

Universidad de Sonora  
División de Ciencias Exactas y Naturales



**Sistema de monitoreo terrestre de campos agrícolas mediante  
una red de sensores terrestres con Raspberry Pi**

Tesis que presenta:

***Luis Emmanuel Peña Murrieta***

Para obtener el Título de

**Ingeniero en Tecnología Electrónica**

Directora de tesis

**DRA. ALICIA VERA MARQUINA**

Hermosillo, Sonora 20 de abril de 2017

# Universidad de Sonora

Repositorio Institucional UNISON



**"El saber de mis hijos  
hará mi grandeza"**



Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

# AGRADECIMIENTOS

Le agradezco principalmente a mis padres que siempre me han apoyado en las decisiones que he tomado y por confiar en mí siempre, a los maestros que siempre están buscando mejores maneras de enseñar y ayudar a los alumnos en su camino profesional, a los compañeros y amigos de la carrera que pasamos grandes momentos.

# DEDICATORIA

Le dedico esta tesis a mis padres que siempre estuvieron ahí y haciéndome una mejor persona, para que tuviera un mejor futuro y nunca se cansaron de apoyarme.

# Contenido

Capítulo 1.....	6
1.1 Introducción general.....	6
1.2 Objetivo.....	6
1.2 Metodología.....	7
Distribución de tesis.....	7
Capítulo 2.....	8
2.1 Introducción.....	8
2.2 Componentes del nodo.....	8
2.3.1 Red de sensores inalámbricos.....	9
2.3.2 Fusión sensorial.....	10
2.3.4 Parámetros de una WSN.....	11
2.3.5 La topología de una red.....	11
2.3.6 Definición y Tipología.....	13
2.4 Desarrollo del sistema.....	13
2.5 Referencias.....	15
Capítulo 3.....	16
3.1 Introducción.....	16
3.2 Arduino.....	16
3.3 Raspberry Pi.....	17
3.4 Otros dispositivos para procesamiento.....	18
3.5 Xbee.....	19
3.5 Referencias.....	22
Capítulo 4.....	23
4.1 Introducción.....	23
4.1 Comunicación – bus I2C.....	23
4.1.2 Descripción de las señales.....	23
4.1.3 Protocolo de comunicación del bus I2C.....	24
4.2 Bus SPI (Serial Peripheral Interface).....	24
4.3 Protocolo 1-WIRE.....	24
4.4 Referencias.....	24
Capítulo 5.....	25

5.1 Introducción .....	25
5.2 Diagrama a bloques.....	25
5.3 Configuraciones y características de los dispositivos utilizados. ....	26
5.3.1 Raspberry Pi.....	26
5.3.2El sensor de temperatura DS18B20. ....	26
5.3.3 Sensor de Luz.....	29
5.3.4 Sensor de Humedad .....	32
5.4 Conexiones de prueba.....	33
5.5 Referencias.....	45
Capítulo 6.....	46
6.1 Conclusiones.....	46
6.2 Desarrollo de trabajos futuros .....	46
Apéndice A.....	47

# Capítulo 1

---

## Introducción general

En empresas dedicadas a dar servicio y soluciones al gremio agrícola, se están utilizando nuevas tecnologías para proporcionar nuevas alternativas y optimizar los recursos, proporcionando al sector agrícola con herramientas necesarias para elevar la producción y mejorando la calidad a un menor costo. Esto a través de soluciones para el cuidado de los cultivos desde la siembra hasta la cosecha.

La agricultura de precisión es una herramienta que permite a los agricultores tener información sobre sus campos, esto a través del uso de tecnologías para su monitoreo, en ellas podemos encontrar el uso de drones y cámaras especializadas para ver el comportamiento que tienen las plantas y así, mediante un análisis, determinar las áreas que se ven afectadas por plagas, falta de nutrientes, o enfermedades en las plantas que pudieran afectar el desarrollo del producto.

En este sentido, la tecnología de drones y cámaras multiespectrales permiten visualizar las áreas afectas. En esta tesis, se plantea utilizar una red de sensores terrestres para tener un control de los diferentes parámetros que se puedan monitorear en tiempo real, como lo son sensores de luz, humedad, conductividad, ph, además se pueden acoplar otros sistemas para obtener información acerca de las condiciones climáticas, que ayudarían al sistema a tener más precisión en afectaciones en el campo que pudieran surgir en la cosecha.

### 1.1 Objetivo

Se plantea crear una red de sensores inalámbricos que permitan el monitoreo del campo que permitan la toma de decisiones en el mismo; para lograrlo el sistema incorpora distintos sensores, los cuales serán controlados por una computadora de placa reducida Raspberry Pi Zero, el cual estará enviando la información a una computadora de placa reducida Raspberry Pi 3 con mayor procesamiento a través de la tecnología de XBee este a su vez estará almacenado en una base de datos, la cual podrá ser consultada desde cualquier ordenador.

## 1.2 Metodología

Investigar cuál es la tecnología más apropiada para la implantación de los circuitos en el campo.

Diseñar una red de sensores que permitan monitorear factores que ayuden al agricultor a saber la situación del campo.

Desarrollar un módulo que permita la utilización de este circuito en el campo.

## 1.3 Distribución de tesis

La propuesta de tesis está dividida básicamente en 5 secciones que describen lo siguiente:

Capítulo 1. Se da una breve introducción sobre el problema de la pérdida de la producción por diversos factores que afectan a la cosecha. Se describe el objetivo y la metodología que se usará.

Capítulo 2. Se describen aspectos importantes relacionados con la estructura básica del nodo.

Capítulo 3. Se presentan las diferentes formas de procesamiento de información con diferentes tecnologías como son Arduino, Raspberry Pi, entre otros.

Capítulo 4. Se da una introducción a los protocolos de comunicación y los tipos más usados. Se ejemplifican comunicación I2C, 1-WIRE Y SPI.

Capítulo 5. Se presentan los resultados obtenidos en base a la configuración elegida y programación Raspberry Pi.

Capítulo 6. Por último, en este capítulo se da la conclusión sobre el desarrollo de este trabajo y el trabajo a futuro que se pudiera implementar en el sistema para mejorarlo.



## Capítulo 2

---

### Estructura del nodo

#### 2.1 Introducción

El sistema de monitoreo se agrupa en una estructura que se le conoce como nodo la cual está diseñada para que envíe información sobre los sensores que estarán recopilando la información para que ésta sea revisada posteriormente, el nodo está compuesto de varios elementos como energía, procesamiento, comunicación y sensores.

La estructura del nodo está diseñada para ser instalada en un área en la cual soporte las condiciones para ser autosuficiente, el nodo deberá tener su propia fuente de alimentación ya sea con baterías, celdas solares o alguna otra fuente de energía externa, éste deberá poder enviar la información a otros nodos o solamente a receptor que estará recopilando los datos de los demás sensores, para procesarlo por medio de una computadora.

#### 2.2 Componentes del nodo

El nodo está constituido por varias secciones como son: energía, comunicación, procesador, memoria y sensores.

En la sección de energía comprende, la fuente de alimentación que tendrá el dispositivo, baterías y celdas solares son las más viables debido a que el nodo estará en el campo, la batería servirá como banco de energía de las celdas que estarán captando la luz solar para generar energía suficiente para mantener en funcionamiento el nodo, a su vez las baterías almacenarán la energía en caso de días nublados o de no ser suficiente para el nodo funcione.

En la sección de comunicación es todo aquello que se utiliza para la enviar y recibir información de los otros nodos.

En la parte de procesamiento, es donde se estará codificando la información para tener datos que proporcionen los sensores, como los de temperatura que estarán en grados centígrados, el porcentaje de la humedad que se encuentre en el área, la luminosidad estará el luxes, también se acoplará el sistema de comunicación para

que este pueda enlazarse.

La memoria es para almacenar los datos en dado caso que es nodo no pueda transmitir y poder obtener los datos de manera independiente.

Los sensores se utilizan para medir distintas variables y condiciones climáticas, como la humedad, temperatura, luminosidad, pH, conductividad, viento, entre otras.

En la figura 2.1 se muestra un diagrama a bloques de una estructura de nodo.

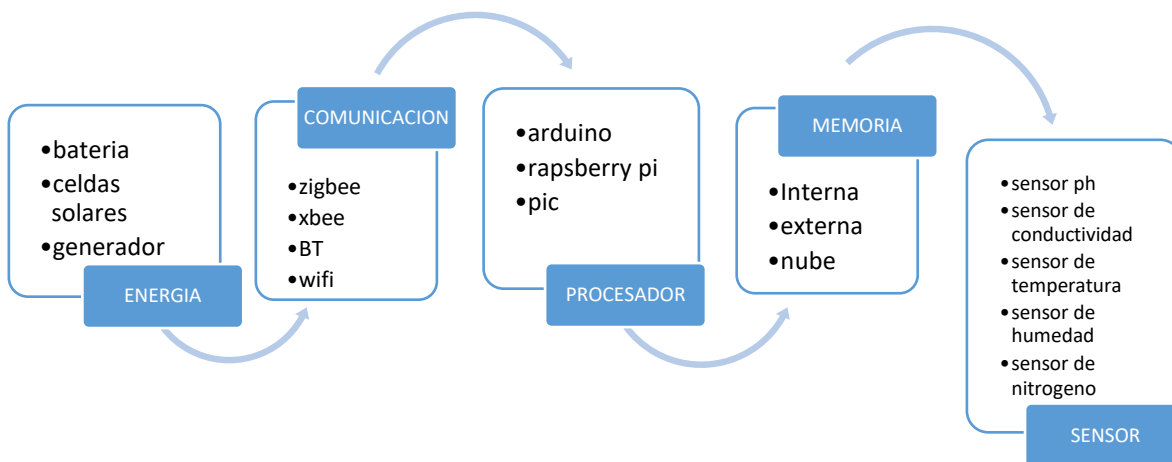


Figura 2.1. Diagrama a bloques de una estructura de nodo.

### 2.3.1 Red de sensores inalámbricos

Las tecnologías de redes inalámbricas han tenido un rápido desarrollo en los últimos años. Se ha pasado de los veteranos infrarrojo (IRDA) para comunicaciones punto a punto a las WPAN de corto alcance y multipuntos como “BlueTooth” o las redes de rango de alcance medio multi-saltos como “ZigBee”. Otras tecnologías inalámbricas que podemos nombrar son, la tecnología WI-FI para redes locales (WLAN), la tecnología “WIMAX” para redes WMAN. También la telefonía celular de largo alcance (GPRS) o el desarrollo de las comunicaciones M2M con tecnología inalámbrica. El desarrollo más interesante es el de las redes de sensores inalámbricos (WSN), debido a sus múltiples aplicaciones, en distintos sectores (seguridad, medio ambiente, industria, agricultura etc.). Los principales analistas tecnológicos, dentro de las tecnologías inalámbricas, valoran las redes inalámbricas

de sensores (WSN) como una de las opciones de futuro más prometedora. Fabricantes como Microsoft, Intel, IBM, Motorola y Texas Instruments, por citar algunos, han lanzado líneas de investigación en esta tecnología. Las redes inalámbricas de sensores (Wireless Sensor Networks) también se encuentra dentro de la llamada Inteligencia Ambiental ("pervasive computing," "ambient intelligence" = computación ubicua). El concepto "inteligencia ambiental" es un terreno fronterizo entre los últimos avances en computación ubicua y los nuevos conceptos de interacción inteligente entre usuario y máquina. En el terreno práctico, la inteligencia ambiental consiste en la creación de una serie de objetos de uso cotidiano con cualidades interactivas "suaves" y no invasiva. El objetivo básico de la inteligencia ambiental es el dotar a objetos de capacidades de adquisición de información (tanto del entorno físico como del estado actual del objeto), procesamiento y comunicación, de tal forma que puedan comunicarse entre ellos y ofrecer nuevos servicios a sus usuarios. [1]

### 2.3.2 Fusión sensorial

Fusión es la combinación de datos sensoriales o datos derivados de datos sensoriales para producir datos mejorados en forma de una representación interna del entorno de proceso. [2]

Objetivos:

- Explotar las sinergias entre medidas: la unión de medidas es mejor que la suma de todas ellas.
- Mejorar fiabilidad: ruido, fallos de algún sensor.

Consideraciones:

- Asociación: determinar si datos de dos o más sensores provienen del mismo fenómeno.
- Filtrado de datos: evitar datos contradictorios.
- Carga computacional: velocidad de repuesta necesaria.

En la Figura 2.2 se muestra un diagrama de fusión sensorial.



Figura 2.2 Diagrama de fusión sensorial.

### 2.3.4 Parámetros de una WSN

Los parámetros de una WSN son factores que se toman en cuenta para saber qué tan viable es el dispositivo de elaborar en los cuales toman en cuenta diferentes valores. [5]

Los valores principales que caracterizan una red inalámbrica de sensores son los siguientes:

- Tiempo de vida
- Cobertura de la red
- Coste y facilidad de instalación
- Tiempo de respuesta
- Precisión y frecuencia de las mediciones
- Seguridad
- Los valores principales que caracterizan al nodo sensor son los siguientes:
  - Flexibilidad
  - Robustez
  - Seguridad
  - Capacidad de comunicación
  - Capacidad de computación
  - Facilidad de sincronización
  - Tamaño y coste
  - Gasto de energía

### 2.3.5 La topología de una red

La topología de una red es la descripción de la forma en la que se conectan sus nodos. [3]

Existen tres topologías básicas:

1. Red centralizada.- Son todos los nodos menos uno. Son periféricos y sólo pueden comunicarse a través del nodo central. La caída del nodo central priva del flujo a todos los demás nodos.

2. Red descentralizada.- En este tipo de red no existe un único nodo central sino un centro colectivo de conectores. La caída de uno de los nodos centralizadores conlleva la desconexión de uno o más nodos del conjunto de la red mientras que la caída del cluster centralizador produciría necesariamente la ruptura y práctica desaparición de la red.

Nota: clúster se le llama al organizador de todos los nodos el cual lleva a cabo la recopilación de todos los nodos conectados a él, este dispositivo estaría en el centro de toda la maya, este contaría con un servidor web o base de datos para todos los dispositivos conectados a él.

3. Red distribuida.- La extracción de cualquiera de los nodos no desconectaría de la red a ningún otro. Todos los nodos se conectan entre sí sin que tengan que pasar necesariamente por uno o varios centros locales. En este tipo de redes desaparece la división centro periferia y por tanto el poder de filtro sobre la información que fluye por ella.

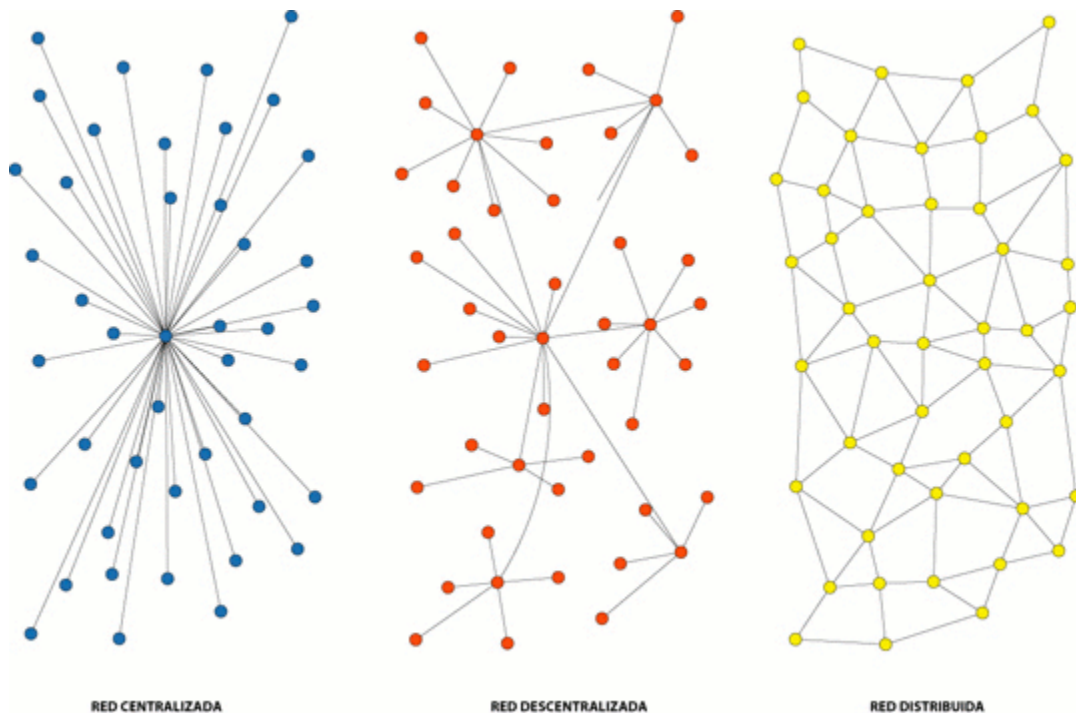


Figura 2.3 Topologías básicas

### 2.3.6 Definición y Tipología

Las tecnologías inalámbricas son aquellas que dependen para su funcionamiento de ondas de radio, de microondas y pulsos de luz infrarroja para transportar las comunicaciones digitales sin cables entre dispositivos de comunicación. En cuanto a su alcance se pueden distinguir los siguientes tipos de red: [6]

#### *WPAN*

WPAN (Wireless Personal Area Networks, Red Inalámbrica de Área Personal o Red de Área Personal o Personal Area Network) es una red para la comunicación entre distintos dispositivos (tanto ordenadores, puntos de acceso a internet, teléfonos móviles, PDAs, dispositivos de audio, impresoras, etc.) cercanos al punto de acceso. Estas redes normalmente son de unos pocos metros y se utilizan para uso personal. Entre las diferentes tecnologías de WPAN destacan Bluetooth y Zigbee.

#### *WLAN*

WLAN (Wireless Local Area Network, Red Inalámbrica de Área Local) es un sistema de comunicación de datos inalámbrico flexible muy utilizado como alternativa a las redes LAN cableadas o como extensión de éstas. Utiliza tecnología de radiofrecuencia que permite mayor movilidad a los usuarios al minimizar las conexiones cableadas. La tecnología asociada a esta forma de red es Wi-Fi.

#### *WMAN*

WMAN (Wireless Metropolitan Area Network o Red Inalámbrica de Área Metropolitana) es una red de alta velocidad que, dando cobertura en un área geográfica extensa, proporciona capacidad de integración de múltiples servicios, mediante la transmisión de datos, voz y vídeo, sobre medios de transmisión inalámbricos. Destacadas tecnologías asociadas son WiMax.

#### *WWAN*

WWAN (Wireless Wide Area Network) son típicamente redes celulares para telefonía móvil y transmisión de datos. Destacadas tecnologías asociadas son GSM (telefonía móvil 2G) y UMTS (telefonía móvil 3G).

## 2.4 Desarrollo del sistema

De acuerdo a la información anterior, en este trabajo de tesis se desarrolló el sistema con los parámetros establecidos.

2 Nodos de FFD (Dispositivo de funcionalidad completa) recolector y 10 Nodos de corto alcance, en la cual se tendría una red descentralizada debido a que se

requiere con dos nodos de recolección y 10 nodos en los cuales estarían los sensores.

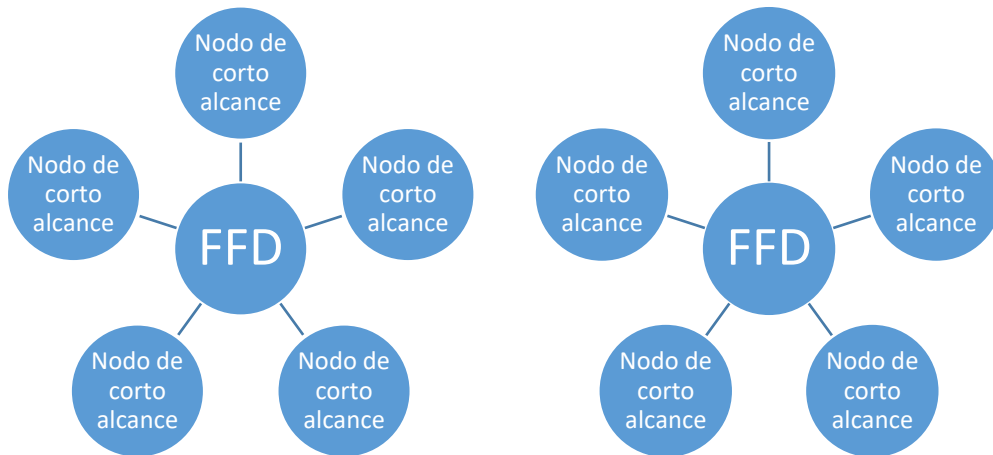


Figura 2.4. Red WSN red descentralizada 2 nodos FFD y 10 de corto alcance

En este sistema de redes se tendrán dos nodos, los cuales cubrirán un área que dependerá del dispositivo utilizado, los dispositivos varían en alcance, existen desde los 20 metros hasta kilómetros, para que las muestras sean más viables, éstas comprenderán estar en posiciones no muy largas por lo que se propone utilizar dispositivos de corto alcance los cuales darán mediciones más regulares entre posiciones. Se colocarán a una distancia de 20 metros del nodo central.



Figura 2.5. Red centralizada con 1 nodo FFD y 10 nodos de corto alcance

En esta red se tendría un círculo de 100mts de radio. En este caso, se está considerando sólo un nodo recolector de datos, por lo cual se tendría un red centralizada, donde se tendría que colocar los nodos a una distancia de 10 metros de separación.

En conclusión, estos tipos de redes centralizadas de corto alcance sólo servirían si se considerara un campo con dimensiones pequeñas.

Los modelos que comprenden estas tecnologías, en el acoplamiento con el dispositivo Arduino, resultan mucho más baratas en lo que respecta a la tecnología ZigBee. En el capítulo 3 se presentarán algunos módulos Arduino.

## 2.5 Referencias

[1] <http://www.mfbarcell.es/conferencias/wsn.pdf>

[2]

[https://mobile.aau.at/~welmenre/papers/elmenreich\\_Dissertation\\_sensorFusionInTimeTriggeredSystems.pdf](https://mobile.aau.at/~welmenre/papers/elmenreich_Dissertation_sensorFusionInTimeTriggeredSystems.pdf)

[3] <https://lasindias.com/indianopedia/topologias-de-red>

[4] <http://new-redes.blogspot.mx/2014/11/red-ad-hoc-inalambrica-una-red-adhoc.html>

[5] <http://informaticaredes2012.blogspot.mx/2012/07/motas-gateway-estacion-base-parametros.html>

[6] [http://www.agenciaidea.es/c/document\\_library/get\\_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157](http://www.agenciaidea.es/c/document_library/get_file?uuid=9e84835b-e749-4b6c-a01d-f60d3399fae9&groupId=10157)



## Capítulo 3

---

# Procesamiento de información

### 3.1 Introducción

Para llevar a cabo el procesamiento de información se investigaron las diferentes plataformas que existen para llevar a cabo el proceso de recopilación de datos de las cuales mostraran algunas de las características, esto con la finalidad compararlas.

En esta sección se dan a conocer los diferentes tipos de procesadores investigados de los cuales, se eligió uno como en el nodo recolector de datos, en los cuales se toman en cuenta diferentes parámetros como los de comunicación, procesamiento y memoria.

### 3.2 Arduino

Arduino es una plataforma electrónica de código abierto basado en hardware y software fácil de usar. Las tarjetas de desarrollo Arduino son capaces de leer las entradas - la luz en un sensor, un dedo sobre un botón o un mensaje de Twitter - y convertirlo en una salida - la activación de un motor, encender un LED, publicar algo en línea, entre otros. Para ello se utiliza el lenguaje de programación de Arduino (basado en el cableado), y el software de Arduino (IDE), sobre la base de procesamiento.<sup>1</sup>

En las tablas 3.1 y 3.2 se muestran algunas de las especificaciones de dos dispositivos para establecer el procesamiento de los nodos.

Tabla 3.1 Especificaciones (Genuino uno)

Voltage de operation	5V
Voltaje de entrada(recomendada)	7-12V
Voltaje de entrada(limite)	6-20V

Pines analogos	6
Memoria flash	32KB
Velocidad de reloj	16MHz
Peso	25g
Longitud	68.6 mm
Ancho	53.4 mm

Tabla 3.2 Especificaciones Arduino Mega

Voltage de operation	5V
Voltaje de entrada(recomendada)	7-12V
Voltaje de entrada (limite)	6-20V
Pines analogos	16
Memoria flash	32KB
Velocidad de reloj	16MHz

### 3.3 Raspberry Pi

La Raspberry Pi es un único ordenador de a bordo de tamaño aproximado a una tarjeta de crédito. Ha sido desarrollado por la Fundación Raspberry PI, una organización benéfica del Reino Unido (Reg. Nº 1.129.409). Se puede conectar a un teclado, ratón y pantalla convirtiéndolo en un potente y pequeño ordenador. <sup>2</sup>

Raspberry Pi cuenta con capacidad de almacenar información, existen versiones que cuentan con comunicación WI-FI y bluetooth además de tener 40 puertos con diferentes tipos de comunicaciones.

La Raspberry Pi 3 es la tercera generación de Raspberry Pi. Se sustituye el Raspberry Pi Modelo B 2 en febrero de 2016. En comparación con el Raspberry Pi 2 se tiene:

- Un 1,2 GHz de 64 bits de cuatro núcleos de CPU ARMv8
- 802.11n Wireless LAN
- Bluetooth 4.1
- Bluetooth Low Energy (BLE)

Al igual que el Pi 2, también tiene:

- 4 puertos USB
- 40 pines GPIO
- puerto lleno de HDMI
- Puerto Ethernet
- conector de audio de 3,5 mm combinado y vídeo compuesto
- interfaz de la cámara (CSI)
- Interfaz de pantalla (DSI)
- ranura para tarjetas micro SD (ahora push-pull en lugar de push-push)
- VideoCore IV 3D núcleo de gráficos
- La Raspberry Pi 3 tiene un factor de forma idéntica a la anterior Pi 2 (1 y Pi Modelo B +) y tiene una compatibilidad completa con Raspberry Pi 1 y 2.

### 3.4 Otros dispositivos para procesamiento

También existen otros dispositivos que se utilizan para realizar las tareas de procesamiento como lo es Wido y Seeendduino Stalker, que son sistemas basados en la plataforma de Arduino pero desarrollados para que cuenten con otros parámetros como memoria y conexión a internet. Seeendduino Stalker permite crear una red de nodos inalámbricos por medio de XBee, el uso de paneles solares de 5V para cargar una batería de litio de 3.7V y utiliza las librerías de Arduino. En las tablas 3.3 y 3.4 se indican algunas de las herramientas con las que cuentan.

Tabla 3.3 Precios de arduino mega en el mercado, características y disponibilidad en México.



	Seeeduino Stalker v3		
	características	Ventajas	desventajas
ATmega328P MicroSD card socket I2C Pin Bee series Solar socket	Arduino compatible	Consumo	

Tabla 3.4 Características de Seeenduo Stalker.

	WiDo		
	características	Ventajas	desventajas
Cloud IoT internet WiFi Node	Arduino compatible	Consumo	

### 3.5 Xbee

XBee es una tecnología muy amigable con dispositivo Arduino y permite crear redes punto a multipunto, o en redes de punto a punto, además que cuenta con Zigbee que es una alianza y un estándar de redes MESH de eficiencia energética y de costos, por lo que se opta por este sistema. [3]

Sistema Regular vs Pro – Hay pocas diferencias entre un XBee regular y un XBee PRO. La diferencia en cuanto a hardware es que el XBee PRO es un poco más largo. Con respecto a comunicación, la versión Pro tiene un mayor alcance (1,6 Km línea vista), pero para ello tiene un mayor consumo de potencia. El criterio para tomar la decisión de cual usar, es la distancia que uno requiera comunicar dos XBee. Los dos modelos se pueden mezclar dentro de la misma red.

Para comenzar, en el acoplamiento de estos dispositivos, se necesita tener por lo menos dos XBee, uno que sería el maestro, el cual se conectaría a un PC para leer la recopilación de información y otro que se tendría en la red de sensores.

Los tipos de XBee que se muestran en las tablas se pueden implementar para las tres tipos de arquitectura propuestas en este trabajo de tesis, además de utilizar las diferentes redes estrella, malla y Ad-hoc.

Tabla 3.5. Precios de XBee Series 1.



	Precios		Características
	Digi-key	32.00 USD	<ul style="list-style-type: none"> <li>• 3.3V @ 215mA</li> <li>• Velocidad de datos 250kbps</li> <li>• Salida 60mW (+18dBm)</li> <li>• Rango 1 milla (1500mts)</li> </ul>
	Digi-key	\$19.00 USD	<ul style="list-style-type: none"> <li>• Rango 30 metros encerrado,</li> <li>• 100 metros al aire libre</li> <li>• Voltaje de alimentación 2.8-3.4 VDC,</li> <li>• Corriente de transmisión 45mA, corriente de recepción 50mA</li> <li>• Antena múltiple opcional</li> </ul>
	Newark	\$38.00 USD	
Fáciles de trabajar, no necesitan ser configurados, para comunicaciones punto a punto y redes MESH			

Tabla 3.6. XBee explorer. Dispositivo que permite establecer comunicación con un PC para su configuración o bien para la adquisición de datos.

	Fabricante	Distribuidor	precio
--	------------	--------------	--------




	SparkFun	Digi-Key <a href="http://www.digikey.com.mx/product-detail/es/sparkfun-electronics/WRL-11697/1568-1116-ND/5318758">http://www.digikey.com.mx/product-detail/es/sparkfun-electronics/WRL-11697/1568-1116-ND/5318758</a>	\$24.95 USD
Dispositivo para conectar con pc para configuraciones			

Tabla 3.7. Los dispositivos XBee serie 2. Permiten un mayor alcance además de tener que ser configurados esto da un mejor control de los dispositivos.

	Distribuidor	precios	Características
	Digi-key [4]	\$17.50 USD	<ul style="list-style-type: none"> <li>• 2.1 a 3.6V</li> <li>• Corriente de tansmision 33mA @ 3.3 VDC</li> <li>• Velocidad de datos 1Mbps</li> <li>• Alimentación-Salida 8 dBm</li> <li>• Rango adentro 60 metros</li> <li>• Rango afuera 1200m</li> </ul>
	Digi-key [5]	\$ 28.50 USD	<ul style="list-style-type: none"> <li>• Voltaje de fuente 2.7V~3.6V</li> <li>• Velocidad de datos 1 Mbps</li> <li>• Antena de cable</li> <li>• Rango dentro 90 metros</li> <li>• Rango afuera 3200m</li> </ul>
Necesitan ser configurados			

### 3.5 Referencias

[1] <https://www.arduino.cc/en/Guide/Introduction>

[2] <http://www.raspberrypi-spy.co.uk/2012/03/what-is-a-raspberry-pi>

[3] <http://xbee.cl/que-es-xbee/>

[4] <http://www.digikey.com.mx/product-detail/es/digi-international/XB24CZ7PIT-004/602-1557-ND/532237>

[5] <http://www.digikey.com.mx/product-detail/es/digi-international/XBP24CZ7WIT-004/602-1564-ND/5322378>

# Capítulo 4

---

## Buses de comunicación

### 4.1 Introducción

Un bus es un medio compartido de comunicación, constituido por un conjunto de líneas (conductores) que conecta las diferentes unidades de un computador. La principal función de un bus será, pues, servir de soporte para la realización de transferencias de información entre dichas unidades

En esta sección se identifican algunas de las redes de que se encuentran internas en la Raspberry Pi para que se puedan comunicar con los sensores en las cuales encontramos, protocolos de comunicación e interfaces de comunicación.

### 4.1 Comunicación – bus I2C

El bus I2C, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos a una velocidad aceptable, de unos 100 Kbits por segundo, aunque hay casos especiales en los que el reloj llega hasta los 3,4 MHz. [1]

La metodología de comunicación de datos del bus I2C es en serie y sincrónica. Una de las señales del bus marca el tiempo (pulsos de reloj) y la otra se utiliza para intercambiar datos.

#### 4.1.2 Descripción de las señales

SCL (System Clock) es la línea de los pulsos de reloj que sincronizan el sistema. SDA (System Data) es la línea por la que se mueven los datos entre los dispositivos. GND (Masa) común de la interconexión entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenaje abierto, es decir, un estado similar al de colector abierto, pero asociadas a un transistor de efecto de campo (o FET). Se



deben polarizar en estado alto (conectando a la alimentación por medio de resistores "pull-up") lo que define una estructura de bus que permite conectar en paralelo múltiples entradas y salidas.

#### 4.1.3 Protocolo de comunicación del bus I2C

Habiendo varios dispositivos conectados sobre el bus, es lógico que, para establecer una comunicación a través de él, se deba respetar un protocolo. Digamos, en primer lugar, lo más importante: existen dispositivos maestros y dispositivos esclavos. Sólo los dispositivos maestros pueden iniciar una comunicación.

#### 4.2 Bus SPI (Serial Peripheral Interface)

SPI es un bus de tres líneas, sobre el cual se transmiten paquetes de información de 8 bits. Cada una de estas tres líneas porta la información entre los diferentes dispositivos conectados al bus. Cada dispositivo conectado al bus puede actuar como transmisor y receptor al mismo tiempo, por lo que este tipo de comunicación serial es full duplex. Dos de estas líneas transfieren los datos (una en cada dirección) y la tercer línea es la del reloj. Algunos dispositivos solo pueden ser transmisores y otros solo receptores, generalmente un dispositivo que tramite datos también puede recibir. [2]

#### 4.3 Protocolo 1-WIRE

La base de 1-Wire® tecnología es un protocolo serie utilizando una sola línea de datos además de referencia de tierra para la comunicación. A 1Wire maestro inicia y controla la comunicación con uno o más 1-Wire esclavos dispositivos en el bus 1-Wire. Cada dispositivo esclavo 1-Wire tiene un programa el fabricante, único, inalterable de 64 bits ID (número de identificación), que sirve como dirección del dispositivo en el bus 1-Wire. El 8 bits de código de la familia, un subconjunto de la ID de 64 bits, identifica el tipo de dispositivo y la funcionalidad. Por lo general, los dispositivos esclavos 1-Wire funcionan en el rango de voltaje de 2.8V (mínimo) de 5.25V (max). La mayoría de los dispositivos 1-Wire no tienen pin de fuente de alimentación; que toman su energía desde el bus 1-Wire (alimentación parásita). [3]

#### 4.4 Referencias

[1] [http://robots-argentina.com.ar/Comunicacion\\_busI2C.htm](http://robots-argentina.com.ar/Comunicacion_busI2C.htm)

[2] <http://www.i-micro.com/pdf/articulos/spi.pdf>

[3] <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>

## Capítulo 5

---

### Resultados

#### 5.1 Introducción

En esta sección se presenta la configuración para el desarrollo del sistema de monitoreo terrestre de campos agrícolas, así como parte de la programación realizada en la Raspberry Pi.

#### 5.2 Diagrama a bloques

El diagrama a bloques del sistema de monitoreo terrestre de campos agrícolas mediante una red de sensores terrestres con Raspberry Pi se muestra en la figura 5.1

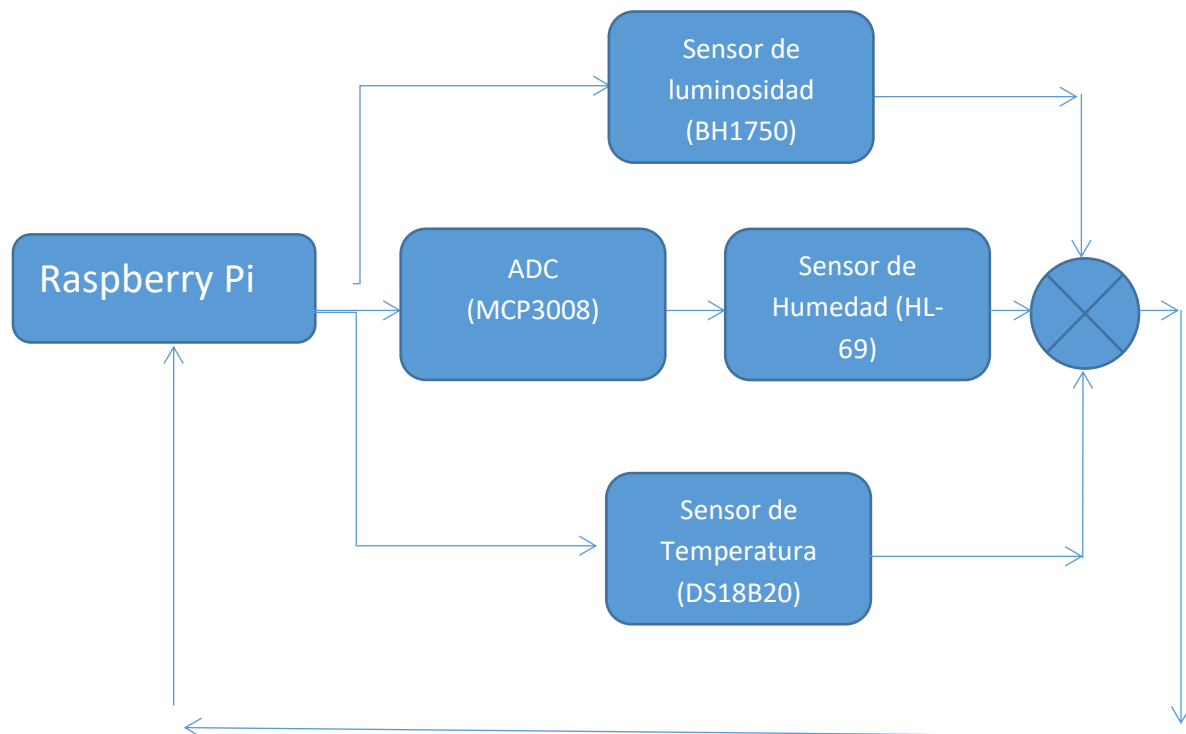


Figura 5.1 Diagrama a bloques del sistema.

### 5.3 Configuraciones y características de los dispositivos utilizados.

El proceso de acoplamiento con los sensores a través de los pines GPIO del Raspberry Pi Zero se llevó a cabo con los sensores de humedad, temperatura y luminosidad, además se acopló una LCD para poder visualizar las mediciones en el campo.

#### 5.3.1 Raspberry Pi

**Raspberry Pi**, el cual utilizan los pines GPIO para la comunicación con los sensores, los cuales sólo usan un voltaje de 3.3V para su funcionamiento. [3]

En la figura 5.2 se muestran los pines GPIO de un Raspberry pi, donde se observan de diferentes colores los distintos puertos de comunicación además de las fuentes de voltaje y tierras.

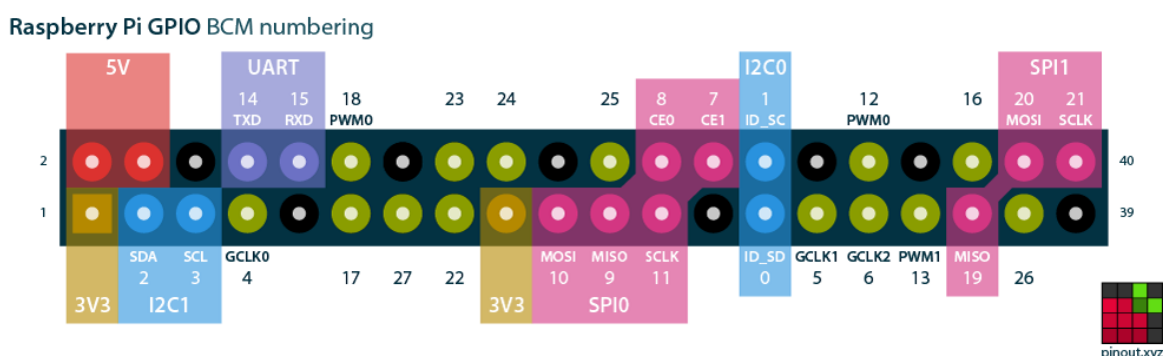


Figura 5.2. Pines de Raspberry Pi [1]

#### 5.3.2 El sensor de temperatura DS18B20.

El sensor DS18B20 permite medir temperaturas de hasta 125°C de forma fácil y además está sellado en un envoltorio estanco que permite sumergirlo en un líquido o protegerlo de la intemperie. Dado que es un sensor digital, la señal leída no se degrada debido a la distancia del cableado. Puede funcionar en modo 1-Wire con una precisión de  $\pm 0.5^\circ\text{C}$  con una resolución de 12 bits. También pueden utilizarse varios sensores sobre el mismo pin ya que internamente viene programado con un ID único de 64 bits para diferenciarlos. El rango de funcionamiento es de 3 a 5V por lo que se puede utilizar en prácticamente cualquier sistema de que use microcontroladores. [2]

NOTA: Incluye una resistencia de 4.7K Ohm para poder utilizarlo.

Características del cable:

- Tubo de acero inoxidable de 6mm de diámetro por 30mm de largo
- Largo: 91cm
- Diámetro: 4mm
- Contiene un sensor de temperatura DS18B20

Conexión:

En función de la producción, los cables del sensor pueden variar pero seguirán según estas especificaciones:

Si tienes 4 cables: Rojo es Vcc (positivo), Negro es GND (negativo) y Blanco es el cable de datos. La malla es GND.

Si tienes 3 cables: Rojo es Vcc (positivo), Azul/Negro es GND (negativo) y Amarillo/Blanco es el cable de datos.

Características del sensor DS18B20:

Rango de temperatura: -55 a 125°C

Resolución: de 9 a 12 bits (configurable)

Interfaz 1-Wire (Puede funcionar con un solo pin)

Identificador interno único de 64 bits

Múltiples sensores puede compartir el mismo pin

Precisión:  $\pm 0.5^{\circ}\text{C}$  (de  $-10^{\circ}\text{C}$  a  $+85^{\circ}\text{C}$ )

Tiempo de captura inferior a 750ms

Alimentación: 3.0V a 5.5V

Configuración

El pin 1 es de tierra. El pin 2 es el pin de datos y el pin 3 es el pin de alimentación. El único componente externo requerido es una sola resistencia 4.7Kohm.



Figura 5.3. Diagrama de conexión de DS18B2

Con el fin de configurar el sensor sólo tiene que hacer un pequeño cambio en el archivo config.txt usando:

```
/boot/config.txt sudo nano
```

añada la siguiente línea a la parte inferior:

```
dtoverlay = W1-GPIO, gpiopin = 4
```

El dispositivo está configurado para reportar su temperatura a través de GPIO4.

Para que los cambios surtan efecto tendrá que reiniciar el uso de:

```
sudo reboot
```

Utilice los comandos de abajo para ir al directorio que contiene los dispositivos 1-Wire detectados:

```
cd /sys/bus/w1/devices  
ls
```

Esto mostrará una lista de los directorios asociados a los dispositivos 1-Wire. Cada uno tiene un identificador único y en mi caso es 28-00000482b243. Su número de identificación será diferente, así que asegúrese de usar que en el código de ejemplo a continuación. El uso de "cd" podemos cambiar al directorio sensor de temperatura, una lista del contenido y luego ver el archivo "w1\_slave":

```
cd 28-00000482b243
ls
cat w1_slave
```

Para la lectura de la temperatura a través de Python que aquí hay una secuencia de comandos básicos:

```
#!/usr/bin/python
def gettemp(id):
    try:
        mytemp = ''
        filename = 'w1_slave'
        f = open('/sys/bus/w1/devices/' + id + '/' + filename, 'r')
        line = f.readline() # read 1st line
        crc = line.rsplit(' ',1)
        crc = crc[1].replace('\n', '')
        if crc=='YES':
            line = f.readline() # read 2nd line
            mytemp = line.rsplit('t=',1)
        else:
            mytemp = 99999
        f.close()

        return int(mytemp[1])

    except:
        return 99999

if __name__ == '__main__':

    # Script has been called directly
    id = '28-00000482b243'
    print "Temp : " + '{:.3f}'.format(gettemp(id)/float(1000))
```

**Nota:** Usted tendrá que reemplazar mi referencia a 28-00000482b243 con el identificador del dispositivo.

### 5.3.3 Sensor de Luz

#### Características

EL módulo BH1750 es un sensor de luz, que a diferencia del LDR es digital y nos entrega valores de medición en Lux ( lumen /m<sup>2</sup> ) que es una unidad de medida estándar para el nivel de iluminación (iluminancia). Tiene alta precisión y un rango ente 1 – 65535 lx el cual es configurable. [3]



Figura 5.4. Módulo BH1750 (sensor de luz)

EL módulo tiene un regulador interno de 3.3V pudiendo alimentar con 5V sin problemas.

La interfaz de comunicación es I2C pudiéndolo implementar en la mayoría de microcontroladores, el módulo aparte de los pines de alimentación y pines I2C tiene un pin para establecer la dirección.

Se tienen dos direcciones para poder trabajar:

Tabla 5.1 Pines de establecimiento de conexión.

Pin ADDR	Dirección I2C
ADDR=HIGH (5V)	0x5C
ADDR=LOW (GND o NC)	0x23

Para utilizar el sensor de luz se utiliza la comunicación I2C como se muestra en la tabla 5.2.

Tabla 5.2 conexión I2C de sensor de luz a Raspberry Pi.

Module PCB	Desc	GPIO Header Pins
GND	Ground	P1-06

ADD	Address select	P1-06
SDA	I2C SDA	P1-03
SCL	I2C SCL	P1-05
VCC	3.3V	P1-01

### Script en Python para leer el nivel de luz del módulo.

```
#!/usr/bin/python
import smbus
import time

# Define some constants from the datasheet

DEVICE      = 0x23 # Default device I2C address

POWER_DOWN  = 0x00 # No active state
POWER_ON    = 0x01 # Power on
RESET       = 0x07 # Reset data register value

# Start measurement at 4lx resolution. Time typically 16ms.
CONTINUOUS_LOW_RES_MODE = 0x13
# Start measurement at 1lx resolution. Time typically 120ms
CONTINUOUS_HIGH_RES_MODE_1 = 0x10
# Start measurement at 0.5lx resolution. Time typically 120ms
CONTINUOUS_HIGH_RES_MODE_2 = 0x11
# Start measurement at 1lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_HIGH_RES_MODE_1 = 0x20
# Start measurement at 0.5lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_HIGH_RES_MODE_2 = 0x21
# Start measurement at 1lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_LOW_RES_MODE = 0x23

#bus = smbus.SMBus(0) # Rev 1 Pi uses 0
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

def convertToNumber(data):
    # Simple function to convert 2 bytes of data
```



```

# into a decimal number
return ((data[1] + (256 * data[0])) / 1.2)

def readLight(addr=DEVICE):
    data = bus.read_i2c_block_data(addr, ONE_TIME_HIGH_RES_MODE_1)
    return convertToNumber(data)

def main():

    while True:
        print "Light Level : " + str(readLight()) + " lx"
        time.sleep(0.5)

if __name__=="__main__":
    main()

```

### 5.3.4 Sensor de Humedad

El módulo HL-69 es un sensor de humedad de suelo. Es otro módulo que utiliza la conductividad entre dos terminales para determinar ciertos parámetros relacionados a agua, líquido y humedad. [4]



Figura 5.5 Modulo HL-69 (Sensor de humedad)

Consiste en dos placas separadas entre sí por una distancia determinada. Ambas placas están recubiertas de una capa de material conductor. Si existe humedad en el suelo se creará un puente entre una punta y otra, lo que será detectado por un circuito de control con un amplificador operacional que será el encargado de transformar la conductividad registrada a un valor analógico

Como Raspberry Pi no cuenta con entradas analógicas, se tendría que usar un convertidor analógico digital. Al utilizar la salida digital, ésta solo nos indicaría 1 y 0,

indicando si está húmedo o no, lo cual sería más conveniente usar el convertidor a través de una comunicación SPI por uno de los GPIO.

SPI significa interfaz periférico en serie y es un mecanismo que permite a los dispositivos compatibles para comunicarse con el Pi.

#### 5.3.4.1 Dispositivos analógicos de entrada

Los Convertidores analógico a digital son dispositivos de entrada analógicos típicos (ADC). Existen varias clases para los chips ADC específicos, como son el circuito MCP3004, MCP3008, MCP3204 y el MCP3208.

En este caso se utilizó el MCP3008, ya que era el más comercial y contaba con 8 entradas analógicas en dado caso que se ocuparan más lecturas de otros sensores que se pudieran acoplar al sistema.

En seguida se muestra la parte de programación incluyendo el convertidor analógico utilizado.

Las siguientes líneas de código nos indican la utilización de la librería del convertidor MCP3008 y el retardo que tendrá el sistema para hacer lectura de los datos y poder visualizarlo de tal forma que podamos leer el dato de temperatura en grados centígrados.

```
from gpiozero import MCP3008
from time import sleep

def convert_temp(gen):
    for value in gen:
        yield (value * 3.3 - 0.5) * 100

adc = MCP3008(channel=0)

for temp in convert_temp(adc.values):
    print('The temperature is', temp, 'C')
    sleep(1)
```

### 5.4 Conexiones de prueba

Se utilizan las siguientes partes para elaborar el circuito:

- Raspberry Pi 2
- MCP3008 Convertidor analógico digital
- HL-69 Sensor de Humedad

- BH1750 Sensor de luminosidad
- DS18B20 Sensor de Temperatura

Inicialmente se construyó un circuito en fritzing [5]

\*Fritzing es una iniciativa de hardware libre que hace que la electrónica accesibles como material creativo para cualquier persona. Ofrece una herramienta de software, un sitio web de la comunidad y los servicios en el espíritu de procesamiento y Arduino , fomentando un ecosistema creativo que permite a los usuarios documentar sus prototipos, comparten con otros, enseñan la electrónica en un salón de clases, y el diseño y fabricación de PCB profesionales.

Esto para poder visualizar antes de armar el circuito de la red de sensores con el Raspberry Pi

Pines GPIO

- BH1750
  1. SDA Pin a GPIO 3 BCM 3 (SDA)
  2. SCL Pin a GPIO 5 BCM 5(SCL)
  3. ADR Pin a GPIO 39 Tierra
  4. VCC 3.3V
  5. GND Tierra
  
- DS18B20
  1. VCC 3.3V
  2. DATA GPIO7 BCM4 (P1-7)
  3. GND Tierra
  
- MCP3008
  1. VDD 3.3V
  2. VREF 3.3V
  3. AGND GROUND
  4. CLK GPIO11 (P1-23)
  5. DOUT GPIO9 (P1-21)
  6. DIN GPIO10 (P1-19)
  7. CS GPIO8 (P1-24)
  8. DGND GROUND

En la figura 5.6 se muestra la imagen de los tres sensores acoplados al Raspberry Pi diseñados en "fritzing".

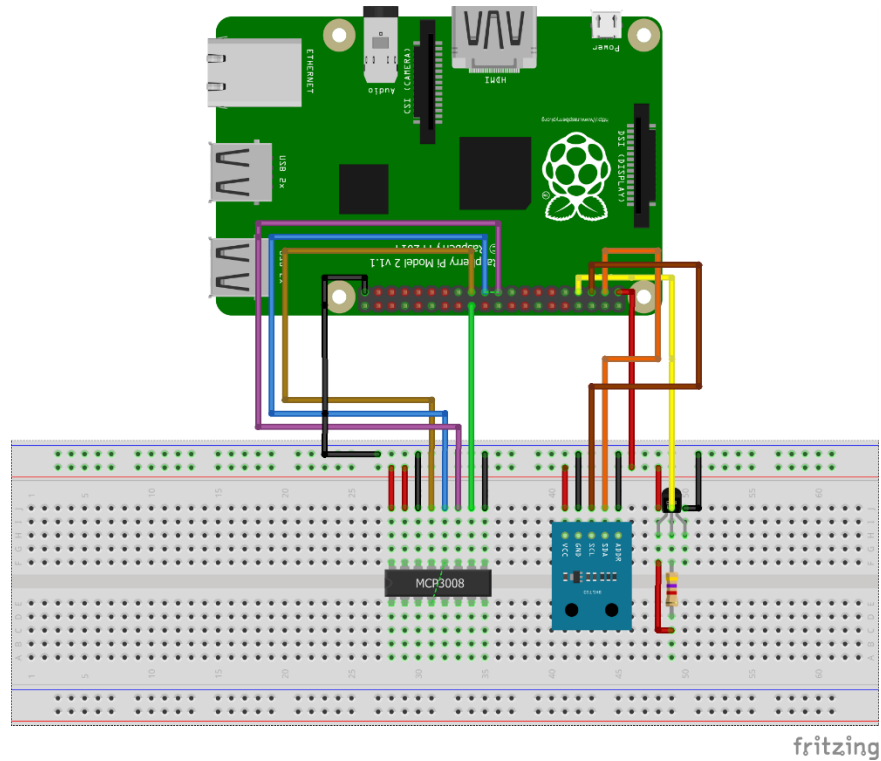


Figura 5.6. Esquema de conexión por cable de los sensores de humedad, luz y temperatura.

Después se construyó el circuito en Proteus sustituyendo las partes por conectores en Schematic Capture. En la figura 5.7 se muestra el diagrama eléctrico del circuito que incluyen los sensores de humedad, temperatura y luz, conectados a las pines GPIO de Raspberry Pi.

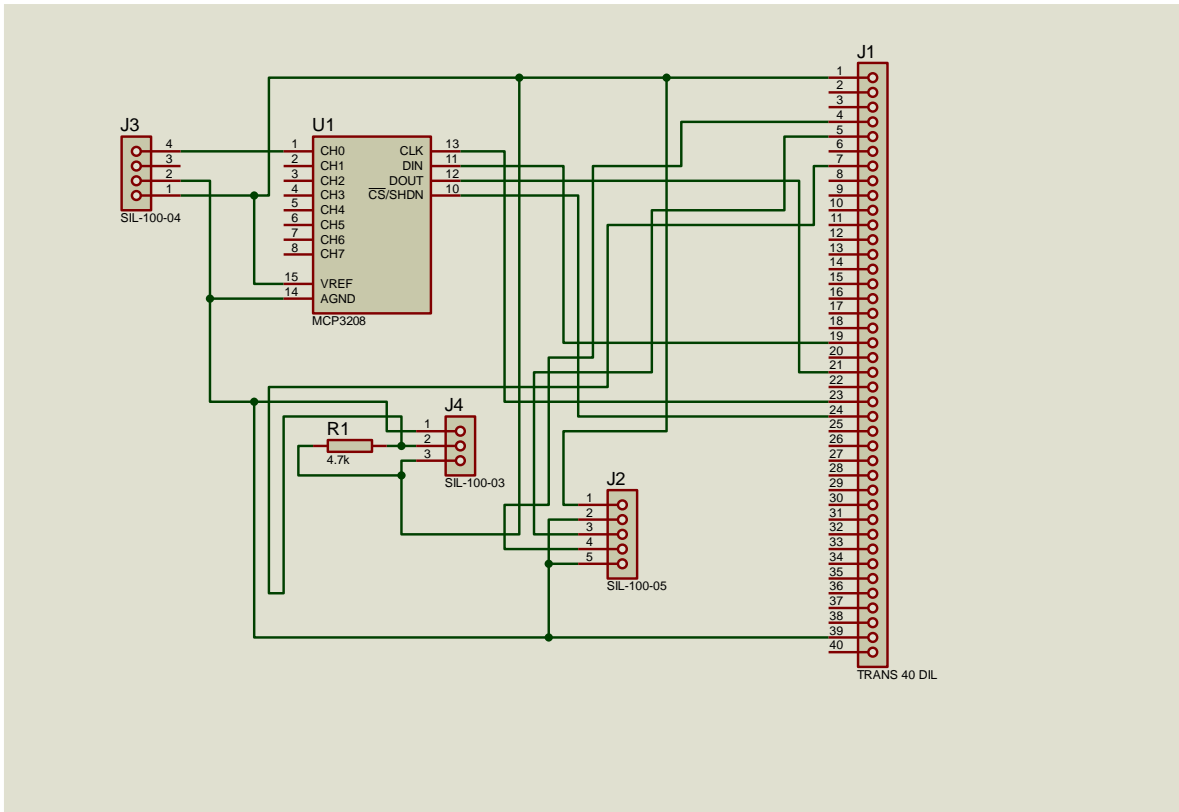


Figura 5.7. Esquema en Proteus de los sensores de humedad, temperatura y luz.

Se usó un Raspberry Pi 2 inicialmente para poder manipular el código del sensor. En la figura 5.8, se muestran 2 imágenes de las pruebas que se hicieron en el laboratorio donde se puede observar una maceta que contiene el sensor de humedad, el de temperatura y el de luz. La información obtenida se muestra en un monitor el cual se encuentra conectado al Raspberry Pi. Esto se puede observar en la figura 5.9.

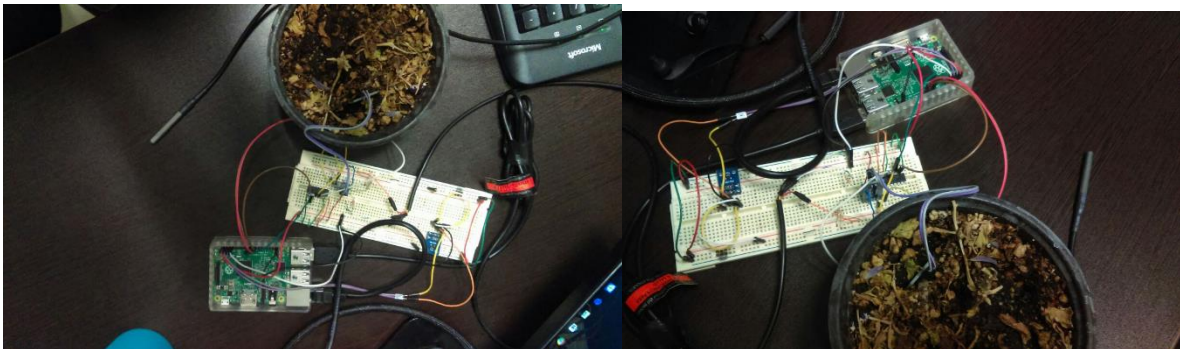


Figura 5.8. Pruebas en laboratorio, sensores de temperatura, humedad y luz

En la imagen de la figura 5.9 se muestran los resultados de humedad en porcentaje, de cuanta humedad se encuentra en la zona, la luminosidad en luxes y la temperatura en grados centígrados

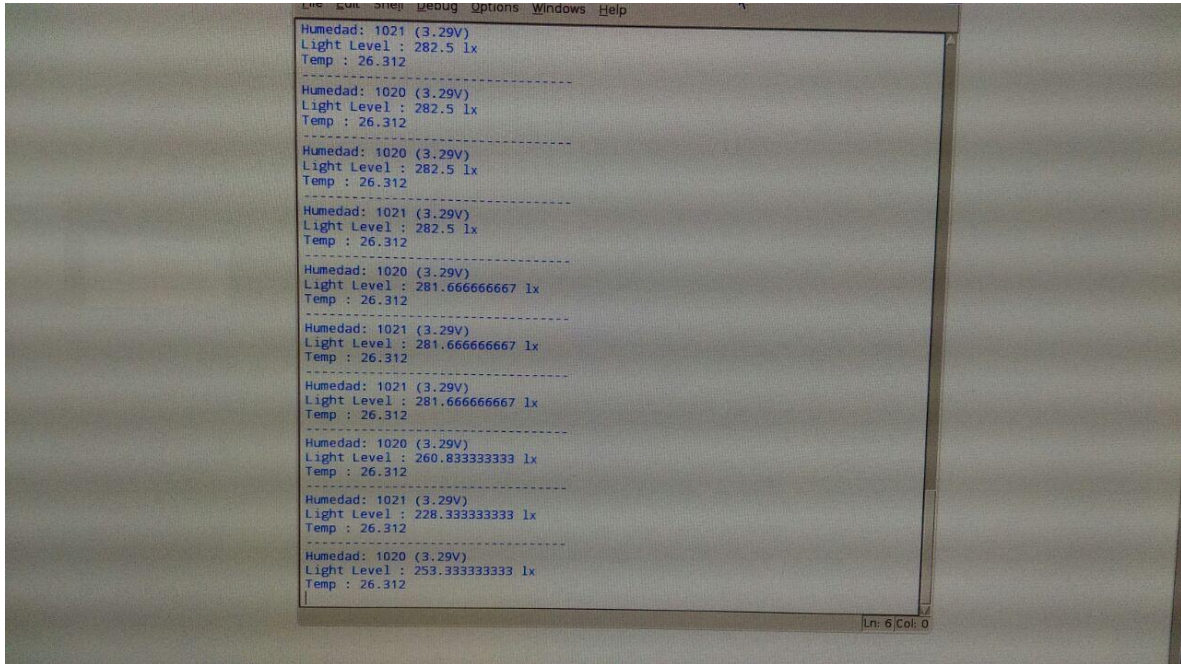


Figura 5.9. Monitor con los resultados. Estos datos están siendo enviados por los sensores de humedad, luz y temperatura a un monitor conectado al Raspberry Pi.



Figura 5.10.Raspberry Pi Zero.

Una vez echas las primeras pruebas con los sensores y un monitor, se utilizó Raspberry Pi Zero para hacer las primeras pruebas de campo a éste, además se le acopla una pantalla de cristal líquido (LCD)) para visualizar los datos sin necesidad de la pantalla. Esto requiere una nueva configuración para que la Raspberry Pi para que pueda mostrar en la LCD las mediciones de los sensores de la misma manera que se mostraba en la pantalla.

Para poder hacer las conexiones de la LCD se identificaron los pines no utilizados y los pines que estuvieran causando interferencia en las comunicaciones abiertas i2c, SPI y 1-wire.

Se utilizan las siguientes partes para elaborar el circuito nuevamente incluyendo la LCD:

- Raspberry Pi Zero
- MCP3008 Convertidor analógico digital
- HL-69 Sensor de Humedad
- BH1750 Sensor de luminosidad
- DS18B20 Sensor de Temperatura
- LCD

De igual manera se construyó un circuito en "fritzing" para visualizar las conexiones

#### Pines GPIO

- BH1750
  - SDA GPIO 3 (SDA) (P1-3)
  - SCL GPIO 5 (SCL)(P1-5)
  - ADR Tierra
  - VCC 3.3V
  - GND Tierra
  
- DS18B20
  - VCC 3.3V
  - DATA GPIO7 BCM4 (Data)(P1-7)
  - GND Tierra
  
- MCP3008
  - VDD 3.3V
  - VREF 3.3V
  - AGND GROUND
  - CLK GPIO11 (P1-23)
  - DOUT GPIO9 (P1-21)
  - DIN GPIO10 (P1-19)
  - CS GPIO8 (P1-24)
  - DGND GROUND
  
- LCD PIN

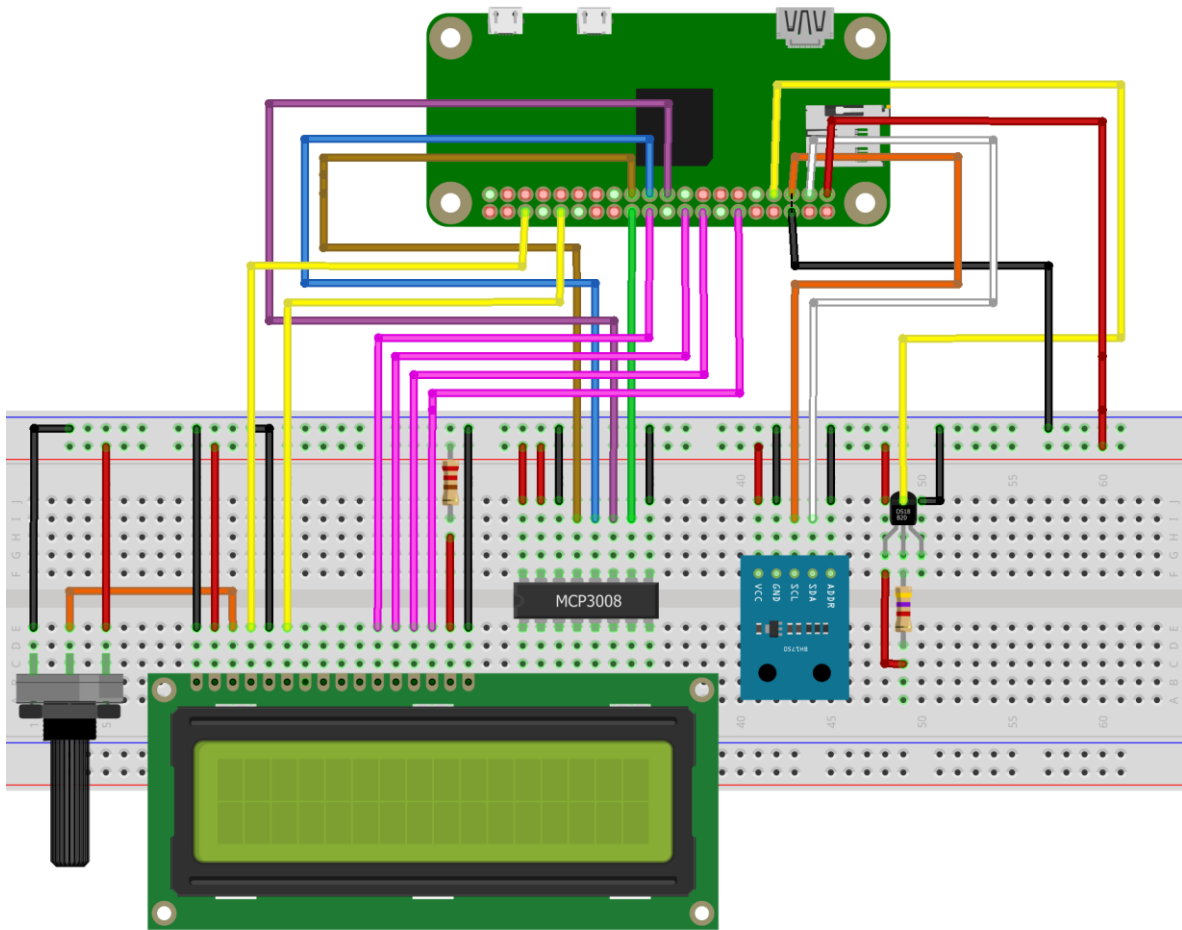
Tabla 5.3. Asignación de pines de conexión LCD a Raspberry Pi utilizada para el sistema de monitoreo

LCD Pin	Function	Pi Function	Pi Pin
01	GND	GND	P1-06



02	+5V	+5V	P1-02
03	Contrast	GND	P1-06
04	RS	GPIO7	P1-26
05	RW	GND	P1-06
06	E	GPIO8	P1-24
07	Data 0		
08	Data 1		
09	Data 2		
10	Data 3		
11	Data 4	GPIO25	P1-22
12	Data 5	GPIO24	P1-18
13	Data 6	GPIO23	P1-16
14	Data 7	GPIO18	P1-12
15	+5V via 560ohm		
16	GND		P1-06

En la figura 5.11 se muestra la imagen del diseño en "fritzing" con todos los sensores ya antes mencionados además de una LCD.



fritzing

Figura 5.11. Imagen se acoplan los sensores de temperatura, humedad y luz en una Raspberry Pi Zero la cual incluye una LCD 16x2

Una vez implementado el sistema completo, se procedió a probarlo y comprobar su funcionamiento. Las primeras pruebas se hicieron en un campo cerca de Empalme, en estas pruebas se usó un banco de baterías para poder utilizar el sistema. En la figura 5.12 se muestran las fotografías de las pruebas realizadas.



Figuras 5.12. Primeras pruebas en el campo. En la LCD se despliegan las mediciones de los sensores de temperatura, luminosidad y humedad.

El segundo viaje se realizó en el campo de UNISON donde también se llevó un luxómetro y un anemómetro. Con el luxómetro se podía observar la luminosidad, esto para comparar los resultados obtenidos con el circuito desarrollado en este proyecto. El anemómetro se utilizó para checar velocidad del viento en ese punto. Cabe mencionar que éste no está incluido en el circuito pero que será una propuesta para incluirlo al sistema. La imagen que se encuentra en la figura 13 toma las medidas conforme al ángulo de posición en donde se encuentra y sus medidas son en candelas.



Figura 5.13. Campos de alfalfa de UNISON.

En las tablas 5.4 y 5.5 se muestra los resultados obtenidos por el nodo de sensores de temperatura, humedad, luz, además se utiliza un luxómetro y un anemómetro comercial para comprar los datos obtenidos. Este registro de datos fue tomado de los campo de alfalfa y de naranjos de la UNISON. En los campos se tomaron las muestras con un solo nodo, por lo cual nos teníamos que desplazar con éste a diferentes puntos para obtener las mediciones.

Tabla 5.4 Pruebas de campo en alfalfa, datos de los sensores de temperatura, humedad, intensidad de luz, luminosidad y velocidad del viento.

Pruebas campo 1, alfalfa, 1 año				
°C	%	cd	lux	km/h
temperatura	humedad	intensidad	luxometro	anemometro
21.75	26	38505	660	1.8
21.68	28	51095		
22.8	28	46151	200	3.6
22.4	28.7	14056		
30.8	98	51995	540	3.9
30.68	98	47878		
24.7	57	54212	560	7.2
24.8	58	54212		
21.37	27	11540	560	5.6
21.2	28	11632		

25.56	21	12816	400	3.6
25.43	22	13180		
26	35	47796	530	9
25	39.4	49.557		
23	93	38491	800	0
23	82	52990		
24.3	37	53900	795	4
24.18	42	54612		

Tabla 5.5 Pruebas de campo en naranjos, datos de los sensores de temperatura, humedad, intensidad de luz, luminosidad y velocidad del viento.

Pruebas campo 2, naranja				
°C	%	cd	lux	km/h
temperatura	humedad	intensidad	luxometro	anemometro
30	95	46800	600	2
30	95	51298		
31	96	17786	200	3.9
20.8	95	12212		
25	88	1870	1	45
24.75	88	2600		
26	98	8386	3.9	96
26.6	98	7537		
23.1	91	3543	7.2	70
22.8	91	3499		
26.2	91.5	4295	2.1	100
26.1	87.6	2095		
29.3	96	5585	1.8	94
29.3	96	5685		

## 5.5 Referencias

- [1] <https://pinout.xyz/resources/raspberry-pi-pinout.png>
- [2] <https://www.adafruit.com/product/381>
- [3] [http://www.naylampmechatronics.com/blog/44\\_Tutorial-m%C3%B3dulo-sensor-de-luz-BH1750.html](http://www.naylampmechatronics.com/blog/44_Tutorial-m%C3%B3dulo-sensor-de-luz-BH1750.html)
- [4] <http://www.picaxe.biz/tienda/producto/HL-69/1/sensor-de-humedad-hl-69-c93->
- [5] <http://fritzing.org/home/>

## Capítulo 6

---

### Conclusiones

- El sistema propuesto sirve para controlar el riego, la luz y otros factores que beneficiarán al agricultor para minimizar sus pérdidas y tener mejor controlada su producción.
- Tener un área controlada sería de gran ayuda ya que las mediciones en campo abierto varían mucho.
- Los sensores utilizados bajan mucho el costo de otras redes de sensores comerciales.
- Utilizar los códigos de estos sensores y acoplarlos al sistema propuesto fue sencillo, debido a la colaboración con integrantes del grupo de trabajo que ya los han utilizado con anterioridad. Se cuenta con suficiente información de los sensores y dispositivos propuestos, además varias personas están colaborando en foros que tienen los mismos problemas que surgen a medida que se va avanzando en la programación

### Desarrollo de trabajos futuros

- Al presente trabajo le faltó desarrollar el aérea de comunicación y energía. El acoplamiento de Xbee y crear un módulo capaz de almacenar los datos y poderlos observar en cualquier dispositivo con internet.
- Agregar otros tipos de sensores como el anemómetro (sensor de aire), y otras condiciones climáticas que pudieran proporcionar datos a tiempo real de tal manera que se pudiera evitar pérdidas por las malas condiciones.
- Acoplar este sistema a drones que interactúen con los nodos para obtener más información del campo por medio de las cámaras infrarrojas o multi-espectrales.
- Ampliar horizontes para que este tipo de redes protejan flora y fauna de zonas de riesgo.

## Apéndice A

---

### Código de programación en python para Raspberry Pi

```
# ds18b20.py
def gettemp(id):
    try:
        mytemp = ""
        filename = 'w1_slave'
        f = open('/sys/bus/w1/devices/' + id + '/' + filename, 'r')
        line = f.readline() # read 1st line
        crc = line.rsplit(' ',1)
        crc = crc[1].replace('\n', "")
        if crc=='YES':
            line = f.readline() # read 2nd line
            mytemp = line.rsplit('t=',1)
        else:
            mytemp = 99999
        f.close()

        return int(mytemp[1])

    except:
        return 99999

#import
import RPi.GPIO as GPIO
import time
import smbus
import time
import spidev
import os

# Open SPI bus
spi = spidev.SpiDev()
spi.open(0,0)

# Function to read SPI data from MCP3008 chip
```



```

# Channel must be an integer 0-7
def ReadChannel(channel):
    adc = spi.xfer2([1,(8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data
def ConvertVolts(data,places):
    volts = (data * 3.3) / float(1023)
    volts = round(volts,places)
    return volts

hum_channel = 0

# Define GPIO to LCD mapping
LCD_RS = 16
LCD_E  = 12
LCD_D4 = 25
LCD_D5 = 24
LCD_D6 = 23
LCD_D7 = 18

DEVICE    = 0x23 # Default device I2C address

POWER_DOWN = 0x00 # No active state
POWER_ON   = 0x01 # Power on
RESET      = 0x07 # Reset data register value

# Start measurement at 4lx resolution. Time typically 16ms.
CONTINUOUS_LOW_RES_MODE = 0x13
# Start measurement at 1lx resolution. Time typically 120ms
CONTINUOUS_HIGH_RES_MODE_1 = 0x10
# Start measurement at 0.5lx resolution. Time typically 120ms
CONTINUOUS_HIGH_RES_MODE_2 = 0x11
# Start measurement at 1lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_HIGH_RES_MODE_1 = 0x20
# Start measurement at 0.5lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_HIGH_RES_MODE_2 = 0x21
# Start measurement at 1lx resolution. Time typically 120ms
# Device is automatically set to Power Down after measurement.
ONE_TIME_LOW_RES_MODE = 0x23

```

```

#bus = smbus.SMBus(0) # Rev 1 Pi uses 0
bus = smbus.SMBus(1) # Rev 2 Pi uses 1

def convertToNumber(data):
    # Simple function to convert 2 bytes of data
    # into a decimal number
    return ((data[1] + (256 * data[0])) / 1.2)

def readLight(addr=DEVICE):
    data = bus.read_i2c_block_data(addr,ONE_TIME_HIGH_RES_MODE_1)
    return convertToNumber(data)

# Define some device constants
LCD_WIDTH = 16 # Maximum characters per line
LCD_CHR = True
LCD_CMD = False

LCD_LINE_1 = 0x80 # LCD RAM address for the 1st line
LCD_LINE_2 = 0xC0 # LCD RAM address for the 2nd line

# Timing constants
E_PULSE = 0.0005
E_DELAY = 0.0005

def main():
    # Main program block

    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM) # Use BCM GPIO numbers
    GPIO.setup(LCD_E, GPIO.OUT) # E
    GPIO.setup(LCD_RS, GPIO.OUT) # RS
    GPIO.setup(LCD_D4, GPIO.OUT) # DB4
    GPIO.setup(LCD_D5, GPIO.OUT) # DB5
    GPIO.setup(LCD_D6, GPIO.OUT) # DB6
    GPIO.setup(LCD_D7, GPIO.OUT) # DB7

    # Initialise display
    lcd_init()

```

```

while True:
    # Read the hum sensor data
    hum_level = ReadChannel(hum_channel)
    hum_volts = ((ConvertVolts(hum_level,2)*100)/3.3)

    # Send some test
    lcd_string("AgroPro",LCD_LINE_1)
    lcd_string("LCD Test",LCD_LINE_2)

    time.sleep(3) # 3 second delay

    # Send some light
    lcd_string("Light Level : " ,LCD_LINE_1)
    lcd_string(str(readLight()) + " lx",LCD_LINE_2)

    time.sleep(3) # 3 second delay

    # Send temp
    id = '28-041662ebd6ff'
    lcd_string("Temperatura : ",LCD_LINE_1)
    lcd_string('{:.3f}'.format(gettemp(id)/float(1000)),LCD_LINE_2)

    time.sleep(3)

    # Send some text
    lcd_string("Humedad:",LCD_LINE_1)
    lcd_string("{} ({}%)".format(hum_level,hum_volts),LCD_LINE_2)

    time.sleep(3)

def lcd_init():
    # Initialise display
    lcd_byte(0x33,LCD_CMD) # 110011 Initialise
    lcd_byte(0x32,LCD_CMD) # 110010 Initialise
    lcd_byte(0x06,LCD_CMD) # 000110 Cursor move direction
    lcd_byte(0x0C,LCD_CMD) # 001100 Display On,Cursor Off, Blink Off
    lcd_byte(0x28,LCD_CMD) # 101000 Data length, number of lines, font size
    lcd_byte(0x01,LCD_CMD) # 000001 Clear display
    time.sleep(E_DELAY)

def lcd_byte(bits, mode):

```

```
# Send byte to data pins
# bits = data
# mode = True for character
#     False for command

GPIO.output(LCD_RS, mode) # RS

# High bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x10==0x10:
    GPIO.output(LCD_D4, True)
if bits&0x20==0x20:
    GPIO.output(LCD_D5, True)
if bits&0x40==0x40:
    GPIO.output(LCD_D6, True)
if bits&0x80==0x80:
    GPIO.output(LCD_D7, True)

# Toggle 'Enable' pin
lcd_toggle_enable()

# Low bits
GPIO.output(LCD_D4, False)
GPIO.output(LCD_D5, False)
GPIO.output(LCD_D6, False)
GPIO.output(LCD_D7, False)
if bits&0x01==0x01:
    GPIO.output(LCD_D4, True)
if bits&0x02==0x02:
    GPIO.output(LCD_D5, True)
if bits&0x04==0x04:
    GPIO.output(LCD_D6, True)
if bits&0x08==0x08:
    GPIO.output(LCD_D7, True)

# Toggle 'Enable' pin
lcd_toggle_enable()
```

```
def lcd_toggle_enable():
    # Toggle enable
    time.sleep(E_DELAY)
    GPIO.output(LCD_E, True)
    time.sleep(E_PULSE)
    GPIO.output(LCD_E, False)
    time.sleep(E_DELAY)

def lcd_string(message,line):
    # Send string to display

    message = message.ljust(LCD_WIDTH," ")

    lcd_byte(line, LCD_CMD)

    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]),LCD_CHR)

if __name__ == '__main__':

    try:
        main()
    except KeyboardInterrupt:
        pass
    finally:
        lcd_byte(0x01, LCD_CMD)
        lcd_string("Goodbye!",LCD_LINE_1)
        GPIO.cleanup()
```