

**INSTITUTO TECNOLOGICO DE TUXTLA GUTIERREZ**



**INGENIERIA ELECTRONICA  
RESIDENCIA PROFESIONAL**

**IMPLEMENTACION Y ANALOGIA ELECTRICA DE UNA RED DE  
DISTRIBUCION DE AGUA POTABLE**

**ASESOR**

DR. HECTOR RICARDO HERNANDEZ DE LEON

**ALUMNOS:**

JOSE SANTIAGO GOMEZ PEREZ  
ROGER ISAAC TOLEDO CRUZ

TUXTLA GUTIERREZ, CHIAPAS A 19 DE ENERO DE 2014.

<b>CAPITULO 1.</b>	
<b>GENERALIDADES.....</b>	<b>4</b>
1.1.- INTRODUCCION.....	5
1.2.- ANTECEDENTES.....	7
1.3.- PLANTEAMIENTO DEL PROBLEMA.....	8
1.4.-OBJETIVOS.....	10
1.4.1.-Objetivo general.....	10
1.4.2.- Objetivos específicos.....	10
1.5.- JUSTIFICACION DEL PROYECTO.....	11
1.6.- ALCANCES Y LIMITACIONES.....	12
1.7.- INFORMACION GENERAL DE LA INSTITUCION O EMPRESA DONDE SE DESARROLLO EL PROYECTO.....	13
1.7.1.- Historia del ITTG.....	13
1.7.2.-Mision.....	14
1.7.3.- Visión.....	14
1.7.4.- Localización.....	14
1.7.5.- Área específica relacionada directamente con el proyecto.....	14
<b>CAPITULO 2. FUNDAMENTO TEORICO.....</b>	<b>15</b>
2.1.- PROGRAMACION EN ARDUINO.....	15
2.1.1.- Arduino uno.....	15
2.1.2.-Características.....	15
2.1.3.-Estructura básica de un programa.....	16
2.1.4.- Entradas/salidas analógicas y digitales.....	16
2.1.6.- Funciones de tiempo.....	17
2.2.- MODULOS X-BEE.....	17
2.2.1.- El coordinador.....	18
2.2.2.- El routers.....	19
2.2.3.- End device.....	19
2.2.4.-Aplicaciones.....	19
2.2.5.-Circuito básico para Xbee.....	20
2.2.6.- Configuración del módulo Xbee.....	21
2.2.7.- Direccionamiento de los módulos.....	21
2.2.8.- Direccionamiento de 16 bits.....	21
2.2.9.- Direccionamiento de 64 bits.....	21
2.2.10.- Punto a punto.....	22
2.2.11.- Punto a multipunto.....	24
2.3.- ¿QUE ES MATLAB?.....	26
2.3.1.-Interfaz gráfica de usuario en Matlab (GUIDE).....	26
2.3.2.-Como crear una GUIDE.....	26
2.3.3.-Propiedades de los componentes.....	29
2.3.4.-Funcionamiento de una aplicación GUIDE.....	30

2.3.5.-Manejo de datos entre los elementos de la aplicación y el archivo .m	30
<b>CAPITULO 3.DESARROLLO E IMPLEMENTACION DEL PROYECTO</b>	<b>32</b>
3.1.-DISEÑO DEL SISTEMA DE ADQUISICION DE DATOS	32
3.1.1.-Características del sensor YF-G21( sensor de flujo)	32
3.1.2.-Diagrama a bloques del proceso que sigue la información del sensor de flujo	33
3.1.3.-Simulación del sistema de adquisición de datos	34
3.1.4.-Materialización del medidor digital	36
3.1.5.-Tratamiento de la señal enviada de ATMEGA328P-PU al módulo Xbee	37
3.1.6.- Diagrama de flujo del programa grabado en el microcontrolador	41
3.2.-IMPLEMENTACION DEL SISTEMA DE ADQUISICION DE DATOS	43
3.2.1.- Estructura de la tubería de pruebas para el medidor digital	43
3.2.2.- Caracterización de los medidores digitales desarrollados	43
3.2.3.-Calibración del medidor	45
3.2.4.- Diseño de la gráfica Matlab	47
3.2.5.- Recibiendo datos en el puerto serial COM10	47
3.2.6.-Diagrama de flujo del programa del programa en Matlab	49
3.4.-RESULTADOS OBTENIDOS	51
3.4.1.- Medidor digital en operación	51
3.4.2.-Graficas generadas por cada medidor digital de forma inalámbrica	51
<b>CONCLUSIONES</b>	<b>53</b>
<b>RECOMENDACIONES DE USO DEL SISTEMA</b>	<b>54</b>
<b>RECOMENDACIONES PARA LA MEJORA DEL SISTEMA</b>	<b>54</b>
<b>REFERENCIAS</b>	<b>55</b>
<b>ANEXO 1.-Diagrama de Arduino UNO</b>	<b>56</b>
<b>ANEXO 2.- Como usar ATMEGA328P-PU fuera de la placa de Arduino</b>	<b>57</b>
<b>ANEXO 3.-Especificaciones del sensor de flujo YF-G21</b>	<b>59</b>
<b>ANEXO 4.-Programa grabado en el micro controlador ATMEGA328P-PU</b>	<b>60</b>
<b>ANEXO 5.- Programa para la gráfica de datos en Matlab</b>	<b>66</b>
<b>ANEXO 6.- Diferencia de potencial en los LED</b>	<b>69</b>
<b>ANEXO 7.-Efecto hall</b>	<b>70</b>

## Capítulo 1. Generalidades

### 1.1.- Introducción

Ningún recurso es más fundamental para la vida y el bienestar de la sociedad que el agua. Y sin embargo, cada año se pierde en total entre el 25 y el 30% del agua potable por culpa de fugas en los sistemas urbanos de distribución hídrica. Por ello se ha pensado en la forma más viable, segura y económica de resolver esta problemática, de una forma innovadora.

Considerando que innovar en el ámbito tecnológico es una de las actividades que la humanidad ha venido haciendo desde hace ya cierto tiempo y gracias a esto se ha podido avanzar, debido a que vemos esto como un rompecabezas, en la cual los primeros investigadores aportan ciertas bases científicas que a su vez los demás utilizan para crear algo mejor, y de utilidad para las personas.

La presente investigación, se basa en el desarrollo de un medidor de caudal o flujo en tubos de media pulgada, considerando pues que tal instrumento ya está a la venta, nuestra mejora se centra en hacerlo de forma digital. Buscando una ventaja en este instrumento, de medición, la cual va a consistir en que además de mostrar de manera visual la velocidad del flujo en un determinado momento, va a ser capaz de enviar datos a una computadora que administre los datos de muchos más sensores de esta índole, pudiendo así ver de manera rápida y confiable la información para quien le sea de utilidad.

El tema de esta investigación es IMPLEMENTACION Y ANALOGIA ELECTRICA DE UNA RED DE DISTRIBUCION DE AGUA POTABLE ya que además de hacer la implementación del sistema de adquisición, con los sensores y tubos de PVC, este operara de forma análoga a la ley de conservación de las masas de Lavoisier para la red hidráulica en donde todo lo que entra en un sistema tiene que salir; el principio será parecido cuando se use la ley de corrientes de Kirchhoff al hacer un circuito simulador electrónico en el monitoreo. Con esto y el interfaz gráfico los proveedores podrán interpretar de manera sencilla y confiable el estado de la red hídrica.

Dado que una de los beneficios del proyecto es el aumento de la cantidad de agua utilizable esperamos que con la implementación real de esta propuesta, mejore notablemente la calidad en el servicio del agua potable, evitando así las demoras en encontrar la falla, fisuras en la tubería, etc.

## 1.2.- Antecedentes

¿Cuenta usted con agua utilizable a su disposición en la cantidad que desea? Muchas amas de casa todavía usan frugalmente su provisión diaria, pues ésta tiene que ser extraída de arroyos cercanos, manantiales o de algún pozo distante. Otros, tal vez, nunca han reflexionado en el asunto debido a que los adelantos modernos han hecho que el agua sea muy fácil de conseguir. Pero no siempre ha sido de esta manera. ¿Sabe usted de qué manera muchos de los habitantes de las ciudades obtenían su agua hace siglos?

La respuesta es los acueductos. Los acueductos antiguos por lo general eran largos y angostos túneles, completamente cerrados, que proveían un canal por encima del nivel de la tierra para que el agua fluyera libremente de un manantial natural a una ciudad. El fluir del agua se efectuaba por gravedad, dándole al acueducto una inclinación gradual de diecinueve a treinta y ocho centímetros por kilómetro. Donde existían valles, era necesario construir una estructura, del tipo de puente, que llevara el agua por la misma inclinación cuesta abajo. Cuando la corriente de agua encontraba cerros o montañas a su paso, era necesario perforar un túnel a través de la montaña.

Es claro que se necesitaban buenos planos de ingeniería. La Biblia relata que Ezequías, rey de Judá (745 a 716 a. de la E.C.), dirigió la construcción de un notable acueducto cortado a través de la roca sólida. Él hizo esto usando dos equipos de hombres que trabajaron el uno hacia el otro desde extremos opuestos hasta que se encontraron en el centro. ¡Y éste no era un túnel pequeño, pues su altura promedio era de 1,83 metros y su largo de unos 542 metros, o más de medio kilómetro

En el presente ya no recurrimos a los acueductos como hicieron otros países en el pasado sino que el suministro de agua se hace a base de cañerías o tubos de diferentes materiales para lograr el mismo fin de una forma más sencilla y rápida, sin embargo el hombre en su intento por sacar agua de otras fuentes y llevarla hasta el lugar de uso se ha encontrado a otros problemas como es el control del flujo, conservar la calidad en el traslado dentro del tubo, así como el cuidado del líquido para que este no se desperdicie o contamine al tener contacto con suelos contaminados cuando el tubo está dañado.

Un proyecto financiado con fondos europeos LEAKCURE ("Intervention for curing pipeline leakage in urban water mains") propone una solución innovadora para la detección automática, el sellado y el curado de las tuberías comunes de dichas redes, sin necesidad de levantar aceras y calzadas.

Se está observando estrés hídrico en zonas donde hasta ahora no solía haber. Según el coordinador del proyecto, Peter Paz de Curapipe System Ltd. (Israel): "La mayoría de las fugas de agua de la red son difíciles de detectar con los medios convencionales. Los métodos habituales, consistentes en arreglar sólo las averías que se encuentren, son ineficaces. Por otra parte, eliminar varias fugas a la vez cambiando cañerías completas resulta caro y genera molestias".

Ya se está observando estrés hídrico -que refleja el balance entre la oferta y la demanda de agua- en zonas donde hasta ahora no solía haber, por ejemplo en el Reino Unido, según informó Paz. Este estrés hídrico está agravándose además en zonas donde ya existía, como en algunas zonas de Europa meridional. Ello se debe en parte al cambio climático y en parte a otros factores como el avance de la urbanización y la industrialización.

Según declaró: "Antes no siempre se daba prioridad a las fugas de tuberías. Ahora se empieza a prestar más atención a la cantidad de fugas por el incremento del estrés hídrico y por la falta de suministros de agua nuevos de acceso fácil".

La solución propuesta por LEAKCURE, llamada "Trenchless, Automated Leakage Repair" (TALR), pasa por implantar unas piezas especiales llamadas lingotes que se introducen para formar un "tren de lingotes".

A grandes rasgos, un lingote es similar a un tapón que se puede hacer que se desplace por un conducto controlando la presión y el caudal. Al introducir dos lingotes en la cañería, dejando un vacío entre ambos, se puede parar el paso de agua por delante y por detrás. En el "tren de lingotes" propuesto por LEAKCURE, ese vacío se rellena con una sustancia viscosa especial de curado que se desplaza por el conducto entre los dos lingotes.

Según explicó Paz: "Este 'tren' se mueve de un modo continuo, no se detiene. Así va sellando las fugas al instante a medida que la sustancia de curado entra en contacto con la fuga, cualquiera que sea su orientación dentro de la tubería. Esta sustancia se endurece y la fuga queda curada de modo permanente a medida que el tren avanza por la tubería".

Paz señaló que las actuales prácticas de reparación están enfocadas sobre todo hacia el control de daños, es decir, evitar que la fuga se agrave, más que reducirla de verdad. "Nuestra propuesta pretende reducir en un grado significativo los niveles actuales de fugas, que son inadmisibles, pues en Europa suponen casi un tercio del agua suministrada".

El equipo del proyecto LEAKCURE está probando su nuevo sistema en el Reino Unido, para lo cual han entablado asociaciones con entidades locales y organizadas ensayos de campo. Al término de estos ensayos, y una vez se introduzcan las adaptaciones necesarias para ajustarse a los contextos locales, el equipo ampliará sus actividades fuera del Reino Unido.

Las actuales prácticas de reparación están enfocadas sobre todo hacia el control de daños. Paz informó que Curapipe comercializará el servicio TALR entre compañías de aguas en colaboración con socios locales que ya se dedican a la reparación y el mantenimiento de cañerías de agua. Y declaró: "Hasta ahora, nuestros resultados indican que podemos reparar los tipos de fugas que causan la mayor parte de las pérdidas hídricas. Se trata de fugas que van de cuarenta a tres

mil litros por hora. Por extensión, si tratamos 40 kilómetros de tuberías con fugas, en teoría podríamos ahorrar cada día 1,7 millones de litros de agua potable".

Según la Autoridad Reguladora de Servicios Hídricos del Reino Unido (OFWAT), la reducción de las fugas de agua tendrá beneficios considerables, entre ellos que se incrementará la cantidad de agua utilizable en el entorno por otros usuarios, como los agricultores, o bien para sostener los hábitats naturales. Asimismo, las empresas y los ayuntamientos deberán gastar menos en obtener nuevos suministros de agua, ahorrarán en costes de tratamiento y bombeo y, al elevar la eficiencia de estas actividades, podrán reducir su gasto energético y sus emisiones de gases de efecto invernadero.

El consorcio responsable de LEAKCURE, que finalizará en 2014, recibió financiación por valor de más de 700.000 euros a través del Programa de Eco-innovación.

### 1.3.- Planteamiento del problema:

El mundo está aprendiendo que la capacidad que el hombre tiene de llevar agua de un lugar a otro sin trastornar gravemente el equilibrio de la naturaleza tiene sus límites.

Cierta revista<sup>1</sup> afirma que con tan solo reducir las pérdidas en la red hidráulica se puede incrementar mucho el abastecimiento de agua. Se ha calculado que, en todo el mundo, por culpa de utilizar métodos ineficaces, se pierde en fugas, evaporación y otras vías entre el 65 y el 70% del agua utilizable.

Tan solo en la ciudad de Tuxtla Gutiérrez, Chiapas la población sufre una severa crisis en su infraestructura de la red de agua potable, la longevidad de las tuberías provoca que diariamente aparezcan de entre 150 y 160 fugas diarias, reconoció director del Sistema Municipal de Agua Potable y Alcantarillado (SMAPA). En una página de internet de noticias fechada el 17 de enero de 2014.

Lo lamentable indicó es que del total de las reportes recibidos, solamente pueden atender cuarenta, esto debido al poco personal encargado que tienen para subsanar este tipo de situaciones, es decir sólo cuatro de 10 fugas se atienden.

Según revela el informe que con toda esta situación de las fugas, aunado a tomas clandestinas, se pierde alrededor del 65 por ciento del líquido bombeado a la ciudad, es decir que de los 3 mil litros por segundo que se bombea, casi 2 mil se van por las calles de la ciudad y a depósitos ilícitos y solamente mil 50 litros llegan a las tomas regularizadas.

Aparte de que la crisis no es solamente en el sistema hidráulico, también en las finanzas; SMAPA tienen una cartera vencida de 40 millones de pesos producto del adeudo de cientos de usuarios que han dejado de pagar el servicio.

Con problemas como este, las empresas buscan soluciones, con las que se pueda seguir dando el servicio y de alguna forma favorecer la economía de la misma; por ello el presente proyecto ofrece sino es la solución completa a esta problemática, un avance de la solución total; aquí se busca aumentar la cantidad de agua utilizable, viéndolo desde una perspectiva razonable tanto para el usuario como el proveedor y es que si esta se derrama por las calles o se contamina mediante contacto con el suelo contaminado en las grietas de la tubería esta ya no es utilizable.

Como usuario debemos entender que el agua potable no puede ser gratuita si te lo llevan hasta tu casa esta tendrá un costo, esto también hace un llamado a aquellos individuos que hacen conexiones clandestinas ocultas que con

---

<sup>1</sup> Revista Times, 1991.



la implementación de este sistema podrán ser detectados de inmediato ya que cada toma autorizada tendrá un medidor de flujo inalámbrico, con el cual se obtendrá un dato en litros por minuto y lo enviara de forma inalámbrica a una computadora central, y esta va a comparar la suma de todos los caudales de las tomas con el flujo de entrada, si este resultara mayor que lo que se está suministrando, el sistema se pondrá en alerta, indicando que pueden estar ocurriendo dos cosas la primera es que existe una fuga en la cuadra de la calle y la otra es que alguien está haciendo uso del agua de forma clandestina.

Con este sistema la compañía abastecedora puede de igual forma hacer dos cosas, una es enviar personal para revisar y corregir el desperfecto, y la otra es invertir un poco más en este proyecto y dotar a la red con actuadores inalámbricos (una especie de solenoides) que cerraran el suministro de agua en rama dañada de la red. Con esto se puede pedir a los usuarios su colaboración en el arreglo o motivarlos a pagar sus mensualidades atrasadas con el fin de restablecer el servicio.

El beneficio reside en que con esto es posible operar todo el sistema con un mínimo de personal, ahorrándose capital, quitar todo aquel usuario que no tenga un contrato oficial con la abastecedora del servicio, así como también evitar encharcamientos en las calles en los cuales se pierde una buena cantidad del líquido y dan mucho que decir a los medios de comunicación.

## **1.4.-Objetivos**

### **1.4.1.- Objetivo general**

Desarrollar un sistema que haga uso de la analogía entre sistemas hidráulicos y eléctricos para detectar y evitar fugas en las calles y tomas clandestinas dentro de la red hidráulica, buscando un aumento en la cantidad de agua utilizable para la empresa.

### **1.4.2.- Objetivos específicos**

- 1.- Recopilación de información e investigación de la analogía eléctrica aplicada a redes de distribución de agua potable y simuladores.
- 2.- Simular el circuito análogo eléctrico a partir de la red hidráulica bajo estudio utilizando software de simulación electrónica.
- 3.- Construcción de la red prototipo de distribución de agua que permita caracterizar su funcionamiento
- 4.- Construcción del sistema de adquisición de datos y la interfaz gráfica del prototipo utilizado implementando un medidor digital capaz de tomar datos analógicamente de flujo de una red de agua potable a datos electrónicos a partir de sensores de caudal, procesarlos y mostrarlos en un pantalla tanto en el LCD local así como de enviarlos inalámbricamente por medio de Xbee para su representación gráfica.
- 5.- Pruebas finales del prototipo, utilizando la interfaz de usuario y una computadora personal.

### **1.5.- Justificación del proyecto.**

El agua potable es un factor de primer necesidad para el ser humano, algunas personas, satisfacen esta necesidad por medio de pozos, pero consideramos que la mayoría opta por el agua potable que proporciona el municipio o región debido a su fácil contratación, y calidad.

Sin tomar en cuenta las fugas, picaduras o corrosión que sufren las tuberías todo iría bien, sin embargo esto no siempre es posible, ya que siempre no faltan estos pequeños problemas, provocando la mala calidad en el servicio y obstruyendo la llegada del líquido hasta el cliente.

A veces no es fácil de hallar en donde precisamente esta la fuga, o esta tapada la tubería, y la empresa proveedor tiene que estar gastando en mano de obra, además que cada vez se rompe el pavimento de la calle debilitando de esta forma su resistencia en el tiempo haciendo esto impráctico y caro.

Esto se traduce en mal servicio para el cliente o perdida para el proveedor, por lo que vemos claramente la importancia de implementar un sistema más eficiente que beneficie tanto el proveedor como al consumidor en el cual se supriman estos inconvenientes.

## 1.6.- Alcances y limitaciones

El proyecto consiste en construir un sistema capaz de mostrar una luz de alerta cuando la suma de todas las salidas o tomas de agua potable sean inferiores al caudal de entrada en un red, así como de mostrar de forma gráfica los datos de diferentes en tiempos determinados por el administrador del sistema en diferentes puntos para una posterior comparación en caso de alguna anomalía dentro de la red hidráulica.

Esto es meramente una analogía, ya que se usara las ley de conservación de las masas de Lavoisier para la explicación del funcionamiento del sistema ya que sin considerar la presión u otros factores nos centramos en el caudal que es el producto de la velocidad por el área de la sección trasversal del tubo según la ecuación de continuidad que igualmente es análoga a la ley de la conservación de las masas. Mientras estas ecuaciones tienen razón de ser físicamente con la tubería la ley de corrientes de Kirchhoff estará presente en el circuito electrónico simulado en la pantalla de monitoreo en donde además de la gráfica y la luz de alarma estará también esta herramienta con el fin de dar información de forma visual simulando que por un elemento pasa mucha corriente cuando en realidad pasa mucha corriente del agua por la parte de la tubería análoga este circuito, esto último queda fuera de los fines de esta investigación ya que no se cuenta con demasiado tiempo para alcanzar tal objetivo.

Sin embargo, nuestra investigación se centra en el monitoreo de la salida como la entrada de caudal en la red hidráulica, mediante una gráfica diseñada con la herramienta de Matlab la cual se vale de un sistema de adquisición de datos, compuesto por un medidor digital capaz de enviar datos de manera inalámbrica para su tratamiento a distancia.

Estos medidores pueden también mostrar caudal en litros por minuto o m<sup>3</sup> totales en una pantalla LCD puesta en el medidor, además tiene la capacidad de funcionar con una pila de 9v y ser independientes de una fuente externa de alimentación.

Aunque en el marco teórico se da referencias a Arduino, los medidores no deben tener un esta placa dentro ya que estos son caros y muy grandes. Así pues se ha usado el mismo circuito integrado de Arduino uno (ATMEGA328P-PU) y el método que se explica en el anexo 4 para grabarle el código hexadecimal y ser montados en una base especialmente diseñada para los requerimientos del medidor adaptando las salidas del sensor de flujo y el módulo de Xbee, la ventaja es programar con Arduino los circuitos integrados ATMEGA328P-PU.

Para la prueba inicial del prototipo se construyó una red Hidráulica pequeña formada de tubos de PVC de media pulgada en un área de 1.5 metros cuadrados, y se grafican los datos recibidos inalámbricamente en una Laptop.

## **1.7.- Información general de la institución o empresa donde se desarrolló el proyecto.**

### **1.7.1 Historia del ITTG**

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública dependiente de la Secretaría de Educación Pública. Imparte 8 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas, que por su gran variedad de climas y suelos es propio para el cultivo de muy diversas especies vegetales nativas y adaptadas, con lo que se puede desarrollar la agroindustria, así como muchos otros procesos industriales a partir del gran potencial que ofrece este estado, en el cual el Instituto debe constituirse en el sujeto de cambio, al presentar alternativas de desarrollo sustentable del estado en forma multidisciplinaria.

Los principales laboratorios con los que cuenta el Instituto Tecnológico de Tuxtla Gutiérrez son:

- Electrónica
- Biotecnología
- Química
- Química pesada
- Mecánica
- Sistemas computacionales
- Ingeniería industrial
- Plantas piloto
- Polo Tecnológico Nacional
- Física
- Microbiología.
- Eléctrica
- Meca trónica

### **Proyectos de vinculación exitosos con la industria**

- Aprovechamiento integral del timbre
- Incremento en el contenido de sacarosa en la caña de azúcar
- Efectos del boro en la producción del cacahuete
- Factibilidad técnica en la extracción de aceites esenciales de la flora chiapaneca

## 1.7.2 Misión

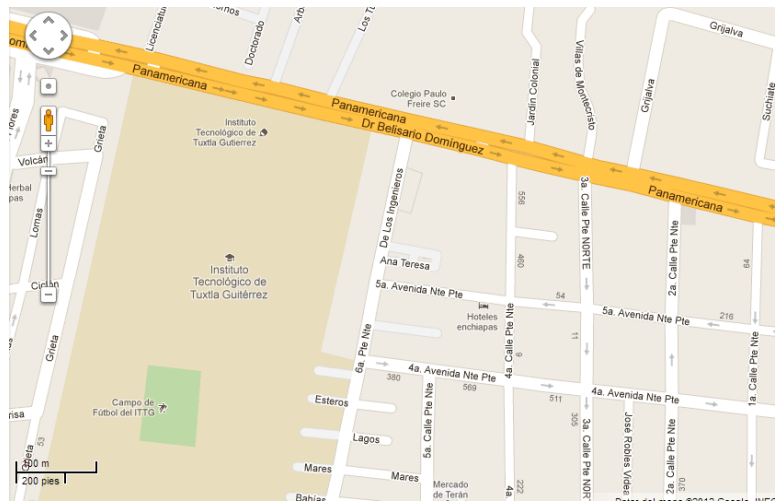
Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

## 1.7.3 Visión

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

## 1.7.4.- Localización

El instituto tecnológico de Tuxtla Gutiérrez, forma parte de la colonia Terán de la ciudad, tal como se muestra en el croquis de la figura 1.1 está a un costado de la carretera panamericana.



**Fig. 1.1.- Carretera Panamericana Km 1080. Tuxtla Gutiérrez Chiapas, C.P. 29050.**

## 1.7.5.- Área específica relacionada directamente con el proyecto

El laboratorio de Ingeniería Electrónica cuenta con 8 áreas destinadas a la investigación, desarrollo e implementación de proyectos escolares, 5 de las cuales son educativas, en las que los docentes imparten clases de: electrónica digital, PLC, electrónica de potencia, instrumentación, mediciones eléctricas, programación en c y electrónicas analógicas. Dos más son para investigación, una pertenece al grupo de IEEE y la otra es de desarrollo de circuitos electrónicos. La última área está destinada para que los alumnos desarrollen sus actividades en extra clase. Las áreas que se ocuparon en la realización de éste proyecto fueron las de IEEE, las de Electrónicas Analógicas y desarrollo de circuitos electrónicos.

## Capítulo 2. Fundamento teórico

### 2.1 Programación en Arduino.

#### 2.1.1 Arduino Uno

Arduino Uno es una placa electrónica basada en el microprocesador ATMEGA328P-PU. Cuenta con 14 pines digitales de entrada / salida (de los cuales 6 pueden ser utilizados como salidas PWM), 6 entradas analógicas, un resonador cerámico 16 MHz, una conexión USB, un conector de alimentación, un header ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador, simplemente conectarlo a un ordenador con un cable USB, o alimentarla con un adaptador de corriente AC a DC para empezar.

El IDE de código abierto puede ser descargado de forma gratuita (en la actualidad para Mac OS X, Windows y Linux).

#### 2.1.2.-Características:

Microcontrolador ATmega328  
Tensión de entrada 7-12V  
14 I/O Pines Digitales (6 salidas PWM)  
6 Entradas Analógicas  
32k de Memoria Flash  
Velocidad del Reloj 16Mhz

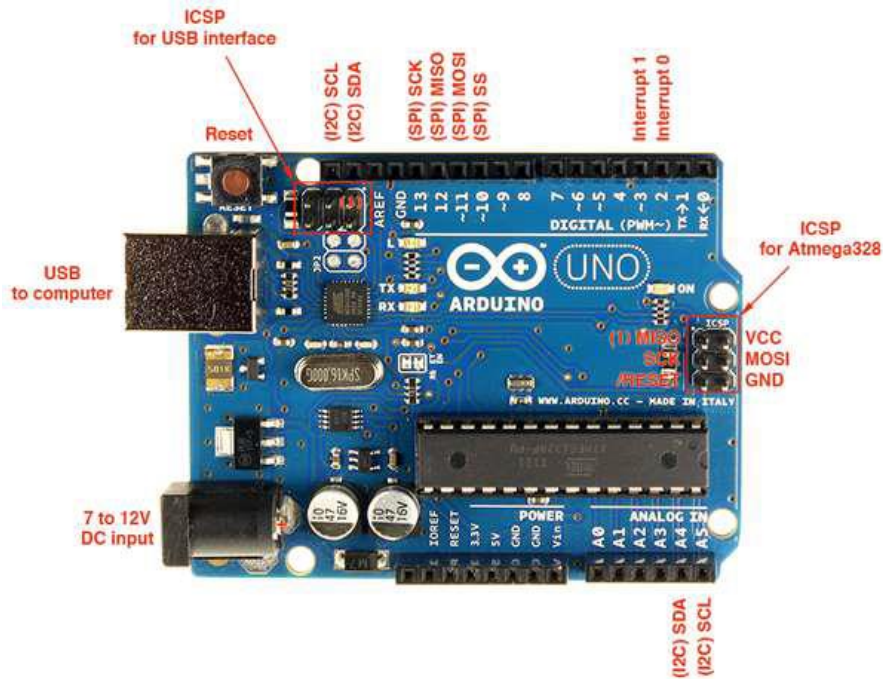


Fig. 2.1.- Partes básicas de una placa Arduino.

### 2.1.3.-Estructura básica de un programa

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes:

Setup  
Loop

Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una única vez y es empleada para configurar las terminales de salida mediante pinMode(). La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.).

```
void setup()  
{  
  //Preparación de programa  
}  
void loop()  
{  
  //Ejecución del programa  
}
```

**Nota:** Al igual que en C se pueden introducir bloques de comentarios con /\* ... \*/.

### 2.1.4.-Entradas/Salidas digitales y analógicas.

**digitalRead(pin).** Lee el valor desde un pin digital específico. Devuelve un valor HIGH o LOW.

**digitalWrite(pin, value).** Introduce un nivel alto (HIGH) o bajo (LOW) en el pin digital especificado. El pin puede ser especificado con una variable o una constante. Por ejemplo para encender el pin 2 de Arduino escribimos digitalWrite(2,HIGH) . Notando que **value** puede ser HIGH o LOW;

**analogRead(pin).** Lee el valor desde el pin analógico especificado con una resolución de 10 bits. Esta función solo funciona en los pines analógicos (0-5). El valor resultante es un entero de 0 a 1023. Los pines analógicos, a diferencia de los digitales no necesitan declararse previamente como INPUT o OUTPUT.

**analogWrite(pin, value).** Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida marcado como PWM. En el caso de Arduino uno los pines marcados con el símbolo (PWM ~) son 11, 10, 9, 6, 5, 3.



### 2.1.6.- Funciones de Tiempo

**delay(ms).** Realiza una pausa en el programa la cantidad de tiempo en milisegundos especificada en el parámetro (máximo 1000, mínimo 1).

**millis().** Devuelve la cantidad de milisegundos que lleva la placa Arduino ejecutando el programa actual como un valor long unsigned. Después de 9 horas el contador vuelve a 0.

### 2.2.- Modulo de Xbee.

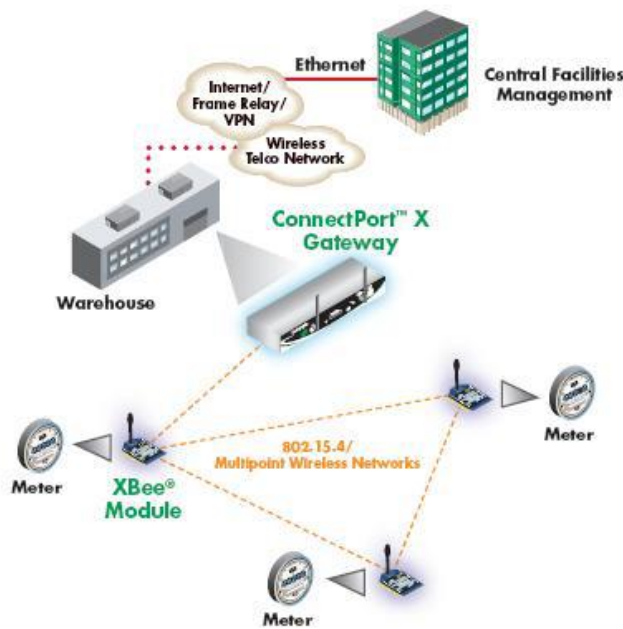
Zigbee es un protocolo de comunicaciones inalámbrico basado en el estándar de comunicaciones para redes inalámbricas IEEE\_802.15.4. Creado por Zigbee Alliance, una organización, teóricamente sin ánimo de lucro, de más de 200 grandes empresas (destacan Mitsubishi, Honeywell, Philips, \_ ODEM\_ do, Invensys, entre otras), muchas de ellas fabricantes de semiconductores. Zigbee permite que dispositivos electrónicos de bajo consumo puedan realizar sus comunicaciones inalámbricas. Es especialmente útil para redes de sensores en entornos industriales, médicos y, sobre todo, domóticas.

Las comunicaciones Zigbee se realizan en la banda libre de 2.4GHz. A diferencia de bluetooth, este protocolo no utiliza FHSS (Frequency hopping), sino que realiza las comunicaciones a través de una única frecuencia, es decir, de un canal. Normalmente puede escogerse un canal de entre 16 posibles. El alcance depende de la potencia de transmisión del dispositivo así como también del tipo de antenas utilizadas (cerámicas, dipolos, etc) El alcance normal con antena dipolo en línea vista es de aproximadamente (tomando como ejemplo el caso de MaxStream, en la versión de 1mW de potencia) de 100m y en interiores de unos 30m. La velocidad de transmisión de datos de una red Zigbee es de hasta 256kbps. Una red Zigbee la pueden formar, teóricamente, hasta 65535 equipos, es decir, el protocolo está preparado para poder controlar en la misma red esta cantidad enorme de dispositivos.

Entre las necesidades que satisface el módulo se encuentran:

- Bajo costo.
- Ultra-bajo consumo de potencia.
- Uso de bandas de radio libres y sin necesidad de licencias.
- Instalación barata y simple.
- Redes flexibles y extensibles.

El uso del protocolo Zigbee va desde reemplazar un cable por una comunicación serial inalámbrico, hasta el desarrollo de configuraciones punto a punto, multipunto, peer-to-peer (todos los nodos conectados entre sí) o redes complejas de sensores. Una conexión típica se muestra en la Figura 1-1, donde se observa que cada módulo Xbee posee algún tipo de sensor, el cual entrega los datos para ser enviados a través de la red a un Centro que administre la información.



**Fig.2.2.- Conexión típica usando Xbee**

Una red Zigbee la forman básicamente 3 tipos de elementos. Un único dispositivo coordinador, dispositivos Routers y dispositivos finales (end points).

### **2.2.1.-El Coordinador.**

El nodo de la red que tiene la única función de formar una red. Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el Coordinador puede formar una red, permitiendo unirse a él a dispositivos Routers y End Points. Una vez formada la red, el Coordinador hace las funciones de Router, esto es, participar en el enrutamiento de paquetes y ser origen y/o destinatario de información.

### 2.2.2.-Los Routers.

Es un nodo que crea y mantiene información sobre la red para determinar la mejor ruta para transmitir un paquete de información. Lógicamente un router debe unirse a una red Zigbee antes de poder actuar como Router retransmitiendo paquetes de otros routers o de End points.

### 2.2.3.-End Device.

Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador o un Router, es decir, no puede enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento

### 2.2.4.-Aplicaciones

Cada módulo Zigbee, al igual que ocurre con las direcciones MAC de los dispositivos ethernet, tiene una dirección única. En el caso de los módulos Zigbee cada uno de ellos tiene una dirección única de 64bits que viene grabada de fábrica. Por otro lado, la red Zigbee, utiliza para sus algoritmos de ruteo direcciones de 16 bits. Cada vez que un dispositivo se asocia a una red Zigbee, el Coordinador al cual se asocia le asigna una dirección única en toda la red de 16bits.

Por eso el número máximo teórico de elementos que puede haber en una red Zigbee es de  $2^{16} = 65535$ , que es el nº máximo de direcciones de red que se pueden asignar.

Estos módulos Xbee, pueden ser ajustados para usarse en redes de configuración punto a- punto, punto-a-multipunto o peer-to-peer. Un ejemplo se muestra en la Figura 2.2, donde se muestra una conexión multipunto, con un coordinador, conectado a varios nodos. Digi 2 tiene 2 series de módulos disponibles. Los módulos Xbee Serie 2 permite hacer redes mesh, La serie 1 no.

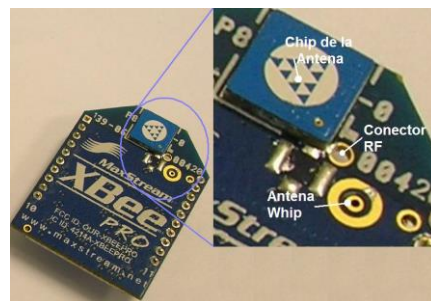


Fig.2.3.- Se observan los elementos del Xbee.

También existen los llamados módulos Xbee PRO de la Serie 1 que se diferencian en la capacidad de alcance, permitiendo en algunos casos doblar la distancia de transmisión, ya que poseen una mayor potencia en la señal. Con los módulos Xbee PRO de la Serie 2, es posible crear redes más complejas, como las llamadas MESH. Estas permiten acceder a un punto remoto, utilizando módulos intermedios para llegar como routers. Además los módulos automáticamente generaran la red entre ellos, sin intervención humana alguna, permitiendo la reparación de la red en caso de que algún nodo falle. Al mismo tiempo la red por sí sola resuelve la mejor ruta para un determinado paquete.

### 2.2.5.- Circuito básico para el Xbee.

La Figura 2.4 muestra las conexiones mínimas que necesita el módulo Xbee para poder ser utilizado. Luego de esto, se debe configurar según el modo de operación que se desea para la aplicación requerida por el usuario.

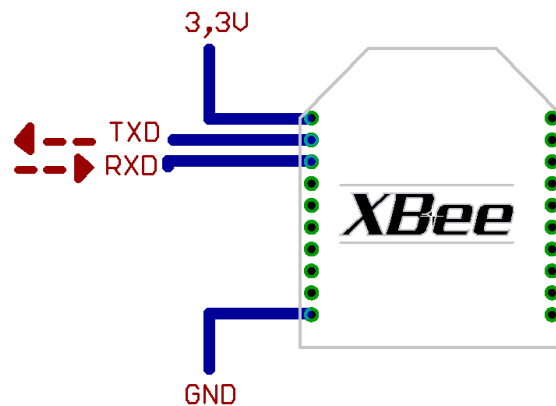


Fig.2.4.- Conexiones mínimas requeridas para el Xbee.

El módulo requiere una alimentación desde 2.8 a 3.4 V, la conexión a tierra y las líneas de transmisión de datos por medio del UART (TXD y RXD) para comunicarse con un microcontrolador, o directamente a un puerto serial utilizando algún convertor adecuado para los niveles de voltaje.

Esta configuración, no permite el uso de Control de Flujo (RTS & CTS), por lo que ésta opción debe estar desactivada en el terminal y en el módulo XBEE. En caso de que se envíe una gran cantidad de información, el buffer del módulo se puede sobrepasar. Para evitar existen dos alternativas:

- Bajar la tasa de transmisión
- Activar el control de flujo.

## **2.2.6.- Configuración del Módulo Xbee.**

En los siguientes subtemas se explicara los tipos de redes que soporta el módulo XBee, así como las configuraciones necesarias. Además se mostrarán algunos ejemplos de topologías de redes.

Para que los cambios realizados tengan efecto se debe ingresar ATCN (sale del modo de comandos) o ATAC (aplica los cambios inmediatamente).

## **2.2.7.-Direccionamiento de los módulos**

Los módulos permiten 2 tipos de direccionamiento. La de 16 bit y la de 64 bits. La principal diferencia es que en la de 64 bit, es posible obtener una mayor cantidad de direcciones y por lo tanto, una mayor cantidad de nodos o equipos funcionando en la misma red. Son a través de estas direcciones que los módulos se comunican entre sí.

### **2.2.8.- Direccionamiento de 16 bit**

El comando MY, define un número de 16 bit como dirección del módulo dentro de la red. El rango se encuentra entre 0x0 y 0xFFFFE (la dirección 0xFFFF y 0xFFFFE son para habilitar la dirección de 64-bit, por lo que si se desea utilizar direccionamiento de 16 bits, estos valores no deben ser usados). Para definirla se ingresa ATMY y el número en formato hexadecimal, pero sin el „0x“. Por ejemplo si a un módulo se le quiere asignar la dirección 0x3BF1 (15345 en decimal), entonces se debe ingresar el comando ATMY3BF1.

El comando DL, permite definir un número de 16 bit como dirección del módulo de destino dentro de la red al cual se va a realizar la comunicación. El rango debe estar entre 0x0 y 0xFFFFE (las direcciones 0xFFFFE y 0xFFFF se utilizan para direccionamiento de 64 bits).

Así para habilitar el direccionamiento de 16 bit, se debe utilizar una dirección menor a 0xFFFFE con el comando MY, de igual modo para DL y se debe dejar en cero el comando DH=0 (ATDH0). No se permite usar la dirección 0xFFFFE ni 0xFFFFE para el direccionamiento de 16.

### **2.2.9 Direccionamiento de 64 bits**

El número 0xFFFF y 0xFFFFE del comando MY, se usa cuando se desea desactivar el direccionamiento de 16 bit, y se habilita el uso de la dirección de 64 bit. Con este direccionamiento ya no es posible definir la dirección de origen del módulo, ya que ésta se asigna automáticamente. En este caso, la dirección del módulo corresponde a su número serial, que viene de fábrica y el cual es imposible de cambiar. Este número se encuentra guardado en dos variables de 32 bit cada una (SL y SH) y es único. SL lee los 32 bit menos significativos del número serial y SH los 32 más significativos.

Cuando se utiliza direccionamiento de 64 bit, para asignar una dirección de destino, se utilizan los comandos DL y DH. Éstos son de 32 bit cada uno (para el direccionamiento de 16 bit, DL se maneja como uno de 16, mientras que DH se mantiene en cero) y juntos (DL+DH) forman el número de 64 bit que debe corresponder con el número serial de otro módulo formado por SL+SH. Así para algún dato, DL debe ser igual a SL y DH debe ser igual a SH, donde SL+SH corresponden al número serial de un módulo destino configurado para direccionamiento de 64 bits.

Para el direccionamiento de 64 bit, se debe dejar MY como 0xFFFF (ATMYFFFF) o 0xFFFE (ATMYFFFE) y elegir una dirección de destino usando DL+DH, que debe corresponder a una dirección de 64 bit de otro módulo, indicando su número serial dado por SL+SH. Para consultar este número se debe ingresar ATSL (32 bit menos significativos) y luego ATSH (32 bit más significativos), entregando como respuesta los números seriales en formato hexadecimal.

### 2.2.10.-Punto a Punto.

Es la conexión ideal para reemplazar comunicación serial por un cable. Sólo se debe configurar la dirección. Para ello se utilizan los comandos MY y el DL. La idea, es que se define arbitrariamente una dirección para un módulo, usando el comando MY, el cual se va a comunicar con otro que tiene la dirección DL, también definida arbitrariamente. Con esto cada módulo define su dirección con MY, y escribe la dirección del módulo al cual se desea conectar usando DL.

En este modo, el módulo receptor del mensaje envía un paquete al módulo de origen llamado ACK (viene de Acknowledge) que indica que el mensaje se recibió correctamente.

En la figura 2.5 se muestra un pequeño ejemplo donde las direcciones se eligieron arbitrariamente: En la siguiente figura se muestra un pequeño ejemplo donde las direcciones se eligieron arbitrariamente:

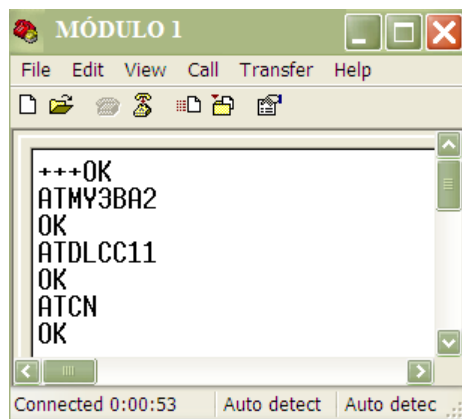


Fig. 2.5.- Configuración módulo 1.

Se observa que en el módulo 1, se ajustó la dirección de origen como 0x3BA2 (ATMY3BA2), mientras que la dirección de destino se asignó como 0xCC11 (ATDLCC11) que corresponde al módulo 2. En el otro módulo se observa lo siguiente:

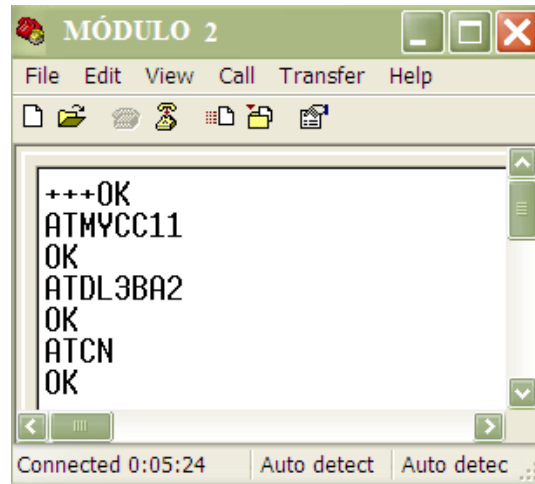


Fig. 2.6.- Configuración módulo 2.

En este módulo se asignó como dirección de origen 0xCC11 (ATMYCC11) y como dirección de destino 0x3BA2 (ATDL3BA2) que corresponde al módulo 1. La figura 2.7 muestra un ejemplo gráfico de lo anterior.

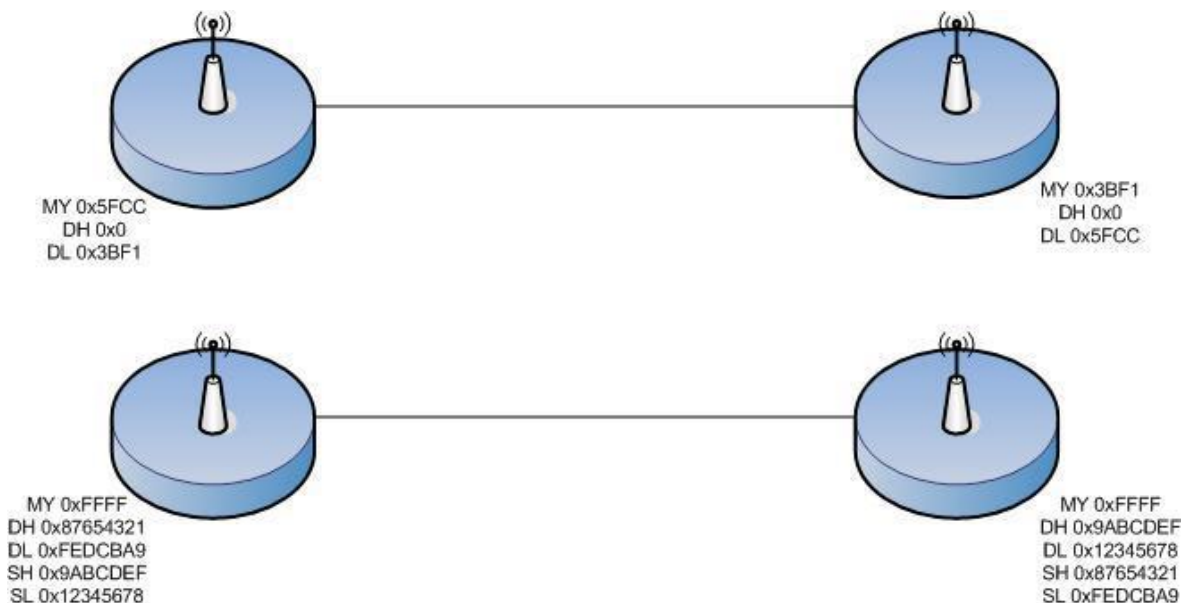


Fig. 2.7.- Ejemplo direccionamiento 16 y 64 bit.

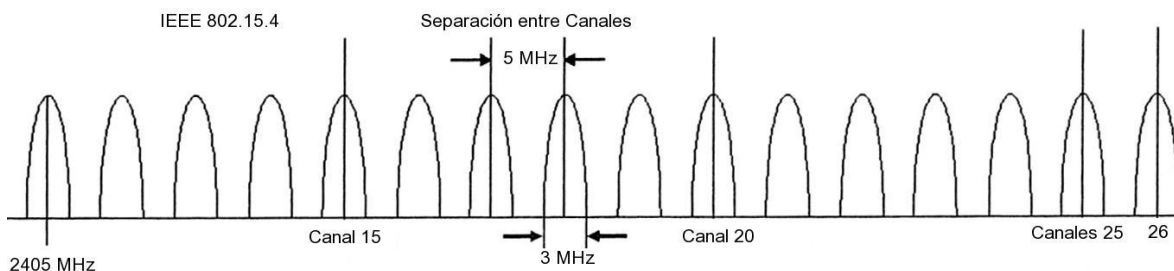
En la figura 2.7 se muestra que la primera conexión es una punto a punto utilizando direccionamiento de 16 bit, mientras que la segunda utiliza direccionamiento de 64 bits. Una vez configurado, el módem se encuentra listo para funcionar. Así todo lo que se transmite por el pin DIN de un módulo, es recibido por el pin DOUT del otro. Para que el modo Punto a Punto funcione, los módulos deben pertenecer a la misma PAN ID y al mismo canal. Más adelante se explica cómo configurar la PAN ID y el Canal.

### 2.2.11.- Punto a Multipunto.

Esta conexión, permite prestaciones extras. Se diferencia del Broadcast, en que permite transmitir información, desde la entrada serial de un módulo (DIN, pin 3) a uno o varios módulos conectados a la misma red de manera más controlada, ya que se necesitan las direcciones de los otros módulos, por lo que existe mayor seguridad. Para esto se necesitan dos comandos más aparte de MY y DL. Se utilizará el direccionamiento de 16 bits.

El primer comando es el ID de la PAN (Personal Area Network- Red de Área Personal). Todos los módulos que tengan idéntico PAN ID, pertenecerán a la misma red. El comando para configurar este valor es ID, es decir, ATID, y su rango va entre 0x0 y 0xFFFF. Por ejemplo si queremos ajustar el PAN ID como 0x3332, se debe ingresar ATID3332. Este parámetro también es arbitrario, al igual que MY y DL.

El otro comando corresponde al canal por el cual se va a comunicar. Según la Figura 10, se disponen de 16 canales según el protocolo IEEE 802.15.4. Esta norma indica que entre cada canal, deben existir 5 MHz de diferencia, partiendo de la frecuencia base 2.405 GHz, se llegan hasta los 2.480 GHz.



**Fig. 2.8.- Canales disponibles para el protocolo IEEE 802.15.4**



Se observa que hay 16 canales disponibles, sin embargo, los valores se asignan desde el 11 hasta el 26. Para calcular la frecuencia central se utiliza la siguiente fórmula:

[GHz] 005.0) 11(405 .2CH Canal Donde CH equivale al número del canal entre 11 y 26. Así para cambiar de canal se utiliza el comando CH con el número de canal en formato hexadecimal. Es decir, si se desea ocupar el canal 15 (0x10), se ingresa ATCH10. La siguiente tabla muestra la frecuencia central de cada canal, así como su límite inferior y superior: Frecuencia [GHz].

Tabla 2.1.- Características de las frecuencias.

<b>Canal</b>	<b>Hexadecimal</b>	<b>Inferior</b>	<b>Central</b>	<b>Superior</b>	<b>Comando AT</b>
<b>11</b>	0x0B	2,4025	2,4050	2,4075	ATCH0B
<b>12</b>	0x0C	2,4075	2,4100	2,4125	ATCH0C
<b>13</b>	0x0D	2,4125	2,4150	2,4175	ATCH0D
<b>14</b>	0x0E	2,4175	2,4200	2,4225	ATCH0E
<b>15</b>	0x0F	2,4225	2,4250	2,4275	ATCH0F
<b>15</b>	0x10	2,4225	2,4250	2,4275	ATCH10
<b>17</b>	0x11	2,4325	2,4350	2,4375	ATCH11
<b>18</b>	0x12	2,4375	2,4400	2,4425	ATCH12
<b>19</b>	0x13	2,4425	2,4450	2,4475	ATCH13
<b>20</b>	0x14	2,4475	2,4500	2,4525	ATCH14
<b>21</b>	0x15	2,4525	2,4550	2,4575	ATCH15
<b>22</b>	0x16	2,4575	2,4600	2,4625	ATCH16
<b>23</b>	0x17	2,4625	2,4650	2,4675	ATCH17
<b>24</b>	0x18	2,4675	2,4700	2,4725	ATCH18
<b>25</b>	0x19	2,4725	2,4750	2,4775	ATCH19
<b>26</b>	0x1A	2,4775	2,4800	2,4825	ATCH1A
<b>Frecuencia Base</b>		<b>2,405</b>			<b>GHz</b>

## 2.3.- ¿Qué es Matlab?

MATLAB es una de las muchas sofisticadas herramientas de computación disponibles en el comercio para resolver problemas de matemáticas, tales como Maple, Mathematica y MathCad. A pesar de lo que afirman sus defensores, ninguna de ellas es “la mejor”. Todas tienen fortalezas y debilidades. Cada una permitirá efectuar cálculos matemáticos básicos, pero difieren en el modo como manejan los cálculos simbólicos y procesos matemáticos más complicados, como la manipulación de matrices. Por ejemplo, MATLAB es superior en los cálculos que involucran matrices, mientras que Maple lo supera en los cálculos simbólicos. El nombre mismo de MATLAB es una abreviatura de Matrix Laboratory, laboratorio matricial. En un nivel fundamental, se puede pensar que estos programas son sofisticadas calculadoras con base en una computadora. Son capaces de realizar las mismas funciones que una calculadora científica, y muchas más. Si usted tiene una computadora en su escritorio, descubrirá que usará MATLAB en lugar de su calculadora incluso para la más simple de sus aplicaciones matemáticas, por ejemplo para el balance de su chequera. En muchas clases de ingeniería, la realización de cálculos con un programa de computación matemático como MATLAB sustituye la programación de computadoras más tradicional. Esto no significa que el lector no deba aprender un lenguaje de alto nivel como C++ o FORTRAN, sino que los programas como MATLAB se han convertido en una herramienta estándar para ingenieros y científicos.

### 2.3.1.-Interfaz gráfica de usuario en Matlab (GUIDE).

GUIDE es un entorno de programación visual disponible en Matlab para realizar y ejecutar programas que necesiten ingreso continuo de datos. Tiene las características básicas de todos los programas visuales como Visual Basic o Visual C++.

### 2.3.2.-Como crear una GUIDE.

Para iniciar nuestro proyecto, lo podemos hacer de dos maneras:

- Ejecutando la siguiente instrucción en la ventana de comandos:  
>> guide
- Haciendo un clic en el ícono que muestra la figura:

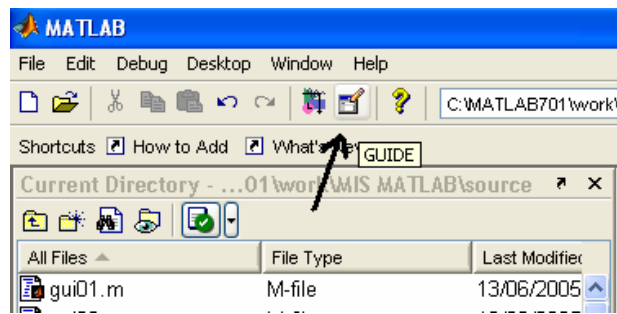


Fig. 2.9. Ícono GUIDE.

Se presenta el siguiente cuadro de diálogo:

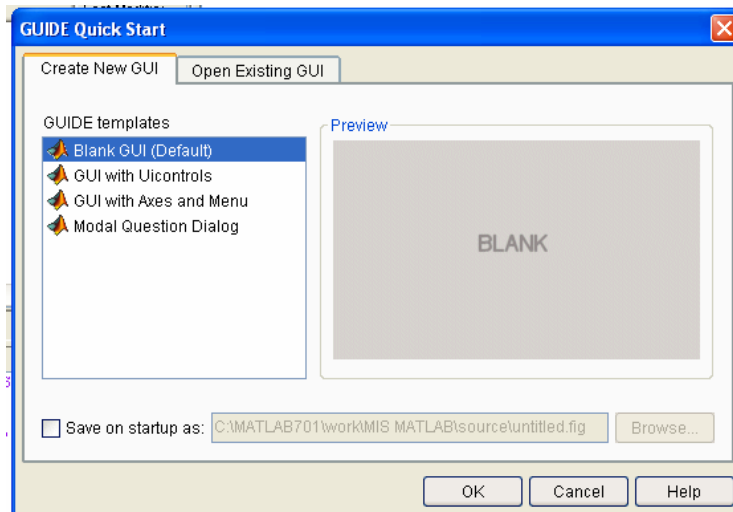


Fig. 2.10. Ventana de inicio de GUI.

Se presentan las siguientes opciones:

**a) Blank GUI (Default)**

La opción de interfaz gráfica de usuario en blanco (viene predeterminada), nos presenta un formulario nuevo, en el cual podemos diseñar nuestro programa.

**b) GUI with Uicontrols**

Esta opción presenta un ejemplo en el cual se calcula la masa, dada la densidad y el volumen, en alguno de los dos sistemas de unidades. Podemos ejecutar este ejemplo y obtener resultados.

**c) GUI with Axes and Menu**

Esta opción es otro ejemplo el cual contiene el menú File con las opciones Open, Print y Close. En el formulario tiene un Popup menu, un push button y un objeto Axes, podemos ejecutar el programa eligiendo alguna de las seis opciones que se encuentran en el menú despegable y haciendo clic en el botón de comando.

**d) Modal Question Dialog**

Con esta opción se muestra en la pantalla un cuadro de diálogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No, dependiendo del botón que se presione, el GUI retorna el texto seleccionado (la cadena de caracteres 'Yes' o 'No').

Elegimos la primera opción, Blank GUI, y tenemos:



Fig. 2.11. Entorno de diseño de GUI

Asimismo, tenemos las siguientes herramientas:

	Alinear objetos
	Editor de menú
	Editor de orden de etiqueta
	Editor del M-file
	Propiedades
	Navegador
	Grabar y ejecutar

Fig. 2.12.- Nombres de algunos iconos de la GUI.

Para obtener la etiqueta de cada elemento de la paleta de componentes ejecutamos: **File>>Preferentes** y seleccionamos **Show names in component palette**. Y tenemos la siguiente presentación:

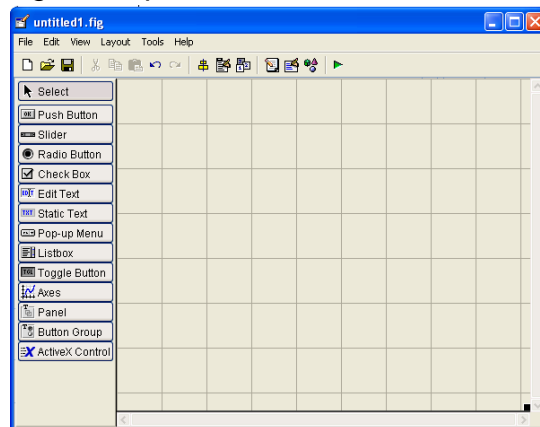


Fig. 2.13.- Entorno de diseño: componentes etiquetados.

### 2.3.3.-Propiedades de los componentes

Cada uno de los elementos de GUI, tiene un conjunto de opciones que podemos acceder con clic derecho.

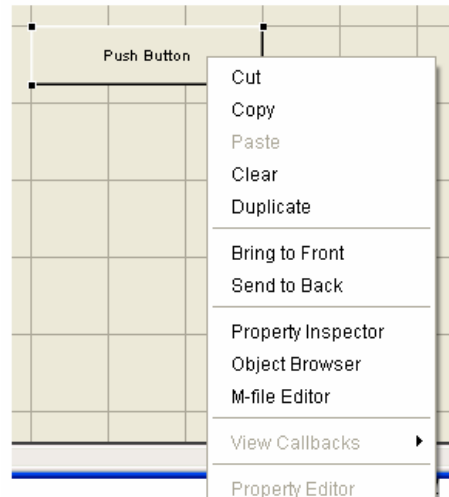


Fig. 2.14.- Opciones del componente.

La opción **Property** Inspector nos permite personalizar cada elemento.

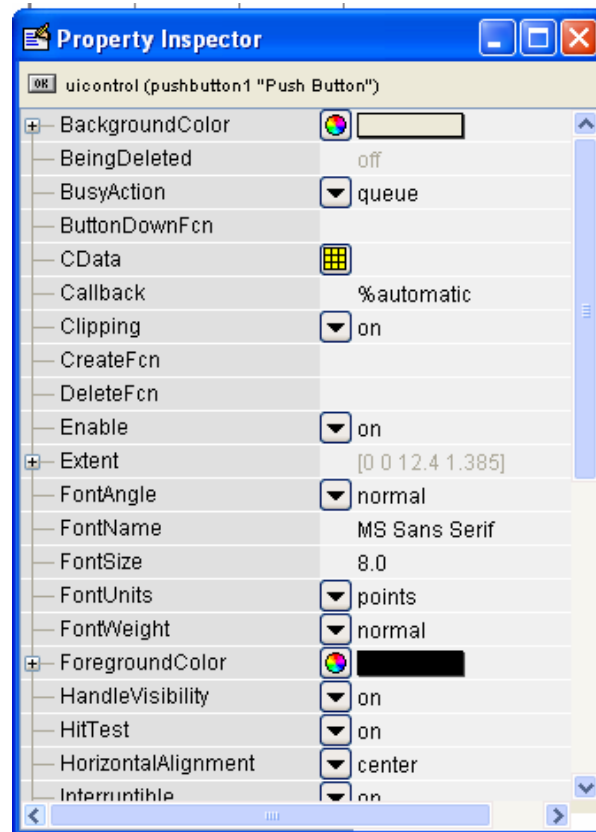


Fig. 2.15. Entorno Property Inspector.

Al hacer clic derecho en el elemento ubicado en el área de diseño, una de las opciones más importantes es View Callbacks, la cual, al ejecutarla, abre el archivo .m asociado a nuestro diseño y nos posiciona en la parte del programa que corresponde a la subrutina que se ejecutará cuando se realice una determinada acción sobre el elemento que estamos editando.

Por ejemplo, al ejecutar View Callbacks>>Callbacks en el Push Button, nos ubicaremos en la parte del programa:

```
function pushbutton1_Callback(hObject, eventdata, handles)
%hObject handle to pushbutton1 (see GCBO)
%eventdata reserved-to be defined in a future version of MATLAB
%handles structure with handles and user data (see GUIDATA)
```

### **2.3.4.-Funcionamiento de una aplicación GUIDE.**

Una aplicación GUIDE consta de dos archivos: .m y .fig. El archivo .m es el ejecutable y el .fig la parte gráfica. Para ejecutar una Interfaz Gráfica, si la hemos etiquetado con el nombre curso.fig, simplemente ejecutamos en la ventana de comandos .

```
>> curso
```

### **2.3.5.-Manejo de datos entre los elementos de la aplicación y el archivo .m**

Todos los valores de las propiedades de los elementos (color, valor, posición, string...) y los valores de las variables transitorias del programa se almacenan en una estructura, los cuales son accedidos mediante un único y mismo puntero para todos estos. Tomando por ejemplo el programa listado anteriormente el puntero se asigna en:

handles.output = hObject; handles, es nuestro puntero a los datos de la aplicación. Esta definición de puntero es salvada con la siguiente instrucción:

guidata(hObject, handles); guidata, es la sentencia para salvar los datos de la aplicación.

Guidata es la función que guarda las variables y propiedades de los elementos en la estructura de datos de la aplicación, por lo tanto, como regla general, en cada subrutina se debe escribir en la última línea lo siguiente:

```
guidata(hObject,handles);
```

Esta sentencia nos garantiza que cualquier cambio o asignación de propiedades o variables quede almacenado.

Por ejemplo, si dentro de una subrutina una operación dio como resultado una variable utpl para poder utilizarla desde el programa u otra subrutina debemos salvarla de la siguiente manera:

```
handles.utpl=utpl;  
guidata(hObject,handles);
```

La primera línea crea la variable utpl a la estructura de datos de la aplicación apuntada por handles y la segunda graba el valor.

## Capítulo 3.- Desarrollo e implementación del proyecto

### 3.1.- Diseño del sistema de adquisición de datos.

#### 3.1.1.- Características del sensor YF-G21 (sensor de flujo)

La variable principal que el medidor digital transmitirá y procesará es el flujo en litros por minuto, de lo cual como elemento primario de considero, el sensor YF-G21 que es un sensor de flujo de agua el cual se compone de una válvula con cuerpo de plástico, un rotor de agua, y un sensor de efecto Hall. (Ver anexo 7). Cuando el agua fluye a través de los rodillos del rotor, el rotor gira. Su velocidad cambia con diferente tasa de flujo, por lo que si aumenta la velocidad del flujo, mayor será la frecuencia de los pulsos enviados por el sensor. El sensor de efecto Hall emite la señal de pulsos, en donde la frecuencia de los pulsos es directamente proporcional al flujo que pasa por él.

Así Frecuencia de pulsos  $\propto$  Flujo que pasa.



Fig.3.1.- Sensor de flujo YF-G21

Al caracterizar este sensor se descubrió que, requiere de una alimentación de 5V y nos regresa una señal cuadrada, con rango de 0 a 5 v por medio del cable amarillo, el cual es conectado al pin 5 del ATMEGA328P-PU que esta etiquetado como INT 1 (pd3) en la hoja de datos mostrada en anexo 1.



### 3.1.2.-Diagrama a bloques del proceso que sigue la información del sensor de flujo.

Para dar una explicación más clara de cómo se procesa la información proporcionada del sensor mostramos en la figura 3.2, un diagrama que muestra bloque a bloque como esta va llegando desde la tubería donde se encuentre el transductor hasta la computadora que recibe y grafica los datos.

Haciendo notar que sin la reducción de rango que muestra el bloque de acondicionamiento de señal esta llegara al módulo emisor x-bee con rango de 0 a 5, con lo cual el modulo tiende a quemarse.

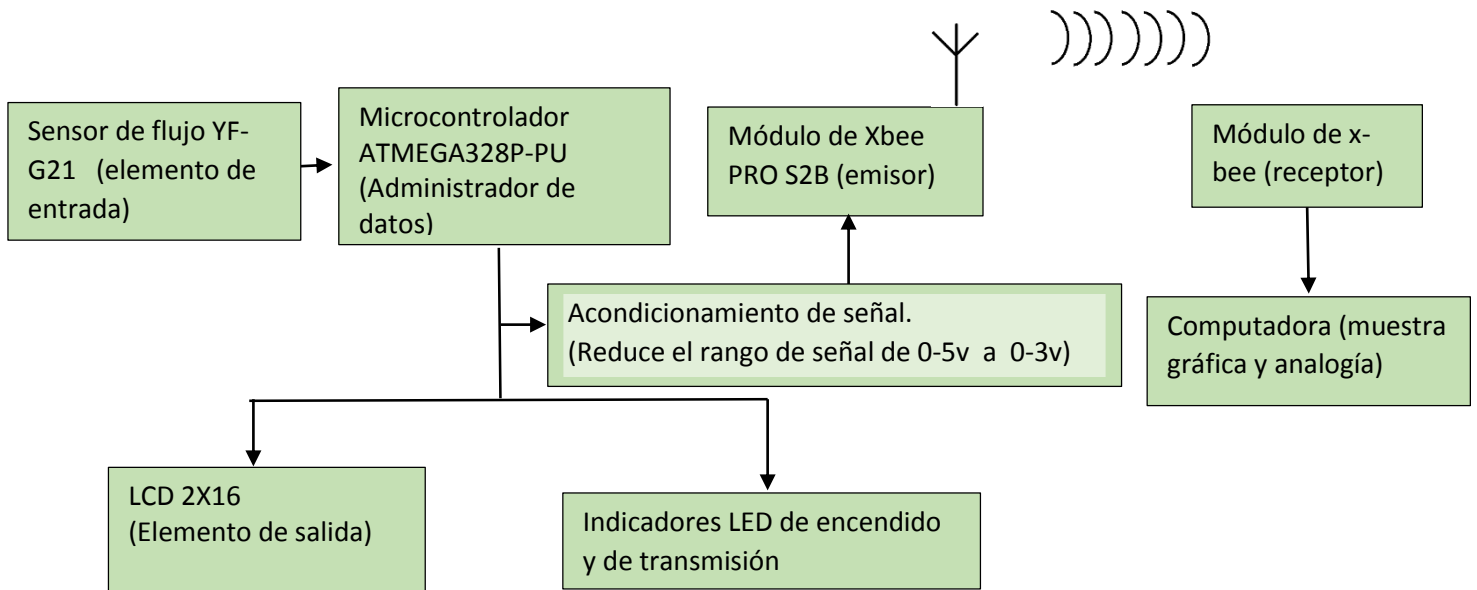


Fig. 3.2.- Diagrama a bloques del procesamiento de la información del sensor de flujo.

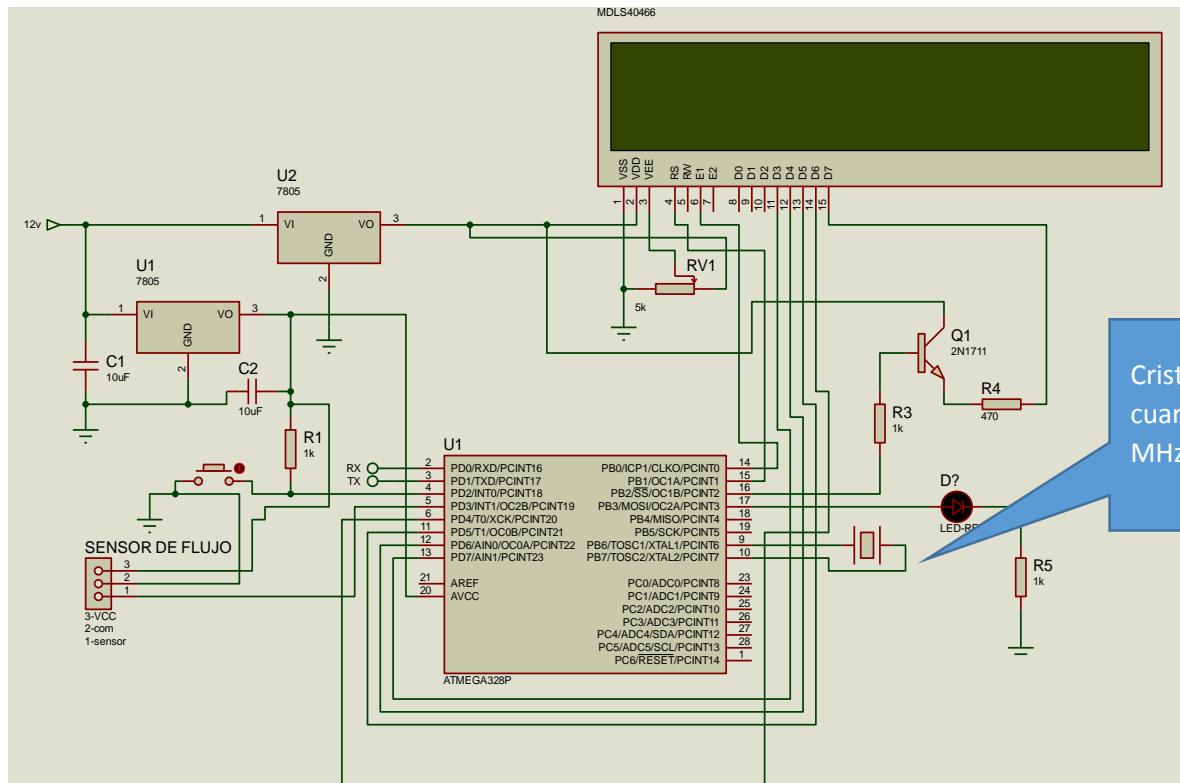
### 3.1.3.- Simulación del sistema de adquisición de datos.

La simulación mostrada en la figura xx muestra el diagrama de la placa 1 presentada en la figura 3.3; si vemos con atención el diagrama veremos que aparecen varios reguladores de voltaje, esto se hizo con el fin de garantizar la estabilización del sistema, así el LCD es alimentado por medio de un regulador 7805 exclusivo para él, así como también se asignó un regulador aparte para alimentar el sensor de flujo y el ATMEGA328P-PU.

Con esto logramos que ambos funcionen simultáneamente sin interferir en la estabilización electrónica del otro ya que en primera instancia se había conectado todo en un mismo regulador, pero el programa de ATMEGA328P-PU se descontrolaba.

Nótese que en la figura 3.3, la conexión del foco de iluminación del LCD esta por medio de un transistor 2N3904, esto con el fin de controlar el apagado y encendido de la luz de pantalla del LCD.

Para una mayor visualización de la conexión de ATMEGA328P-PU puede ver diagrama completo de Arduino UNO en el anexo 1 de esta investigación.



Cristal de cuarzo 16 MHz.

Fig. 3.3.- Diagrama de la placa 1 simulada con Proteus.

Por razones de espacio se pensó en hacer dos placas, la placa 1 y la placa dos la cual mostramos el diagrama esquemático en la figura 3.4 y su materialización en la figura 3.6 b, con el espacio dentro de la cojita que se logra separando el circuito en dos placas ha de haber una pila de 9v que alimentara todo el medidor digital de forma autónoma.

De la figura 4.3 notamos que posee dos reguladores el primero reduce de 12 a 9 volt y el que le sigue reduce de 9 a 3.3v exactos para el funcionamiento del módulo xbee, No esta demás decir que el regulador ams1117 tiene la capacidad de soportar los 12 v directamente, sin embargo el arreglo que se implementó con el regulador 7809 se hizo con el fin de evitar calentamiento excesivo en el primero.

Por otra parte hacemos notar que la conexión RX de ATMEGA328P-PU debe ir conectado a TX del módulo de xbee, aunque tal conexión no tiene función alguna en este proyecto, mientras que la conexión Tx de ATMEGA328P-PU debe ir conectado al pin 3 de Xbee que Rx. Una cuestión importante que queremos mencionar es el arreglo que se hizo para adaptar la señal cuadrada que envía ATMEGA328P-PU al módulo xbee emisor, tal arreglo tiene la simple función de reducir el rango de la señal, de 0 a 5 y dejarlo de 0 a 3.2 v aproximadamente, esto hace que el módulo xbee no se queme.

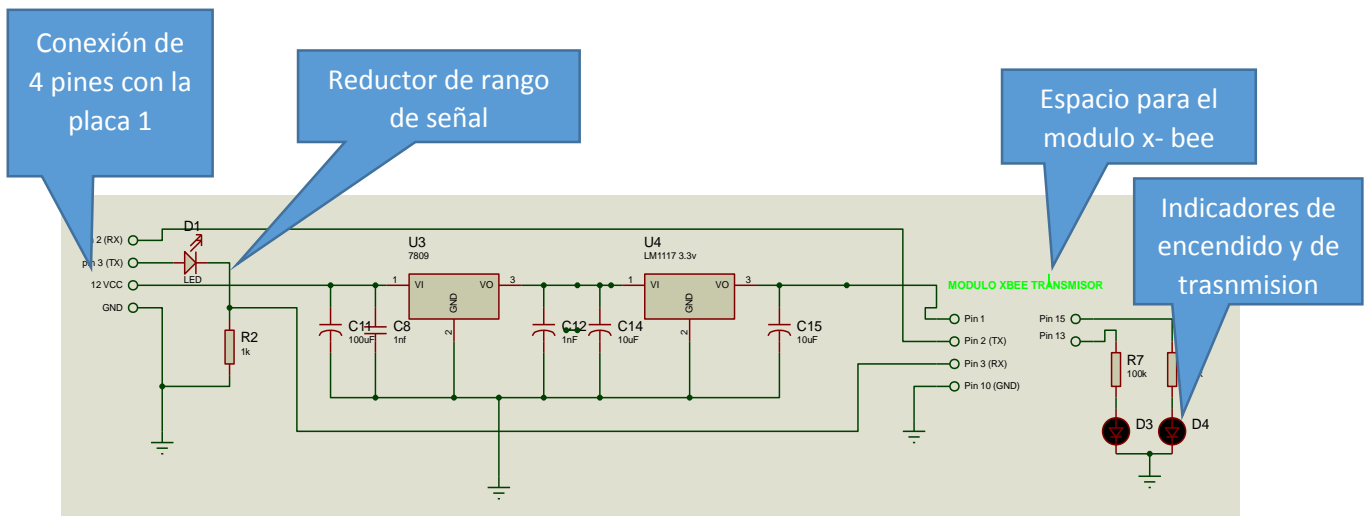
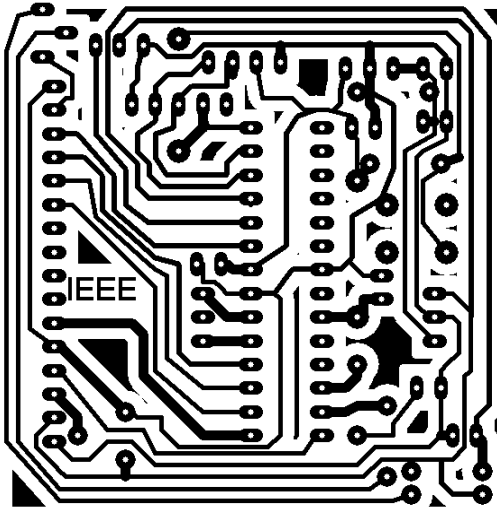


Fig. 3.4.- Diagrama de la placa 2, Simulada con Isis Proteus.

### 3.1.4.-Materialización del medidor digital.

Para la implementación del medidor digital se procedió a revisar el datasheet (hoja de datos) propio de Arduino uno, para montar ATMEGA328P-PU de forma independiente, (ver anexo 1) se procedió a realizar el rutado siguiente el cual obedece a la simulación hecha con el programa Isis Proteus que se explicó en el tema anterior.



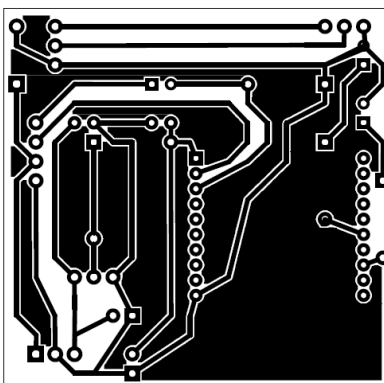
a)



b)

Fig.3.5.- Placa 1, a) Rutado en PCB Wizard , b) Materialización.

Una vez montado el circuito de la placa 1 y funcionando, se procedió a realizar placa 2 mostrada en la figura 3.6, que tendrá la posibilidad de ensamblarse a la placa 1. Y es en esta placa donde se instalará un zócalo de 20 pines para el módulo xbee, este diseño fue realizado con el programa de fritzing.



a)



b)

Fig.3.6.- Placa 2. a) Rutado en fritzing. b) Materialización.

### 3.1.5.-.-Tratamiento de la señal enviada de ATMEGA328P-PU al módulo Xbee.

Es importante remarcar que el módulo de Xbee es un tanto delicado en cuestiones de voltaje de alimentación ya que aunque este soporta un rango de operación tal como se comentó en el marco teórico este está entre 2.8 v a 3.3 v tan importante es esto como las señales de entrada estas no deben de ser mayores al voltaje de alimentación, por lo que debemos cuidar detalladamente este hecho, ya que el pasar por alto esta aclaración podría significar quemar el módulo de Xbee, para resolver este inconveniente se procedió a dar un pequeño tratamiento a la señal reduciendo su rango de operación, pasando de 0 - 5v a 0 – 3.2v, para lograrlo se implementó el circuito de abajo, en donde el pin 3 de ATMEGA328P-PU etiquetado en la hoja de datos como TX está conectado al ánodo de un LED color rojo de 5mm, el cual tiene conectado su cátodo al pin 3 del módulo xbee etiquetado como RX, pero tal como se ve en la figura 3.7 hay una resistencia de  $\frac{1}{2}$  watt, 1 k $\Omega$  en paralelo con la carga, en este caso el módulo de transmisión, y en serie con el Diodo emisor de luz tal resistor tiene la función de limitar la corriente.

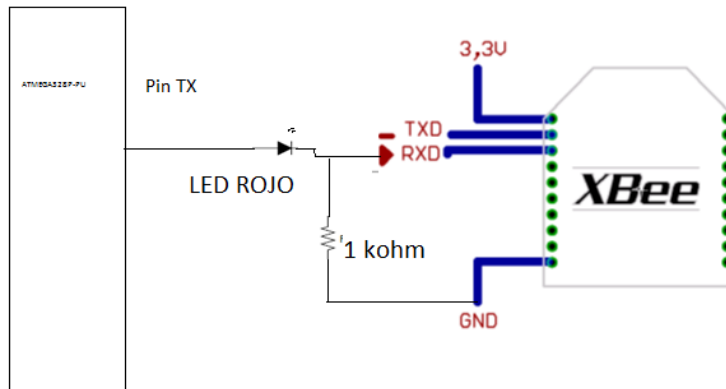


Fig.3.7.- Arreglo para darle tratamiento o la señal Tx de ATMEGA328P-PU para Xbee.

Según información descrita en el anexo 6, dice que el rango de caída de voltaje para un LED rojo esta entre 1.8v a 2.2v, este fue comprobado al medir la caída con un multímetro, para este caso el LED tenía una caída de 1.8 con lo que se procedió a realizar el siguiente calculo.

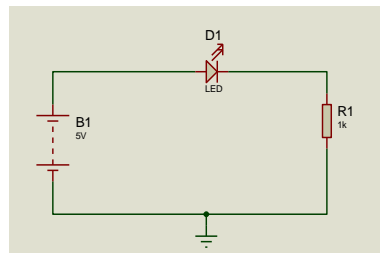


Fig. 3.8.- Diagrama usado en el cálculo.

Si consideramos el circuito de la figura 3.8 en donde la fuente de 5 volts es el pin 5 del micro controlador ATMEGA328-PU y calculamos la caída de voltaje en R1 hallaremos el nuevo rango de la señal, entonces.

$$V_T = V_D + V_{R1} \quad \text{Por ley de Kirchhoff}$$

Donde

$V_T =$  Voltaje total del entrada.

$V_D =$  Voltaje de diodo.

$V_{R1} =$  Voltaje de caída en la resistencia.

Considerando que  $V_D$  es 1.8 según la experimentación y el anexo 5, tenemos.

$$5v = 1.8v + I_T (1 \times 10^3 \Omega) \quad \text{al aplicar ley de ohm en } V_{R1}.$$

$$I_T = \frac{5v - 1.8v}{1 \times 10^3 \Omega} \quad \text{al resolver para } I_T \text{ (corriente total del circuito).}$$

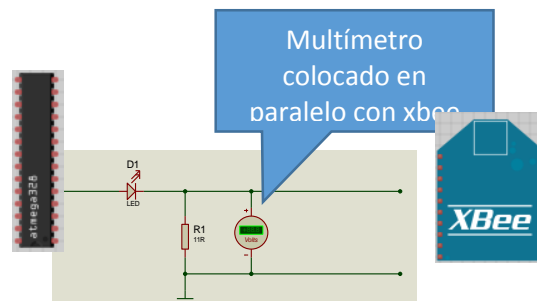
Por lo tanto  $I_T = 3.2 \text{ mA}$ .

Conociendo  $I_T$  resolvemos  $V_{R1}$ .

$$V_{R1} = (1 \times 10^3 \Omega) (3.2 \times 10^{-3} A)$$

Encontramos que  $V_{R1}$  es **3.2v** el cual es el nuevo valor máximo de la señal.

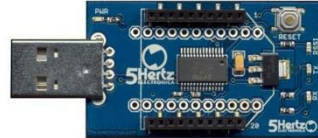
Y de esta forma si se procediera a medir el voltaje en la señal de entrada con el multímetro conectado como se muestra en la figura 3.9, y en el modo CD este marcaría 3.2 v como valor máximo y decimos máximo ya que estaría variando continuamente debido a que es una señal variante.



**Fig. 3.9.- Medición del rango de la señal.**

Recordando que una de los objetivos de este proyecto es hacer de este medidor digital más versátil, por lo que después de tener listas ambas placas se procedió a meter ambas en una cajita tal como se muestra en la figura 3.11.

Con esto el emisor del sistema está listo y enviando datos a un módulo xbee receptor que estará conectado al puerto de la computadora por medio de un adaptador y programador de xbee como el que se muestra en la figura 3.10, el cual se puede conseguir en mercado libre, y luego conectar a la computadora a la que nos servirá para graficar los datos recibidos en su puerto.



**Fig. 3.10.- Adaptador y programador Xbee.**

Una vez conectado dicho receptor se procedió a administrar los datos recibidos con Matlab programa que tiene la herramienta para procesar datos y mostrarlos en un grafica en tiempo real.

Sin embargo antes de recibir los datos en este receptor estos deben haber sido procesados dentro del ATMEGA328P-PU que se encuentra en el medidor ubicado en su respectivo nodo de la red hidráulica, el circuito integrado posee un programa que fue compilado con Arduino antes de ser trasladado a la placa 1, de la figura mm; la lógica de dicho programa se encuentra en el diagrama de flujo de la figura 3.5, y su tarea es censar cada segundo las interrupciones que hace el aspa del sensor de flujo YF-G21 guardarlo en una variable tipo volatile int ya que esta se actualiza cada segundo.

Tal proceso también cumple con la tarea de llevar la cuenta de cuantos metros cúbicos de agua se transferido al cliente así como también el costo que este va acumulando, datos que forman un total de tres, que son enviados al módulo x-Bee para que este lo envía de forma inalámbrica hasta el modulo receptor, este lo hará en forma de una cadena de 15 caracteres precedido por una presentación. La cadena se verá en el receptor como la siguiente.

### **A0.00V0.00F0.00**

Esta cadena de caracteres se repetirá a una frecuencia de uno por segundo haciendo la analogía A litros por segundo V metros cúbicos F el costo en pesos, así pues el programa que graficara estos datos no tendrá que hacer cálculos de transformación o calibración, ya que estos estarán determinado dentro de la cajita del medidor digital que se está desarrollando siguiendo el siguiente algoritmo.

1. Se crean las variables RPM, flujo y acumulado.
2. Se crea la rutina rpm, que se correrá cada vez que uno cualquiera de los brazos del aspa del sensor de una vuelta considerando que este tiene 8.
3. Dentro de la rutina RPM una variable flujo se incrementara en 1 por cada vez que se corra.
4. Los datos de la variable flujo se actualizan cada segundo.
5. Al hacer un análisis de la tabla mm se determina que litros por minuto es igual a  $\text{flujo}/7.5$ ;
6. Los litros por minuto se guardan en la variable acumulado.
7. Acumulado guarda el valor actual y suma los nuevos cada segundo dividiendo estos entre 1000 se obtiene un valor de m<sup>3</sup>.
8. Multiplicando acumulado por el costo del cada litro de agua se obtiene costo.

Una forma sencilla de saber que al dividir 1000 el valor de acumulado resulte el valor de metros cúbicos fue colocar nuestro medidor en serie con otro medidor analógico de flujo, en un tubo de PVC de ½ pulgada, bombear el agua y en un minuto ver cuantos litros tenia censado el analógico, con esto sabíamos que nuestro prototipo debería tener el mismo valor en la pantalla del LCD. Luego para saber cuántos M<sup>3</sup> de agua había usamos el dato que dice que por cada 10cm<sup>3</sup> de agua es un litro para hacer los cálculos de calibración necesarios.

En cuanto al costo que aquí se calcula, habría que saber en específico cual es el costo que tiene un litro de agua en la empresa proveedora ya que únicamente se multiplica el valor acumulado de litros de agua por el precio unitario del litro del producto dando el costo por mes o por el intervalo de tiempo que el proveedor considere necesario, observando que no hay de qué preocuparse por la cantidad de caracteres que contenga la variable acumulado ya que esta es de tipo float en el programa.





**Fig.3.11.- Parte interna del medidor digital**

### **3.1.6.- Diagrama de flujo del programa grabado en el micro controlador ATMEGA328P-PU**

La lógica del proceso de la información del sensor dentro del circuito integrado ATMEGA328P-PU mediante el programa mostrado en el anexo a, está dada en la figura 3.12, recordando que este programa fue hecho con Arduino, luego se procedió a cargarlo en la placa Arduino, para que con mucho cuidado sacar el circuito integrado y trasladarlo a la placa 1 de la figura 3.5.

En el cuadro comentado de la figura 3.12, como void setup es la parte del programa de Arduino, la cual solo se corre una vez, se enciende el foco del LCD y se muestra una presentación, después sigue el ciclo infinito, comentada como void setup, esta dara ciclos de 1 por segundo, por lo que el envío de la cadena de datos se enviara cada segundo así como el censado de las vueltas del sensor de flujo YF-G21.

Aparte este diagrama de flujo muestra como el botón de muestreo que posee la cajita del medidor funciona dentro de la lógica del programa, el cual tiene la capacidad de cambiar la forma en se muestran los datos en la pantalla LCD, ya sea metros cúbicos totales, o litros por minuto y costo.

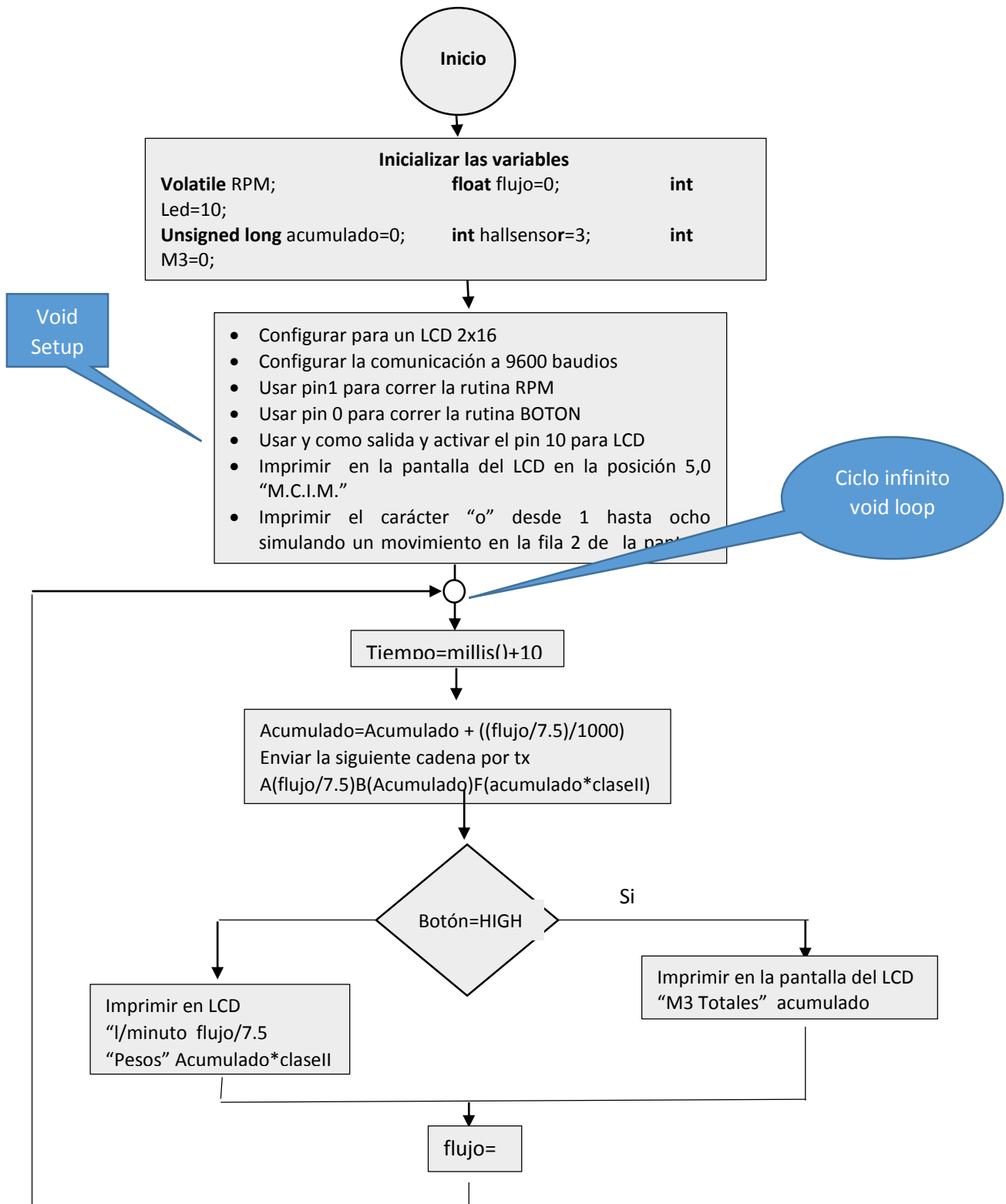


Fig. 3.12.- diagrama de flujo del programa, grabado en ATMEGA328P-PU

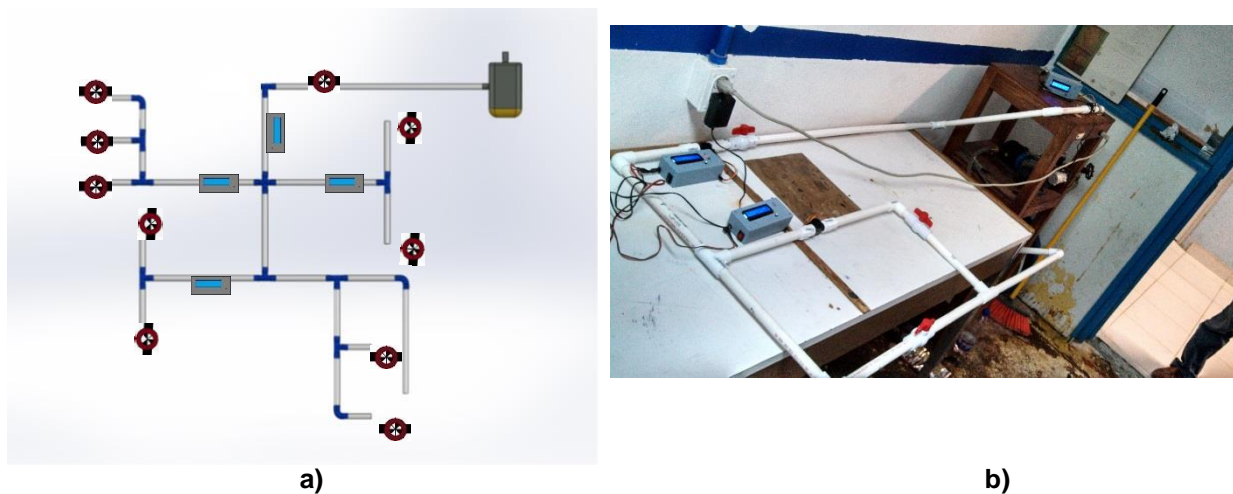
## 3.2. Implementación del sistema de adquisición de datos

### 3.2.1- Estructura de la tubería de pruebas del medidor digital.

Para realizar el diagrama de la figura 3.13b se usó tubo de PVC de media pulgada, En esta prueba se pretendía visualizar los datos enviados de del medidor digital desarrollado, y comparar el cambio del flujo al colocar los medidores en diferentes partes de la red.

Considerando que este prototipo está en su primera etapa, cabe aclarar que para esta prueba se experimentó únicamente con un medidor, sin olvidar nuestro objetivo de comunicar los xbee de forma punto multipunto, en el cual uno recibe y varios módulos envían.

Este medidor fue calibrado, colocándolo en serie con otro medidor analógico industrial, comparando datos e igualando la cantidad de flujo leída en ambos.



b) Fig. 3.13.- Tuberías a) Diagrama para etapa 2 b) diagrama etapa 1

### 3.2.2.-Caracterización de los medidores digitales desarrollados.

Con miras al objetivo principal se procedió a ensamblar más prototipos de los medidores de flujo digitales todos ellos con capacidad de enviar datos de forma inalámbrica a un solo receptor (ver figura 3.14) los cuales tiene una medida de 6cm X 12.5cm X 4cm.

Las partes específicas de tal prototipo se comentan en la figura 3.15 con el fin de dar una idea más clara de ellos.



Fig. 3.14.- Grupo de medidores digitales listos para ser usados.

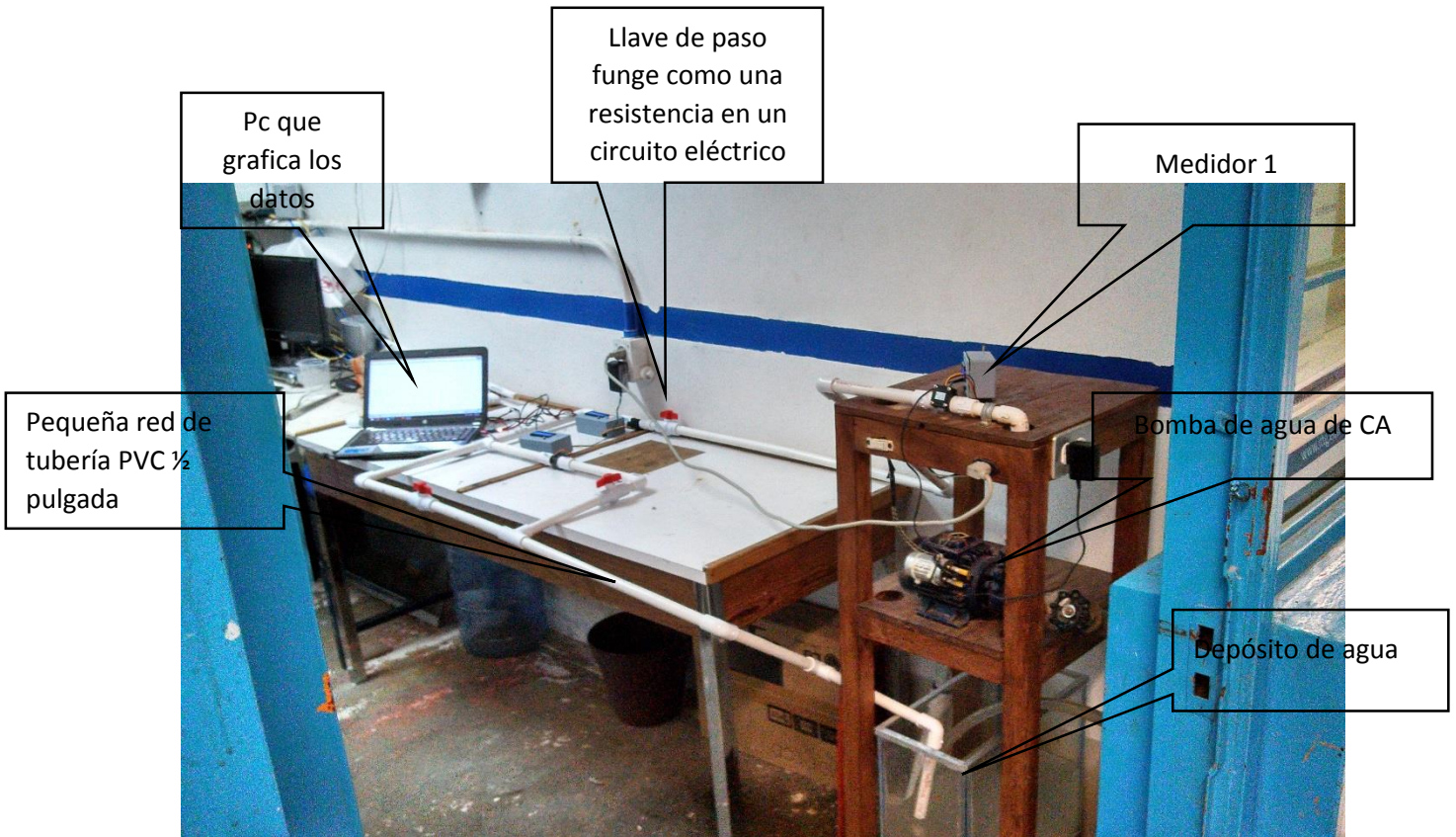


Fig.3.15.- Medidor desarrollado

Esto permitió hacer las pruebas con el prototipo, conectamos el medidor en una parte del circuito hidráulico y se procedió a bombear el agua con motor que en efecto el líquido siempre era el mismo ya que regresaba al recipiente que lo contenía inicialmente, esto permitió hacer el experimento con una mínima cantidad de agua.

Con una corriente de agua pasando por el sensor del medidor, este fue capaz de mostrar los datos en la pantalla del LCD tal como se pretendía, además que envió la cadena de datos deseada, la cual se muestra en la figura 3.18.





**Fig. 3.16.- Foto tomada en el cubículo IEEE, prueba del prototipo**

### **3.2.3.- Calibración del medidor.**

Fue necesario encontrar el coeficiente que multiplicaría la variable en el programa del medidor, para ello se llevó a cabo una serie de pruebas, con el medidor analógico (el de aguja y engranaje) y el que queremos implementar en serie en una pequeña red en donde una pequeña bomba suministra en flujo contante de agua.

Este proceso de calibración consistió en conectar el sensor del medidor digital que se está desarrollando en una tubería de PVC de  $\frac{1}{2}$  pulgada en serie con un medidor de caudal analógico. Con esta configuración se bombeo el agua y se procedió a recibirlo en la salida de tubo con una botella de refresco de dos litros y a tomar el tiempo con un cronometro, este procedimiento se hizo 14 veces seguidas de las cuales se llegó a la conclusión que en promedio la botella se llenaba en 12 segundos.

Una recopilación de los datos obtenidos se muestra en la tabla 3.1.

Tabla 3.1.-Recopilacion de datos de calibración.

No. Prueba	TIEMPO DE LLENADO EN UN BOTE DE 2 LITROS	
	TIEMPO EN SEGUNDOS	VUELTAS ANALOGICAS
1.-	30	2
2.-	17	2
3.-	18	2
4.-	13	2
5.-	13	2
6.-	13	2
7.-	12	2
8.-	12	2
9.-	12	2
10.-	14	2
11.-	14	2
12.-	13	2
13.-	11	2
14.-	12	2

Marcamos las filas en donde el bote se llenó en 12 segundos ya que fue el dato que se eligió debido a que este fue el que más se repitió en la práctica, y la información que nos queda de esto es que pasan dos litros por cada 12 segundos bajo la presión que la bomba ejerce en el tubo de PVC de ½ pulgada.

Con el dato de 12 segundos se emplean para pasar dos litros de agua, la interrogante es ahora, cuantos litros pasan en 60 segundos, para la solución de esta pregunta se hizo una regla de tres. De la cual se obtuvo.

12 segundos → 2 litros

60 segundos → x litros.

$$X \text{ litros} = \frac{2 \text{ litros} \times 60 \text{ segundos}}{12 \text{ segundos}}$$

X litros = 10 litros.

Entonces en 60 segundos (un minuto) pasan 10 litros de agua bajo las condiciones que el tubo y la bomba mantiene en la corriente de agua. Lo que nos indicaba que debíamos hacer que bajo estas condiciones de la corriente de

agua, nuestro medidor de flujo debía marcar 10 litros/minuto. Para estar igualmente calibrado que el medidor de flujo analógico industrial.

### 3.2.4.- Diseño de la gráfica de Matlab

Considerando que el proyecto se encuentra en su primera etapa se recurrió a realizar la gráfica de la cadena de datos que envía un medidor digital, mientras el prototipo estaba funcionando, fue posible visualizar los datos mediante el monitor serial de Arduino, con el programador conectado al puerto serie. Que por defecto este estaba configurado para el puerto COM10.

### 3.2.5.-Recibiendo datos en el puerto serial COM10.

Ya que el medidor envía la señal al receptor Xbee previamente configurado, los datos ahora se encuentra en el puerto serial de la pc.

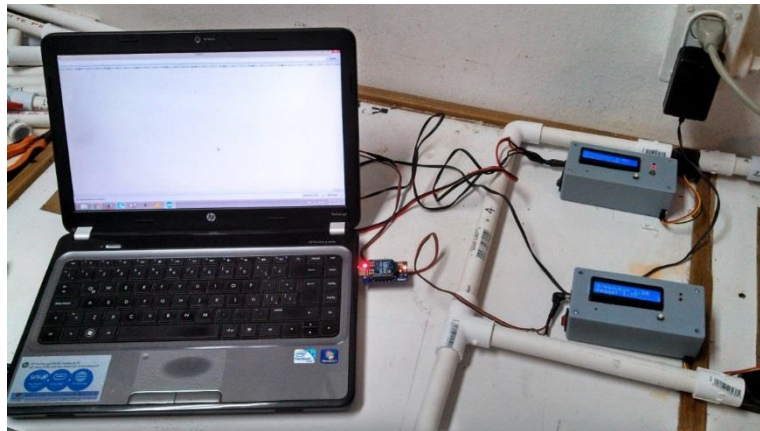


Fig. 3.17.- Recibiendo los datos en el puerto serial.

Tal como se muestra en la figura 3.18; de esta forma los datos pueden verse fácilmente desde el monitor serial de Arduino, los datos vistos en pantalla son una cadena de datos enviados un renglón cada segundo, ya que el programa emisor así se ha configurado.

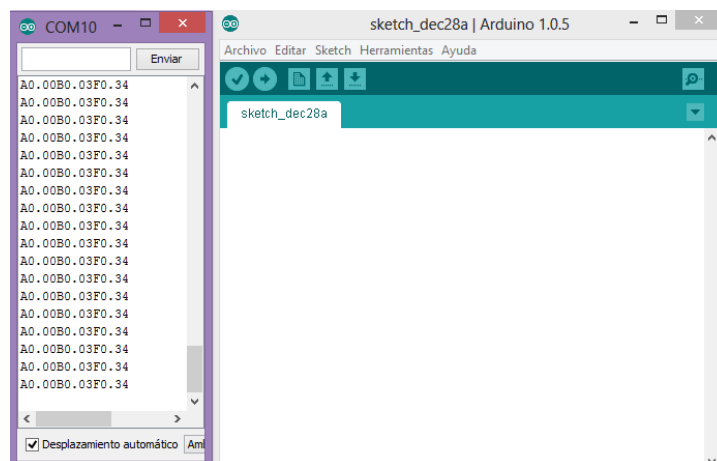


Fig. 3.18.- Datos vistos en el monitor serial de Arduino

Una vez con los datos listos en forma de cadena como A(litros por minuto)B(metros cúbicos)F(cantidad a pagar), el programa de Matlab estará listo para procesar estos datos bajo las siguientes condiciones iniciales.

Un dispositivo para este caso es el medidor digital desarrollado durante esta residencia mediante un módulo de Xbee envía al puerto serial de la PC (a 9600 bauds) los datos en código ASCII como siguen:

- El dispositivo envía primero una "A" (código. ASCII 65) así el programa en MATLAB sabe que llegarán los datos de una muestra de litros/minuto que se guardará en el vector A.
- Para la muestra de litros/minuto siempre llegarán tres caracteres, por ejemplo, para 0.13 litros por minuto, el dispositivo enviará al puerto serial los códigos Ascii de los caracteres "0", ".", "1" y "3".
- El dispositivo envía a continuación una "B" (código. ASCII 66) así el programa en MATLAB sabe que llegarán los datos de una muestra de metros cúbicos totales que se guardará en el vector V.
- Para la muestra de metros cúbicos totales siempre llegarán tres caracteres, por ejemplo, para 5.25 metros cúbicos, el dispositivo enviará al puerto serial los códigos ASCII de los caracteres "5", ".", "2" y "5" ya que consideramos que una familia no ha de gastar más de 999 metros cúbicos al mes.
- El dispositivo envía a continuación una "F" (código. ASCII 70) así el programa en MATLAB sabe que llegarán los datos de una muestra de costo en pesos que se guardará en el vector F.
- Para la muestra de costo no se sabe cuántos caracteres llegarán. Por qué los costos pueden variar, entonces si se miden 50 pesos llegarán dos caracteres: "5" y "0". Si se miden 1550 pesos llegarán cuatro caracteres: "1", "5", "5" y "0". Nadie sabe cuántos caracteres llegarán, sólo se sabe que éstos se terminan cuando se recibe nuevamente una "A" y todo el ciclo (desde el paso 1) se vuelve a repetir.

El programa almacenará diez muestras por cada parámetro leído (litros por minuto, M3 totales, pesos), las almacenará en sus respectivos vectores y mostrará estos en una gráfica en pantalla. Todo esto se repetirá ochocientas veces. Entonces el programa cierra el puerto y finaliza.

Esto nos proporcionara la facilidad de guardar los datos de alguno de los vectores para crear un gráfica, a continuación mostramos el diagrama de flujo del programa con el cual funciona el programa compilado en matlab mostrado en el anexo 5.

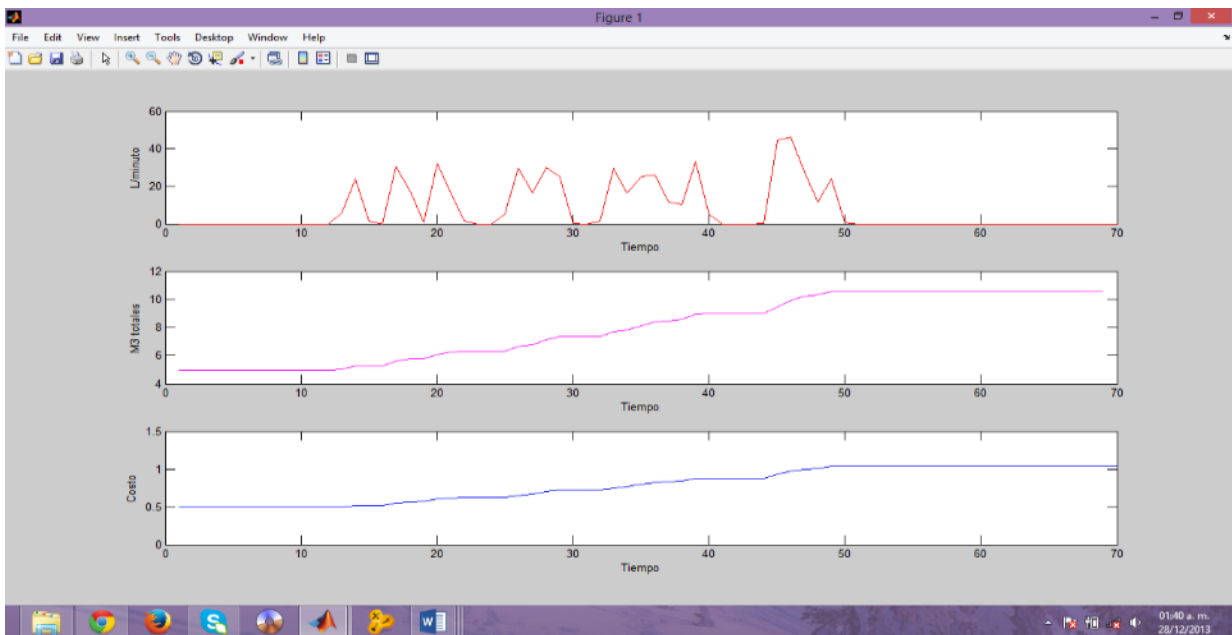


### 3.2.6.-Diagrama de flujo del programa en Matlab.

Este diagrama de flujo de la figura 3.20 tiene como misión mostrar de forma clara como el programa toma los datos del puerto serial COM10 y procede a graficarlos, es claro que este solo va a graficar los datos que envié uno de los medidores como primer etapa, la forma de la gráfica se puede ver en la figura 3.19, en donde matlab abre el puerto, y se sincroniza a 1 por segundo, así pues almacena 10 datos de cada uno (litros por minuto. M3 totales, y costo de lo acumulado), en sus vectores  $V[ ]$ ,  $A[ ]$ ,  $F[ ]$ , y procede a graficar; luego sin borrar la presente grafica vuelve a juntar 10 datos de cada uno y grafica nuevamente haciendo cada vez más grande el valor en y.

Una gráfica generada con la lógica de este diagrama es como la figura 3.19, de la cual podemos observar que se estableció una ventana exclusiva para cada una de las variables, así una es para los litros por minuto, la otra es para ver los metros cúbicos que el usuario lleva consumiendo, y el ultimo es el de costo.

Nótese que según la información del grafico se estuvo variando la velocidad del flujo de agua además que mientras que la primera sube y baja en las últimas dos permanece en el lugar sin bajar otra vez, ósea que se van acumulando con el tiempo.



**Fig.3.19- Grafico de Matlab obtenido.**

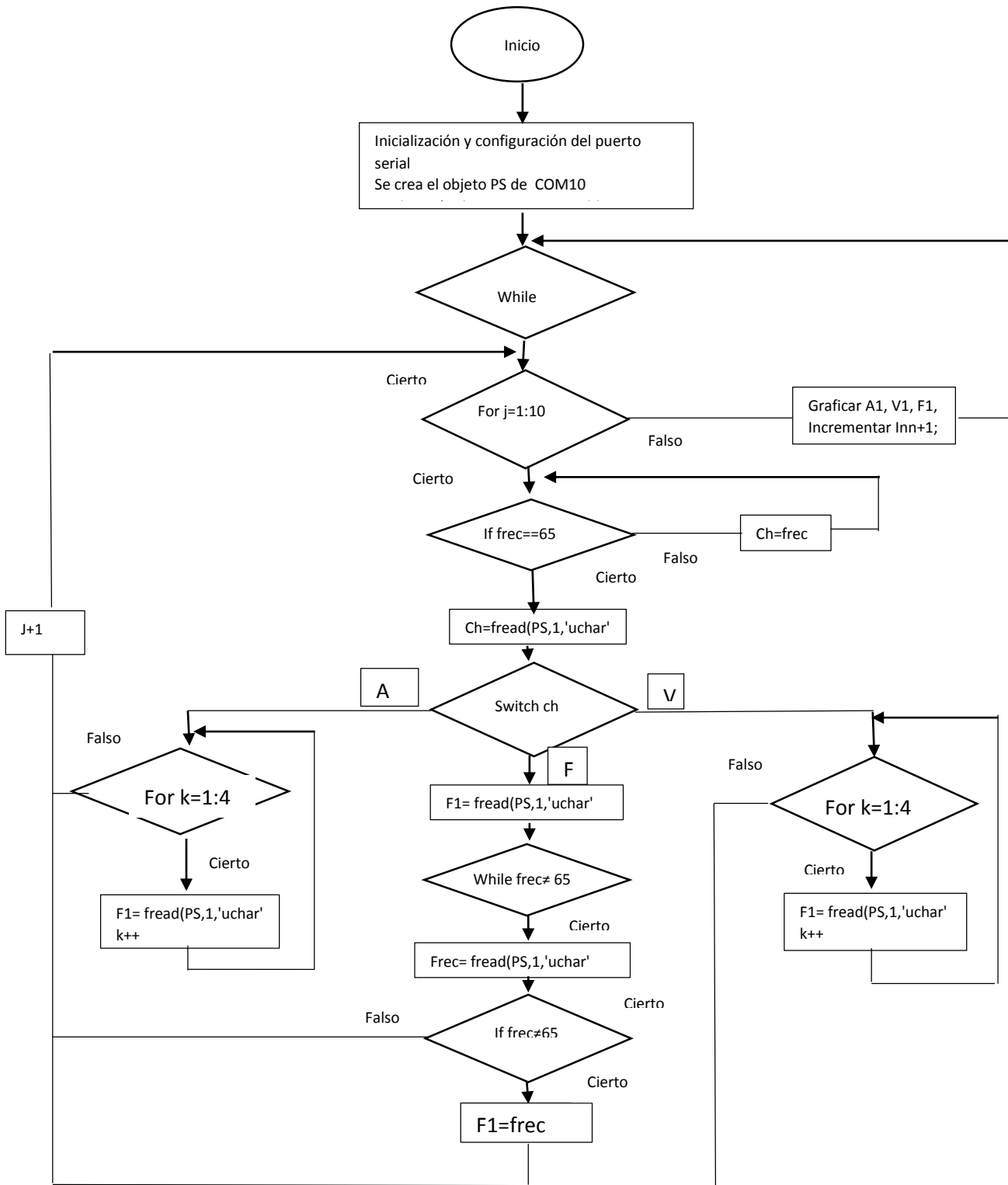


Fig. 3.20.- Diagrama de flujo del programa graficado en Matlab.

### 3.4.- Resultados obtenidos.

#### 3.4.1.- Medidor digital, en operación.

Una vez probada y terminada la funcionalidad del primer medidor digital se procedió a realizar los otros tres que sugiere el diagrama de tuberías. Y a ponerlos en operación, corrimos el programa MATLAB, obteniéndose una gráfica como la de la figura 3.22.



Fig. 3.21.- Medidor digital en operación

#### 3.4.2.- Graficas generadas por cada medidor digital de forma inalámbrica.

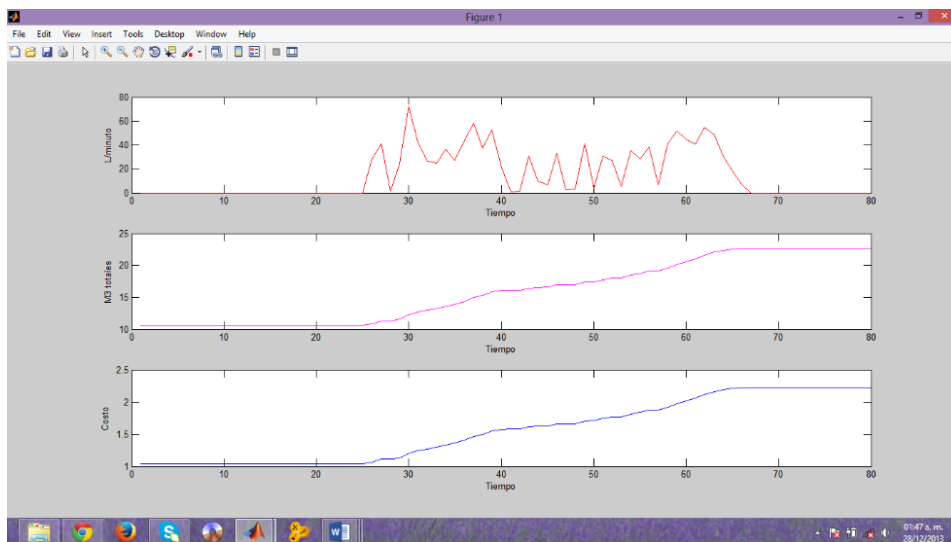
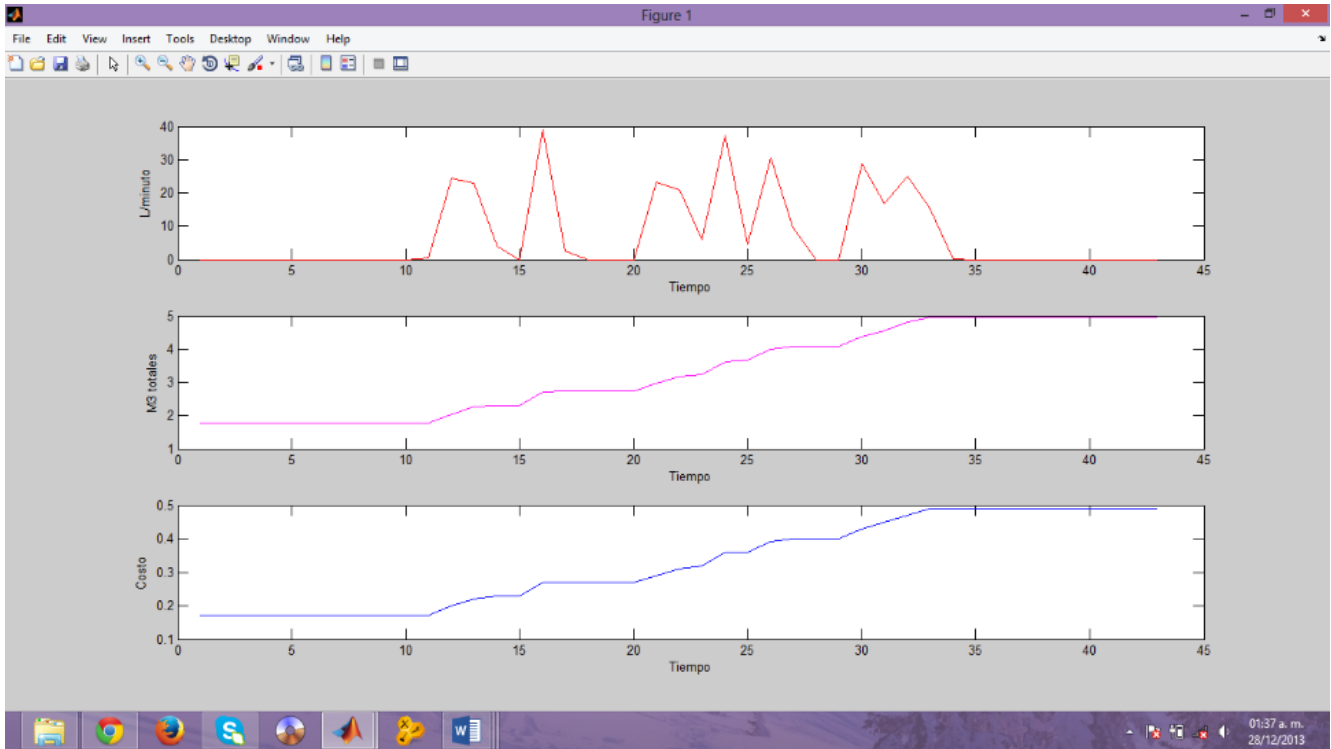
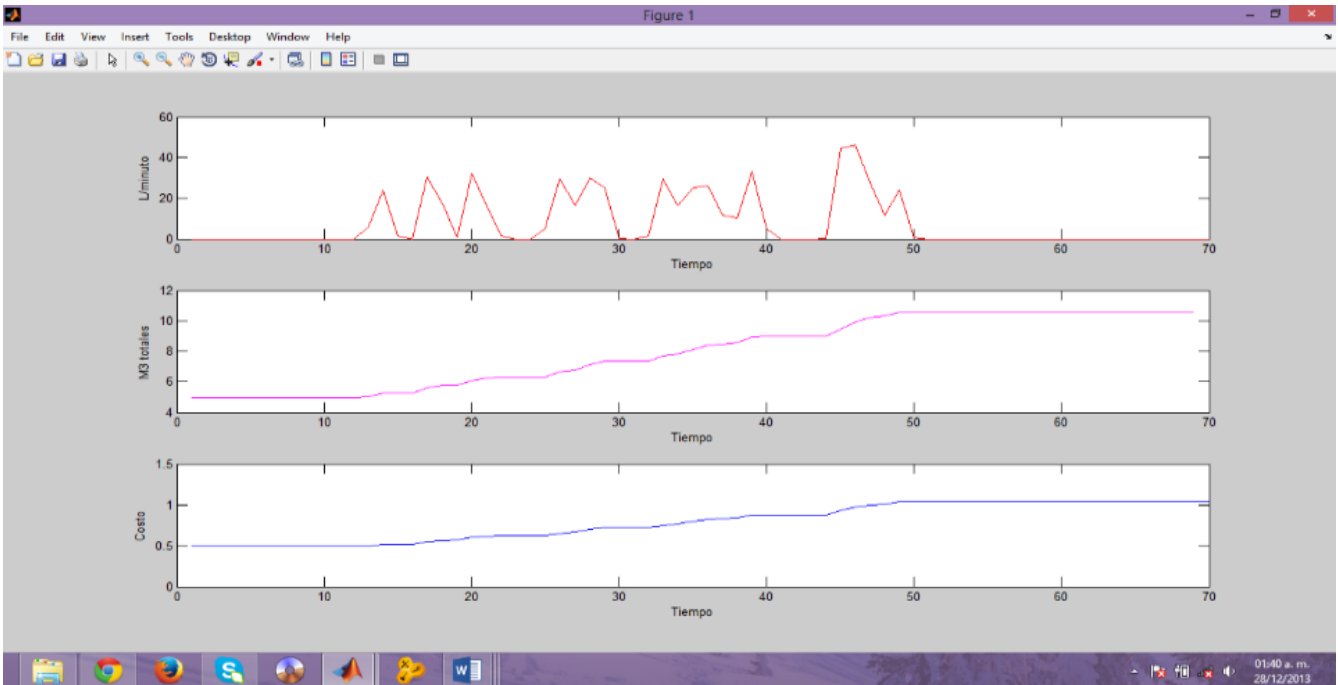


Fig. 3.22 Medidor digital 3 bajo una variación en un minuto



**Fig. 3.23 Medidor digital 2 bajo una variación en un minuto**



**Fig. 3.24 Medidor digital 3 bajo una variación en un minuto.**

## **Conclusiones.**

Es de sentirse una satisfacción cuando un sistema funciona y tiene una aplicación como es este caso, ya que la implementación de este sistema en el sector industrial o sistema de agua potable mejorara de forma directa la calidad en el servicio y los ahorros por perdida además que si esto se llevara usar para control de otras sustancias puede resultar en una mayor seguridad para el ser humano.

Gracias a la característica inalámbrica de este sistema electrónico, será muy posible evitar los encharcamientos y desperdicios de agua por las calles, sin tener que ir rompiendo gran área de pavimento en las calles, considerando el problema socio ambiental que esto provoca.

No dudamos pues ver esta investigación quizá no tan diferente de la que aquí se muestra, aplicándose en los sistemas hidráulicos de nuestro país, y así mejorar en el ámbito tecnológico.

## **Recomendaciones de uso del sistema.**

En cuanto el receptor Xbee además del programador sea cual sea también es posible usar como receptor un placa Arduino y el shield de Xbee montado, solo que esta vez se debe retirar el ATMEGA328P-PU de la placa, y con ello obtendremos una comunicación entre el computador y el Xbee.

Un hecho importante en el uso del medidor digital para el envío de datos es que primero debe estarse enviando datos desde el emisor ya que si este comienza después el programa que hace la gráfica no podrá arrancar.

Por otra parte del emisor se interrumpe mientras el programa está construyendo la gráfica esta marcará error.

## **Recomendaciones para la mejora del sistema.**

Como se mencionó al principio el objetivo principal de este proyecto es hacer un sistema que quite los problemas de agua potable de forma efectiva, como lo son el uso clandestino de las tomas de agua potable y el encharcamiento en las calles, hemos iniciado en primera etapa tal proyecto dejando las siguientes etapas que se puedan generar hasta llevar esto hasta su implementación física en el sistema hidráulico de la ciudad para aquellos los estudiantes e investigadores que tengan interés en mejorar las condiciones actuales de nuestro sistema hídrico..

Mencionando pues que para mejorar hay que invertir, y si en este proyecto se tiene que invertir en los trasmisores, tubos etc sabiendo que la calidad casi siempre va de la mano con el precio..

## Referencias.

### **Abastecimiento de agua y disposición y eliminación de excretas.**

Pedro López alegría.  
Alfa-omega.

### **Tratamiento de aguas industriales**

Aguas de proceso y residuales  
Miguel rigola Lapeña  
Alfa-omega Marcombo

### **Potabilización del agua 3 edicion**

Jairo Alberto romero rojas  
Alfa-omega

### **Manual de tratamiento de agua**

Departamento de sanidad del estado de Nueva York  
Limusa. Noriega

### **Despertad**

A donde ha ido el agua.  
Testigos de Jehová.

### **Xbee Series 1**

Guía de usuario  
MCI ELECTRONICS.

### **Manual de interfaz gráfica de usuario en Matlab**

Republica de ecuador  
Diego Armando Barragán Borrego

### **Matlab para ingenieros**

Holly Moore  
Pearson Prentice Hall

<file:///H:/LEAKCURE%20%20Nuevo%20mecanismo%20para%20reducir%20las%20fugas%20de%20agua%20en%20las%20redes%20urbanas%20de%20distribuci%F3n.htm>

<file:///H:/Sensores%20de%20caudal%20%20%20caudal%EDmetros.htm>

<file:///H:/Nuevo%20contador%20de%20agua%20de%20Kamstrup.htm>

El efecto hall.  
[www.wikipedia.com](http://www.wikipedia.com)

## Anexo 1.- Diagrama esquemático de Arduino UNO

Este esquema fue tomado de una página de internet, para lograr hacer las placas de ATMEGA328P-PU.

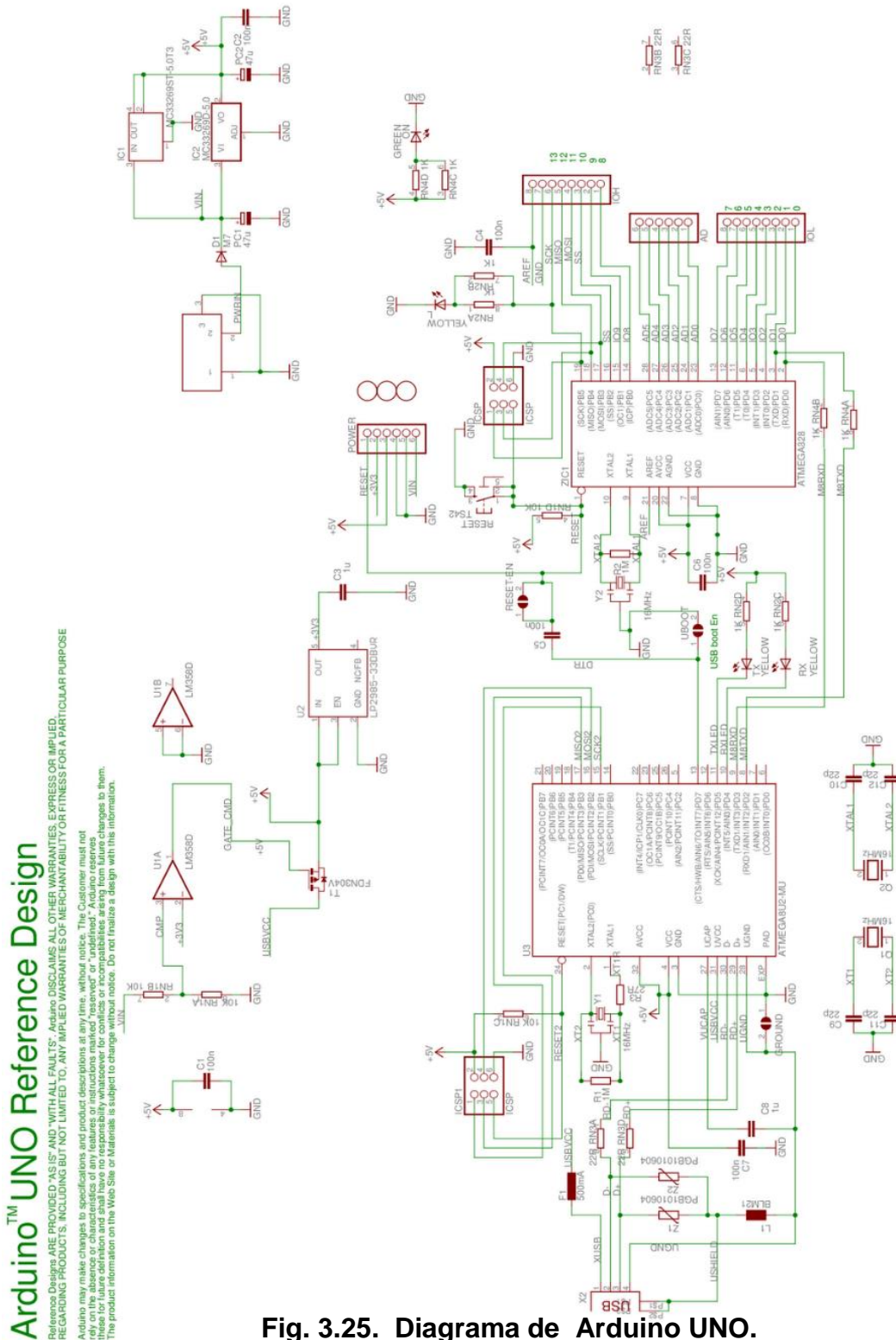


Fig. 3.25. Diagrama de Arduino UNO.



## Anexo 2.-Como usar el ATMEGA328P-PU fuera de la base de Arduino.

Para sustituir el microcontrolador ATMEGA328P-PU de la placa de Arduino por otro virgen, es necesario grabarle la secuencia de arranque que se menciona en el siguiente punto. Ya que sin esto a la hora de grabar el programa de Arduino el programa marcará un error.

### Cableado y forma de quemar la secuencia de arranque en el ATMEGA328P-PU

Con este proceso, es posible usar el CI ATMEGA328P-PU y programarlo desde la plataforma de Arduino uno, proceso que consiste en llevar a cabo cierto cableado que más abajo se explica quedando nuestro Arduino uno como quemador de PIC.

1. Tamaño: Si queremos un circuito pequeño si incluimos un Arduino nos limitamos al tamaño de él, con un micro el tamaño será mucho más pequeño.
2. Volumen de proyectos: Si somos aficionados a la electrónica y montamos numerosos proyectos definitivos que no llevamos idea de desmontar y no tenemos una economía boyante será mejor emplear \$100.00 en un micro que \$450.00 en un Arduino.

El inconveniente es que para poder programar el chip ATMEGA32P-8PU con el software de Arduino debe tener cargado el bootloader (secuencia de arranque) el micro, podemos comprarlo precargado o bien cargarlo nosotros con la ayuda de un Arduino.

¿Cómo lo hacemos? o bien realizamos este montaje en una placa de prototipos.

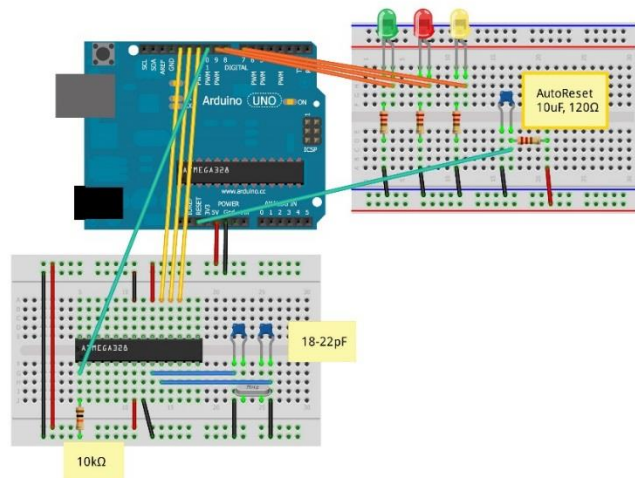


Fig.3.26.- Conexión para quemar la secuencia de arranque

### **Algoritmo para quemar el bootloader (secuencia de arranque) para el Arduino uno.**

- 1.- No quitar el ATMEGA 32-8PU de la plataforma de Arduino (el original).
- 2.- Conectar Arduino normalmente.
- 3.- Abrir el programa de Arduino, ir a **archivo**, luego en **ejemplos** después en **Arduino ISCP** y luego de cargar programa.
- 4.- Una vez cargado el programa ISCP hacer el conexionado mostrado en la fig. 3.
- 5.- Dentro del programa ir a **herramientas**, luego **grabar secuencia de inicio** y esperar.
- 6.- Al terminar Boot loader (secuencia de arranque) listo.

### Anexo 3.- Especificaciones del sensor de flujo YF-G21.



Fig.3.27- Sensor de flujo YF-G21.

Voltaje de funcionamiento	5V-24V
Corriente máxima	15 mA (DC 5V)
Peso	43 g
Rango de caudal	1 ~ 30 L / min
Temperatura de funcionamiento	0C ~ 80C
Temperatura del líquido	<120C
Humedad de funcionamiento	35% ~ 90% RH
Presión de trabajo	bajo 1.75Mpa
Temperatura	-25C ~ +80 C
Humedad	25% ~ 90% RH

#### Anexo 4.- Programa grabado en el micro controlador ATMEGA328P-PU

Damos valor a la variable Flujo tomada de **attachInterrupt(1,RPM,FALLING)**; //Pin 3 del Arduino y la dividimos entre 7.5 según la calibración en el sensor utilizado, tal valor será enviado por medio de la rutina envía(), al módulo de Xbee de forma inalámbrica por medio de **Serial.print(flujo/7.5)**;

Para fines prácticos formamos las variables desde aquí y enviamos además de este los otros valores como son.

```
Acumulado=Acumulado+((flujo/7.5)/1000);  
Serial.print(" A ");  
Serial.print(flujo/7.5);  
Serial.print(" V");  
Serial.print(Acumulado);  
Serial.print(" F");  
Serial.println(Acumulado*clasell);
```

Dentro de la rutina envía ();

Cada módulo de medición, se le ha grabado el siguiente programa, el cual tiene la ventaja de ser procesado por una placa Arduino uno, y además de ser compilado por las placas independientes que se desarrollaron durante esta residencia con el fin de no dejar los Arduinos, y Shield pegados al proyecto de prueba.

```

/*
*****
**                               I.T.T.G                               **
** Programación para medidor digital de agua en Registro de patente **
**   Grupo del Evento de innovación Tecnológica 2013   **
**                               SISMAE SA CV 2013                               **
*****

*/
#define Sensor 3
#define clasell 10.2
#include <LiquidCrystal.h>
LiquidCrystal lcd(9, 8, 7, 6, 5, 4);

volatile int Rpm;                //Revoluciones por minuto.
float flujo=0;
int hallsensor = 3;             //Pin donde se conectara el sensor.
unsigned long acumulado=0;
int x=0;
boolean Boton=HIGH;
boolean A=HIGH;

void setup()
{
  lcd.begin(16,2);
  lcd.setCursor(0,0);
  Serial.begin(9600);
  attachInterrupt(1,RPM,FALLING); //Pin 3 del Arduino
  attachInterrupt(0,BOTON,FALLING); //Pin 2 del Arduino
  pinMode(10,INPUT);
  muestralInicio();
  lcd.clear();
}

unsigned long Tiempo;
float Acumulado=0;
int m3=0;
int bandera=0;

void loop()
{
  Tiempo=millis()+1000;

```

```

while(millis() <= Tiempo)
{
}
Envia();
Imprime();
flujo=0;
}
//Rutinas utilizadas para este programa
//Botón
//-----
-----
void BOTON()
{
if(A == HIGH)
{
Boton=LOW;
A = LOW;

}
else{
Boton = HIGH;
A=HIGH;
}
lcd.clear();
}
//-----
-----
//Delay
//-----
-----
void Delay(int x)
{
unsigned long _mili = millis() + x;
if (_mili == 4294967295)
{
_mili=_mili-4294967295;
}
do{
BOTON();
Imprime();
}while(_mili >= millis());
}

```

```

void Delay2(int x)
{
  unsigned long _mili = millis() + x;
  if (_mili == 4294967295)
  {
    _mili=_mili-4294967295;
  }
  do{

    }while(_mili >= millis());
  }

//-----
-----
//Envía
//-----
-----
void Envia()
{

  Acumulado=Acumulado+((flujo/7.5)/1000);
  Serial.print(flujo/7.5);
  Serial.print(" L/minuto ");
  Serial.print(Acumulado);
  Serial.print(" m3 Totales ");
  Serial.print(Acumulado*clasell);
  Serial.println(" pesos");

}
//-----
-----

//Imprime
//-----
-----
void Imprime()
{
  if(Boton == HIGH)
  {
    lcd.setCursor(0,0);
    lcd.print("m3 totales: ");
    lcd.setCursor(3,1);
    lcd.print(Acumulado);
  }
}

```

```

    }
    else if (Boton == LOW)
    {
        lcd.setCursor(0,0);
        lcd.print("/minuto: ");
        lcd.print(fluj0/7.5);
        lcd.setCursor(0,1);
        lcd.print("pesos: ");
        lcd.print(Acumulado*clasell);
    }
}

```

```

//-----
-----
//Otros
//-----
-----

```

```

void muestralInicio()
{
    lcd.setCursor(5,0);
    lcd.print("M.C.I.M.");
    for(int i=0; i < 4; i++)
    {
        lcd.setCursor(8,1);
        lcd.print("o");
        delay(300);
        lcd.setCursor(8,1);
        lcd.print(" ");

        lcd.setCursor(9,1);
        lcd.print("o");
        delay(300);
        lcd.setCursor(9,1);
        lcd.print(" ");

        lcd.setCursor(6,1);
        lcd.print("o");
        delay(300);
        lcd.setCursor(6,1);
        lcd.print(" ");

        lcd.setCursor(7,1);
        lcd.print("o");
    }
}

```



```
delay(300);  
lcd.setCursor(7,1);  
lcd.print(" ");  
  
}  
  
Serial.println("*****I.T.T.G.*****");  
Serial.println("*      M.C.I.M.      *");  
Serial.println("* Medidor digital de agua *");  
Serial.println(" ");  
}
```

```
void RPM()
```

```
{  
  flujo=flujo+1;  
}
```

```
//-----  
-----
```

## Anexo 5.- Programa usado para la gráfica de datos en Matlab.

```
clear all;
close all;
clc;

PS=serial('COM10');
set(PS,'Baudrate',9600); % se configura la velocidad a 9600 Baudios.
set(PS,'StopBits',1); % se configura bit de parada a uno.
set(PS,'DataBits',8); % se configura que el dato es de 8 bits, debe
estar entre 5 y 8
set(PS,'Parity','none'); % se configura sin paridad.
set(PS,'Terminator','CR/LF'); % "c" caracter con que finaliza el envío.
set(PS,'OutputBufferSize',1); % "n" es el número de bytes a enviar.
set(PS,'InputBufferSize',1); % "n" es el número de bytes a recibir.
set(PS,'Timeout',1); % segundos de tiempo de espera.

fopen(PS); %abre el puerto serial

A=[]; %Litros por minuto.
V=[]; %M3 totales.
F=[]; %Costo en pesos.

A1=[]; %temp litros por minuto.
V1=[]; %temp m3 totales.
F1=[]; %temp costo en pesos.
Ch=0;

disp('presione enter para iniciar captura (solo se puede parar la ejecución
mediante control-C)');
pause;

inn=1;
frec=0;

while inn<800

    for j=1:10

        if frec==65
            Ch=frec;
        else
            Ch=fread(PS,1,'uchar'); % Ch: variable donde se almacena el dato leído del
puerto
        end; %fin de if.
    switch Ch
```

```

case 65
    for k=1:4
        A1 = [A1 fread(PS,1,'uchar')];
    end;                                %fin de for.
frec=0;
case 66
    for k=1:4
        V1 = [V1 fread(PS,1,'uchar')];
    end;                                %fin de for.
frec=0;
case 70
    F1 = [F1 fread(PS,1,'uchar')];
    while frec ~=65
        frec= fread(PS,1,'uchar');
        if frec ~=65
            F1 = [F1 frec];
        end;                            %fin de if.
    end;                                %fin de while.
Otherwise                             %default.
    continue;
end;                                %fin de switch.

A1=char(A1); V1=char(V1); F1=char(F1); % convierte ascii a carácter.
A=[A str2num(A1)];                    % convierte de caracter a numero
V=[V str2num(V1)];
F=[F str2num(F1)];

A1=[];
V1=[];
F1=[];
end;                                %fin de for.

figure(1)
colordef black
subplot(3,1,1)
plot(A,'r');
xlabel('Tiempo')
ylabel('L/minuto');
subplot(3,1,2)
plot(F,'m');
xlabel('Tiempo')
ylabel('M3 totales');

subplot(3,1,3)
plot(V,'b');
xlabel('Tiempo')

```

```
ylabel('Costo');  
pause(0.1);
```

```
inn=inn+1;  
end; %fin de while.
```

```
fclose(PS); %cerrar puerto serial.  
delete(PS);  
clear PS;  
INSTRFIND
```

## Anexo 6.- Diferencia de potencia en los LED.

La diferencia de potencial varía de acuerdo a las especificaciones relacionadas con el color y la potencia soportada.

En términos generales, pueden considerarse de forma aproximada los siguientes valores de diferencia de potencial:[7]

- Rojo = 1,8 a 2,2 voltios.
- Anaranjado = 2,1 a 2,2 voltios.
- Amarillo = 2,1 a 2,4 voltios.
- Verde = 2 a 3,5 voltios.
- Azul = 3,5 a 3,8 voltios.
- Blanco = 3,6 voltios.

Luego, mediante la ley de Ohm, puede calcularse la resistencia R adecuada para la tensión de la fuente V fuente que utilizemos.

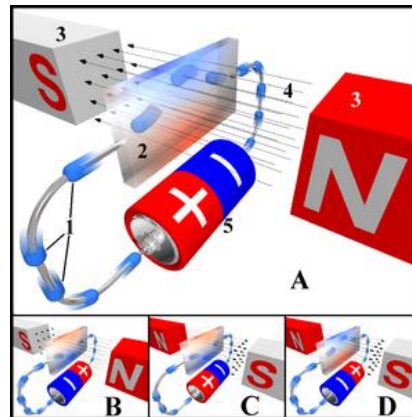
$$R = \frac{V_{fuente} - (V_{d1} + V_{d2} + \dots)}{I}$$

El término I en la fórmula se refiere al valor de corriente para la intensidad luminosa que necesitamos. Lo común es de 10 miliamperios para ledes de baja luminosidad y 20 mA para ledes de alta luminosidad; un valor superior puede inutilizar el led o reducir de manera considerable su tiempo de vida.

Otros ledes de una mayor capacidad de corriente, conocidos como ledes de potencia (1 W, 3 W, 5 W, etc.), pueden ser usados a 150 mA, 350 mA, 750 mA o incluso a 3000 mA dependiendo de las características optoeléctricas dadas por el fabricante.

Cabe recordar que también pueden conectarse varios en serie, sumándose las diferencias de potencial en cada uno. También se pueden hacer configuraciones en paralelo, aunque este tipo de configuraciones no son muy recomendadas para diseños de circuitos con ledes eficientes.

## Anexo 7.- Efecto Hall



Fin 3.28.- Diagrama del efecto Hall, mostrando el flujo de electrones.

### Descripción

En la imagen tt, una carga negativa aparece en el borde superior del sensor Hall (simbolizada con el color azul), y una positiva en el borde inferior (color rojo). En B y C, el campo eléctrico o el magnético están invertidos, causando que la polaridad se invierta. Invertir tanto la corriente como el campo magnético (imagen D) causa que la sonda asuma de nuevo una carga negativa en la esquina superior.

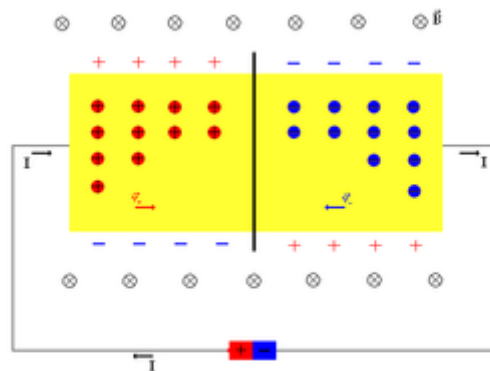


Fig. 3.29.- Dibujo explicativo.

En un conductor por el que circula una corriente, en presencia de un campo magnético perpendicular al movimiento de las cargas, aparece una separación de cargas que da lugar a un campo eléctrico en el interior del conductor, perpendicular al movimiento de las cargas y al campo magnético aplicado. A este campo eléctrico se le denomina **campo Hall**. Llamado efecto Hall en honor a su modelador Edwin Herbert Hall.

En épocas contemporáneas (1985) el físico alemán Klaus von Klitzing y colaboradores descubrieron el hoy conocido como efecto Hall cuántico que les valió el premio Nobel de Física en 1985. En 1998, se otorgó un nuevo premio Nobel de Física a los profesores Laughlin, Strömer y Tsui por el descubrimiento de un nuevo fluido cuántico con excitaciones de carga fraccionarias. Este nuevo efecto ha traído grandes problemas a los físicos teóricos y hoy en día, constituye uno de los campos de investigación de mayor interés y actualidad en toda la física.