentro del ecosistema ABB GROUP.....	73
Figura 61: Sistema RFID de monitoreo de agua potable de IUSA..	74

ÍNDICE DE TABLAS

Tabla 1: Ventajas y Desventajas de los sensores Analógicos o Convencionales.....	14
Tabla 2: Ventajas y Desventajas de los sensores Electromagnéticos.	19
Tabla 3: Características Raspberry Pi.	24
Tabla 4: Puertos GPIO de la Raspberry Pi.....	25
Tabla 5: Diferencias entre RAM vs. X86.	26
Tabla 6: Tabla de valores del sensor de Flujo.	30
Tabla 7: Tabla de valores de las pruebas realizadas con el medidor convencional.	63
Tabla 8: Costos del prototipo implementado.	69
Tabla 9: Comparación precios medidores comerciales con el prototipo implementado.	70
Tabla 10: Comparación técnica medidores comerciales con el prototipo implementado.	71

TEMA

“DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO DE UN SISTEMA DOMÓTICO PARA LA MEDICIÓN DEL CONSUMO DE AGUA POTABLE A TRAVÉS DE INTERNET Y CORREO ELECTRÓNICO”.

RESUMEN

Este trabajo investigativo fue diseñado con el objetivo de poder medir el consumo de agua potable en un hogar a través de internet en tiempo real y con reportes de correo electrónico hacia el usuario. Se estudia e implementa la ingeniería de hardware y software que el prototipo necesitara para su correcto funcionamiento.

El hardware del prototipo está compuesto por un sensor de flujo basado en el fenómeno de efecto de Hall, este sensor envía los datos de medición al núcleo principal del sistema de medición que es una Raspberry Pi. Este sistema embebido procesa la información, la almacena y la envía hacia un servidor en internet que desplegará los datos hacia el usuario en una página web. Además la Raspberry Pi permite visualizar la información a través de una pantalla LCD tipo resistiva.

El software se basa en el sistema operativo Linux embebido usando la distribución Raspbian, dentro de esta distribución se usó varios lenguajes de programación como: C/C++, Python, HTML, PHP y Java Script. Cada uno de estos lenguajes tiene un fin específico: manejo de los sensores, envío de la información hacia internet, la creación de reportes de consumo por correo electrónico, alarmas por mensajes de texto y la creación de la interfaz web que funciona en la red local.

El prototipo de medición se implementó en una vivienda común de la ciudad de Loja a un lado del medidor de agua convencional. El sistema funciona las 24 horas sin interrupciones. Todos los días a las 23:59 se crean tablas de consumo diario que después son consultadas por el sistema para los reportes de correo electrónicos y cada minuto el sistema envía el consumo agua potable hacia un servidor en internet. Las alarmas por medio de mensajes ocurren bajo parámetro específicos como por ejemplo: la temperatura del CPU de la Raspberry Pi o la velocidad de flujo del agua potable rebasa un límite.

SUMMARY

This research work was designed with the aim of measuring the consumption of drinking water in a home, through Internet in real time with email reports to the user. It studies and implements the engineering of hardware and software, which prototype needs for proper functioning.

The prototype hardware is composed by a flow sensor based on the Hall Effect phenomenon, this sensor sends the measurement data to the main body of the measuring system, which is a Raspberry Pi. This embedded system processes information, stores it and sends it to a server on the Internet, which will display the data to the user in a web page. Furthermore Raspberry Pi allows to display the information through a resistive LCD monitor.

Software is based on Linux operating system embedded using the Raspbian distribution, in this distribution it was used various programming languages: C/ C++, Python, HTML, PHP and Java Script. Each of these languages has a specific purpose: managing the sensors, sending information to the Internet, creating consumer reports by email, text message alerts and creating of the web interface, which running on the local network.

The prototype measurement was implemented in a common house in Catamayo city, next to conventional water meter. The system operates 24 hours without interruption. Every day at 23:59 daily consumption tables are created and then queried by the system for electronic mail reports and every minute the system sends the water consumption to a server on the Internet. Alarms through text messages occur under specific parameters such as: the temperature of the Raspberry Pi CPU or when the flow rate of water exceeds a limit.

OBJETIVOS

OBJETIVO GENERAL

- Diseñar e implementar un sistema domótico que permita monitorear el consumo de agua potable a través de correo electrónico y acceso Web en una vivienda de la ciudad de Loja.

OBJETIVOS ESPECÍFICOS

- Estudiar y analizar las diferentes tecnologías en sistemas domóticos tanto para sensores, dispositivos de recepción y transmisión inalámbrica de datos, microcontroladores y sistemas embebidos.
- Implementar el hardware y software para la lectura digital del consumo de agua potable.
- Desarrollar una interfaz gráfica amigable para el usuario que facilite el monitoreo para proporcionar mayor confort y seguridad.

INTRODUCCIÓN

El rápido avance de las tecnologías de la información ha hecho que todas las personas, hogares y ciudades enteras estén conectadas a la red Internet. Una nueva tendencia es el Internet Of Things, con esto se refiere a una red de objetos cotidianos interconectados, es decir que las cosas tengan conexión a Internet en cualquier momento y lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. De esta manera, cualquier objeto es susceptible de ser conectado y «manifestarse» en la Red.

El Internet Of Things no solo es aplicable a objetos sino también a parámetros como: consumo de energía eléctrica o el consumo de agua potable.

Poder tener información en tiempo real desde cualquier parte del mundo sobre el consumo de agua potable en una vivienda es el primer paso en el campo de las casas inteligentes o “SMART HOME”. Esto le permite al usuario saber con exactitud cómo y cuándo se consume el líquido vital. Teniendo a la mano datos estadísticos confiables puede tomar decisiones más efectivas para un ahorro de consumo.

Pero este tipo de sistemas no solo se puede aplicar en hogares sino también a empresas de manufactura, sistemas de riego o los gobiernos municipales que deseen hacer que su red de distribución de agua potable sea más inteligente y robusta.

CAPITULO 1

1. FUNDAMENTOS TEÓRICOS

1.1 DOMÓTICA

Domótica es el conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes interiores y exteriores de comunicación, cableadas o inalámbricas, y cuyo control goza de cierta ubicuidad, desde dentro y fuera del hogar [1]. La domótica es una tecnología que tiene como objetivo fundamental proporcionar un estilo de vida más cómodo en el hogar. Puede definirse como la integración y aplicación de tecnologías como:

- Dispositivos electrónicos: Sensores, actuadores, etc.
- Tecnologías de Telecomunicaciones: Ethernet, Wi-Fi, Bluetooth, Zigbee, Fibra, etc.
- Sistemas de procesamiento central: Computadoras, Sistemas embebidos, Microcontroladores, etc.

Sus principales características son:

- Interacción.
- Facilidad de uso.
- Monitoreo a distancia.
- Fiabilidad.

En la actualidad la domótica se asocia al ámbito de las telecomunicaciones, hogar digital u hogar conectado. El gran proceso tecnológico sufrido por los sistemas de telecomunicación y el crecimiento exponencial de Internet, han incrementado la capacidad para generar información, almacenarla, transmitirla, recibirla y procesarla. El

mayor acceso a la información, ha venido además asociado a una mayor facilidad para comunicarnos en cualquier momento y desde cualquier lugar.

Ahora estamos en los comienzos de una revolución de servicios para el hogar, donde los hogares apoyados de conexiones de banda ancha, conectarán inteligentemente todos los dispositivos del hogar, añadiendo servicios de valor agregado de diversa índole pasando a formar parte de las Sociedades de la Información. Su arquitectura puede ser centralizada, distribuida o mixta tal como se puede observar en la figura 1.

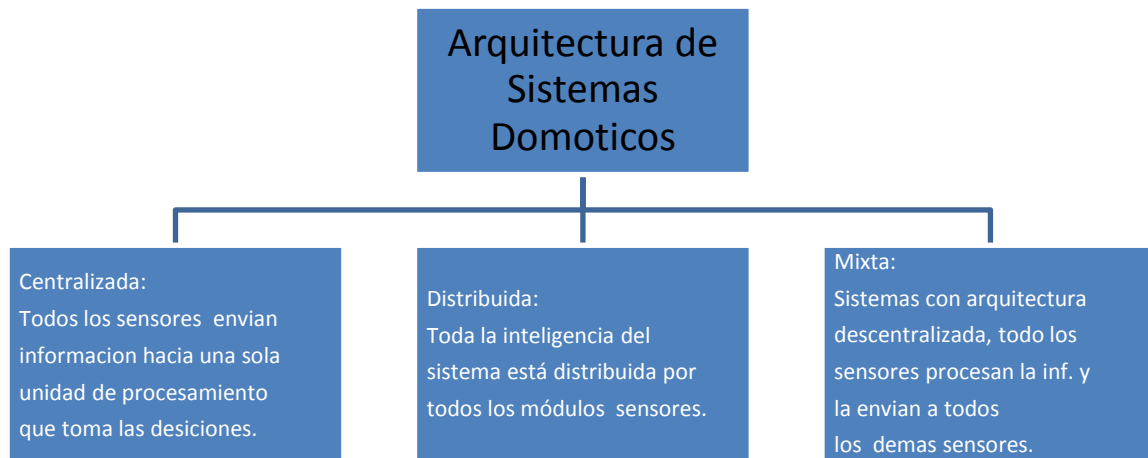


Figura 1: Tipos de Arquitecturas en Sistemas Domóticos. Fuente: [El Autor].

Algunas de las ventajas de vivir en un hogar domótico y conectado a la red Internet son:

- Programación del encendido y apagado de sistemas de luz, gas, agua, etc. Ya sea por el usuario o por un evento en específico.
- Contadores de consumo energético y de agua potable con reportes, alarmas, recordatorios, etc.
- Control de dispositivos electrónicos.
- Monitoreo de cámaras de seguridad y micrófonos.

- Servicios WEB para los usuarios.

Todos estos servicios en tiempo real y desde cualquier dispositivo que tenga acceso a internet como: Teléfonos inteligentes, computadoras portátiles, Televisores inteligentes, etc. Unas de las tecnologías que están teniendo éxito en el campo de la domótica aplicada a la red Internet es la corriente llamada: Internet Of Things (Internet de las Cosas).

El Internet de las cosas es un término que se refiere a una red de objetos cotidianos interconectados. El concepto de Internet de las cosas se atribuye a Auto-ID Center (grupo de investigadores de tecnologías RFID), fundado en 1999 y basado en el MIT (Instituto Tecnológico de Massachusetts) [2].

1.2 INTERNET OF THINGS

El Internet of Things consiste en que los objetos tengan conexión a Internet en cualquier momento y lugar. En un sentido más técnico, consiste en la integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes fijas e inalámbricas. El hecho de que Internet esté presente al mismo tiempo en todas partes permite que la adopción masiva de esta tecnología sea más factible. Dado su tamaño y coste, los sensores son fácilmente integrables en hogares, entornos de trabajo y lugares públicos. De esta manera todo tipo de sensores como: luz, humedad, temperatura, energía eléctrica, flujo de agua, etc. es susceptible de ser conectado e integrarse a Internet.

El correcto empleo del Internet of Things es por tres fenómenos recientes:

- La miniaturización de dispositivos electrónicos y de comunicación, lo que facilita que se pueda conectar prácticamente cualquier cosa en cualquier momento.
- La superación de la limitación de la infraestructura de telefonía móvil e Internet.

- La proliferación de las aplicaciones y los servicios que ponen en uso la gran cantidad de información creada a partir del Internet of Things (plataformas como Xively, Open Sen.se, EMMONS, etc.).

El Internet of Things constituye un avance con gran impacto sobre la sociedad y los negocios. Hasta el momento de la redacción de este documento cerca de tres mil millones de usuarios de todo el mundo utilizan Internet [3] tanto en su vida laboral como en la social y gracias a la tecnología Wireless se han ampliado las posibilidades de acceso a la Red. A medida que la información y las personas están cada vez más conectadas, la tecnología sirve como herramienta de colaboración y toma de decisiones en un mundo en el que converge lo físico con lo digital. Paul Horn, Vicerrector de Investigación de la Universidad de Nueva York y ex vicepresidente sénior de IBM, afirma que el mundo está siendo instrumentado e interconectado, a la vez que se vuelve más inteligente [4]. Los objetos que forman parte de nuestra vida cotidiana siempre han generado gran cantidad de información, pero esa información estaba fuera de nuestro alcance. Con el Internet of Things, pequeños sensores están siendo integrados en los objetos del mundo real y son instrumentos que proporcionan información de prácticamente todo lo que es posible medir. De esta manera, cada vez estamos más interconectados y las personas y objetos pueden interactuar de manera completamente distinta. Mediante radiofrecuencia (o por un medio cableado), la información viaja a un ordenador, dispositivo móvil o sistema embebido con acceso a Internet. Dicha información puede ser recibida por un usuario para su interpretación. También existe la posibilidad de que el extremo final sea otra máquina que interprete los datos y actúe según parámetros preestablecidos, tal como se observa en la figura 2.

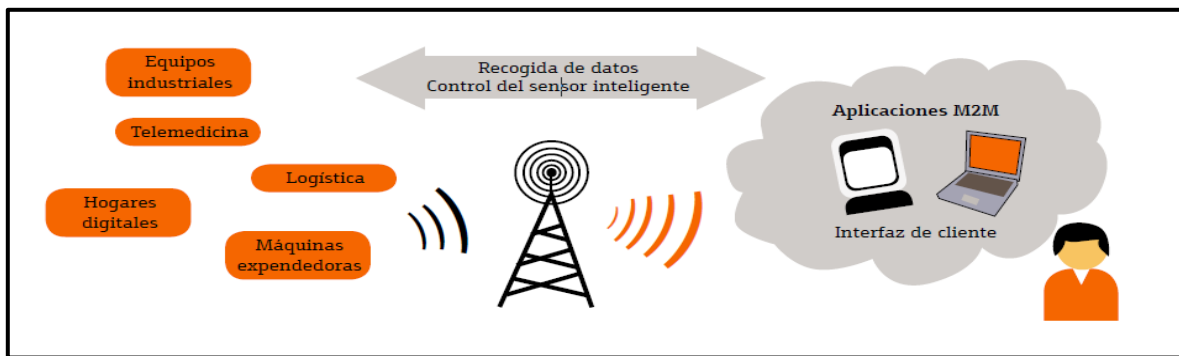


Figura 2: Diagrama general de comunicación de la plataforma Internet of Things. Fuente: [5].

Los sectores de la logística y el transporte han sido de los primeros en sumergirse en el concepto del Internet de las Cosas con su adopción de las etiquetas de identificación por radiofrecuencia. En 2010, cerca de 3.000 millones de etiquetas RFID se encontraban en circulación en el mundo. Las compañías de logística pueden optimizar sus cadenas de suministro al conocer con precisión la posición de todas sus mercancías y los vehículos pueden circular sin detenerse en los peajes de las autopistas. Sin embargo, no son más que los primeros pasos en el Internet of Things, a pesar de que estos dispositivos hayan revolucionado ambos sectores (ver figura 3).

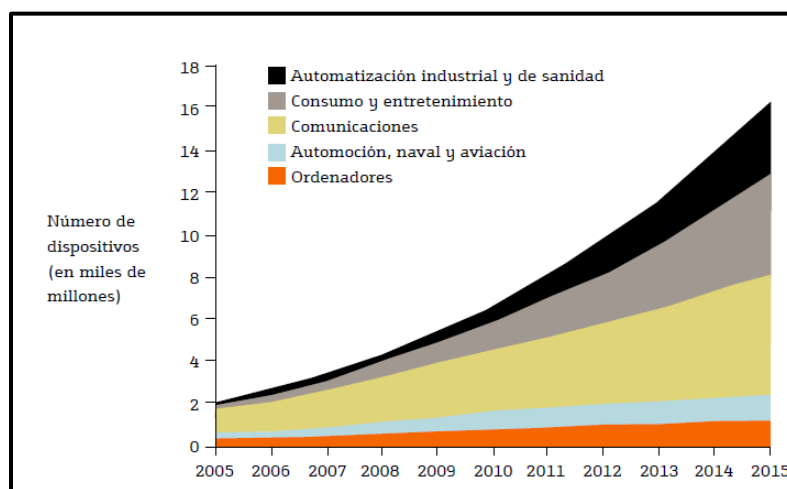


Figura 3: Dispositivos que se comunican por una red mundial. Fuente: [6].

Al dotar a los objetos con la posibilidad de comunicarse, se ha empezado a generar información en cantidades incalculables, abriendo todo un universo de posibilidades para negocios que se dediquen a su extracción, interpretación y análisis, lo que supone enormes avances a lo largo y ancho de las industrias. Los expertos del Future Trends Forum [7] prefieren ser cautos y señalar que el impacto no va a ser el mismo en todos los sectores. Señalan a la domótica, la logística, la medicina y la industria farmacéutica como los principales en los que el Internet Of Things va a tener mayor repercusión.

1.2.1 APLICACIONES DEL INTERNET OF THINGS: MONITOREO REMOTO DE SENSORES

El concepto de conectar objetos a Internet abrió una enorme ventana de oportunidades para la creación de aplicaciones en las áreas de la automatización y el uso de sensores.

Uno de los desafíos en el mundo actual es la eficiencia energética y ahorro, esto no solo se resume a energía eléctrica sino también al consumo de agua potable. Este recurso natural renovable es de gran importancia en los hogares, en los gobiernos y en el planeta en general. Una de las maneras de poder tener eficiencia en el consumo de agua es a través de un correcto monitoreo.

El uso de tecnologías combinadas como Domótica y el Internet de las cosas facilitan las tareas de monitoreo remoto, no solo para usuarios del hogar sino también para compañías que quieren tener información más detallada del uso de recursos y así poder tomar decisiones basadas en los valores proporcionados por los diferentes tipos de sensores instalados en la infraestructura de la empresa.

En un hogar el monitoreo remoto permite contar con información en tiempo real del uso de recursos. A esto se le añade la posibilidad de implementación de alarmas cuando un evento en específico se cumpla o el inicio de algún procedimiento particular ya sea por el usuario o por el cumplimiento de una regla en específico.

Los medidores tradicionales no pueden ser integrados fácilmente a la corriente de la Internet of Things, por ello se requiere de nuevas tecnologías de sensores de flujo de agua que permitan una fácil interacción, con sistemas de lectura y conexión a la Internet.

1.3 MEDIDORES CONVENCIONALES DE AGUA Y SENSORES DE FLUJO

1.3.1 CAUDAL

El caudal es la cantidad de fluido que pasa en una unidad de tiempo, llamada también flujo volumétrico o volumen que pasa por un área dada en la unidad de tiempo. El caudal de una tubería u objetos similares puede calcularse a través de la siguiente fórmula:

$$Q = A * \bar{v} \quad (1)$$

Dónde:

Q es el caudal en metros cúbicos por segundo.

A es el área en metros cuadrados

v es la velocidad lineal promedio en metros por segundo.

Dada una sección de área **A** atravesada por un fluido con velocidad uniforme **v**, si esta velocidad forma con la perpendicular a la superficie **A** un ángulo θ , entonces el flujo se calcula como:

$$\phi = A * V * \cos\theta \quad (2)$$

El caso particular de que el flujo sea perpendicular al área **A** (por tanto $\theta = 0$ y $\cos \theta = 1$) entonces el flujo es:

$$\phi = A * V \quad (3)$$

Si la velocidad del fluido no es uniforme o si el área no es plana, el flujo debe calcularse por medio de una integral:

$$\phi = \iint_S \mathbf{v} \cdot d\mathbf{S} \quad (4)$$

Donde $d\mathbf{S}$ es el vector superficie, que se define como:

$$d\mathbf{S} = \mathbf{n} dA \quad (5)$$

Donde \mathbf{n} es el vector unitario normal a la superficie y dA un elemento diferencial de área.

Si se tiene una superficie S que encierra un volumen V , el teorema de la divergencia establece que el flujo a través de la superficie es la integral de la divergencia de la velocidad \mathbf{v} en ese volumen:

$$\iint_S \mathbf{v} \cdot d\mathbf{S} = \iiint_V (\nabla \cdot \mathbf{v}) dV \quad (6)$$

En física e ingeniería, caudal es la cantidad de fluido que circula por unidad de tiempo en determinado sistema o elemento. Se expresa en la unidad de volumen dividida por la unidad de tiempo.

1.3.2 MEDIDORES DE AGUA POTABLE

Un medidor de agua es un dispositivo que, instalado en una tubería, permite conocer el flujo volumétrico (m^3/s) o caudal que está circulando por la misma. Los medidores tradicionales que se usan en la mayoría de hogares son de tipo mecánicos y estáticos al usuario, es decir no tienen ningún tipo de integración con las nuevas tecnologías de comunicación. A este grupo pertenecen los medidores de agua convencionales. La evolución de los métodos de medición ha desarrollado tecnologías como sensores electromagnéticos, de efecto Hall, de efecto Doppler y otras que permiten fácilmente ser

integradas a sistemas de procesamiento digital y con ello ampliar el abanico de servicios de los sensores tradicionales. A este grupo pertenecen los medidores de agua electrónicos.

1.3.2.1 TIPOS DE MEDIDORES CONVENCIONALES

Existen dos importantes grupos de medidores de agua convencionales como se observa en la figura 4:

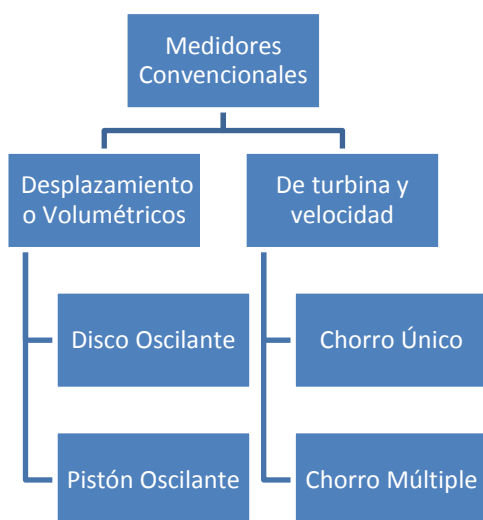


Figura 4: Diagrama de clasificación principal de los medidores convencionales. Fuente: [El autor].

1.3.2.1.1 MEDIDORES DE DESPLAZAMIENTO O VOLUMÉTRICOS

En un medidor de gasto por desplazamiento, el gasto se determina subdividiendo la corriente total del fluido en fracciones más o menos grandes de volumen conocido. La medición se efectúa entonces mediante el recuento del número de fracciones en la unidad de tiempo. Los dos tipos más corrientes de medidores por desplazamiento para líquidos son los contadores de pistón oscilante y de disco oscilante.

MEDIDORES DE PISTÓN OSCILANTE

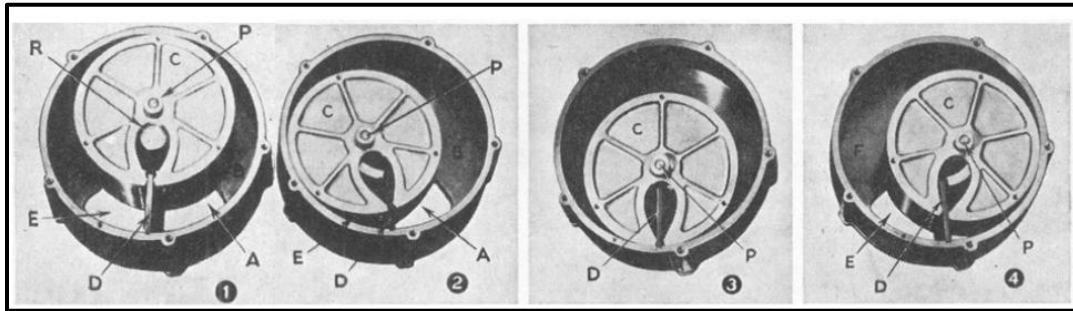


Figura 5: Funcionamiento de un contador de pistón oscilante. Fuente: [8].

La figura 5 representa la cámara de medida de un contador de pistón oscilante, mostrando a esta pieza en cuatro posiciones de una misma revolución, equidistantes entre sí. Todo el fluido penetra por el orificio de entrada y pasa alrededor del espacio anular, entre los anillos externo e interno, hacia el orificio de descarga. Parte del fluido recorre esta distancia por el espacio existente entre el pistón y el anillo externo, y parte circula por el espacio entre la pared del pistón hueco y el anillo interior.

El pistón va guiado por el eje, el cual sigue una trayectoria circular entre el anillo interno y un rodillo central. Una prolongación del eje que atraviesa la cubierta de la cámara de medida, comunica movimiento a una serie de ruedas dentadas que registran sobre una esfera graduada el volumen total de fluido que ha circulado por el contador a partir del momento en que se ajustó a cero.

MEDIDORES DE DISCO OSCILANTE

La cámara de medida de un contador de disco oscilante posee una forma aproximadamente cilíndrica y lleva un diafragma que se extiende radialmente desde un lado de la pared de la cámara hasta el rodamiento esférico del centro, del mismo modo que ocurre en la cámara del contador de pistón oscilante. Tan pronto como el fluido penetra en el espacio entre el disco y las paredes de la cámara, empuja al disco hacia

adelante imprimiéndole un movimiento de oscilación rotatoria. El movimiento del disco y del eje es similar al de un trompo en el momento en que llega al final de su impulso de giro y va a caer. Este tipo de contador se utiliza mucho en las conducciones de agua de capacidad media. Como se representa en la figura 6; existe un disco ranurado para dejar paso al diafragma que lleva un rodamiento esférico en su centro; formando un ángulo recto con él, hay un eje corto que une esta pieza al sistema de engranajes. Las superficies internas superior e inferior de la cámara de medida son cónicas, con inclinación hacia el centro, proporcionando así una línea de contacto con el disco.

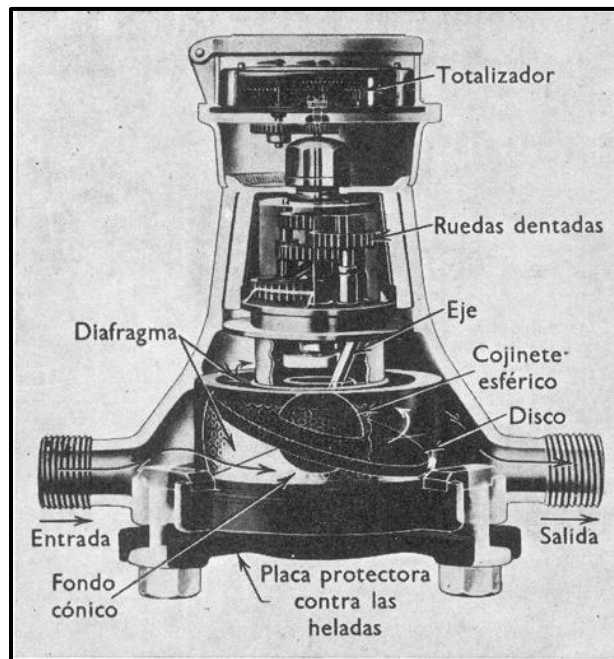


Figura 6: Aspecto interior del contador de disco oscilante. Fuente: [9].

1.3.2.1.2 DE TURBINA Y VELOCIDAD

Estos medidores emplean un procedimiento mecánico y que por acción de la velocidad del agua giran un mecanismo móvil, el cual puede ser una turbina o hélice

MEDIDORES DE TURBINA A CHORRO ÚNICO

Su mecanismo es accionado por medio de un chorro único de agua. El funcionamiento de este tipo de contador se basa en la incidencia tangencial de un único chorro sobre una turbina alojada en el interior de un cuerpo. La velocidad de giro de la turbina depende de la velocidad de impacto del chorro de agua. La figura 7 muestra detalladamente su funcionamiento.

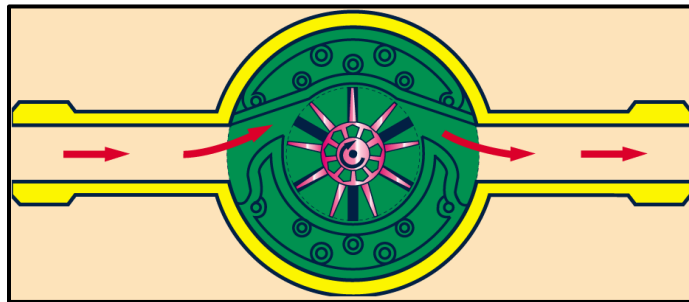


Figura 7: Interior de un medidor de turbina a chorro único. Fuente: [10].

MEDIDORES DE TURBINA A CHORRO MÚLTIPLE

El mecanismo es accionado por varios chorros tangenciales de agua. Se distinguen del chorro único, en que la turbina está dentro de la cámara con varios orificios de entrada y salida, diametralmente opuesta.

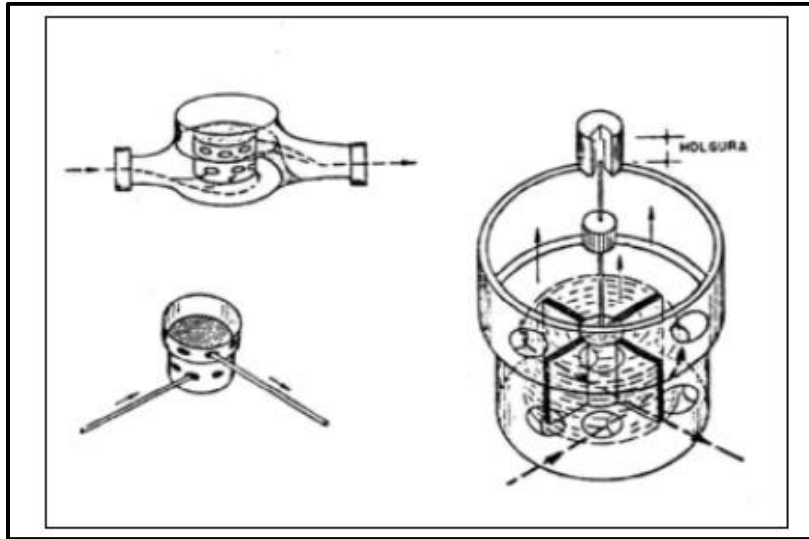


Figura 8: Interior de un medidor de turbina a chorro múltiple. Fuente: [11].

Ambos tipos de medidores: chorro único y chorro múltiple, miden el volumen de agua por la velocidad de la turbina. Tanto en el sistema único como en el múltiple, la turbina está en contacto con el agua, puesto que es en la que recibe el impulso que trasmite el movimiento al mecanismo indicador. Además, están dotados de un colador, por las posibles materias que pudieran pasar a través de él.

En la tabla 1 se muestra la información comparativa sobre las ventajas y desventajas de los sensores convencionales.

Tabla 1: Ventajas y Desventajas de los sensores Analógicos o Convencionales.

	Ventajas	Desventajas
Medidor de Pistón Oscilante	Sensible a gastos bajos.	Deben instalarse en sistemas de aguas con un alto grado de filtración, pues cualquier cantidad, por insignificante que sea de materiales en suspensión en el agua tales como arenas, partículas vegetales, sales de calcio, etc., detienen el medidor, obturando el paso del agua. Si no están debidamente instalados producen vibraciones en la línea de salida con las consecuentes molestias para el consumidor.
Medidor de Disco Oscilante	Sensible a gastos bajos.	Su funcionamiento en aguas no filtradas es deficiente además los medidores volumétricos no se deben colocar en cualquier posición ya que su vida media se reduce significativamente
Medidor de Chorro Único	Sensibilidad. Al tener una sola cámara de medición y un solo chorro.	La durabilidad de la cámara depende de la velocidad del agua entrante.
Medidor de Chorro Múltiple	Durabilidad. Al contar con varios chorros la cámara no está todo el tiempo en contacto con el agua.	Menor sensibilidad.

Fuente: [El autor].

1.3.3 MÉTODO DE LECTURA DE LOS MEDIDORES CONVENCIONALES

Los medidores tienen un mecanismo interno que es accionado con el paso del agua, este mecanismo es el que mueve de manera continua las agujas y los números ubicados en la carátula del medidor. El principio de funcionamiento se basa en que el agua penetra en la cámara de medición produciéndose el movimiento de una turbina o un pistón por el empuje de agua. Este movimiento se transmite durante un tren de engranes al registro donde se realiza la lectura.

La caratula del medidor presenta algunos valores de medición que se detallan a continuación:

- **Los números:** Todos los medidores son hechos con un código de colores, en el cual se usa el color negro para los indicadores de los metros cúbicos, y el rojo para los indicadores de los litros.

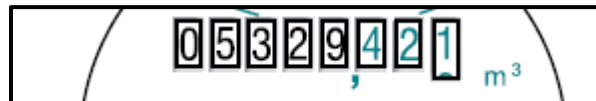


Figura 9: Indicador numérico de un sensor convencional. Fuente: [12].

- **Las agujas:** Para saber las cantidades de agua que está marcando la aguja de un reloj, primero se debe observar los números que se encuentran junto al reloj; estos números se llaman factores y según el factor será la cantidad de agua que indica la aguja.

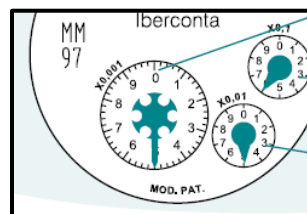


Figura 10: Indicadores de agujas en medidor convencional. Fuente: [13].

Finalmente, hay que tomar en cuenta que los indicadores trabajan en conjunto y que si un indicador comienza a marcar un número, este no está completo, si el indicador anterior no ha pasado de 9 a 0.

- El número $\times 0,0001$: Indica que la aguja marca la décima parte de un litro o un decilitro.

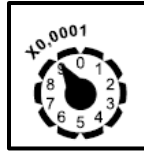


Figura 11: Indicador de aguja para decenas de litro. Fuente: [14].

- El número $\times 0,001$: Aparece siempre junto al reloj donde se marcan los litros.

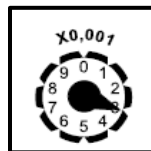


Figura 12: Indicador de aguja para litros. Fuente: [15].

- El número $\times 0,01$: Es ubicado junto al reloj donde se miden las decenas de litros.

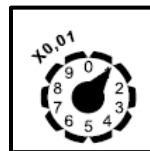


Figura 13: Indicador de aguja para decenas de litros. Fuente: [16].

- El número $\times 0,1$: Significa que la aguja de ese reloj va marcando los cientos de litros.

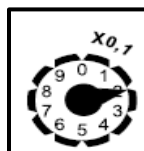


Figura 14: Indicador de aguja para cientos de litros. Fuente: [17].

1.3.4 TIPOS DE MEDIDORES DE AGUA ELECTRÓNICOS

Existen tres importantes grupos de medidores de agua electrónicos como se observa en la figura 15:

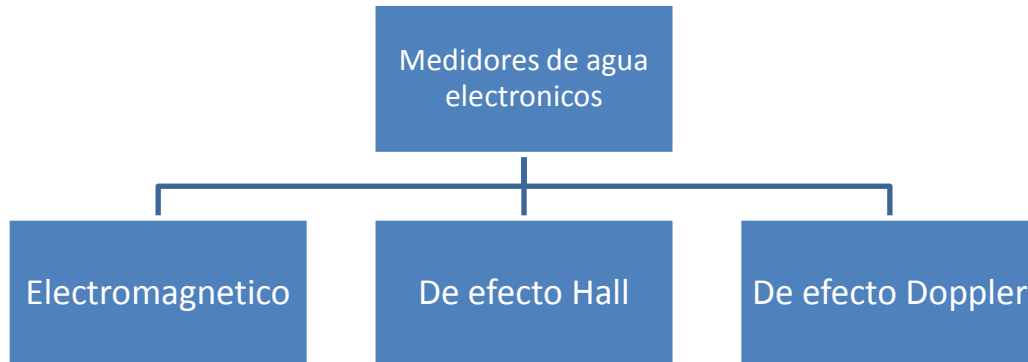


Figura 15: Clasificación de los medidores de agua electrónicos. Fuente: [El autor].

Estos medidores cuentan con sensores que son capaces de convertir el flujo de agua en algún tipo de voltaje eléctrico a su salida. Este voltaje eléctrico siguiendo unas reglas específicas puede entrar a un sistema de medición electrónico y convertirlo a valores reales de medición para el usuario.

1.3.4.1 MEDIDOR ELECTROMAGNÉTICO

El medidor de flujo electromagnético es un sistema diseñado para medir el caudal volumétrico de líquidos con conductividad eléctrica en sistemas de tuberías cerradas. Su principio de operación, basado en la ley de Faraday [18] de inducción electro-magnética, junto con su diseño de tubo libre de obstrucciones, provee a este tipo de instrumento con una serie de ventajas muy importantes comparado con otros sistemas de medición de flujo. A medida que un líquido conductor pasa a través del campo magnético existente dentro de un medidor, se genera un voltaje. Este voltaje es directamente proporcional a la velocidad promedio del flujo y Al ser el diámetro del tubo una variable conocida, el medidor magnético “calcula” el caudal que se desplaza por la tubería.

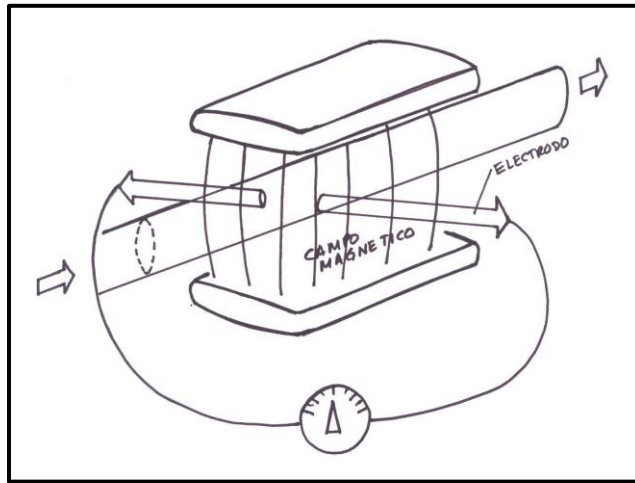


Figura 16: Esquema del funcionamiento de un medidor magnético de flujo. Fuente [19].

Alrededor del tubo se encuentran una serie de bobinas de diseño parecido al devanado de un motor, y con un núcleo semejante a los que se usan en un transformador, siendo las que producen el campo magnético. También cuenta con un par de electrodos que detectan la fuerza electromotriz que genera el agua a su paso por el campo magnético, enviando la señal para medición a un registrador que traduce la señal en información de caudales o volúmenes.

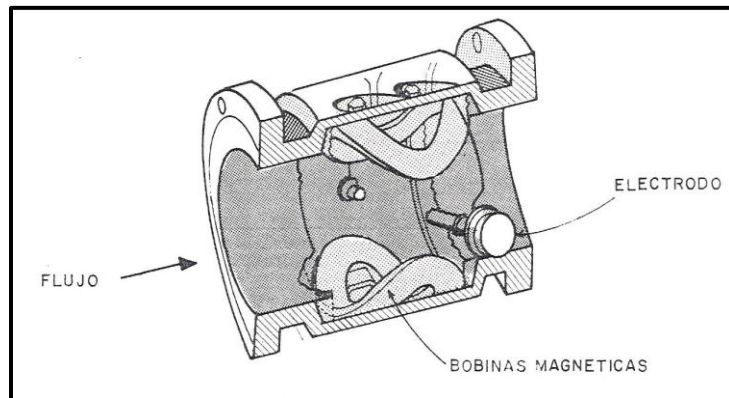


Figura 17: Interior del tubo por donde circula el fluido en un medidor electromagnético.

Fuente: [20].

En la tabla 2 se puede observar las ventajas y desventajas de los sensores electromagnéticos:

Tabla 2: Ventajas y Desventajas de los sensores Electromagnéticos.

Ventajas	Desventajas
<ul style="list-style-type: none"> • No posee partes móviles en contacto con el agua. • Para su instalación requiere una pequeña longitud de tramo recto aguas arriba. • Pérdida de carga despreciable. • Error de medición $\frac{1}{4}$ - $\frac{1}{2}$ %. • Puede manejar líquidos con sólidos en suspensión. • Instalación muy sencilla. 	<ul style="list-style-type: none"> • Alto costo. • Mano de obra especializada para su instalación, mantenimiento y calibración. • Requiere cuidados con respecto a las fuentes de energía externa que puedan provocar distorsiones en la operación normal. • Necesidad de mantenimiento periódico de los electrodos, pues las partículas metálicas que son arrastradas por el agua se van adhiriendo e interfiriendo en la medición.

Fuente: [El autor].

1.3.4.2 SENSOR DE FLUJO DE AGUA ELECTRÓNICO DE EFECTO HALL

El efecto Hall es la medición del voltaje transversal en un conductor cuando es puesto en un campo magnético. Mediante esta medición es posible determinar el tipo, concentración y movilidad de portadores. Es decir el sensor de efecto Hall crea un voltaje saliente proporcional al producto de la fuerza del campo magnético y de la corriente.

Este tipo de medidores tienen un sensor que funciona a modo de switch ON/OFF. Cuando un imán pasa cerca del sensor se produce un voltaje en sus terminales, si el imán se retira el voltaje cae a 0V. Este imán está integrado en el rotor. En la figura 18 se muestra un sensor de la empresa SEED STUDIO que dispone de un cuerpo plástico, 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.



Figura 18: Sensor de Flujo de la marca SEEDSTUDIO. Fuente: [21].

1.3.4.3 MEDIDORES ULTRASÓNICOS UTILIZANDO EL EFECTO DOPPLER

El efecto Doppler es el aparente cambio de frecuencia de una onda producida por el movimiento relativo de la fuente en relación a su observador. Este concepto no solo es aplicable para las ondas acústicas sino a muchos otros tipos de ondas entre ellas las electromagnéticas. Los medidores de tipo Doppler utilizan el concepto de que si se deja pasar el ultrasonido en un fluido en movimiento con partículas, el sonido será reflejado de nuevo desde las partículas. La variación de frecuencia del sonido reflejado será proporcional a la velocidad de las partículas. Al igual que en el caso de los medidores magnéticos, los medidores de caudal por ultrasonidos no presentan obstrucciones al flujo, no dan lugar a pérdidas de carga, por lo que son adecuados para su instalación en grandes tuberías de suministro de agua, donde es esencial que la pérdida de carga sea pequeña.

Los transductores son incorporados en el cuerpo del medidor, sin necesidad de juntas de contacto con el fluido. Respecto a la precisión, los medidores de tipo magnético pueden llegar hasta un 0.25% del caudal real, mientras que los de tipo de ultrasonidos hasta un

0.5%. Su fácil instalación reduce los costos de mantenimiento, y además la medición, sin apenas pérdida de carga, reduce los costes energéticos.

1.4 SISTEMAS EMBEBIDOS

Un sistema embebido se define como un sistema electrónico diseñado específicamente para realizar unas determinadas funciones. La característica principal es que emplea para ello uno o varios procesadores digitales (CPU) en formato microprocesador, microcontrolador o DSP lo que le permite aportar inteligencia más avanzada a un sistema.

1.4.1 HARDWARE

Normalmente un sistema embebido se trata de un módulo electrónico alojado dentro de un sistema de mayor entidad al que ayuda en la realización tareas tales como el procesamiento de información generada por sensores, el control de determinados actuadores, etc. El núcleo de dicho módulo lo forma al menos una CPU en cualquiera de los formatos conocidos:

- Microprocesador.
- Microcontrolador de 4, 8, 16 o 32 bits.
- DSP de punto fijo o punto flotante.
- Diseño a medida tales como los dispositivos FPGA.

Todos los sistemas embebidos tienen como característica fundamental los pines GPIO (Entradas y salidas de propósito general por su significado en español). Estos pines pueden funcionar como entrada de datos, salida de datos, contador de pulsos, conversores de datos analógicos a digitales o de digitales a analógicos, según los requerimientos de los usuarios. Las tecnologías de comunicación van desde interfaz serial, Usb, I2C, Ethernet, Wi-Fi, incluso algunos cuentan con interfaces de fibra óptica.

Algunos ejemplos de sistemas embebidos son:

- Microcontroladores MICROCHIP.
- Microcontroladores ATMEL.
- FPGA.
- Plataforma Arduino.
- Raspberry Pi.
- Intel Edison.
- Cubieboard.
- A13-OLinXino

1.4.2 SOFTWARE

En lo que se refiere al software, se tendrán requisitos específicos según la aplicación. En general para el diseño de un sistema embebido no se dispone de recursos ilimitados sino que la cantidad de memoria será escasa, la capacidad de cálculo y dispositivos externos será limitada, etc. Podemos hablar de las siguientes necesidades:

- Trabajo en tiempo real.
- Optimizar al máximo los recursos disponibles.
- Disponer de un sistema de desarrollo específico para cada familia de microprocesadores empleados.
- Programación en ensamblador, aunque en los últimos años, los fabricantes o empresas externas han mejorado la oferta de compiladores que nos permiten trabajar en lenguajes de alto nivel, tales como C.

El empleo de un sistema operativo determinado viene dado por el tipo de sistema que estemos empleando. La mayoría de microcontroladores, usa derivaciones del lenguaje

C/C++. Pero si son arquitecturas un poco más “complejas” tales como ARM o X86 estaremos ligados a un sistema operativo, por ejemplo Linux Embebido.

1.4.3 RASPBERRY PI

La Raspberry Pi es un sistema embebido de arquitectura ARM que es gobernada por Linux Embebido. El procesador en el interior de la Raspberry Pi es un procesador multimedia Broadcom BCM2835 system-on-chip (SoC). Esto quiere decir que la mayor parte de los componentes del sistema, incluidos la unidad central de procesamiento y la de gráficos junto con el audio y el hardware de comunicaciones, se encuentran integrados dentro de aquel único componente oculto ubicado justo debajo del chip de memoria al centro de la tarjeta. El Broadcom BCM2835 nos proporciona procesamiento, manejo de gráficos y control sobre los GPIO.

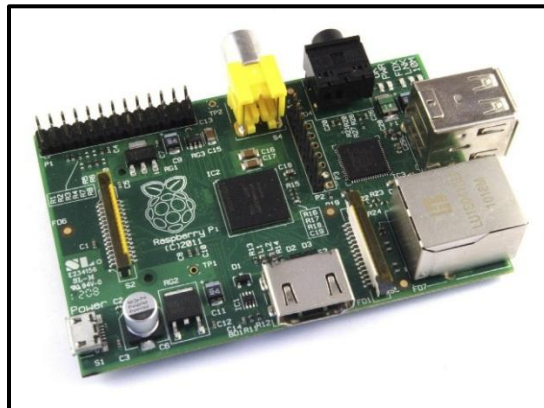


Figura 19: Raspberry Pi Modelo B. Fuente: [22].

La Raspberry de la figura 19 es una de modelo B, existe una de modelo A que tiene menor capacidad de Memoria RAM.

En la tabla 3 mostramos las principales características de la Raspberry Pi.

Tabla 3: Características Raspberry Pi.

SOC (System on Chip)	Broadcom BCM2835
CPU	ARM 1176JZF-S a 700 MHz -- 1000MHz
GPU	Broadcom VideoCore IV
RAM	512MB
Puerto USB 2.0	2
Salidas Multimedia	HDMI, RCA, AUDIO 3.5 mm
Conectividad	Fast Ethernet, USB Wi-Fi
Tipo de Almacenamiento	Tarjeta SD externa
Periféricos de bajo nivel:	8 x GPIO, SPI, I ² C, UART
RTC	No (Puede sincronizarse con un servidor NTP)
Alimentación	5V Micro USB – 1 Amp
SO	Debian (Raspbian) Fedora (Pidora) Arch Linux (Arch Linux ARM) Slackware Linux. RISC OS2
Dimensiones	85.60mm × 53.98mm

Fuente: [El autor].

1.4.4 FUENTE DE VOLTAJE Y ALMACENAMIENTO DE LA RASPBERRY PI

Para que pueda funcionar de manera normal la Raspberry Pi, se tiene que hacer uso de una fuente de poder de 5V con salida USB micro que proporcione un amperaje mayor a 800mA recomendable 1A. El consumo de este nivel de corriente se debe a que medida que se conecten dispositivos como: ratón, teclado, HDMI, RCA, audio de 3.5mm, cámara, GPIO, etc. la demanda de corriente aumenta y es posible que el aparato se quede en estado 'freeze'. La Raspberry Pi no dispone de ninguna interfaz de almacenamiento interno (excepto el firmware que se almacena en la GPU), por lo tanto es necesario hacer arrancar el SO desde una tarjeta SD (MicroSD si se hace uso de un adaptador).

1.4.5 LECTURA DE DATOS DESDE LA GPIO

La Raspberry Pi dispone de un puerto de expansión de entradas y salidas de propósito general conocido como GPIO (Tabla 4) y que se accede a él a través de un conector de cabecera rotulado como P1 en la placa. Todos los pines menos los de alimentación y tierra son reconfigurables y por tanto podremos adaptarlos a nuestras necesidades, aunque por defecto una serie de pines están configurados para la conexión de las interfaces UART, I2C y SPI. Los pines GPIO 4, GPIO 17, GPIO 27, GPIO 22, GPIO18, GPIO23, GPIO 24 y GPIO 25 son los que vienen configurados por defecto para entradas y salidas digitales.

Tabla 4: Puertos GPIO de la Raspberry Pi.

USO	PIN	PIN	USO
3.3 V	1	2	5 V
I2C SDA	3	4	DNC
I2C SCL	5	6	GND
GPIO4	7	8	UART TXD
DNC	9	10	UART RXD
GPIO17	11	12	GPIO 18
GPIO21	13	14	DNC
GPIO22	15	16	GPIO 23
DNC	17	18	GPIO 24
SP10 MOSI	19	20	DNC
SP10 MISO	21	22	GPIO 25
SP10 SCLK	23	24	SP10 CE0 N
DNC	25	26	SP10 CE1 N

Fuente: [23].

1.4.6 ARM VERSUS x86

El procesador es un circuito integrado cuya misión es permitir la comunicación entre las diferentes partes de un dispositivo, ya sean elementos de entrada como el teclado, el acelerómetro, o la pantalla táctil, o elementos internos tales como espacio disponible o

memoria RAM. Existen dos grandes modelos de procesadores en la actualidad, x86 y ARM. En la tabla 5 se muestra las principales características de estos modelos de procesadores.

Tabla 5: Diferencias entre RAM vs. X86.

ARM	x86
Arquitectura RISC	Arquitectura CISC
Bajo consumo energético	Alto consumo energético
Procesadores de 32 bits	Procesadores de 16,32, 64 bits
Arquitectura licenciable	Arquitectura Propietaria

Fuente: [El autor].

La arquitectura ARM está especialmente diseñada para aplicaciones donde el bajo consumo sea la prioridad, por ello la acogida de esta tecnología en la mayoría de teléfonos inteligente, calculadoras, tablets, etc. El BCM2835 basado en ARM es el secreto que explica cómo la Raspberry Pi es capaz de funcionar con tan sólo una fuente de alimentación de 5V 1A suministrada por el puerto micro-USB a bordo. Esto, sin embargo, significa que la Raspberry Pi no es compatible con el software de las PC tradicionales. La mayoría del software para computadoras de escritorio y laptops se construye teniendo en cuenta la arquitectura de conjunto de instrucciones x86, presente en los procesadores como AMD e Intel. Por consiguiente, este software no funciona en la Raspberry Pi que se basa en la arquitectura ARM.

1.4.7 ENTORNOS DE PROGRAMACIÓN

El éxito de la Raspberry Pi es que cualquier programador la puede empezar a usar. Al estar basada en Linux tenemos a nuestro alcance varios lenguajes como:

- C/C++
- Java
- Java Script

- Python
- PHP
- HTML
- ASM

Este universo de lenguajes hace que podamos llevar nuestros proyectos a límites un poco más allá de los tradicionales. Por ejemplo C/C++ se usa para el manejo del hardware directamente, Python se lo puede usar para aplicaciones gráficas y HTML o Java Script para aplicaciones orientadas a la WEB. Si juntamos estos lenguajes, tenemos un interesante proyecto en funcionamiento.

CAPITULO II

2. DISEÑO Y CONSTRUCCIÓN DEL HARDWARE

2.1 DESCRIPCIÓN GENERAL DEL SISTEMA

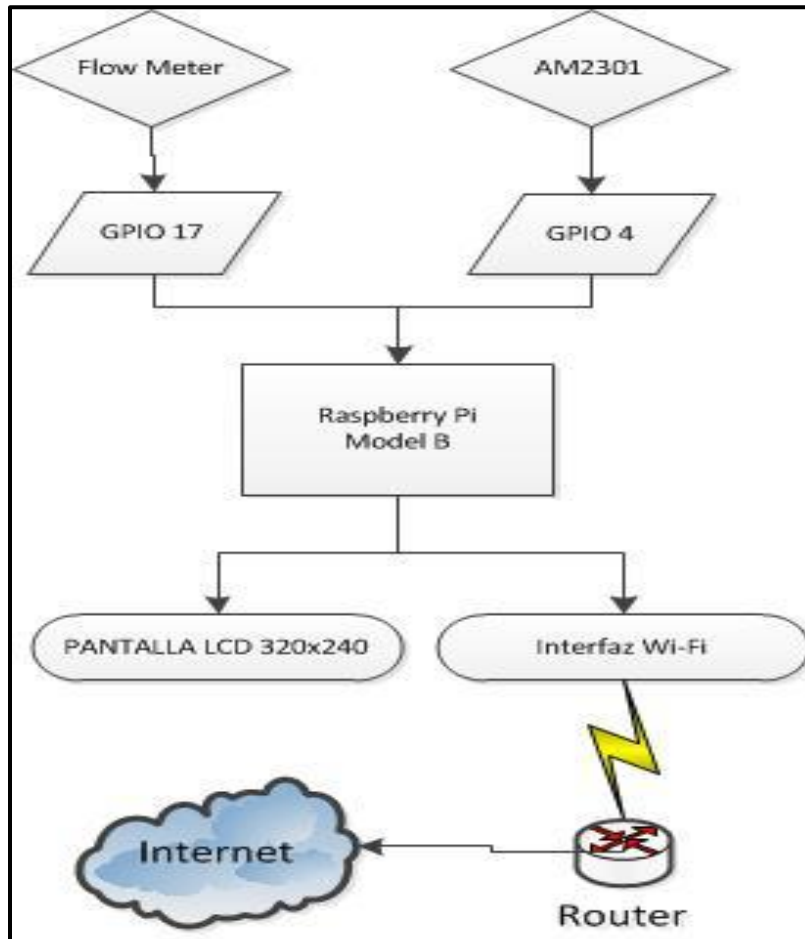


Figura 20: Diagrama de bloques del prototipo de medición de agua potable electrónico.

Fuente: [El autor].

El sistema de medición de agua potable hace uso de un sensor de efecto Hall tipo turbina que entrega un tren de pulsos, este tren de pulsos es leído por el sistema embebido Raspberry Pi que cuenta estos pulsos y aplicando una formula especifica obtiene el flujo de agua que pasa por el sensor en ese instante. Además se ha implementado un sensor

en tiempo real de temperatura y humedad ambiental. La Raspberry Pi se encarga de manipular los datos proporcionados por los sensores y en base a reglas definidas previamente almacena la información, la muestra y la envía por la interfaz inalámbrica hacia los diferentes servicios basados en Internet of Things. La etapa de Hardware del sistema de medición cuenta con 3 ejes fundamentales:

- Etapa de Sensores.
- Sistema embebido Raspberry Pi.
- Salida de Información: Pantalla LCD e interfaz inalámbrica.

En el diagrama de la figura 20 se observa que los sensores se conectan a los puertos GPIO de la Raspberry Pi, el sistema administra la información proporcionada y la envía como información visual a la pantalla LCD y al internet a través de la interfaz inalámbrica Wi-Fi.

2.2 ETAPA DE SENSORES

2.2.3 SENSOR DE FLUJO

El sensor de flujo implementado es manufacturado por la empresa SEEED STUDIO y usa el efecto Hall para su funcionamiento. Este sensor se acopla fácilmente a las tuberías de media pulgada que es la que se encuentra en la mayoría de hogares. La tabla 6 muestra las especificaciones técnicas del sensor.

Tabla 6: Tabla de valores del sensor de Flujo.

SENSOR DE EFECTO HALL	
Voltaje de Trabajo Mínimo	DC 4.5V
Corriente máxima de trabajo	15mA (DC 5V)
Voltaje de trabajo	DC 5V~24V
Rango de medición	1~30L/min
Capacidad de Carga	≤10mA (DC 5V)
Temperatura de Operación	≤80°C
Temperatura del Liquido	≤120°C
Humedad de Operación	35%~90%RH
Presión de Agua	≤1.75MPa
Ciclo de trabajo	40% - 60%
1L	>4.5V
0L	<0.5V
Error de medición	+/- 3%

Fuente: [El autor].

El sensor fue instalado a un lado del medidor comercial, esto con el objetivo de poder comparar las mediciones y su fácil manipulación en el periodo de pruebas. En la figura 21 se puede observar como quedo instalado el sensor.

El caudal puede calcularse por la siguiente formula:

$$Pulsos = 7.5 * Q \quad (7)$$

Donde **Q** es igual al flujo en litros por minuto con una precisión de +/- 3%.

El sensor de flujo de agua tiene tres pines: VCC, GND y DATA. Para la lectura del tren de pulsos se usa el pin GPIO 17 de la Raspberry Pi. La comunicación es unidireccional desde el sensor al GPIO, es decir el sensor siempre envía un tren de pulsos (en aun ausencia de flujo de agua el valor a la salida de 0V) por lo que pin GPIO siempre está en modo “listen”.



Figura 21: Sensor de flujo de efecto Hall instalado a una lado del medidor convencional.

Fuente: [El autor].

2.2.4 SENSOR AM2301

La medición de los valores de temperatura y humedad ambiental se realizó con el fin de poder tener parámetros en tiempo real que permitan saber si el sensor de flujo y la Raspberry Pi están trabajando en condiciones favorables y no sobrepasan las recomendaciones de los fabricantes. Esto con el fin de proteger a los dispositivos electrónicos y evitar un mal funcionamiento.

El fabricante recomienda usar una resistencia de Pull-Up (figura 22), esto con la finalidad de que los valores lógicos nunca queden en un estado “flotante”, es decir siempre tenga un valor de 1L o 0L.

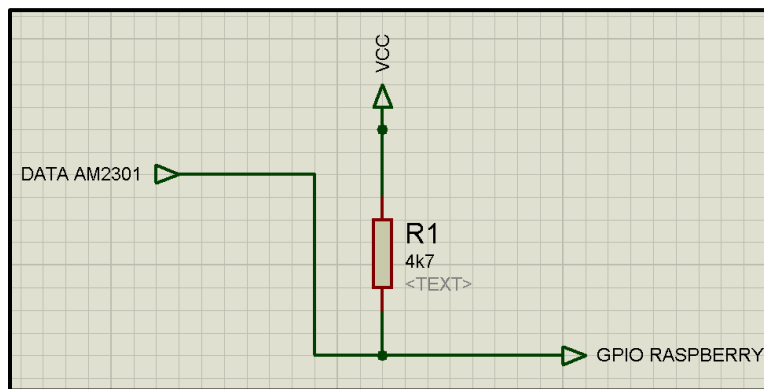


Figura 22: Diagrama de conexión de la resistencia Pull-Up. Fuente: [El autor].

El sensor AM2301 usa una comunicación de “single bus”, es decir una única línea de datos para el intercambio de información. La configuración es master-slave, La Raspberry Pi envía una petición y el sensor responde la petición. Este proceso debe tener tiempos de intervalo mayores a 2 segundos para obtener datos confiables. El sensor responde con 40 bits de información donde cada 8 bits tiene la información correspondiente tal como se muestra en la figura 23.

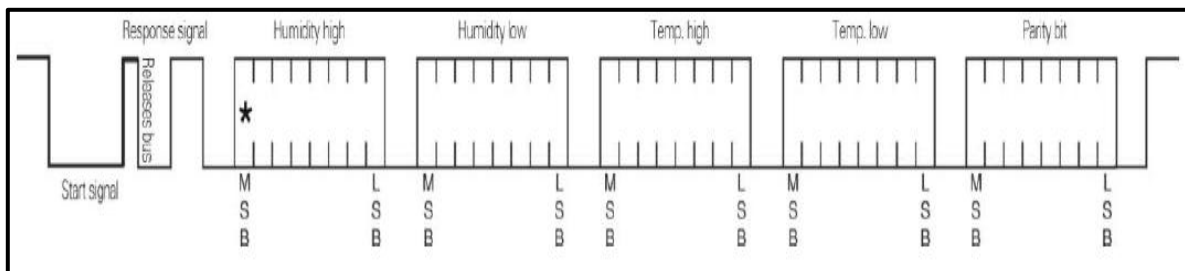


Figura 23: Protocolo de comunicación del AM2301. Fuente: [24].

La “start signal” de petición por parte de la Raspberry Pi y la “response signal” por parte del sensor se muestra en la figura 24.

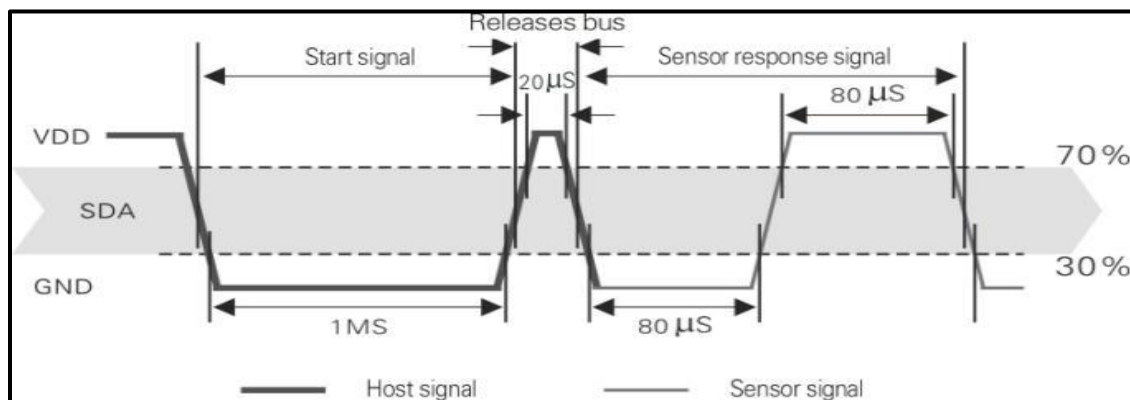


Figura 24: Protocolo de inicio de comunicación del AM2301. Fuente: [25].

Después que el sensor responde, envía inmediatamente los datos tal como se muestra en la figura 25. Como se puede apreciar el GPIO de la Raspberry Pi tiene que pasar de salida de datos a entrada de datos, además realizar la conversión de los valores Hexadecimales a valores decimales reales. El GPIO 4 se usa para conectar el sensor.

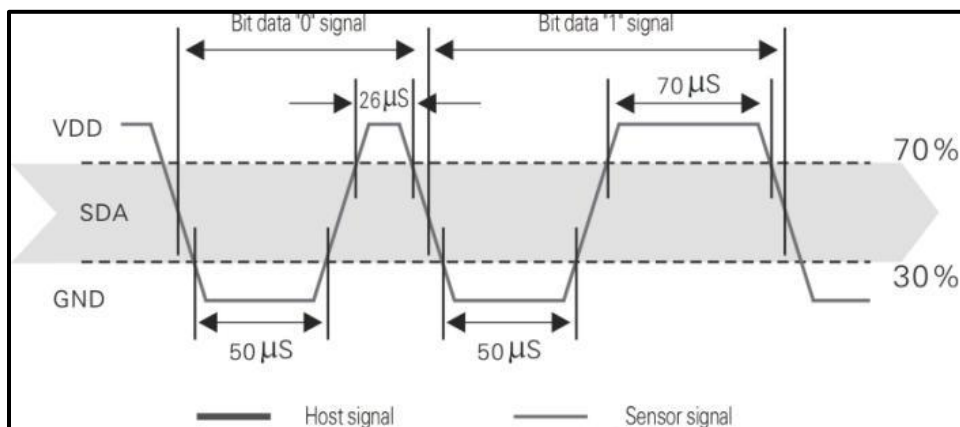


Figura 25: Valores 0L y 1L del sensor AM2301. Fuente: [26].

2.2.5 PANTALLA LCD

La pantalla que se implementó en el sistema es manufacturada por la empresa Adafruit (ver figura 26), esta pantalla es de 320x240 (2.8 pulgadas), tipo capacitiva a 16 bits de color por pixel. Para la comunicación se usa la interfaz SPI de la Raspberry Pi además de los pines GPIO 24 y GPIO 25. De la misma manera para la alimentación se usa los pines 1 (VCC) y 3 (GND) que proporciona la Raspberry Pi.

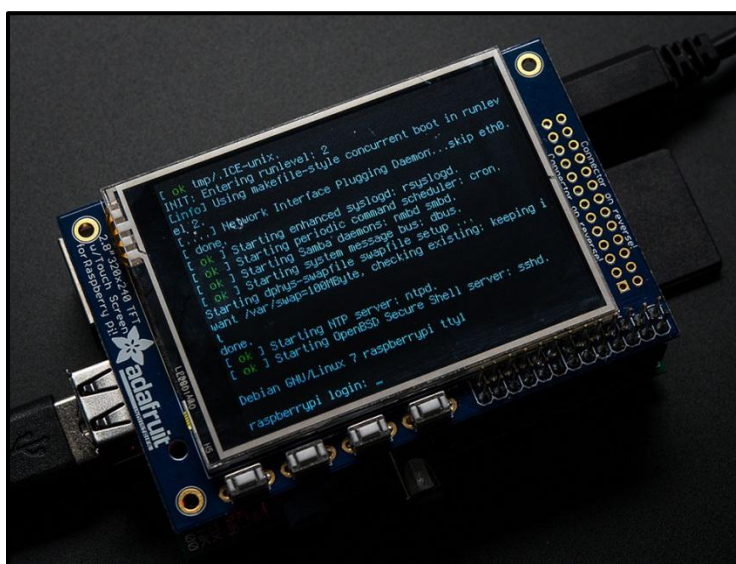


Figura 26: Pantalla LCD manufacturada por la empresa ADAFRUIT Inc. Fuente: [27].

2.2.6 INTERFAZ INALÁMBRICA WI-FI

La Raspberry Pi cuenta con una interfaz FASTETHERNET, pero para este sistema se decidió usar un adaptador USB Wi-Fi (ver figura 27) que es completamente compatible con la arquitectura ARM-Linux.



Figura 27: USB Wi-Fi de la marca EDIMAX. Fuente: [El autor].

Este adaptador tiene un alcance de hasta 70m en ambientes sin ninguna interferencia. En ambientes con interferencia se llega hasta 15m. Todo esto depende también de la potencia de radio transmisor del router. Para poder agrupar todos los sensores y la pantalla se diseñó un circuito donde se pueda conectar todos los componentes. En la figura 28 se puede observar el circuito.

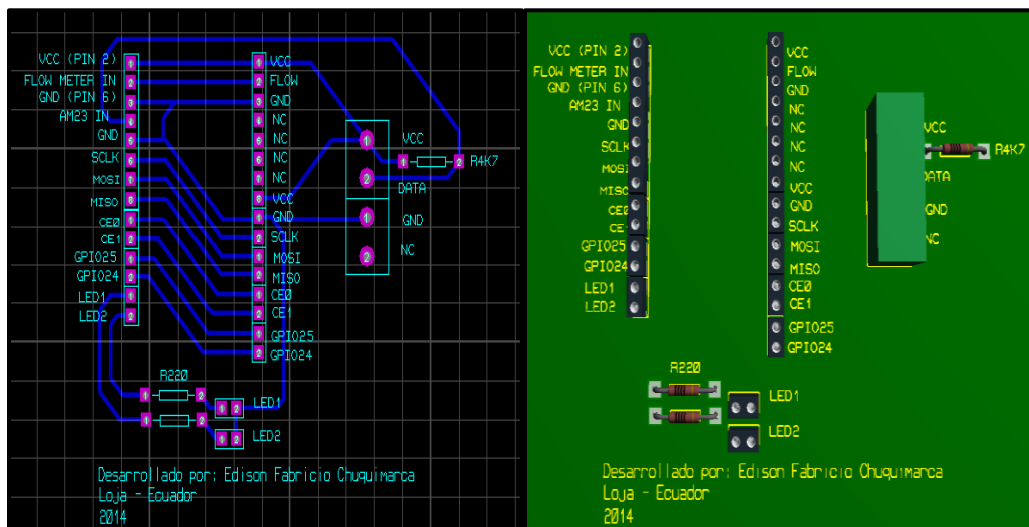


Figura 28: Circuito de conexión de sensores y Pantalla LCD. Fuente: [El autor].

Como se puede observar en la figura 28 se añadió dos LEDS informativos para el usuario: Un led se activa solamente cuando existe algún tipo de flujo en el sensor, si no existe flujo él led permanece apagado. Otro led cumple la función de informar que el sistema está operando correctamente, si el sistema de medición tiene algún problema el led se apaga.

En el momento de realizar las pruebas de funcionamiento surgió un problema referente a la temperatura del CPU de la Raspberry Pi que llegaba a valores incluso mayores de 60°C, por lo que se decidió colocar disipadores sobre el SOC principal y el concentrador de los puertos USB y FastEthernet, acompañados de un ventilador de 12V que no comparte alimentación con la Raspberry Pi por motivos de corriente. Para poder proteger el circuito se diseñó y se implementó una caja de acrílico como se puede observar en la figura 29 y 30.

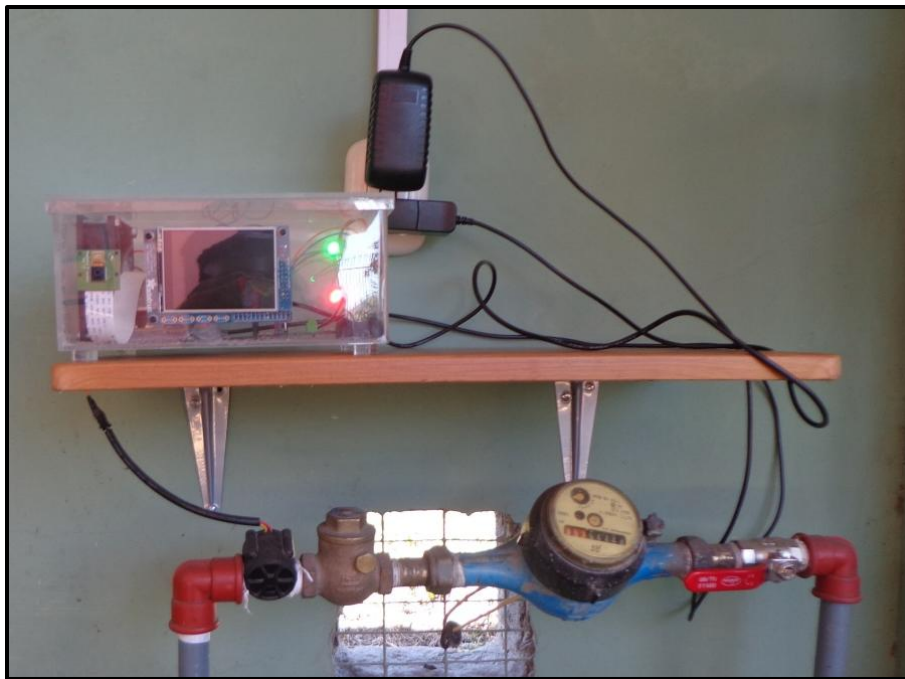


Figura 29: Montaje del prototipo de medición de agua potable. Fuente: [El autor].

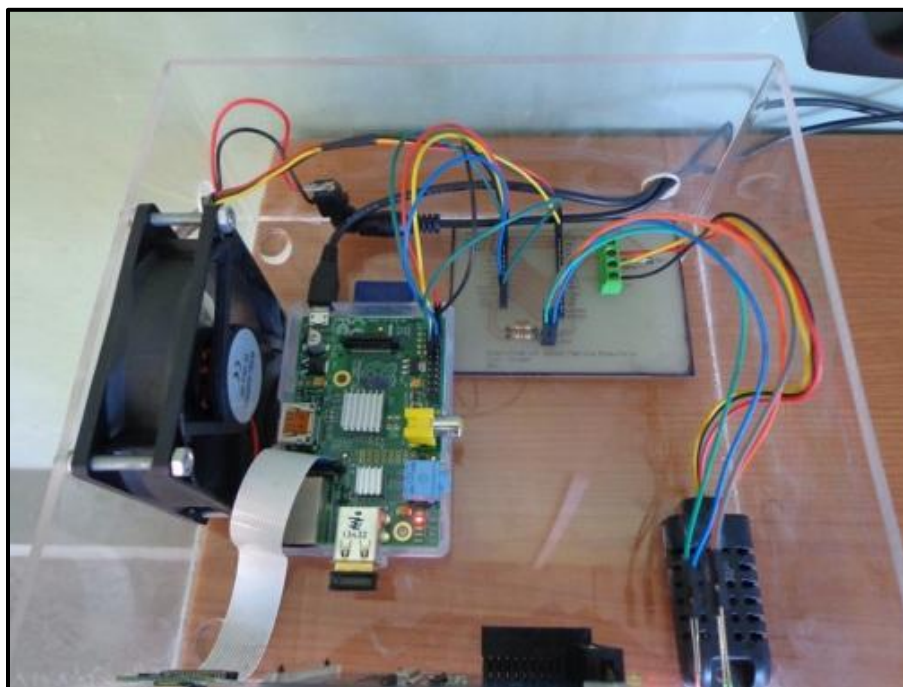


Figura 30: Interior de la caja de acrílico donde se encuentran los componentes del sistema. Fuente: [El autor].

CAPITULO III

3. DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

El hardware de la Raspberry Pi se maneja desde un sistema operativo basado en Linux Embebido. Para el desarrollo del proyecto se decidió usar Raspbian. La razón se debe a que esta es la única distribución que recibe actualizaciones oficiales por parte de la empresa propietaria, además en la comunidad de usuarios es la que más se usa por lo tanto hay mucha información disponible. A continuación se detalla los pasos seguidos para la implementación del software del sistema de medición de agua potable.

3.1 INSTALACIÓN SISTEMA OPERATIVO

La Raspberry Pi no dispone de ninguna interfaz de almacenamiento interno (excepto el firmware que se almacena en la GPU), por lo tanto es necesario hacer arrancar el SO desde una tarjeta SD (MicroSD si se hace uso de un adaptador). En un primer momento se usó una tarjeta MicroSD Clase 4 Kingston. Pero con el pasar del tiempo la tarjeta empezó a corromperse en algunos sectores de memoria. Estos problemas se solucionaban usando la herramienta de Linux *fsck* que se encarga de comprobar los bloques de memoria de la tarjeta y tratar de repararlos, aun así los problemas persistían por lo que se optó por usar una tarjeta SD de 8 GB Clase 10 Transcend (ver figura 31). Esta última al tener mayor velocidad, 10MB/s, y al ser de mayor calidad tiene un mejor rendimiento que se comprueba fácilmente al iniciar el SO y ejecutar los programas.

Las imágenes de los SO están disponibles para su descarga desde la página oficial de la fundación Raspberry Pi: [**http://www.raspberrypi.org/download**](http://www.raspberrypi.org/download)

En este caso instale el SO Raspbian usando el programa para Windows Win32 Disk Imagen: [**http://sourceforge.net/projects/win32diskimager/**](http://sourceforge.net/projects/win32diskimager/)



Figura 31: Tarjeta Kingston y Tarjeta Transcend. Fuente: [El Autor].

Es posible realizar la grabación del SO Raspbian en la SO Linux, a través de líneas de comandos, pero es un poco más tedioso y no muy práctico. Desde Windows solo hay que localizar la imagen del SO y la ubicación de la tarjeta SD y dar click en el botón “Write” y esperar a que se complete la grabación tal como se muestra en la figura 32.

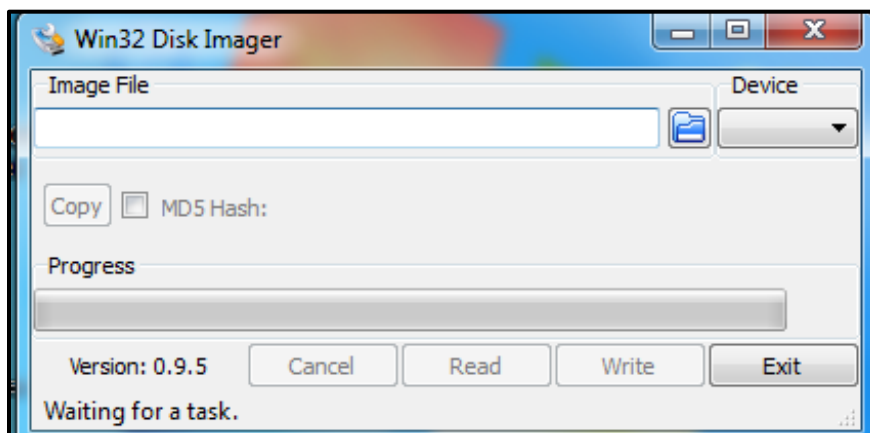


Figura 32: Utilidad Win32 Disk Imager. Fuente: [El Autor].

3.2 CONEXIÓN REMOTA USANDO EL PROTOCOLO SSH

Para poder acceder a la Raspberry Pi se puede optar por cualquiera de las siguientes configuraciones:

- Conectando una pantalla a la salida HDMI o el conector de video analógico, conectar teclado y ratón a los puertos USB.
- Por conexión remota desde otra computadora usando el protocolo SSH.

La primera opción es un poco molesta ya que ocuparía mucho espacio, la segunda y mejor opción es la conexión remota SSH. El protocolo SSH hace uso del puerto 22 y permite acceder a un host remoto y enviar órdenes a través de un intérprete de comandos. Para ello se necesita conocer la dirección IP que se le asigna en la red local a la Raspberry Pi. Es preferible para la Raspberry Pi asignar una IP estática por identificación de MAC en el router que se esté haciendo usando para no estar averiguar en cada momento que IP dinámica se le ha asignado a la Raspberry Pi. Un cable cruzado (Router -- Host) FastEthernet de máximo 100m se puede usar para poder dar acceso remoto e Internet a la Raspberry Pi. Usando un cliente SSH como Putty (que es multiplataforma) o usar la línea de comandos en una terminal en Linux. El desarrollo de este proyecto se lo realizo en una PC Linux asi que se usó la línea de comandos. En la terminal se ingresa lo siguiente:

ssh pi@172.16.10.250

Donde “pi” es el nombre del usuario default, como es la primera vez que se da la autenticación a ese host nos pedirá una confirmación, se acepta con “yes” y se ingresa el password por default que es ***raspberry***.

```
[edison@edisonfabricio ~]$ ssh pi@172.16.10.250
The authenticity of host '172.16.10.250 (172.16.10.250)' can't be established.
RSA key fingerprint is 24:5e:56:9a:ec:e7:57:bb:6a:e7:b5:42:d3:d1:8d:11.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.10.250' (RSA) to the list of known hosts.
pi@172.16.10.250's password:
Linux raspberrypi 3.10.25+ #622 PREEMPT Fri Jan 3 18:41:00 GMT 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan  7 23:33:01 2014
pi@raspberrypi ~ $
```

Figura 33: Ingreso por SSH a la Raspberry Pi desde terminal Linux. Fuente: [El Autor].

No es necesario ingresar como root al sistema pues el usuario “pi” tiene los mismos privilegios de root, para cualquier comando que requiera privilegios solo se debe anteponer **sudo**. El sistema operativo Raspbian no está totalmente listo, hay realizar algunas configuraciones que se detallan a continuación.

EXPANDIR TAMAÑO DEL SO:

Como la imagen de Raspbian para la Raspberry tiene un tamaño de 2.6 GB es posible extender ese tamaño para que ocupe toda la tarjeta SD, en este caso una de 8GB, para poder tener más espacio disponible para actualizaciones, aplicaciones, multimedia, etc. Para ello se ingresa el Comando `sudo raspi-config >> Expand File system`, seguidamente de un reboot al sistema.

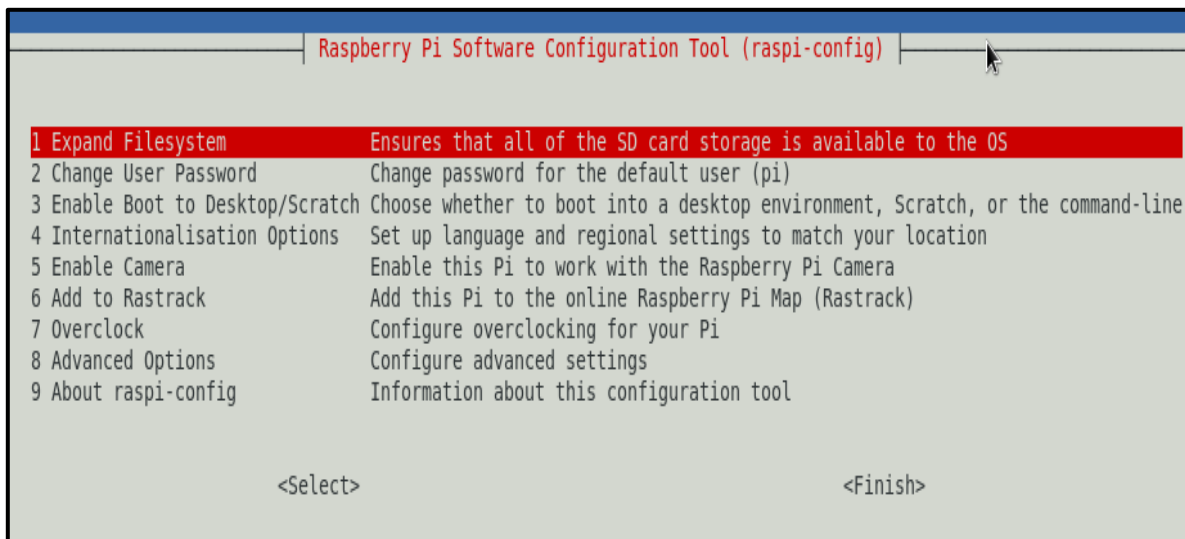


Figura 34: Menú de configuración general de la Raspberry Pi. Fuente: [El Autor].

REAL CLOCK TIME O SERVIDOR NTP

La Raspberry Pi no dispone de un RTC (Real Time Clock) interno. Existe la posibilidad de usar un DS1307 y solucionar este problema. Pero hay una solución más eficaz y económica: Servidores NTP (Network Time Protocol). Si se tiene acceso siempre a internet (fundamento principal del Internet of Things) podemos realizar peticiones para la sincronización de la fecha y hora. Para la sincronización mediante NTP es necesario decirle a la Raspy en que zona horaria estamos. Para ello por líneas de comandos se ingresa: `sudo raspi-config >> Internationalisation Options >> Change Timezone >> None of the above >> GMT-5` (zona horaria de Ecuador).

3.3 AÑADIR CONEXIÓN WIFI

Se recomienda conectar el adaptador Wi-Fi cuando la Raspberry Pi no esté conectada a la corriente eléctrica. Una vez encendida se debe ingresar por SSH y en el terminal se escribe ***lsusb***.

```
edison@raspberrypiedison:~$ lsusb
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
Bus 001 Device 004: ID 7392:7811 Edimax Technology Co., Ltd EW-7811Un 802.11n Wi
reless Adapter [Realtek RTL8188CUS]
edison@raspberrypiedison:~$
```

Figura 35: El comando 'lsusb' nos permite ver los dispositivos conectados a nuestra interfaz de USB. Fuente: [El autor].

Para la conexión con una red inalámbrica es necesario saber el nombre exacto de la red y la contraseña de la misma y editar estos archivos de esta manera:

sudo nano /etc/network/interfaces

El fichero a editar define las interfaces de red ya sean loop, ethe, wlan, etc. Observar la figura 36.

```
auto lo
iface lo inet loopback
iface eth0 inet dhcp
allow-hotplug wlan0
iface wlan0 inet dhcp
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf
iface default inet dhcp
```

Figura 36: Fichero "interfaces". Fuente: [El autor].

Una vez editado este fichero se guardan los cambios y se ingresa en la terminal este comando:

sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

El fichero `wpa_supplicant` es multiplataforma con soporte para WPA y WPA2 (IEEE 802.11). Está adaptado tanto para los ordenadores de sobremesa/portátiles como para sistemas embebidos basados en Linux [28]. Implementa las negociaciones entre la clave y un WPA Authenticator, y controla el roaming y la asociación/authenticación IEEE 802.11 del controlador wlan. Observar la figura 37.

```
ctrl_interface=DIR=/var/run/wpa_supplicant
GROUP=netdev
update_config=1
network =
{
  ssid="Raspberry-Pi"
  psk="fabricio1991"
  proto=WPA
  key_mgmt=WPA-PSK
  pairwise=TKIP
  auth_alg=OPEN
}
```

Figura 37: Fichero “wpa_supplicant”. Fuente: [El autor].

3.4 INSTALACIÓN WIRINGPI

Para manejar los pines GPIO de la Raspberry Pi se utiliza el lenguaje C/C++ ya que al ser un herramienta de bajo nivel permite acceder de manera más eficaz al hardware del dispositivo sin consumir demasiada memoria y recursos.

WiringPi es una biblioteca correspondiente a Arduino escrito en C y de distribución libre que se puede utilizar desde C/C++. La biblioteca original, que funciona en Arduino, se encarga de gestionar de manera eficiente los pines de entrada y salida así que WiringPi es lo mismo pero replicado hacia los GPIO de la Raspberry Pi.

Para instalar y compilar wiringPi en la terminal se debe ingresar estos comandos:

```
sudo apt-get install git-core  
sudo git clone git://git.drogon.net/wiringPi  
cd wiringPi  
git pull origin  
sudo ./build
```

3.5 LECTURA DE VALORES DEL SENSOR DE TEMPERATURA/AMBIENTE AM2301

Para la lectura de datos del sensor AM2301 se debe tomar en cuenta los parámetros del protocolo de comunicación master-slave entre la Raspberry Pi y el sensor AM2301 que se detallaron en el capítulo II sección 2.2.4. La figura 38 muestra el diagrama de flujo correspondiente a la lectura de los valores del sensor.

Se debe crear un fichero de extensión .C con el nombre am2301.c donde se alojara el código, este código se encuentra disponible para su consulta en el anexo A. Para ejecutar el código se debe primeramente compilar, para ello se usa la herramienta GCC (GNU Compiler Collection):

```
sudo gcc -o "archivo de salida" am2301.c -L/usr/local/lib -lwiringPi
```

Donde "archivo de salida" es el programa final y am2301.c es el fichero de extensión .C donde está escrito el código. Para la ejecución del código se ingresa este comando: **sudo ./"archivo de salida"**.

Este código no se ejecutara siempre, solo será llamado cuando la Raspberry Pi esté lista para enviar la información hacia internet. Funcionará como una librería externa al programa principal.

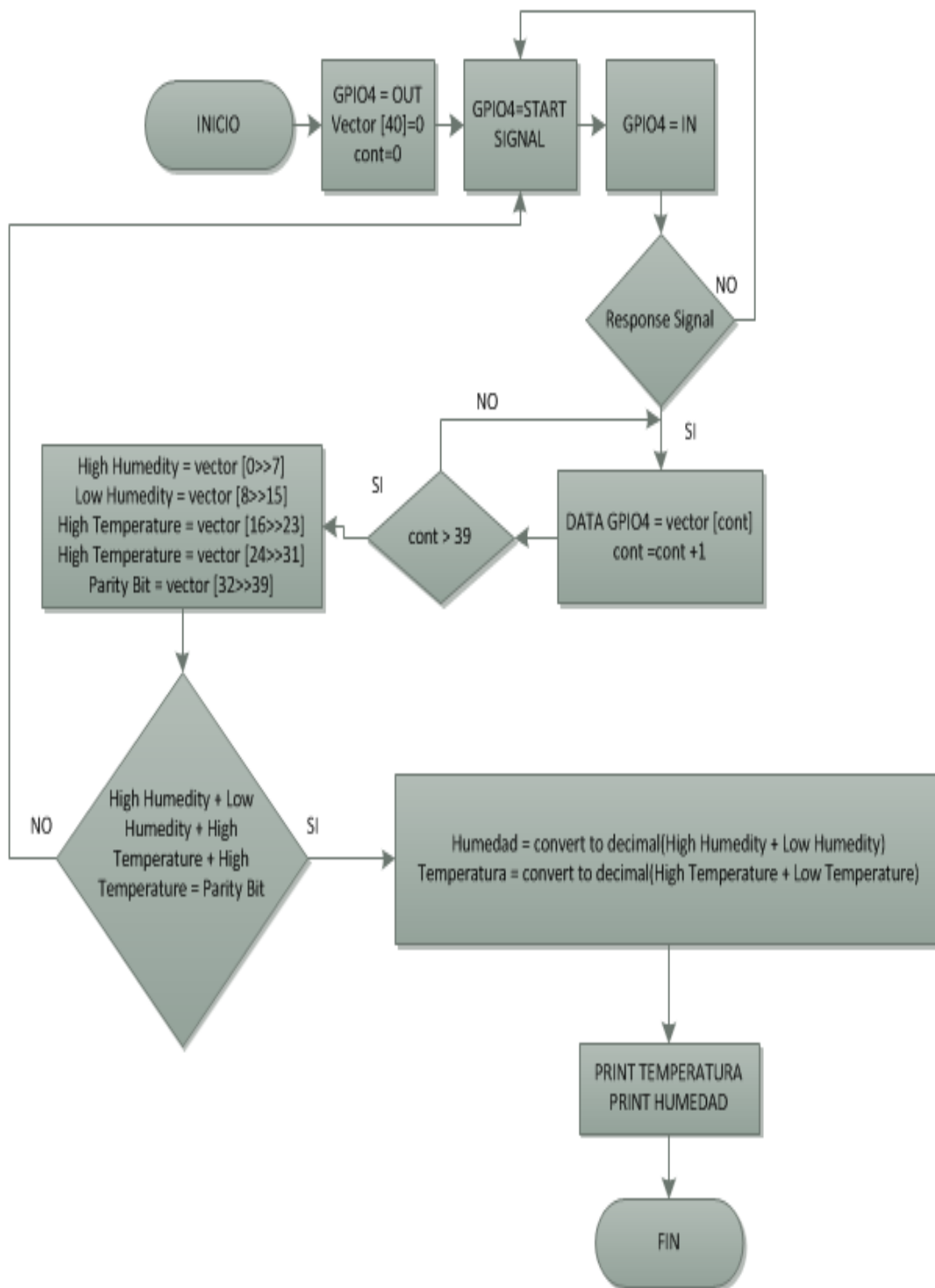


Figura 38: Diagrama de flujo del algoritmo para obtener la temperatura y humedad relativa. Fuente: [El autor].

3.6 LECTURA DE VALORES DEL SENSOR DE EFECTO HALL

En la figura 39 se muestra el diagrama de flujo correspondiente a la lectura del sensor.

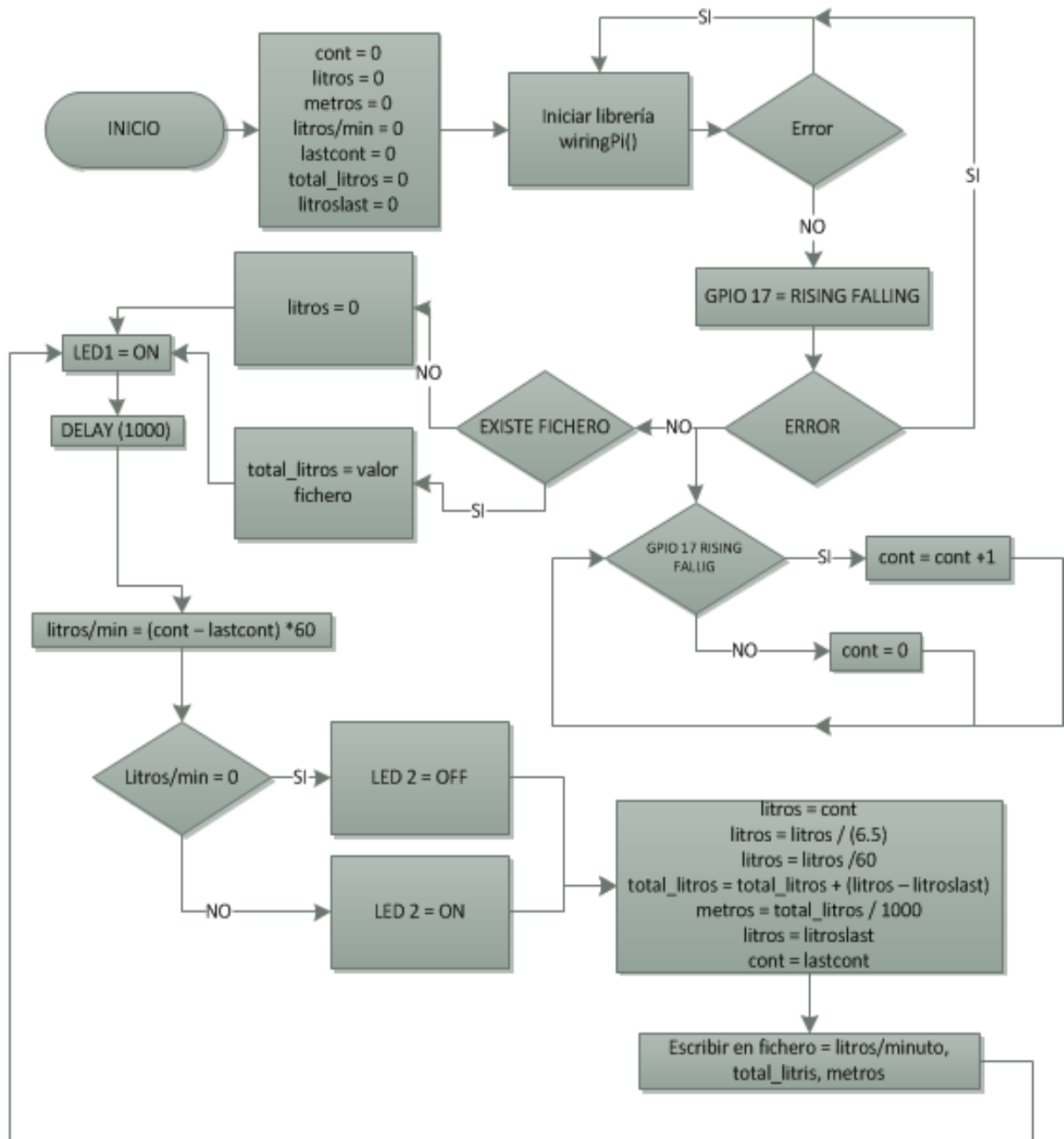


Figura 39: Diagrama de flujo del algoritmo para medir la cantidad flujo que circula por el sensor. Fuente: [El autor].

El sensor de flujo entrega un tren de pulsos que es proporcional al flujo de agua que circula por ese momento dada en la ecuación 7. Este valor dado en litros por minuto hay que convertirlo a litros y a su vez a metros cúbicos y almacenarlo en un fichero para tener de respaldo la información en el caso que exista un corte de energía y en el momento que se reanude la energía consultar ese archivo y sumar acumulativamente el consumo de agua potable. En la figura 39 se muestra el diagrama de flujo correspondiente. Este código funciona en un ciclo infinito, es decir en ningún momento el código se detiene. Este es el programa principal del sistema por lo que tiene que ser ejecutado cada vez que la Raspberry Pi se inicializa. Para ello se crea un script de inicialización en el directorio `/etc/init.d/`, con el comando:

sudo nano /etc/init.d/programa

Observar la figura 40 donde se indica las líneas de código que debe tener el script.

```
#!/bin/sh

case "$1" in
    start)
        echo "Arrancando programa"
        # Aquí hay que poner el programa que quiere
        sudo ./flowmeter
        ;;
    stop)
        echo "Deteniendo programa"
        ;;
    *)
        echo "Modo de uso: /etc/init.d/flowmeter"
        {start|stop}
        exit 1
        ;;
esac

exit 0
```

Figura 40: Fichero que iniciará el programa del contador del flujo. Fuente: [El autor].

El fichero tiene que hacerse ejecutable de la siguiente manera:

sudo chmod 755 /etc/init.d/programa

Y por último se debe activar el arranque automático:

sudo update-rc.d programa defaults

Cuando se reinicie la Raspberry el programa se ejecuta de forma automática. El fichero donde estará el código se nombró como *flowmeter.c*, el programa completo se puede encontrar en el anexo B.

El fichero donde se guardarán los registros de los valores de agua potable es de suma importancia en el sistema ya que los demás servicios consultarán la información almacenada aquí para poder enviarla hacia internet, algo parecido a una base de datos.

3.7 ENVIÓ DE DATOS A LA PLATAFORMA XIVELY

Xively Cloud Services ha sido diseñada para ayudar a las compañías a simplificar el desarrollo de una nueva generación de productos conectados que pueden ser rápidamente desplegados para el Internet Of Things, además permite interactuar fácilmente con otras ofertas conectadas y aplicarse sin esfuerzo a billones de dispositivos. Como resultado los desarrolladores independientes pueden enfocarse en sus innovaciones a la vez que reducen de forma significativa los recursos, costes y tiempo destinado a lanzar nuevas ofertas conectadas al mercado. Xively Cloud Services proporciona una plataforma abierta con mensajería en tiempo real, servicios de directorio y servicios de datos, que incluyen base de datos de series cronológicas y analíticas, siendo accesibles a través de una API estándar que soporta docenas de lenguajes de programación y dispositivos. La finalidad del uso de Xively en este proyecto es la de interactuar con los servicios de monitorización de estadísticas en la nube es decir poder acceder desde internet a datos que se generan de manera local.

Para poder comunicar la Raspberry Pi con la plataforma Xively es necesario crear una cuenta pública donde se almacenarán los datos. Todo esto se detalla en el anexo C. Además se debe instalar algunos módulos en la Raspberry Pi antes de empezar a escribir el código Python. Por terminal SSH se escribe lo siguiente:

```
sudo apt-get install python-dev  
sudo de apt-get install python-pip  
wget-O Geekman pythoneeml.tar.gzhttps://github.com/geekman/pythoneeml/tarball/master
```

Explicando lo anteriormente hecho el primer comando instala Python para desarrolladores. El segundo comando instala *pip* que es un sistema de gestión de paquetes que se utiliza para instalar y gestionar paquetes de software escritos en Python. El tercer comando se descarga desde un repositorio de Github la herramienta EEML que sirve básicamente para modelar sistemas basado en una estructura jerárquica de capas que es que el que usaremos para enviar los datos a Xively.

Se usa el comando ***tar zxvf Geekman-python-eeml.tar.gz*** para extraerlo; y luego ***cd Geekman-python-eeml ****, donde el signo (*) corresponde a la versión de EEML. Y finalmente ***sudo Python setup.py install*** para instalarlo (figura 41).

```
edison@edisonfabricio:~  
Archivo Editar Ver Buscar Terminal Ayuda  
copying build/lib.linux-armv6l-2.7/ee/ml/_init_.py -> build/bdist.linux-armv6l/egg/ee/ml  
byte-compiling build/bdist.linux-armv6l/egg/ee/ml/datastream.py to datastream.pyc  
byte-compiling build/bdist.linux-armv6l/egg/ee/ml/_init_.py to _init_.pyc  
creating build/bdist.linux-armv6l/egg/EGG-INFO  
copying Python_EEML.egg-info/PKG-INFO -> build/bdist.linux-armv6l/egg/EGG-INFO  
copying Python_EEML.egg-info/SOURCES.txt -> build/bdist.linux-armv6l/egg/EGG-INFO  
copying Python_EEML.egg-info/dependency_links.txt -> build/bdist.linux-armv6l/egg/EGG-INFO  
copying Python_EEML.egg-info/top_level.txt -> build/bdist.linux-armv6l/egg/EGG-INFO  
zip_safe flag not set; analyzing archive contents...  
creating dist  
creating 'dist/Python_EEML-0.1-py2.7.egg' and adding 'build/bdist.linux-armv6l/egg' to it  
removing 'build/bdist.linux-armv6l/egg' (and everything under it)  
Processing Python_EEML-0.1-py2.7.egg  
creating /usr/local/lib/python2.7/dist-packages/Python_EEML-0.1-py2.7.egg  
Extracting Python_EEML-0.1-py2.7.egg to /usr/local/lib/python2.7/dist-packages  
Adding Python-EEML 0.1 to easy-install.pth file  
  
Installed /usr/local/lib/python2.7/dist-packages/Python_EEML-0.1-py2.7.egg  
Processing dependencies for Python-EEML==0.1  
Finished processing dependencies for Python-EEML==0.1  
edison@raspberrypiedison:~/geekman-python-ee/ml-a7d2949$
```

Figura 41: Salida del terminal de comandos cuando la configuración para Xively esta lista. Fuente: [El autor].

Se crea un documento de texto con extensión “.py” donde se alojara el código de Python. La figura 42 muestra el diagrama de flujo del código para el envío de datos a Xively.

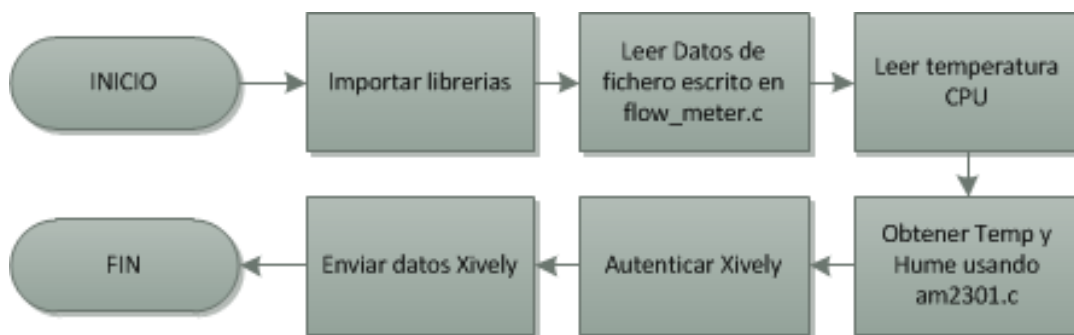


Figura 42: Diagrama de flujo del algoritmo para enviar datos hacia XIVELY. Fuente: [El autor].

Este código escrito en Python permite leer los datos del fichero donde están los valores del consumo de agua potable que ejecuta en el código *flowmeter.c*, además se obtiene la temperatura de la CPU.

Para obtener los datos de la temperatura y humedad ambiental, se ejecuta desde Python el código *am2301.c* que proporciona esos valores. Una vez que se tiene todos los datos se realiza la autenticación en Xively usando el *API KEY* y el *FEED ID*, y se procede a enviar los datos a la plataforma.

El link: <https://xively.com/feeds/1538624376> es el correspondiente a la página en Xively como lo muestra la figura 43.

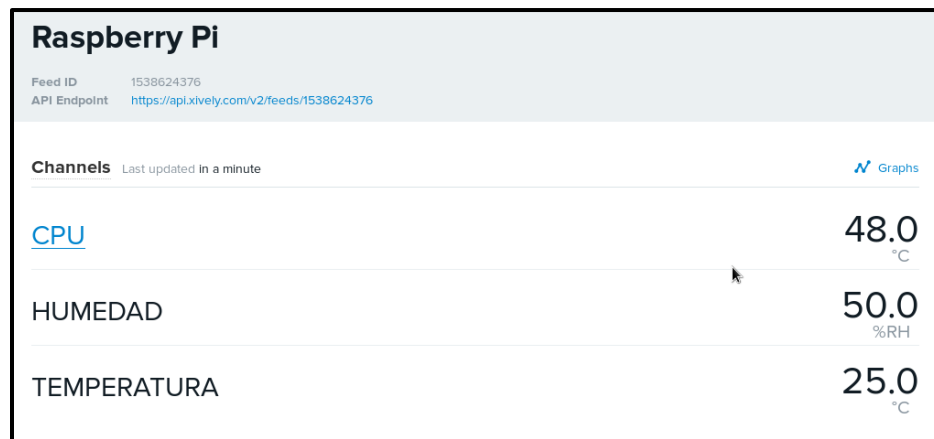


Figura 43: Datos almacenados en Xively. Fuente: [El Autor].

Para poder enviar los datos periódicamente a la plataforma Xively se empleará la herramienta cron. En el sistema operativo Linux, cron es un administrador regular de procesos en segundo plano que ejecuta procesos o guiones a intervalos regulares por ejemplo, cada minuto, día, semana o mes. El código escrito en Python que se encarga de enviar los datos a la plataforma Xively se ejecutará cada minuto las 24 horas al día, 7 días a la semana, el mismo está disponible para su consulta en el anexo D. En la terminal de comandos se escribe ***sudo nano crontab -e***, al final de este archivo ingresar la siguiente línea:

****/01 * * * * python "programa.py" #cada minuto se ejecutara esta acción***

3.8 CREACIÓN DE REGISTROS DIARIOS Y MENSUALES

Para poder tener un registro diario y mensual de manera local del consumo de agua potable en la Raspberry Pi se implementó un código en Python que todos los días a las 23:59 guarda los datos del consumo de agua potable de ese día en un fichero con la fecha correspondiente, por ejemplo si el día 18 de Mayo del 2014 hubo un consumo de 0,96 m³ el programa crea un fichero con el nombre del mes correspondiente: “Datos_Agua_Mayo2014” y dentro de este archivo escribe la línea “18—05—2014—0,96”, y así se repite para todos los días en el fichero del mes correspondiente.

Estos ficheros son texto plano y ocupan en memoria un espacio de 10 KB aproximadamente. La capacidad de almacenamiento del sistema es 8GB, lo cual queda reducido a 5GB por el espacio ocupado por el sistema operativo. Se tiene que **5GB = 5242880KB**, por lo tanto tenemos espacio para **524288** ficheros aproximadamente lo que garantiza que el sistema tiene capacidad de almacenar información para **43690** años, hipotéticamente.

De la misma manera que con el programa de Xively, se usa la herramienta cron para que el código se ejecute todos los días. En el archivo **cron** agregar lo siguiente:

59 23 * * * python datos_diarios.py #Creación de Datos para inventario diario

Este procedimiento se realiza con la finalidad de tener información adicional a la de la plataforma Xively y poder tener datos para los reportes mensuales hacia el usuario a través de correo electrónicos. El código *datos_diarios.py* lo podemos consultar en el anexo E, además este código se encarga de resetear las variables del consumo diario de agua potable.

3.9 AÑADIR SERVICIO DE SMTP: GMAIL

El sistema creará reportes mensuales el día 5 de cada mes sobre los valores de agua potable del mes anterior, es decir el día 5 de Mayo enviará un mail con los datos del consumo de agua potable del mes de Abril.

Los días 15 de cada mes también enviará reportes, por ejemplo el día 15 de Mayo enviará un reporte de los valores del 1 al 15 de Mayo del consumo de agua potable. Para ello el sistema usa el servicio de SMTP y se debe crear una cuenta exclusiva para la Raspberry Pi en GMAIL. En la terminal de comandos se ingresa lo siguiente:

sudo apt-get install ssmtp mailutils

Mailutils es un multiservicio de gestión del correo electrónico. Ofrece un completo conjunto de utilidades y demonios para el procesamiento de las direcciones de correo. Mailutils se orienta hacia tres tipos de audiencias: los usuarios regulares, administradores de sistemas y desarrolladores de software de manejo de correo electrónico. Seguidamente se modifica el archivo de configuración de mailutils.

sudo nano /etc/ssmtp/ssmtp.conf

Se debe cambiar este campo:

Mailhub=mail

Por

Mailhub=smtp.gmail.com:587

Donde 587 indica el puerto estándar de SMTP donde existe el cifrado SSL.

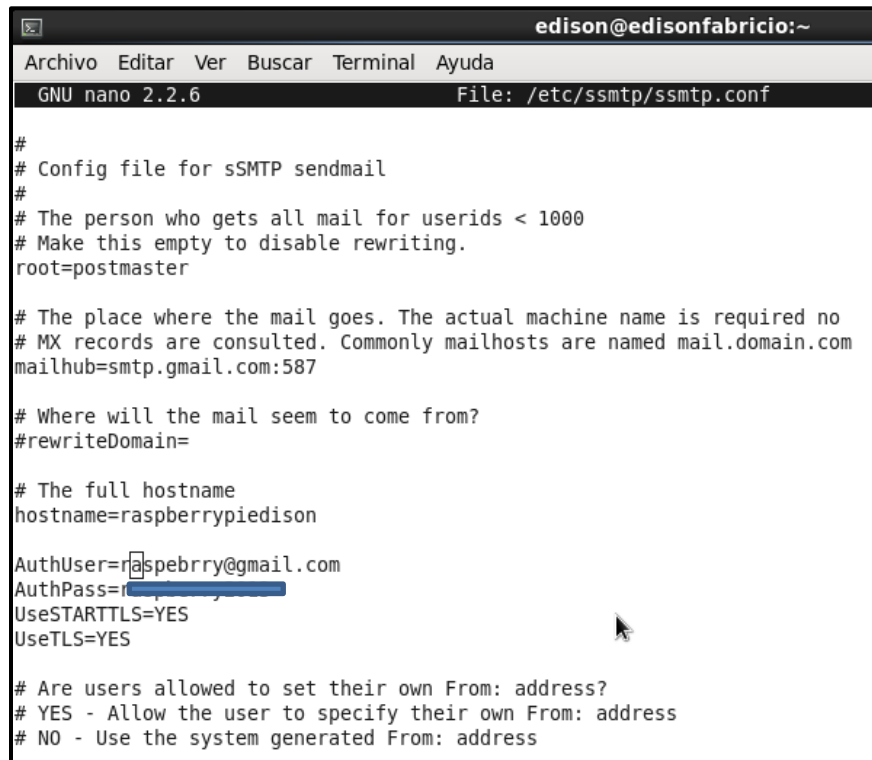
Además agregar estos campos:

AuthUser=correo@electronico.com

AuthPass=contraseña

UsesSTARTTLS=YES

UseTLS=YES



```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=postmaster

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=raspberrypiedison

AuthUser=raspebrry@gmail.com
AuthPass=
UseSTARTTLS=YES
UseTLS=YES

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
```

Figura 44: Fichero de configuraciones de SSMTP. Fuente: [El Autor].

Con esto la Raspberry Pi está lista para enviar correos electrónicos. Usando Python se envía correos a más de un destinatarios, y de ahí usar el demonio cron para que la tarea se realice automáticamente. El código escrito en Python se encuentra en el anexo F, en el fichero ***crontab***, añadir lo siguiente:

45 9 05 * * python reportes_mensuales.py

45 9 15 * reportes_quincenales.py

3.10 CONEXIÓN CON GOOGLE DRIVE

Ya que se tiene el servicio de notificaciones por correo electrónico usando GMAIL es posible también usar una función interesante: Google Docs, crear una tabla de registros diarios del consumo de agua potable en un documento público que puede ser visto, no editado, por el usuario desde internet. Usando la herramienta Drive (de la cuenta propia del Raspberry Pi) se crea un documento de Hoja de Cálculo, se le asigna un nombre al documento, este mismo nombre se usará para enviar los datos desde la Raspberry Pi. En la terminal de comandos se instalan las herramientas necesarias para lograr la comunicación:

```
wget http://pypi.python.org/packages/source/g/gspread/gspread-0.1.0.tar.gz  
tar -zxvf gspread-0.1.0.tar.gz  
cd gspread-0.1.0  
sudo python setup.py install
```

Usando la herramienta **cron** se ejecuta un código para que envíe a las 23:57 los datos de agua potable y lo almacene en Google Drive. El código está para su consulta en el anexo E.

3.11 ENVÍO DE ALERTAS USANDO WHATSAPP

Usando la herramienta *Yowsup Cli* es posible enviar mensajes de WhatsApp a uno o varios números de operadoras telefónicas de todo el mundo, en este caso se implementó el servicio para temperaturas mayores a 60°C y cuando el flujo de actual sobrepase los 16 litros/min.

El valor de 16 litros/min se dedujo de manera experimental. Cuando las llaves de agua potable están abiertas por un periodo largo de tiempo se da este valor. Así mismo el valor de 60°C se determinó de manera experimental cuando la Raspberry Pi estaba funcionando a plena capacidad y la temperatura ambiente superaba los 30°C .

Usando la herramienta **cron** se ejecuta cada minuto un programa (ver anexo G) donde consulta el estado de la CPU y el flujo de corriente.

3.12 TWITTERBOT

Twitterbot es un programa usado para producir los mensajes automatizados a través del servicio de microblogging Twitter. Usando la API de Tweepy que combinado con Python me permite acceder de manera más fácil a Twitter. Se envía tuits cada hora con los datos de temperatura CPU, temperatura/humedad y el flujo de agua sin interrupciones las 24 horas al día. El código está disponible en el anexo H. En la figura 45 se muestra los mensajes enviados por el sistema.

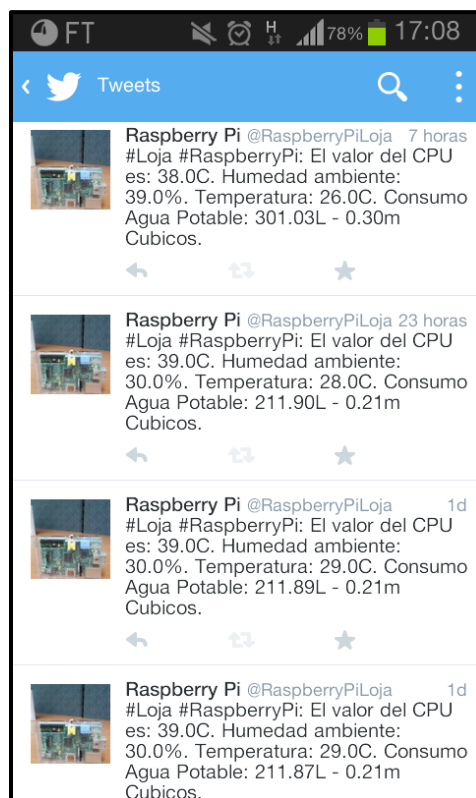


Figura 45: Mensajes generados automáticamente en Twitter. Fuente: [El Autor].

3.13 PANTALLA LCD

La pantalla LCD permitirá al usuario visualizar los valores de medición de agua potable de manera simple y sin necesidad de tener acceso a una computadora con conexión a Internet. Además se añadió los parámetros de temperatura CPU, temperatura Ambiente y humedad ambiente. Para poder usar la pantalla es necesario modificar el kernel de la Raspberry Pi para que los pines de la comunicación SPI puedan enviar información a la Raspberry Pi. Para ello en nuestra terminal ingresamos los siguientes comandos:

```
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-bin-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-dev-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi-doc-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/libraspberrypi0-adafruit.deb
wget http://adafruit-download.s3.amazonaws.com/raspberrypi-bootloader-adafruit-112613.deb
```

Los comandos anteriormente ingresados permiten descargar los módulos de configuración de la pantalla, después de esto es necesario construir e instalar estos nuevos módulos. Ingresamos el siguiente comando:

```
sudo dpkg -i -B *.deb
```

Se debe reiniciar el sistema con **sudo shutdown -h now**. Para encender la pantalla ingresamos los siguientes comandos:

```
sudo modprobe spi-bcm2708
sudo modprobe fbtt_device name=adafruitts rotate=90
export FRAMEBUFFER=/dev/fb1
startx
```

Estos comandos se pueden ejecutar automáticamente cada vez que se enciende el sistema si se lo agrega a un script de inicialización.

Para poder crear programas gráficos empleamos el entorno desarrollo Qt Creator (ver anexo I) que se basa en lenguaje C/C++. En la figura 46 y 47 se muestra la pantalla en funcionamiento.

HORA: 14 : 11 : 23

FECHA: Saturday 21, June 2014

TEMPERATURA CPU : 42.0 C

TEMPERATURA AMB : 31.0 C

HUMEDAD AMB : 55.0 %

CONSUMO DIARIO LITROS: 150.77

CONSUMO DIARIO METROS: 0.15

FLUJO DE AGUA: 0.00

Figura 46: Interfaz Gráfica de Usuario creado en QT Creator. Fuente: [El Autor].



Figura 47: Pantalla LCD en funcionamiento. Fuente: [El Autor].

CAPITULO IV

4. PRUEBAS Y RESULTADOS.

El prototipo de medición de agua potable funciona las 24 horas al día, siete días a la semana. Su funcionamiento empezó desde el día 01 de Mayo del 2014. Durante este tiempo hasta el día de la redacción de la tesis el sistema nunca ha tenido un estado de fuera de línea por motivos de mal funcionamiento, excepto cuando la conexión a Internet no existe o no hay una fuente de energía que alimente al sistema. Cuando no existe una conexión a Internet el sistema sigue funcionando normalmente. Una vez que el servicio de conexión a Internet esta restablecido los datos de medición no enviados anteriormente se envían de nuevo. Si ocurre algún tipo de corte de energía los valores no se pierden, cuando se restituye el servicio de energía los valores de litros consumidos anteriormente serán recuperados y se puede seguir con las mediciones. Hay que tener en cuenta que los valores de litros consumidos durante el corte de energía no serán contados por el sistema de medición. Para comprobar el correcto funcionamiento del sensor de flujo de Agua se hizo uso de una tarjeta Arduino UNO. La elección de la tarjeta Arduino se basa en que el fabricante del sensor de flujo pone a disposición de los usuarios el código de lectura de datos para usarlo con la plataforma Arduino (ver figura 48). De esta manera se asegura primeramente que el sensor funciona correctamente antes de unirlo al sistema embebido Raspberry Pi.

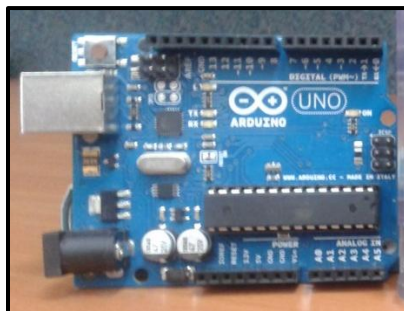


Figura 48: Tarjeta Arduino UNO R3. Fuente: [El autor].

Conectando una pantalla LCD a la tarjeta Arduino y el sensor de flujo unido a un sistema de medición provisional, se iniciaron las pruebas llenando un contenedor de 20 litros y comparando que las mediciones lleguen a los 20 litros, que es la capacidad del contenedor comercial, tal como se puede observar en la figura 49.



Figura 49: Comprobación del sensor de flujo junto con la tarjeta Arduino UNO. Fuente: [El autor].

Una vez comprobado el correcto funcionamiento del sensor se procedió a usar el sistema de medición provisional pero remplazando la tarjeta ARDUINO por el sistema embebido Raspberry Pi. En la figura 50 se muestra las mediciones realizadas.

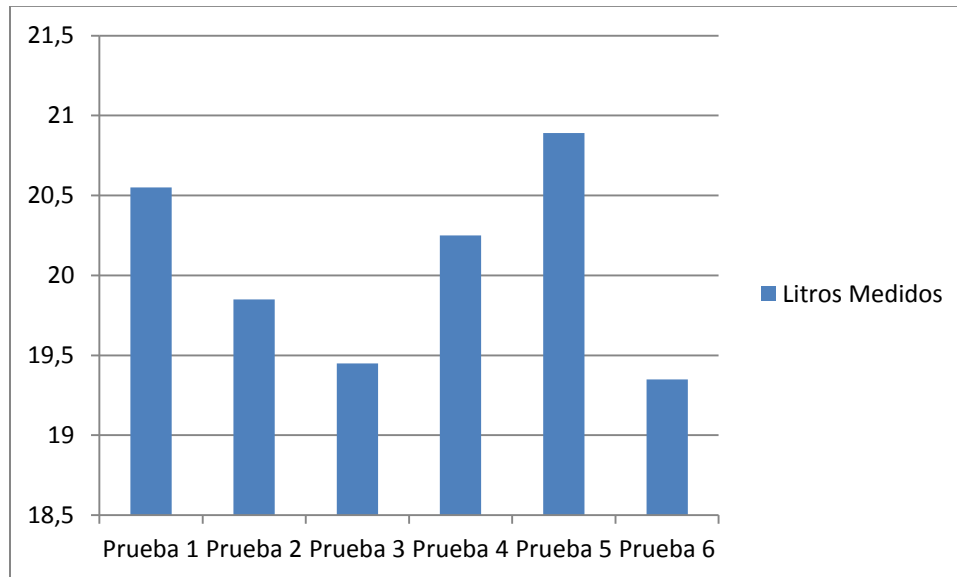


Figura 50: Valores obtenidos por el prototipo en seis mediciones diferentes. Fuente: [El autor].

Como podemos observar en la figura 50 los valores medidos se aproximan significativamente a los 20 litros. Para las mediciones reales se precedió a comparar los valores dados por el sistema de medición de agua potable implementada y los valores dados por el medidor convencional.

Durante 7 días a las 06:00 se observó el valor en litros del medidor de agua potable y a las 19:00 se procedió a realizar nuevamente una lectura. En la tabla 7 se muestra los valores medidos en esos días y en la figura 51 y 52 se muestra cuadros estadísticos de estas medidas.

Tabla 7: Tabla de valores de las pruebas realizadas con el medidor convencional.

	MEDIDOR CONVENCIONAL (LITROS)	MEDIDOR IMPLEMENTADO (LITROS)	ERROR ABSOLUTO	ERROR RELATIVO
Día 1	843	840,54	2,46	0,29%
Día 2	710	713,05	3,05	0,43%
Día 3	514	512,05	1,95	0,38%
Día 4	625	629,75	4,75	0,76%
Día 5	678	680,58	2,58	0,38%
Día 6	635	638,01	3,01	0,47%
Día 7	589	593,75	4,75	0,81%

Fuente: [El autor].

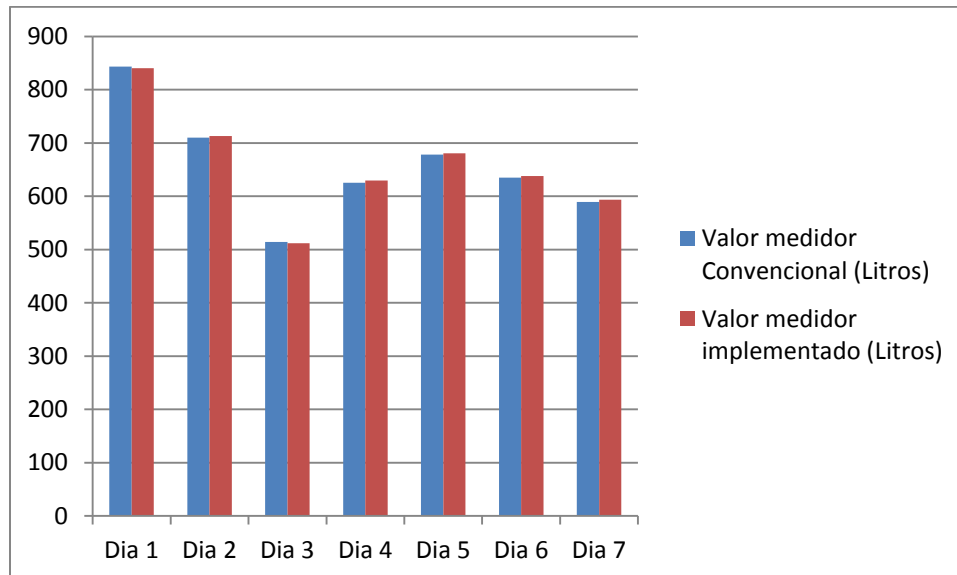


Figura 51: Comparación de las medidas de consumo diario entre el medidor convencional y el prototipo implementado. Fuente: [El autor].

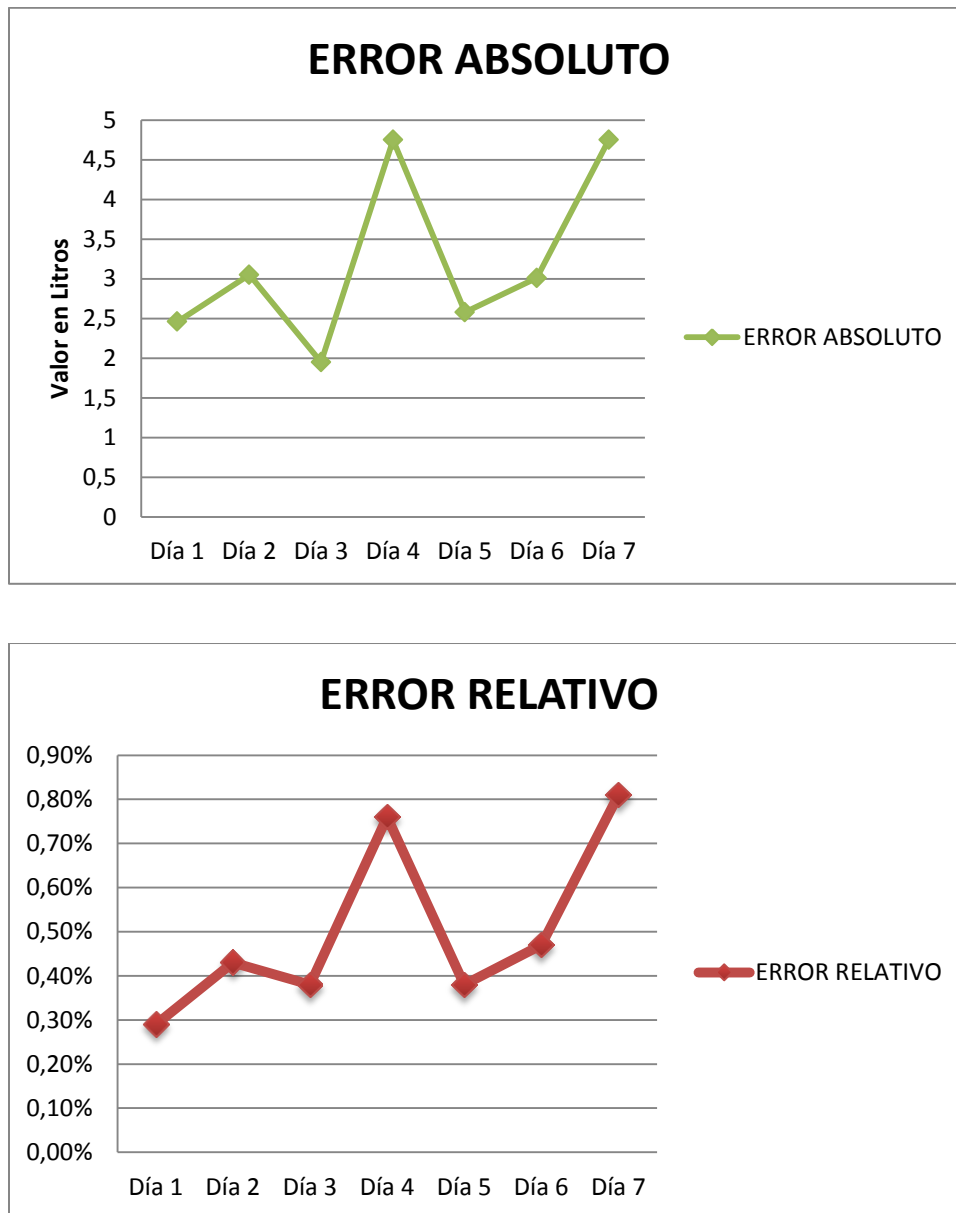


Figura 52: Error porcentual del prototipo calculado en cada medición. Fuente: [El autor].

Se realizó dos cálculos de errores: El error absoluto y el error relativo. El primero es la diferencia entre el valor real (medidor convencional) y el valor obtenido por el sensor de flujo. El segundo es el cociente entre el error absoluto y el valor real de la magnitud.

Ambos indican que la calidad de la medida y la imprecisión que se comete en el momento de la medición. Ambos son complementarios y se deben analizar conjuntamente.

Como se puede observar en la figura 52 el error relativo está dentro del margen de error del sensor de flujo que es 3% por lo que se deduce que el sistema está operando correctamente.

Una vez que los datos proporcionados por el sistema de medición son válidos, se procede a enviar estos datos a los diferentes servicios que se basan en Internet.

4.1 ENVIÓ DE DATOS A XIVELY

Cada minuto las 24 horas al día los 7 días a la semana el sistema envía el valor de agua potable a la plataforma Xively.

Xively registra estos datos en un historial que puede ser consultado en cualquier momento a través de internet. Por ejemplo en la figura 53 se puede observar el consumo de un día en específico y en la figura 54 se muestra un registros de 7 días de consumo de agua potable.

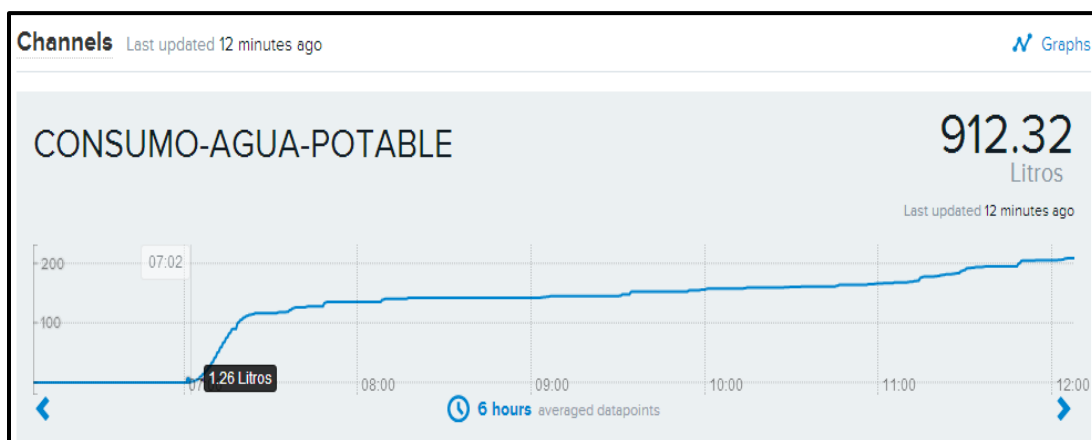


Figura 53: Historial diario del consumo de agua potable en Xively. Fuente: [El autor].

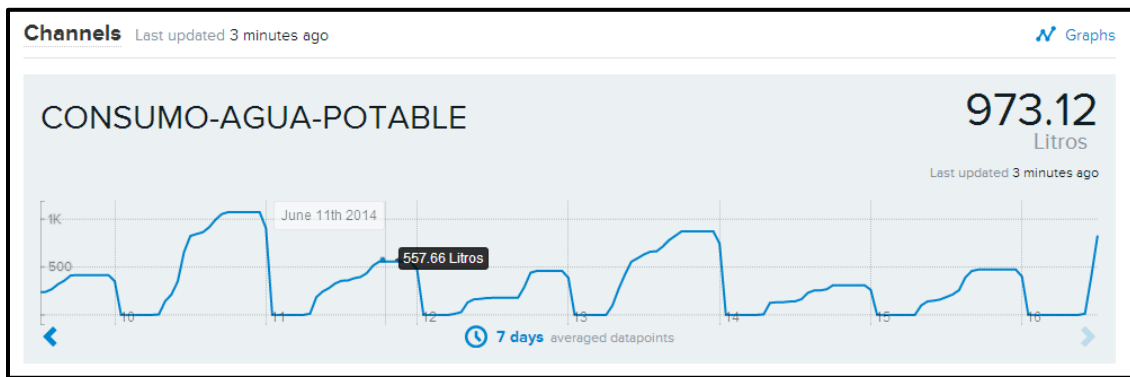


Figura 54: Historial semanal del consumo de agua potable en Xively. Fuente: [El autor].

4.2 ENVIÓ DE CORREOS ELECTRÓNICOS

El envío de correos electrónicos se realiza automáticamente los días 5 y 15 de cada mes a varios usuarios que están en una lista “beta”. En la figura 55 se muestran los correos electrónicos que se envían a los usuarios con la información del consumo de agua potable.

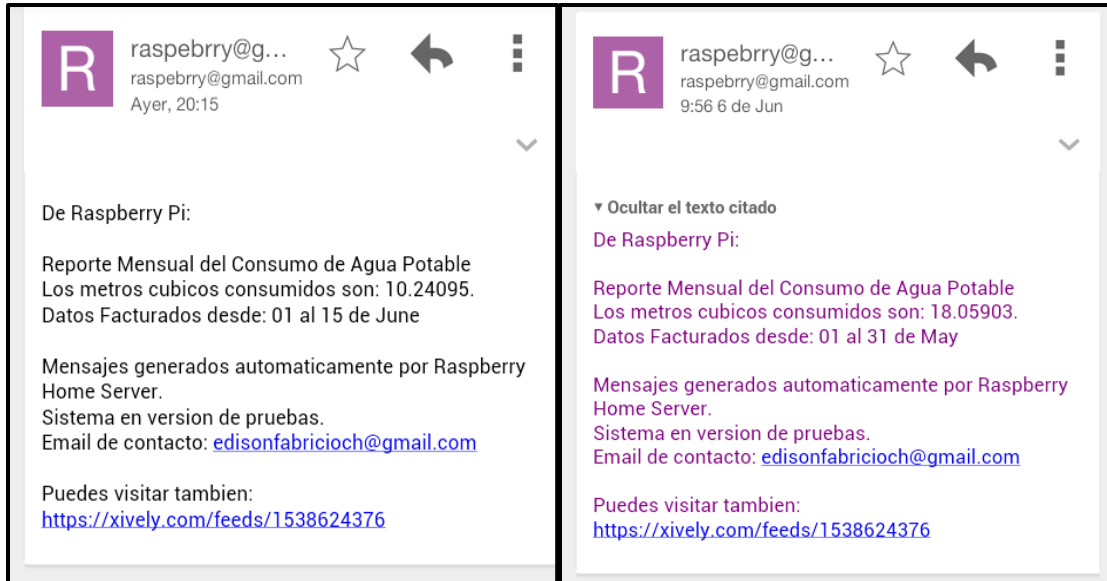
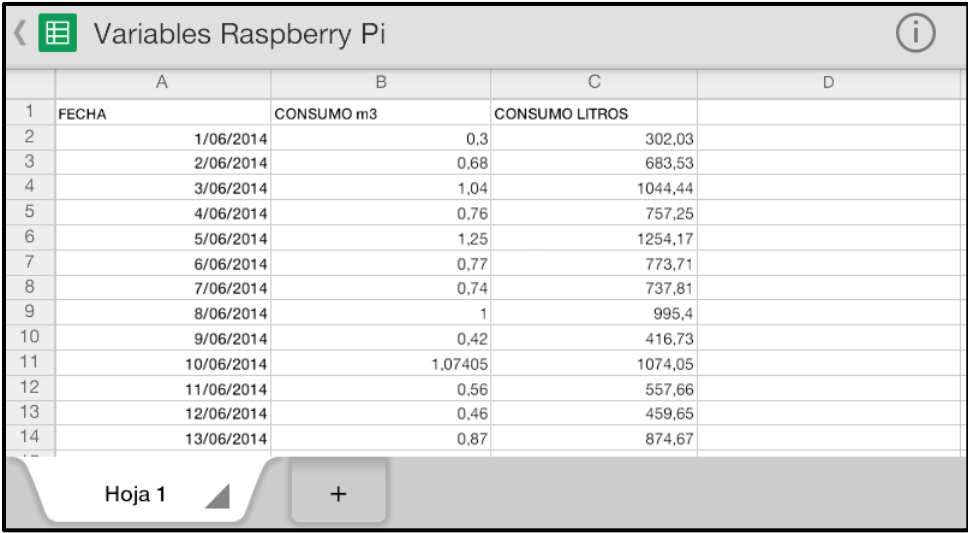


Figura 55: Modelo de correos electrónicos recibidos por los usuarios. Fuente: [El autor].

Estos valores son fácilmente comparados con los registros diarios y mensuales que almacena en su memoria interna la Raspberry Pi.

4.3 REGISTRO DE VALORES DIARIOS EN GOOGLE DRIVE

El sistema registra los valores diarios de consumo de agua potable en una tabla anexada a Google Drive tal como se muestra en la figura 56 que puede ser consultada por el usuario además de tener la posibilidad de proyectar gráficas estadísticas mensuales del consumo de agua potable (ver figura 57).



	A	B	C	D
1	FECHA	CONSUMO m3	CONSUMO LITROS	
2	1/06/2014	0,3	302,03	
3	2/06/2014	0,68	683,53	
4	3/06/2014	1,04	1044,44	
5	4/06/2014	0,76	757,25	
6	5/06/2014	1,25	1254,17	
7	6/06/2014	0,77	773,71	
8	7/06/2014	0,74	737,81	
9	8/06/2014	1	995,4	
10	9/06/2014	0,42	416,73	
11	10/06/2014	1,07405	1074,05	
12	11/06/2014	0,56	557,66	
13	12/06/2014	0,46	459,65	
14	13/06/2014	0,87	874,67	

Figura 56: Registro del consumo diario de agua potable en un documento en Google Docs. Fuente: [El autor].

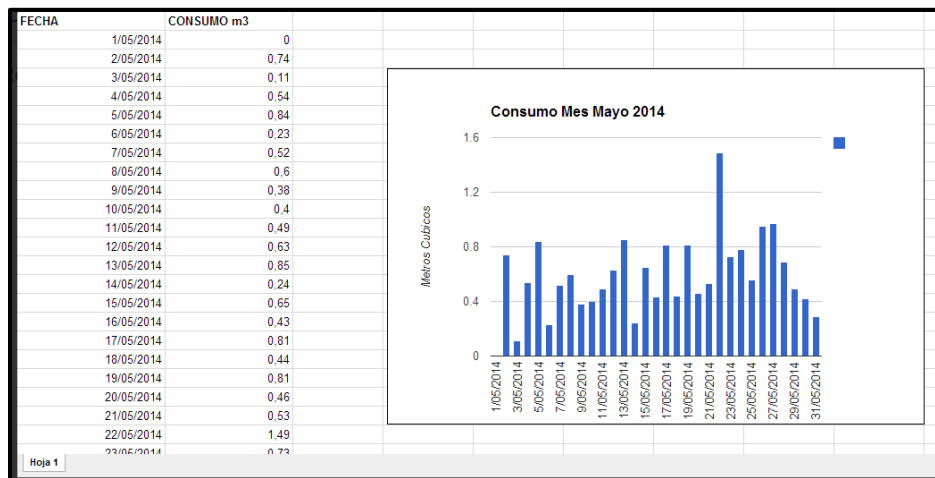


Figura 57: Estadísticas del consumo mensual realizadas en Google Docs. Fuente: [El autor].

4.4 ENVIÓ DE NOTIFICACIONES A WHATSAPP

Las notificaciones WhatsApp se muestran a continuación y ocurren solo bajo reglas específicas que han sido creadas con anterioridad en la Raspberry Pi.



Figura 58: Notificaciones por medio de mensajes usando la plataforma Whatsapp.

Fuente: [El autor].

CAPITULO V

5. ANÁLISIS DE COSTOS

El dispositivo de medición de agua potable implementado fue sometido a un análisis de costos donde se incluyen los precios de todos los materiales e ingeniería que fueron necesarios para su elaboración, además de una comparativa técnica y económica con otros productos similares existentes en el mercado.

5.1 ANÁLISIS DE COSTOS DEL DISPOSITIVO IMPLEMENTADO

Tabla 8: Costos del prototipo implementado.

DESCRIPCIÓN	CANTIDAD	VALOR (USD)
Raspberry Pi Clase B	1	80.00
Tarjeta SD Clase 10	1	25.00
USB-WiFi	1	15.00
Sensor Temperatura/Humedad	1	16.00
Sensor de Flujo	1	16.00
Disipadores	Varios	8.00
Ventilador	1	4.00
Pantalla LCD	1	80.00
Adaptador AC-DC	1	12.00
Circuito Electrónico	1	20.00
Caja de Fibra de Vidrio	1	28.00
Carcasa Raspberry pi	1	20.00
Cables y Conectores	Varios	10.00
	TOTAL	334.00

Fuente: [El autor].

En la tabla 8 se muestra los costos de cada uno de los componentes. El valor correspondiente a los estudios de ingeniería de hardware y de software requiere un estudio más concienzudo que por el momento no serán analizados.

Si el objetivo es la producción en masa para la venta comercial, los costos de ingeniería en software y hardware no serían incluidos. Además la producción y adquisición de componentes en volúmenes grandes hará que el valor final del sistema baje considerablemente. Realizando una estimación del costo, el valor prototipo para su venta sería de **325\$** aproximadamente. En la tabla 9 se muestra una comparación económica del sistema de medición de agua potable implementado con otros medidores comerciales actuales en el mercado.

Tabla 9: Comparación de precios de los medidores comerciales con el prototipo implementado.

MARCA	PRECIO
Prototipo Implementado	325\$
Medidor Electrónico Assured Automation	325\$
Medidor Electrónico Bell Flow Systems	500\$
Medidor Mecánico Convencional	80\$

Fuente: [El autor].

5.2 COMPARACIÓN TÉCNICA DEL SISTEMA DE MEDICIÓN IMPLEMENTADO CON MEDIDORES COMERCIALES

En la tabla 10 se realiza una comparación de las prestaciones de los medidores de agua electrónicos residenciales que hay en el mercado actual con el dispositivo implementado.

Tabla 10: Comparación técnica de los medidores comerciales con el prototipo implementado.

PARÁMETRO	MEDIDOR 1	MEDIDOR 2	MEDIDOR 3
			
Marca	Dispositivo Implementado	Assured Automation	Bell Flow Systems
Modelo	Dispositivo Implementado	LM50P	Agrimag
Voltaje de Operación	110V	2 Baterías AAA	6 Baterías AA
Sistema de Back Up	No	Si	Si
Rango de Medición	1-30 Litros/minuto	1-30.2 Litros/Minuto	1-40 Litros/minuto
Porcentaje de Error de la Medición		+0.5%	+1%
Tamaño Tubería	½ Pulgada	½ Pulgada	½ Pulgada
Alarmas al Usuario	Si	No	No
Envío de datos a Internet	Si	No	No
Envío de Reportes Mensuales	Si	No	No
Interfaz de Comunicación	Wi-Fi o Ethernet	No	No

Fuente: [El autor].

En el mercado actual no existe un equipo de medición de agua potable para uso doméstico que cubra todas las características del dispositivo implementado. En la mayoría de los casos existen soluciones de monitoreo integral que son proporcionadas por grandes empresas que venden toda la solución completa o bajo requerimientos específicos del cliente, en este caso municipios, grandes empresas de manufactura entre otras. Algunas de las empresas que implementan este tipo de soluciones son:

ABB GROUP

Esta multinacional suiza de monitoreo y control industrial, tiene entre sus servicios la implementación de un sistema de monitoreo integral de flujo de agua potable que se basa en la colocación de medidores de agua potable electrónicos (ver figura 59) en las tuberías principales o en las casas de los usuarios y a través de un sistema de comunicación inalámbrica, con infraestructura propia, envían los datos a un grupo de servidores centrales donde los operadores tratan la información (ver figura 60).



Figura 59: Medidor Electrónico de ABB GROUP. Fuente: [29].

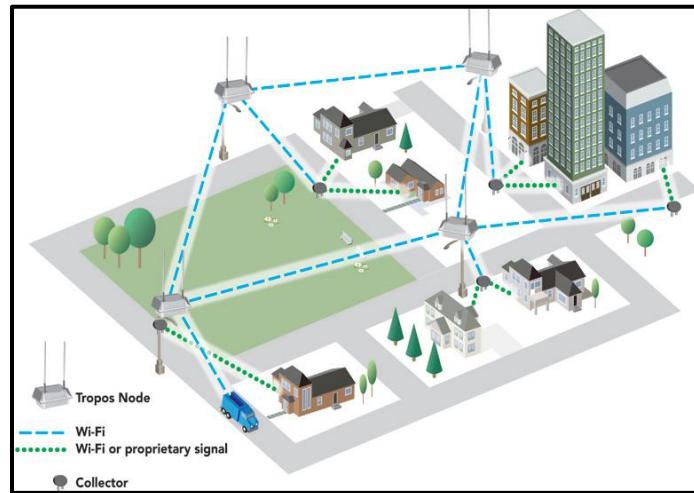


Figura 60: Sistema de comunicación entre sensores y servidores dentro del ecosistema ABB GROUP. Fuente: [30].

TADAKU

Tadaku es una empresa de Israel que ofrece un servicio de software y Hardware utilizado por las empresas de suministro de agua de todo el mundo para supervisar y gestionar sus redes de distribución de agua en tiempo real. TaKaDu envía alertas sobre fugas, roturas y anomalías, así como también proporciona información de la ubicación de éstas, permitiendo que los equipos de detección y reparación, localicen y reparen rápidamente los problemas.

IUSA

Este consorcio mexicano implementa el Sistema de Medición Inteligente de Autogestión IUSA, el cual fue creado con el objetivo de hacer más eficientes los procesos de atención y disminuir los costos financieros y operativos de las compañías prestadoras de servicios de agua, al mismo tiempo que facilita al usuario el control de su consumo y le proporciona formas de pago más accesibles.

Este sistema está patentado internacionalmente y opera vía un medidor electrónico multifunción, una tarjeta inteligente bidireccional y un punto de venta autorizado con un software que administra el proceso de cobranza (ver figura 61).

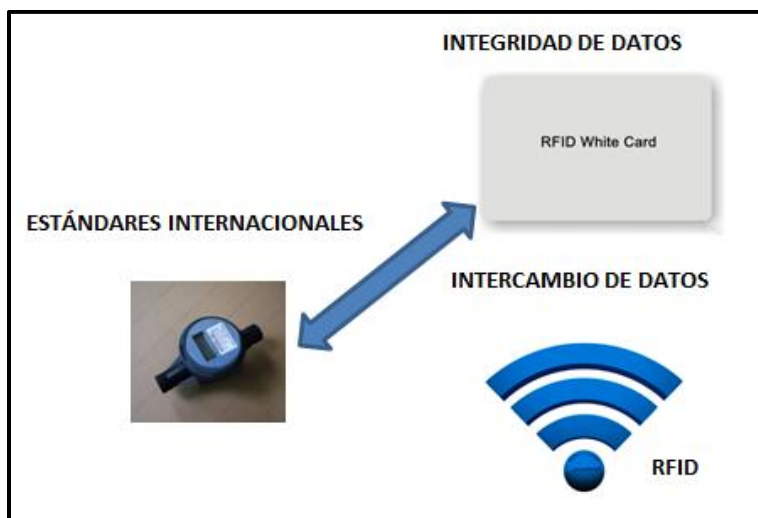


Figura 61: Sistema RFID de monitoreo de agua potable de IUSA. Fuente: [31].

Cuenta con una comunicación bidireccional usuario-compañía de agua, que permite “monitorear” al usuario y obtener estadísticas de frecuencia de uso, consumos y otras informaciones técnicas útiles para la compañía de agua. Además, la compañía prestadora de servicios de agua puede programar funciones en el medidor vía la tarjeta inteligente. Su sistema de seguridad asegura y protege de manera confiable toda la información de los usuarios y de la misma compañía de agua.

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- El sistema de medición electrónico implementado permite monitorear de manera local y remota el consumo de agua potable usando plataformas WEB orientadas a la Internet Of Things y a través de reportes de correo electrónico.
- Analizadas las diferentes tecnologías en sensores de flujo se determinó que el sensor por efecto de Hall cumple perfectamente con los requerimientos para un sistema de medición electrónica ya que combina la mecánica con la teoría electromagnética y la señal de voltaje de salida puede ser acoplada a numerosos sistemas electrónicos.
- El sistema embebido Raspberry Pi puede ser usado para interactuar con cualquier sensor electrónico aplicando la ingeniería de hardware y software correspondiente. Además al usar un kernel de Linux como sistema operativo tenemos todos los beneficios del software libre y no requerir ninguna licencia comercial.
- El monitoreo del consumo de agua potable residencial permite al usuario contar con información segura, asequible y de mayor variedad que en los sistemas tradicionales de medición. Al tener registros de valores diarios de consumo se puede construir automáticamente graficas estadísticas para tener bases en la toma de decisiones futuras.
- El Internet of Things como nueva tendencia tecnológica es la gran oportunidad para monitorear y controlar remotamente dispositivos de toda clase, lo que abre un gran campo de nuevas oportunidades de negocios.

- La investigación en nuevas tecnologías de sistemas embebidos es fundamental para el desarrollo de nuevos métodos de medición de parámetros no solo a nivel residencial si no a futuro aplicado al campo industrial.
- Los antiguos medidores mecánicos residenciales pueden ser fácilmente reemplazados por el sistema de medición electrónico ya que usa la misma infraestructura existente.
- El sistema al no contar con una infraestructura propia de intercomunicación ni con servidores propios no permite al administrador un control total y solución de errores si ocurre algún fallo de comunicación o el servidor no está disponible.

6.2 RECOMENDACIONES

- Implementar baterías que trabajen conjuntamente con la alimentación eléctrica del sistema de medición electrónico, para cuando ocurra un corte de energía las baterías sean las encargadas de alimentar al sistema.
- Montar una red de intercomunicación exclusiva entre el sistema de medición y un servidor o grupo de servidores privados. Así se puede tener total control sobre el sistema y efectuar diagnósticos más rápidos y confiables cuando el sistema presente problemas.
- Las actualizaciones del software y firmware de la Raspberry Pi se deben realizar semestralmente por un administrador ya que estas implementan mejoras en la ejecución de programas, kernel y añaden seguridades a vulnerabilidades en la red.
- Como todo sistema informático que está conectado a la Internet. la Raspberry Pi es susceptible a intentos de infiltración para apoderarse o alterar la información que esta almacena, por lo que un administrador debe cambiar las contraseñas de acceso regularmente, configurar correctamente el servidor Firewall, ya sea de la Raspberry Pi (que al ser Linux tiene uno incluido) o el de la red local.

- Socializar el proyecto con las diferentes municipalidades locales haciendo énfasis en las ventajas del monitoreo y control remoto y como esto ayudaría el ahorro de recursos, tiempo y dinero.
- Diseñar una aplicación móvil multiplataforma que implemente información más detallada y nuevas características del sistema de medición electrónico como: precio del consumo, alarmas por eventos definidos por el usuario, periodo de reportes, etc.
- Someter al sistema de medición electrónico a pruebas y requerimientos más rigurosos con el fin de obtener resultados más reales e implementar posibles mejoras al sistema.
- Implementar la característica de telecontrol que permita al administrador controlar el flujo de agua usando una electroválvula.
- Analizar la ingeniería de software necesaria para implementar un algoritmo que sea capaz de aprender del sistema las particulares del consumo de agua potable y pueda detectar anomalías y enviar alarmas hacia el usuario y el administrador.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Luigi de Russis. (2011). Domotics: An Open Approach. Recuperado de: <http://www.slideshare.net/luigidr/domotics-an-open-approach>
- [2] Kevin Ashton. (2009, 22 de Junio). That 'Internet Of Things' Thing: In the real world, things matter more than ideas. *RFID JOURNAL*. Recuperado de: <http://www.rfidjournal.com/articles/view?4986>
- [3] Internet Live Stats. (2014). Internet Users. Recuperado de: <http://www.internetlivestats.com/internet-users/>
- [4] Fundación Bankinter. (2012). Un internet de las cosas, por Pual Horn. Recuperado de: <http://www.fundacionbankinter.org/es/videos/an-internet-of-everything>
- [5] Fundación de la Innovación Bankinter. (2011). El Internet de las Cosas: En un mundo conectado de objetos. Recuperado de: http://www.fundacionbankinter.org/system/documents/8168/original/XV_FTF_El_internet_de_las_cosas.pdf
- [6] Fundación de la Innovación Bankinter. (2011). El Internet de las Cosas: En un mundo conectado de objetos. Recuperado de: http://www.fundacionbankinter.org/system/documents/8168/original/XV_FTF_El_internet_de_las_cosas.pdf
- [7] Fundación de la Innovación Bankinter. (2011). El Internet de las Cosas: En un mundo conectado de objetos. Recuperado de: http://www.fundacionbankinter.org/system/documents/8168/original/XV_FTF_El_internet_de_las_cosas.pdf
- [8] George Granger Brown. (1965). Operaciones básicas de la Ingeniería Química. Barcelona. Recuperado de: <http://es.scribd.com/doc/143343855/Operaciones-Basicas-de-La-Ingenieria-Quimica-Brown>
- [9] George Granger Brown. (1965). Operaciones básicas de la Ingeniería Química. Barcelona. Recuperado de: <http://es.scribd.com/doc/143343855/Operaciones-Basicas-de-La-Ingenieria-Quimica-Brown>

[10] Marco A. Toledo Gutiérrez. (2011) 5 Tipos de medidores de agua domiciliarios. Recuperado de:
<http://www.aneas.com.mx/contenido/xxvpresent/Cap%205%20Tipos%20de%20Med%20ANEAS.pdf>

[11] Marco A. Toledo Gutiérrez. (2011) 5 Tipos de medidores de agua domiciliarios. Recuperado de:
<http://www.aneas.com.mx/contenido/xxvpresent/Cap%205%20Tipos%20de%20Med%20ANEAS.pdf>

[12] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

[13] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

[14] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

[15] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

[16] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

[17] Lara Naranjo Rolando Fabricio, Molina Salazar Carlos Andrés. (2011). Diseño e implementación de un medidor digital para el consumo de agua potable en lugares residenciales y envío de datos vía SMS. Escuela Politécnica del Ejército. Latacunga , Ecuador.

- [18] Arthur William Poyser. (2014). Magnetism and Electricity: A Manual for Students in Advanced Classes. Recuperado de:
http://books.google.com.ec/books?id=JzBAAAAAYAAJ&pg=PA285&redir_esc=y
- [19] Marco A. Toledo Gutiérrez. (2011) 5 Tipos de medidores de agua domiciliarios. Recuperado de:
<http://www.aneas.com.mx/contenido/xxvpresent/Cap%205%20Tipos%20de%20Med%20ANEAS.pdf>
- [20] Marco A. Toledo Gutiérrez. (2011) 5 Tipos de medidores de agua domiciliarios. Recuperado de:
<http://www.aneas.com.mx/contenido/xxvpresent/Cap%205%20Tipos%20de%20Med%20ANEAS.pdf>
- [21] SeedStudio. (2014). G1/2" Water Flow Sensor. Recuperado de:
http://www.seeedstudio.com/depot/G12-Water-Flow-Sensor-p 635.html?cPath=25_32
- [22] Eben Upton, Gareth Halfacree. (2013, 2 de Diciembre). Raspberry Pi: Guía Del Usuario. Recuperado de: <http://es.scribd.com/doc/225682222/Raspberry-Pi-Guia-Del-Usuario-2da-Ed-en-Espanol>
- [23] HobbyTronics Ltd . (2014). Raspberry Pi GPIO Pinout. Recuperado de:
<http://www.hobbytronics.co.uk/raspberry-pi-gpio-pinout>
- [24] AOSONG. Temperature and Humidity Module. (2014). AM2301 Product Manual. Recuperado de: <http://meteobox.tk/files/AM2301.pdf>
- [25] AOSONG. Temperature and Humidity Module. (2014). AM2301 Product Manual. Recuperado de: <http://meteobox.tk/files/AM2301.pdf>
- [26] AOSONG. (2014). Temperature and Humidity Module AM2301 Product Manual. Recuperado de: <http://meteobox.tk/files/AM2301.pdf>
- [27] Adafruit. (2014). Adafruit PiTFT Mini Kit. Recuperado de:
<http://www.adafruit.com/products/1601>
- [28] ABB Inc. (2014). AquaMaster: Medidor electrónico para agua. Recuperado de:
[http://www05.abb.com/global/scot/scot203.nsf/veritydisplay/d20072c9ddb7a575c1257b0c0053fffb/\\$file/DS_AMAS-E_6.pdf](http://www05.abb.com/global/scot/scot203.nsf/veritydisplay/d20072c9ddb7a575c1257b0c0053fffb/$file/DS_AMAS-E_6.pdf)

[29] ABB Inc. (2014). Application Data Sheet Automated: Water Meter Reading. Recuperado de: <https://www.tropos.com/pdf/application-datasheets/automated-water-meter-reading.pdf>

[30] IUSSA. (2014). SMART SELF MANAGEMENT METERING SYSTEM FOR WATER. Recuperado de: http://www.iusa.com.mx/med02_en.asp?ic=23

ANEXOS

ANEXO A: CÓDIGO SENSOR TEMPERATURA HUMEDAD Y HUMEDAD.

```
#include <wiringPi.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#define MAXTIMINGS 85
#define DHTPIN 7 //GPIO 4
int dht11_dat[5] = {0,0,0,0,0};
void read_dht11_dat()
{
    uint8_t laststate = HIGH;
    uint8_t counter = 0;
    uint8_t j = 0, i;
    dht11_dat[0] = dht11_dat[1] = dht11_dat[2] = dht11_dat[3] = dht11_dat[4] = 0;
    pinMode(DHTPIN, OUTPUT);
    digitalWrite(DHTPIN, LOW);
    delay(5);
    digitalWrite(DHTPIN, HIGH);
    delayMicroseconds(40);
    pinMode(DHTPIN, INPUT);
    for ( i=0; i< MAXTIMINGS; i++) {
        counter = 0;
        while (digitalRead(DHTPIN) == laststate) {
            counter++;
            delayMicroseconds(1);
            if (counter == 255) {
                break;
            }
        }
        laststate = digitalRead(DHTPIN);
        if (counter == 255) break;
        if ((i >= 4) && (i%2 == 0))
        {
            dht11_dat[j/8] <= 1;
            if (counter > 16)
                dht11_dat[j/8] |= 1;
            j++;
        }
    }
    if ((j >= 40) && (dht11_dat[4] == ((dht11_dat[0] + dht11_dat[1] + dht11_dat[2] + dht11_dat[3]) & 0xFF)) )
```

```

{
    float f, h;
    h = dht11_dat[0] * 256 + dht11_dat[1];
    h /= 10;
    f = (dht11_dat[2] & 0x7F) * 256 + dht11_dat[3];
    f /= 10.0;
    if (dht11_dat[2] & 0x80) f *= -1;
    printf("%.1f--%.1f\n", f, h);
}
else
{
    printf("00.0--00.0\n");
}
}

int main (void)
{
    if (wiringPiSetup () == -1)
        exit (1) ;
    read_dht11_dat();
    return 0 ;
}

```

ANEXO B: CÓDIGO SENSOR DE FLUJO

```
#include <wiringPi.h>
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <sys/time.h>
#define FLOW_METER_PIN 0 //PIN 11 GPIO17
static volatile float litros = 0; //Variable donde Almacenare los litros leidos
static volatile float litroslast = 0;
static volatile float total_litros = 0 ; //Variable donde estara el valor global de los litros
static volatile int flowCounter = 0; //Contador de pulsos
static volatile double lastTimeMillis = 0;
static volatile float litersperminute = 0; //Variable de litros/min
static volatile float meter = 0; //Variable donde almaceno los metros
void incrementPulseCount(void) { //si ocurre una interrupcion un contador trabajara
    ++flowCounter;
}
int timevalToMillis(int secs, int usecs) {
    return secs * 1000 + usecs / 1000 ;
}
double calculateFlowRate(int pulses, double elapsedMillis) {
    return ( ((double)pulses / 450) / elapsedMillis) * 1000; // pulsos por segundo
}
int main()
{
    int lastCount = 0 ;
    struct timeval tv;
    double nowMillis = 0;
    gettimeofday(&tv, NULL);
    lastTimeMillis = timevalToMillis(tv.tv_sec, tv.tv_usec);
    FILE* archivo = fopen("/home/pi/Datos_Agua.txt", "r");
    if (archivo) //Si el archivo existe leo los datos que tiene; quiere decir que ya hay datos en esa txt
    {
        fscanf(archivo,"%f",&total_litros); //Leo el valor y lo almaceno en la variable litros
        fclose(archivo);
    }
    if (wiringPiSetup () < 0)
    {
        fprintf (stderr, "Problema configurando la funcion WiringPi: %s\n", strerror (errno)) ;
        return 1 ;
    }
    pinMode (1, OUTPUT) ; //GPIO 18
    pinMode (2, OUTPUT) ; //GPIO 27
    if (wiringPiISR (FLOW_METER_PIN, INT_EDGE_FALLING, &incrementPulseCount) < 0)
```

```

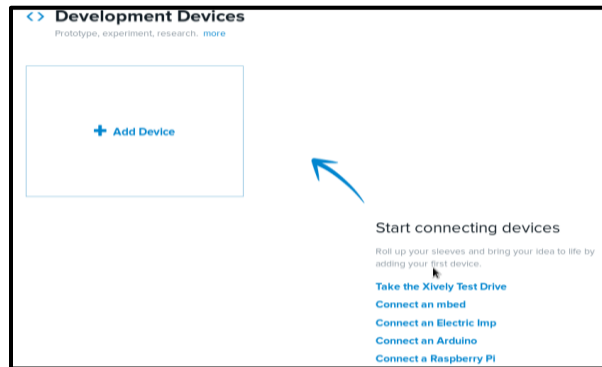
{
    fprintf(stderr, "Problema configurando la funcion ISR: %s\n", strerror(errno));
    return 1;
}

for (;;) //ciclo infinito
{
    printf("Waiting...."); fflush(stdout);
    digitalWrite(1, HIGH);
    delay(1000);
    gettimeofday(&tv, NULL);
    nowMillis = timevalToMillis(tv.tv_sec, tv.tv_usec);
    litersperminute = (calculateFlowRate((flowCounter - lastCount), nowMillis - lastTimeMillis)*60);
    if (litersperminute == 0.00)
    {
        digitalWrite(2, LOW);
    }
    else
    {
        digitalWrite(2, HIGH);
    }
    printf(" Flow Rate: %.2f L/min", litersperminute);
    litros = flowCounter;
    litros /= 6.5;
    litros /= 60.0;
    total_litros = total_litros + (litros - litroslast);
    printf(" %.2f", litros);
    printf(" Total Litros: %.2f", total_litros);
    meter = total_litros/1000;
    printf(" Total Metros Cubicos: %.2f\n", meter);
    FILE *file_write;
    file_write = fopen("/home/pi/Datos_Agua_Xively.txt", "w"); //txt para uso de Xively y demas
servicios
    fprintf(file_write, "%.2f--%.2f--%.2f", total_litros, litersperminute, meter);
    fclose(file_write);
    litroslast = litros;
    FILE *file_write_;
    file_write_ = fopen("/home/pi/Datos_Agua.txt", "w");
    fprintf(file_write_, "%.2f", total_litros);
    fclose(file_write_);
    gettimeofday(&tv, NULL);
    lastCount = flowCounter;
    lastTimeMillis = timevalToMillis(tv.tv_sec, tv.tv_usec);
}
return 0;
}

```

ANEXO C: CREAR UNA CUENTA EN XIVELY

- Tenemos que crear una cuenta en <https://xively.com/>. A continuación, comprobar en el correo electrónico el link para validar la cuenta.
- En la nueva página podemos añadir nuestra Raspberry Pi



- Le asignamos un nombre y una descripción. Por último la convertimos en un dispositivo público.

- Esto nos llevará a nuestra página de inicio del dispositivo, donde podemos empezar a añadir canales a nuestros datos, tales como la temperatura, humedad, temperatura PC, litros, metros cúbicos, etc. Hay algunos detalles clave en esta página. Tomar en cuenta 'ID Feed' en la parte superior, y "API keys" en el medio. Estos son necesarios para vincular nuestra Raspberry Pi a Xively a través de código Python.

Channels

Last updated in a minute

Graphs

Add Channel

required

e.g. sensor1

Tags

Use a comma to separate tags.

Units

Symbol

e.g. energy, project:name=my_pi

e.g. Watts

e.g. W

Current Value

Save Channel

Cancel

Activated

Deactivate

at 11-02-2014 15:07:35

Deploy

Feed ID

388505930

Feed URL

<https://xively.com/feeds/388505930>

API Endpoint

<https://api.xively.com/v2/feeds/388505930>

Auto-generated Raspberry Pi test device key for feed

388505930

tjffqbm01CRk5rj2GN4yhD4RbxPSUYmYB2CSkTVH2cYLlID

permissions READ,UPDATE,CREATE,DELETE

private access

ANEXO D: ENVIÓ DE DATOS A XIVELY

```
#!/usr/bin/python3
import os, glob, time, sys, datetime, eeml, syslog, json, math, smtplib
file = open('/home/pi/Datos_Agua_Xively.txt','r')
linea = file.readline()
file.close()
file1 = open('/home/pi/Datos_Agua.txt','r')
linea1 = file1.readline()
file1.close()
metro = float (linea1)/1000
datos = linea.split("--")
litros = (datos[0]) #litros
flowratelitros = (datos[1]) #flowrate
metros = (datos[2]) #metros
cmd1 = '/opt/vc/bin/vcgencmd measure_temp'
line1 = os.popen(cmd1).readline().strip()
c = line1.split('=')[1].split('"')[0]
d = float(c)
e = math.ceil(d)
temp_cpu = str(e)
file_one = open("/home/pi/temp_cpu_GUI.txt", "w")
file_one.write(temp_cpu)
file_one.close()
cmd = os.popen("sudo /home/pi/AM2301/am2301")
line = cmd.readline().strip()
flag = 0
while flag == 0:
    if line == "00.0--00.0":
        cmd = os.popen("sudo /home/pi/AM2301/am2301")
        line = cmd.readline().strip()
        flag = 0
    else:
        data = line.split("--")
        f = float (data[0])
        g = float (data[1])
        temp_amb = math.ceil(f)
        hume_amb = math.ceil(g)
        flag = 1
API_KEY = 'FYaUp2y7vEsWJ4qhtV0fPpEHaBe5c1r5xxxxxxxxxxxxxx'
FEED = 15386xxxxx
API_URL = '/v2/feeds/{feednum}.xml'.format(feednum = FEED)
pac = eeml.Pachube(API_URL, API_KEY)
pac.update([eeml.Data("CPU", temp_cpu, unit=eeml.Celsius())])
pac.update([eeml.Data("FLUJO-AGUA", flowratelitros)])
pac.update([eeml.Data("CONSUMO-AGUA-POTABLE", litros)])
```

```

if temp_amb >= 40.00 or temp_amb <= 0.00 :
    i = 0
else:
    pac.update([eeml.Data("TEMPERATURA", temp_amb, unit=eeml.Celsius())])
    file_two = open("/home/pi/temp_amb_GUI.txt", "w")
    file_two.write(str(temp_amb))
    file_two.close()
if hume_amb >= 90.00 or hume_amb <= 0.00 :
    i = 0
else:
    pac.update([eeml.Data("HUMEDAD", hume_amb, unit=eeml.RH())])
    file_three = open("/home/pi/hume_amb_GUI.txt", "w")
    file_three.write(str(hume_amb))
    file_three.close()
pac.update([eeml.Data("CONSUMO-AGUA-POTABLE-", metros)])
pac.at = datetime.datetime.now()
pac.put()

```

ANEXO E: REPORTES DIARIOS CON GOOGLE DRIVE Y BASE DE DATOS LOCAL

```
import os, glob, time, gspread, sys, datetime, eeml, syslog, json, math, smtplib
from datetime import datetime
ahora = datetime.now()
month = ahora.strftime("%B")
date = str(ahora.day) + "--" + str(ahora.month) + "--" + str(ahora.year)
name_file = "/home/pi/Datos_Agua_" + month + str(ahora.year) + ".txt"
file_read = open('/home/pi/Datos_Agua.txt','r')
line = file_read.readline()
file_read.close()
datos = line.split("--")
litros = (datos[0]) #litros
metro = float (litros)/1000
metro1 = "{0:.2f}".format(metro)
file_txt = open(name_file,'a')
file_txt.write(date + '-' + litros + '\n')
file_txt.close()
fecha = str(ahora.day) + "-" + str(ahora.month) + "-" + str(ahora.year)
email = 'raspebrry@gmail.com'
password = 'xxxxxxx'
spreadsheet = 'Variables Raspberry Pi'
try:
    gc = gspread.login(email,password)
except:
    print('fail')
    sys.exit()
worksheet = gc.open(spreadsheet).sheet1
values = [fecha, metro1, litros]
worksheet.append_row(values) #write to the spreadsheet
```

ANEXO F: REPORTES MENSUALES DE CONSUMO CON ENVIÓ DE E-MAILS

```
from datetime import datetime
import os, glob, time, sys, eeml, syslog, json, math, smtplib
ahora = datetime.now()
month = ahora.month - 1
if month == 0: #DECEMBER
    name_file = "/home/pi/Datos_Agua_" + "Dec" + str(ahora.year)+ ".txt"
    name_month = "Dec"
elif month == 1: #JANUARY
    name_file = "/home/pi/Datos_Agua_" + "Jan" + str(ahora.year)+ ".txt"
    name_month = "Jan"
elif month == 2: #FEBRUARY
    name_file = "/home/pi/Datos_Agua_" + "Feb" + str(ahora.year)+ ".txt"
    name_month = "Feb"
elif month == 3: #MARCH
    name_file = "/home/pi/Datos_Agua_" + "Mar" + str(ahora.year)+ ".txt"
    name_month = "Mar"
elif month == 4: #APRIL
    name_file = "/home/pi/Datos_Agua_" + "Apr" + str(ahora.year)+ ".txt"
    name_month = "Apr"
elif month == 5: #MAY
    name_file = "/home/pi/Datos_Agua_" + "May" + str(ahora.year)+ ".txt"
    name_month = "May"
elif month == 6: #JUNE
    name_file = "/home/pi/Datos_Agua_" + "Jun" + str(ahora.year)+ ".txt"
    name_month = "Jun"
elif month == 7: #JULY
    name_file = "/home/pi/Datos_Agua_" + "Jul" + str(ahora.year)+ ".txt"
    name_month = "Jul"
elif month == 8: #AUGUST
    name_file = "/home/pi/Datos_Agua_" + "Aug" + str(ahora.year)+ ".txt"
    name_month = "Aug"
elif month == 9: #SEPTEMBER
    name_file = "/home/pi/Datos_Agua_" + "Sep" + str(ahora.year)+ ".txt"
    name_month = "Sep"
elif month == 10: #OCTUBER
    name_file = "/home/pi/Datos_Agua_" + "Oct" + str(ahora.year)+ ".txt"
    name_month = "Oct"
elif month == 11: #NOVEMBER
    name_file = "/home/pi/Datos_Agua_" + "Nov" + str(ahora.year)+ ".txt"
    name_month = "Nov"
cont = 0
suma = 0
metros = 0
i = 0
```

```

infile = open(name_file, 'r')
for line in infile:
    datos = line.split("--")
    cont = float(datos[3])
    suma = suma + cont
    i = i + 1
infile.close()
metros = suma/1000
smtpUser = 'raspebrry@gmail.com'
smtpPass = 'XXXXXXXXXX'
toAdd = 'Test Users'
emails = ["lis98eve@gmail.com", "Fa_byEd@hotmail.com", "edisonfabricioch@gmail.com",
"edisonfabricio@electritelecom.com", "david.0189.chamba@gmail.com",
"jen_93_flakis@hotmail.com", "jonathanagreda1990@gmail.com", "da_incho560@hotmail.com",
"Mfvaldiviezo@gmail.com"]
fromAdd = smtpUser
subject = 'Raspberry Home Server'
header = 'To: ' + toAdd + '\n' + 'From: ' + fromAdd + '\n' + 'Subject: ' + subject
body = 'De Raspberry Pi: ' + '\n' + '\n' + 'Reporte Mensual del Consumo de Agua Potable\n' + 'Los metros cubicos consumidos son: ' + str(metros) + '\n' + 'Datos Facturados desde: 01 al ' + str(i) + ' de ' + name_month + '\n' + '\n' + 'Mensajes generados automaticamente por Raspberry Home Server.' + '\n' + 'Sistema en version de pruebas.' + '\n' + 'Email de contacto: edisonfabricioch@gmail.com' + '\n' + '\n' + 'Puedes visitar tambien: ' + '\n' + 'https://xively.com/feeds/1538624376' + '\n'
s = smtplib.SMTP('smtp.gmail.com', 587)
s.ehlo()
s.starttls()
s.ehlo()
s.login(smtpUser, smtpPass)
s.sendmail(fromAdd, emails, header + '\n\n' + body)
s.quit

```

ANEXO G: NOTIFICACIONES DE WHATSAPP

```
#!/usr/bin/env
# -*- coding: utf-8 -*-
import base64
import os
import math
from Examples.EchoClient import WhatsappEchoClient
password = "YaKy1ouQK8+6mUxxxxxxx"
password = base64.b64decode(bytes(password.encode('utf-8')))
username = '59396928xxx'
keepAlive= False
cmd1 = '/opt/vc/bin/vcgencmd measure_temp'
line1 = os.popen(cmd1).readline().strip()
e = line1.split('=')[1].split('')[0]
f = float(e)
g = math.ceil(f)
temp_cpu = str(g)
if g > 60:
    whats = WhatsappEchoClient("593993796xxx","¡Alerta! La temperatura de la CPU es de: " +
temp_cpu + " °C." + "\n" + "Para mas informacion visite: https://xively.com/feeds/1538624376" ,
keepAlive)
    whats.login(username, password)
file = open('/home/pi/Datos_Agua_Xively.txt','r')
linea = file.readline()
file.close()
datos = linea.split("--")
litros = (datos[0]) #litros
flowratelitros = (datos[1]) #flowrate
metros = (datos[2]) #metros
metros1 = float (flowratelitros) #valor numerico
if metros1 >= 16:
    whats = WhatsappEchoClient("593993796xxx","¡Alerta! ¡La velocidad del Flujo de agua es de: "
+ str(metros1) + " Litros por minuto! " + "\n" + "Para mas informacion visite:
https://xively.com/feeds/1538624376" , keepAlive)
    whats.login(username, password)
```

ANEXO H: BOT TWITTER

```
#!/usr/bin/python3
import os, glob, time, sys, datetime, eeml, syslog, json, tweepy, math, smtplib
cmd1 = '/opt/vc/bin/vcgencmd measure_temp'
line1 = os.popen(cmd1).readline().strip()
c = line1.split('=')[1].split('')[0]
d = float(c)
e = math.ceil(d)
temp_cpu = str(e)
cmd = os.popen("sudo /home/pi/AM2301/am2301")
line = cmd.readline().strip()
flag = 0
while flag == 0:
    if line == "00.0--00.0":
        cmd = os.popen("sudo /home/pi/AM2301/am2301")
        line = cmd.readline().strip()
        flag = 0
    else:
        data = line.split("--")
        f = float (data[0])
        g = float (data[1])
        temp_amb = math.ceil(f)
        hume_amb = math.ceil(g)
        flag = 1
file1 = open('/home/pi/Datos_Agua.txt','r')
linea1 = file1.readline()
file1.close()
metro = float (linea1)/1000
metro1 = "{0:.2f}".format(metro)
consumer_key = 'MP4rxYEznQEzcOXXXXXX'
consumer_secret = 'R4am8IKo3V7z2g6wVqvN9iG7rQo5FYjXXXXXXXXXX'
access_token = '1890716154-fl4PwXBeVKFZF7Inma7SNIEYXXXXXXXXXX'
access_token_secret = '8GPKYFO1A0fGjSPm0aTKVrk79wpZp39XXXXXXXXXX'
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)
api.update_status('#Loja #RaspberryPi: El valor del CPU es: ' + temp_cpu + 'C. ' + 'Humedad ambiente: ' + str(hume_amb) + '%. ' + 'Temperatura: ' + str(temp_amb) + 'C. ' + 'Consumo Agua Potable: ' + linea1 + 'L - ' + metro1 + 'm Cubicos.')
```

ANEXO I: INTERFAZ GRÁFICA QT CREATOR

```
#include <QApplication>
#include "mainwindow.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.setWindowFlags(Qt::Window | Qt::FramelessWindowHint);
    w.show();
    return a.exec();
}
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QTimer>
#include <QDateTime>
#include <QFile>
#include <QTextStream>
#include <QMessageBox>
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    QTimer *timer = new QTimer (this);
    connect(timer, SIGNAL(timeout()),this, SLOT(show_time()));
    timer->start(1000);
}
void MainWindow::show_time()
{
    QTime time = QTime::currentTime();
    QString time_text = time.toString("hh : mm : ss");
    ui->label_time->setText(time_text);
    QDateTime date_time = QDateTime::currentDateTime();
    QString date_time_text = date_time.toString("dddd d, MMMM yyyy");
    ui->label_date->setText(date_time_text);
    QFile file ("/home/pi/hume_amb_GUI.txt");
    QFile file2 ("/home/pi/temp_amb_GUI.txt");
    QFile file3 ("/home/pi/temp_cpu_GUI.txt");
    QFile file4 ("/home/pi/Datos_Agua_Xively.txt");
    if(!file.open(QIODevice::ReadOnly))
        QMessageBox::information(0,"info",file.errorString());
    if(!file2.open(QIODevice::ReadOnly))
        QMessageBox::information(0,"info",file2.errorString());
    if(!file3.open(QIODevice::ReadOnly))
        QMessageBox::information(0,"info",file3.errorString());
```



```

        if(!file4.open(QIODevice::ReadOnly))
            QMessageBox::information(0,"info",file4.errorString());
        QTextStream data(&file);
        QTextStream data2(&file2);
        QTextStream data3(&file3);
        QTextStream data4(&file4);
        QString texto = data4.readAll();
        QStringList lista;
        lista = texto.split("--");
        ui->label_hum->setText(data.readAll() + " %");
        ui->label_amb->setText(data2.readAll() + " C");
        ui->label_cpu->setText(data3.readAll() + " C");
        ui->label_litros->setText(lista[0]);
        ui->label_metros->setText(lista[2]);
        ui->label_flujo->setText(lista[1]);
    }
MainWindow::~MainWindow()
{
    delete ui;
}
#ifndef MAINWINDOW_H
#define MAINWINDOW_H
#include <QMainWindow>
namespace Ui {
class MainWindow;
}
class MainWindow : public QMainWindow
{
    Q_OBJECT
public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();
private slots:
    void show_time();
private:
    Ui::MainWindow *ui;
};
#endif // MAINWINDOW_H

```

IMÁGENES:



Raspberry Pi Modelo B.



Kit de Desarrollo Raspberry Pi.



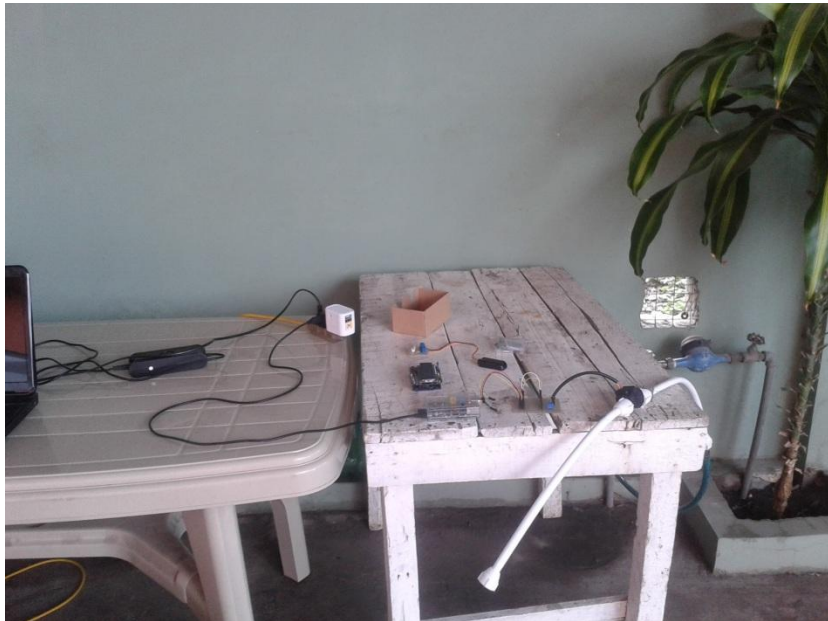
Sistema de medición provisional para pruebas del sensor.



Sistema de medición provisional armado.



Pruebas del sensor usando un Osciloscopio.



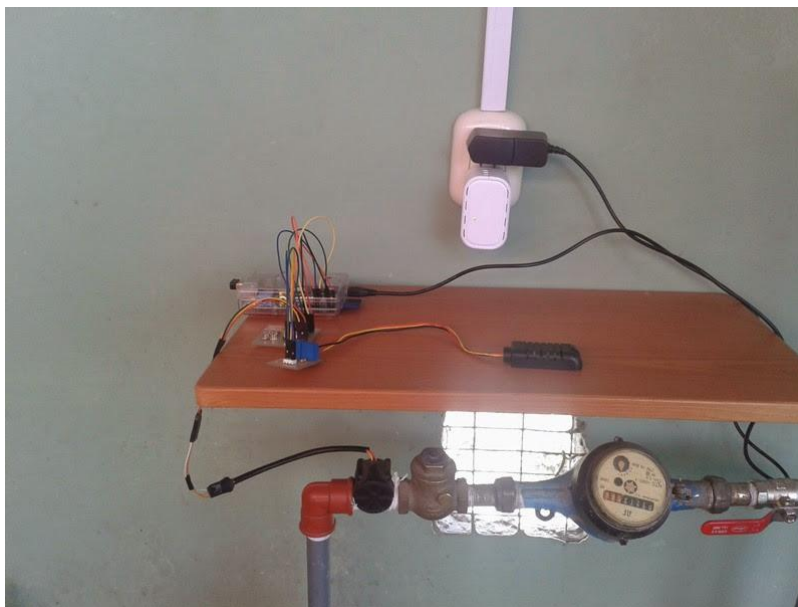
Pruebas del sensor con el sistema de medición provisional.



Pruebas del prototipo usando un recipiente de 20L.



Instalación del sensor.



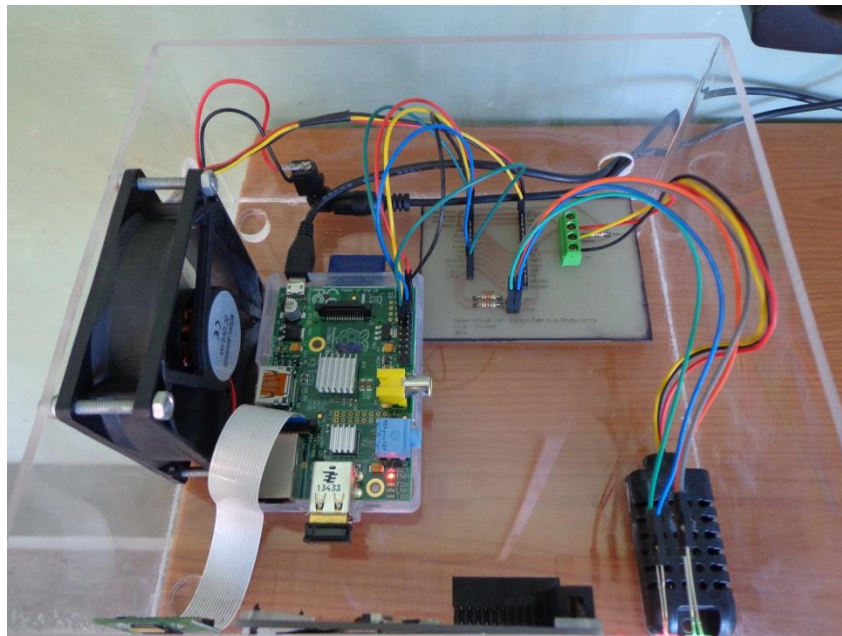
Instalación provisional del prototipo de medición.



Vista frontal del prototipo de medición.



Vista completa del prototipo implementado.



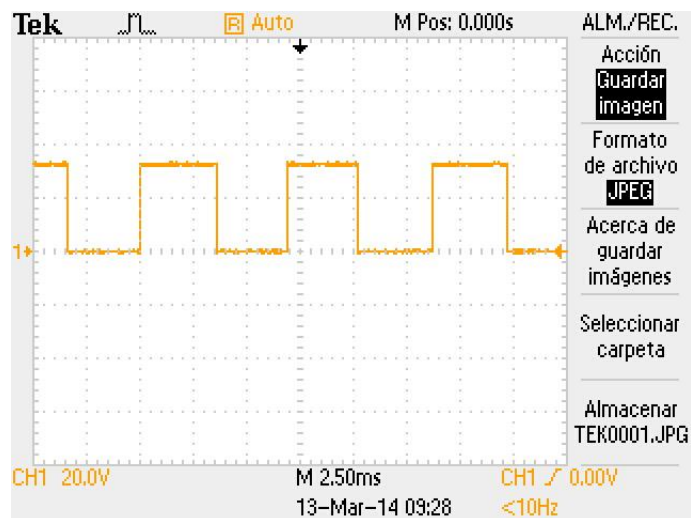
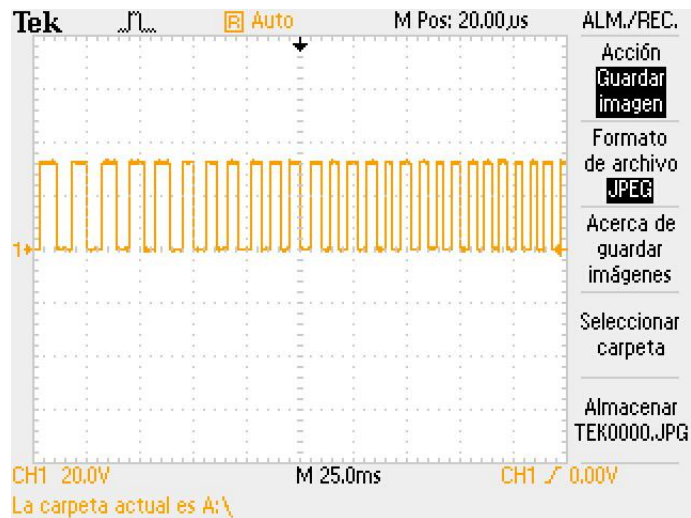
Vista superior del prototipo de medición.



Vista lateral izquierda.



Vista lateral derecha.



Tren de pulsos generado por el sensor de flujo. La imagen se obtiene usando un Osciloscopio TEKTRONIX.