

## COMUNICACIÓN DE SENSOR MQ-135 A INTERNET, Y RESPUESTA ONLINE

### INTRODUCCION

El siguiente proyecto fue con propósito de concluir el examen práctico de Instrumentación II, de la carrera de ICE en IPN ZACATENCO.

Las siguientes referencias se idealizara que contienen información de partes de mismo proyecto, pero complicadas porqué ejercen conocimientos en áreas específicas no proyectadas en los cursos de la especialidad de electrónica. Sin embargo se desarrolló así, porque la creación de manera particular de electrónica "IOT" en la modernidad está muy basada en código abierto y programas específicos, pero que no dejan modificar sus estructuras. Por entenderse así se buscó la creación de una página diseñada por el usuario, que podría ser usada genéricamente no solo por los instrumentos en este proyecto. *Por lo cual se propone la distribución de este reporte de manera digital, para seguir los vides tutoriales insertados y comprender a detalle dichos procesos.*

### Teoría e Instalaciones

#### Qué es Internet de las cosas?

El Internet de las Cosas o IoT (por sus siglas en inglés: Internet of Things) es un término que se refiere a la conexión de los objetos que utilizamos todos los días a Internet, no sólo hablamos de computadoras y celulares, sino también de electrodomésticos, automóviles y básicamente cualquier artículo de uso diario que puede conectarse a una red.

#### ¿Cómo funciona?

El IoT funciona gracias a otras tecnologías inalámbricas, que permiten que todos los dispositivos se conecten a Internet y entre sí. Las más conocidas son el Wi-Fi, Bluetooth, NFC y RFID. (Que Es IOT?, 2017)

#### Qué tarjeta de desarrollo elegir?

Antes de Arduino y Raspberry Pi, las cosas eran más complicadas. De cara al futuro, las cosas no sólo son complicadas de nuevo — son desconcertantes. Ahora estamos viendo una explosión de nuevas tarjetas que salen al mercado, y no hay razón para esperar que la tendencia se desacelere en el próximo año o dos. En todo caso, se espera que aparezcan más nuevas tarjetas, no menos — aunque la mayoría de ellas van a desaparecer con la misma rapidez

#### Alternativas de Arduino

El fenómeno Arduino y sus derivados ha conducido hacia una innovación interesante, y una convergencia en el mercado del microcontrolador.

- **Launchpad MSP430**
- PICAXE contraataca
- Wiring
- Netduino
- Parallax Propeller

### Raspberry Pi

La computadora de tarjeta reducida con Linux, existía mucho antes de la llegada de la Raspberry Pi; por ejemplo las tarjetas Gumstix. Recientemente, sin embargo, al igual que Arduino lo hizo

anteriormente, la Raspberry Pi solo ha reiniciado el mercado, esta vez para las computadoras de tarjeta reducida (SBC's). También al igual que Arduino, se ha generado una explosión de posibles competidores.

A diferencia de la Arduino, la Raspberry Pi nunca fue diseñada como una plataforma para makers. Pero el bajísimo precio de \$35 dólares ha creado un gran mercado para las placas computadoras de la noche a la mañana.

Fue diseñada desde el principio como una plataforma de bajo costo para que los niños aprendieran programación — una herramienta educativa barata. Pese a eso, más que a causa de ello, miles de proyectos creativos de computación embebidos se están construyendo en torno a la tarjeta. Al igual que con el Arduino, es la floreciente comunidad de la Pi lo que la ha hecho exitosa.

#### *Como empezar a conocer Raspberry*

Para tener la Raspberry Pi lista para usar Python no necesitamos hacer nada, ya que viene incluido de serie. Para usarla como plataforma para HTML, CSS y JavaScript creo que lo más interesante puede ser tener una SD instalada con Coder de Google, o añadir Coder a nuestro sistema operativo ya existente.

No sé el misterio completo, pero el apellido de la Raspberry, vamos el famoso Pi se lo pusieron por el lenguaje de programación Python.

#### *Lenguaje de Programación Python*

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

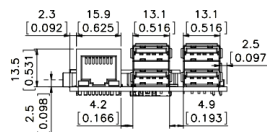
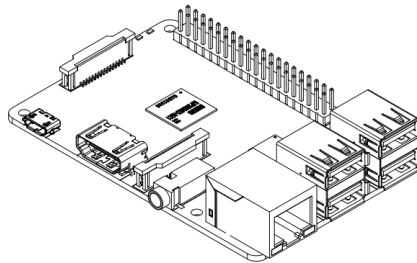
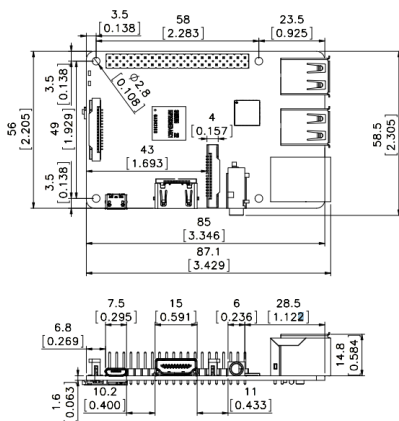
El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, <http://www.python.org/>, y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

(Rossum, 2009)

#### *Especificación técnica*

La Raspberry Pi 3 es la tercera generación de Raspberry Pi. Reemplazo a la Raspberry Pi 2 Modelo B Es Febrero de 2016.

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth baja energia (BLE) incluido
- 40-pin extended GPIO
- 4 USB 2 puertos
- 4 Puertos de salida estéreo y componentes de video HDMI
- CSI Puerto para conexión de cámara a Raspberry Pi cámara
- DSI conexión para pantalla a Raspberry Pi pantalla táctil.
- Micro SD puerto para cargar un sistema operativo y almacenaje
- De datos con un interruptor Micro USB fuente de poder de 2.5<sup>a</sup>



(MOCQ, 2016, Enero)

## Instalación

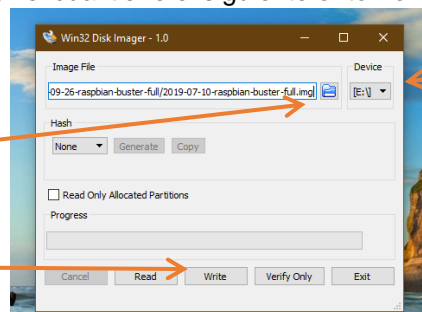
Una vez adquirida la Raspberry®, se necesitara la instalación del sistema operativo en la tarjeta MicroSD. Para ello se dejaron los pasos en las siguientes a desarrollar en las siguientes paginas

- Vamos a la página de <https://www.raspberrypi.org/downloads/raspbian/>  
Ahí encontraremos la versión más actualizada de Raspbian, lo que en particular recomiendo es la versión con Ordenador a seleccionar

En particular se recomienda el uso del siguiente programa para la instalación del sistema operativo, en mi experiencia es cómodo y el recomendado por la Fundación Pi, es más tardado qué este **“win32 disk maker”**, el cual tiene el siguiente entorno.

Aquí seleccionamos el archivo imagen descargado de la página oficial

Una vez con todo seleccionado procedemos a dar **“write”**



Seleccionamos la unidad a instalar la imagen del sistema operativo

Formatear la unidad en **“Fat 32”**

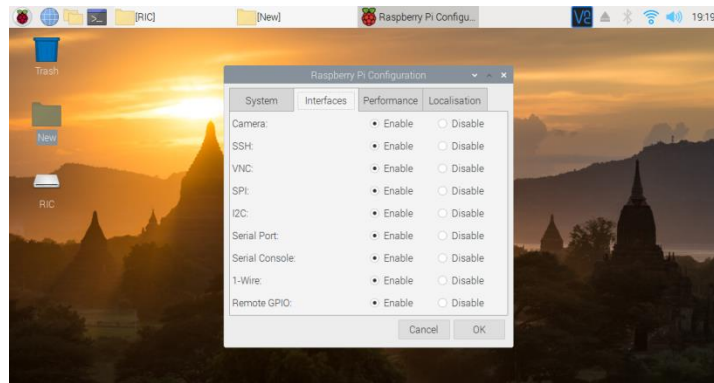
Una vez instalado este archivo, procedemos a conectar a una pantalla (para ello es vital el uso de una fuente de alimentación de 5 V. a 2.5 Amp mínimo)

En el siguiente video podrá ver una referencia sin usar el programa de instalación

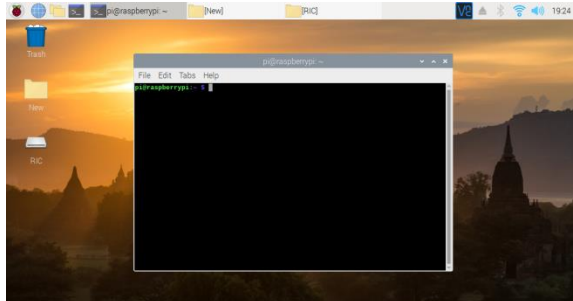
<https://www.youtube.com/watch?v=Vd3KFNzbRGM>

Cuando se abre por primera vez que inicia la Raspberry, se tendrá que configurar idioma y ciudad, es un proceso rápido. Recomiendo en enlace a una red Wifi, para actualizar el sistema y usar por un puerto virtual, un monitor de la Raspberry.

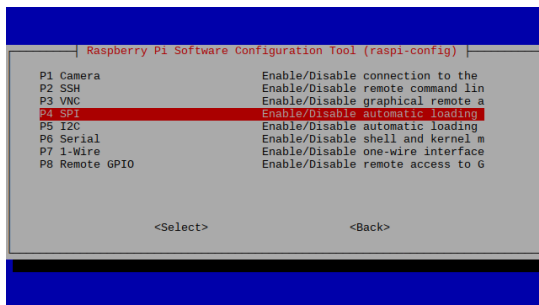
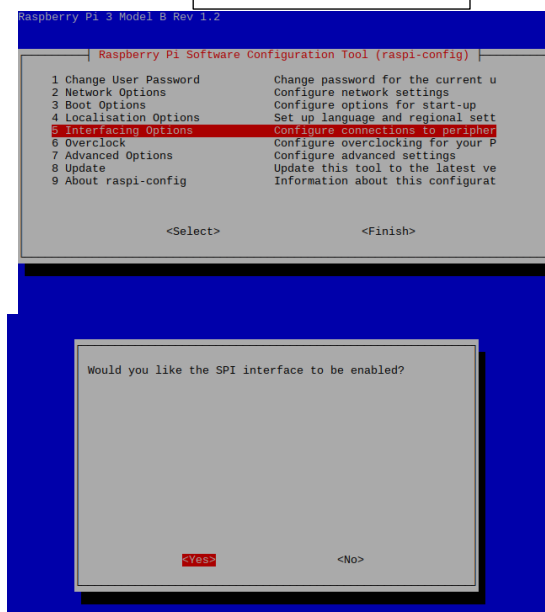
Prosiguiendo en los pasos veremos como abrir los puertos de configuración así en el escritorio vamos a símbolo del sistema -> Preference -> Interfaces. Ahí habilitamos lo que deseemos usar, pero el escritorio es activar el SSH



La otra opción es por medio de comandos, ahí abrimos la consola



sudo raspi-config



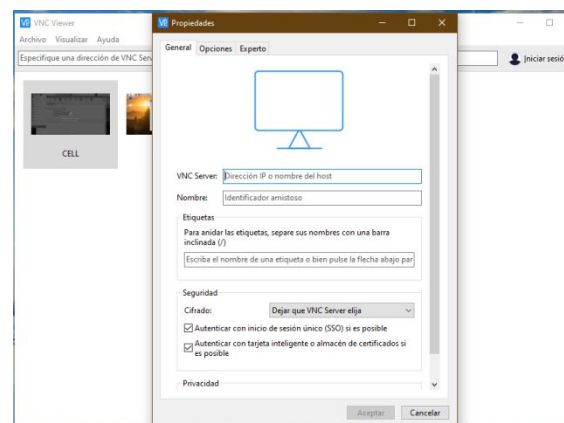
Así habilitamos de igual manera, serial y remote GPIO

Para el escritorio móvil se deberá descargar la versión de prueba de "VNC VIEWER", ahí solamente deberemos poner "NUEVA CONEXIÓN" y agregar el IP asignado a nuestra Raspberry. Cuando nos pida cuenta de usuario podemos por default

Usuario: pi

Contraseña : raspberry

A menos que sea cambiada por el usuario al inicio de la configuraciones



## Microcontrolador Texas Instruments, MSP430

Como sabemos un microcontrolador es un dispositivo de estado sólido que contiene un microcontrolador dentro pero con periféricos con el afán de automatizar procesos mediante programación. El especial el desarrollo con Texas Instruments nos da una serie de facilidades, como puertos analógicos y digitales, que a diferencia de Raspberry no se contemplan puerto analógico.

Como sabemos la mayoría de nuestros transductores físicos son de variable analógica lo cual podemos hacer el muestro por tiempo de las señales analógicas y procesarlas por conexión UART o SPI, a la Raspberry y comunicar lo que hemos leído en estos puertos, además de contener demás sistemas.

### Que es IAR system?

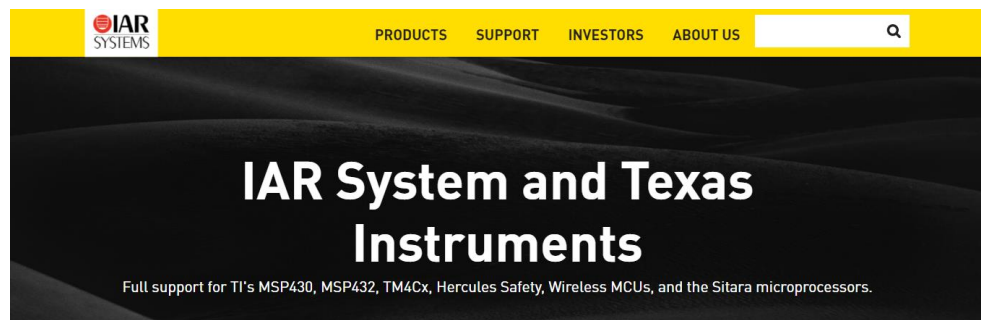
Están dedicados a proporcionar tecnología y servicios superiores, permitiendo a nuestros clientes crear los productos de hoy y las innovaciones del mañana.

Es una plataforma con asociación con empresas diversas, donde crean programas para interconexiones de diferentes Hardware y Software

## INSTALACION DE IAR system para Texas Ins.

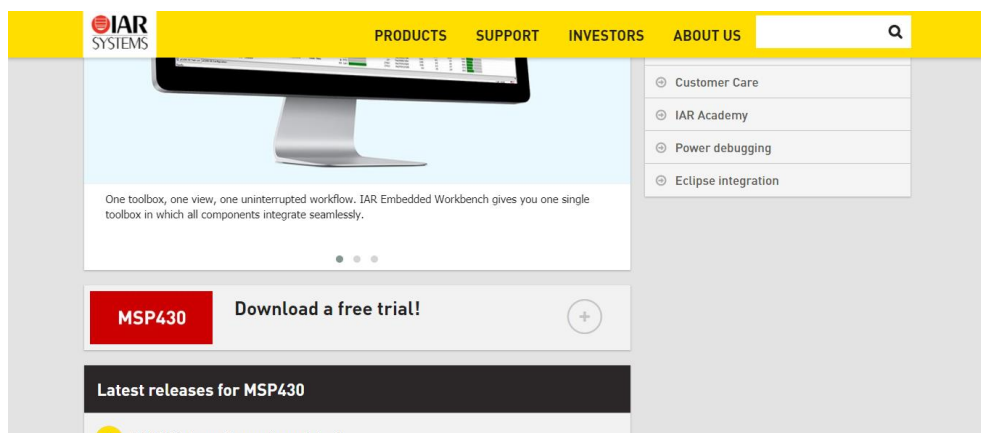
La instalación de programa **IAR**, se hará el proceso para la versión de prueba, para ello visitaremos la siguiente Página.

<https://www.iar.com/iar-embedded-workbench/partners/texas-instruments/>



Damos clic en la parte inferior en el recuadro **"MSP430"**

En la nueva página se baja hasta la parte inferior hasta el menú donde está la versión TRIAL.



## Condiciones de uso

La licencia de evaluación es completamente gratuita y le permite probar el entorno de desarrollo integrado y evaluar su eficiencia y facilidad de uso. Cuando inicie el producto por primera vez, se le pedirá que se registre para obtener su licencia de evaluación.

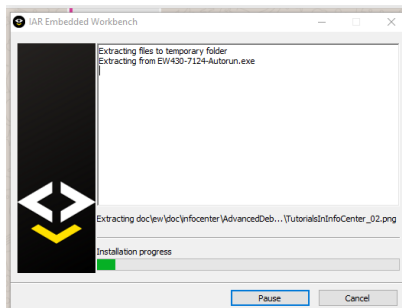
Después de la instalación, tiene las siguientes opciones de evaluación para elegir:

Una licencia de 30 días por tiempo limitado pero totalmente funcional

Una licencia de Kickstart de tamaño limitado sin límite de tiempo

Restricciones a la evaluación por tiempo limitado de 30 días.

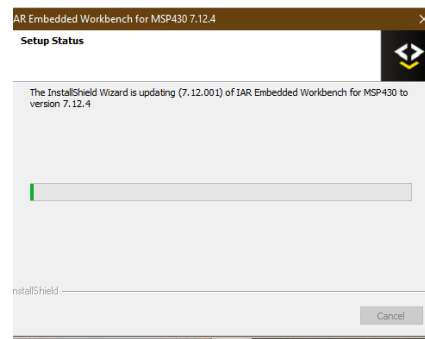
- Tras la descarga iniciara el proceso de instalación



Damos clic en Instalación



## \*Concedemos Permisos de Administrador



## Servidores Web.

El desarrollo del proyecto se utilizó las tecnologías más nuevas. Y todo el armado del proyecto fue por decisión propia y con todas las variables utilizadas se basaron en los cursos online tomados, pero solo fueron las bases.

Un servidor Web es un programa que utiliza el protocolo de transferencia de hiper texto, HTTP (Hypertext Transfer Protocol), para servir los archivos que forman páginas Web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras.

El servidor más conocido para métodos de desarrollo, es **Apache®. Org**

**<http://www.apache.org/>**

Como referencia dejar este video.

[https://www.youtube.com/watch?v=kfRguPngJwk&list=PLpOqH6AE0tNjh\\_agCrY02NIswODL2XBWL&index=2](https://www.youtube.com/watch?v=kfRguPngJwk&list=PLpOqH6AE0tNjh_agCrY02NIswODL2XBWL&index=2)

Es un servidor web, más básico. Pero al mismo tiempo el más sencillo de comprender, y utilizar para trabajar.

Sin embargo para estar en sintonía de los avances tecnológicos y para empezar a practicar con esta tecnología, se utilizó el servidor web **Node js** ®

<https://nodejs.org/es/>

Que es un servicio de compilación en lenguaje de programación **Json**, que aparte de los conceptos básicos, se puede agregar un servicio de motores de transformación.

Para su instalación del servicio web se deja el siguiente video.

[https://www.youtube.com/watch?v=jsPZIGTUIQQ&index=14&list=PLpOqH6AE0tNjh\\_agCrY02NISwODL2XBWL](https://www.youtube.com/watch?v=jsPZIGTUIQQ&index=14&list=PLpOqH6AE0tNjh_agCrY02NISwODL2XBWL)

### Creación de las páginas web.

El mismo curso señalado anteriormente en los primero 10 video te dan una introducción a programación de HTML 5, pero al igual se utiliza los objetos de desarrollo Responsable , en paginas web.

Como ejemplo el uso de CSS y Bootstrap, como desarrollo responsable o las animaciones que son parte de HTML5

Liga

[https://www.youtube.com/playlist?list=PLpOqH6AE0tNjh\\_agCrY02NISwODL2XBWL](https://www.youtube.com/playlist?list=PLpOqH6AE0tNjh_agCrY02NISwODL2XBWL)

[https://codigofacilito.com/cursos/primera\\_pagina](https://codigofacilito.com/cursos/primera_pagina)

### Base de Datos.

El uso de bases de datos, fue un poco compleja debido al uso de DOCUMENTOS a comparación de las típicas que son tabla de datos.

Para ello se utilizaron de nuevo los programas más nuevos y usados a nivel global.

Este programa de base de datos se llama Mongoose db ® , como nota extraordinaria se ejemplifican las principales compañías que lo utilizan.

Proceso de Instalación

[https://www.youtube.com/watch?v=lgrYE77pf1s&list=PLpOqH6AE0tNjh\\_agCrY02NISwODL2XBWL&index=14&t=3s](https://www.youtube.com/watch?v=lgrYE77pf1s&list=PLpOqH6AE0tNjh_agCrY02NISwODL2XBWL&index=14&t=3s)

[https://codigofacilito.com/videos/mi\\_primera\\_pagina\\_web\\_nodejs\\_instalar\\_mon](https://codigofacilito.com/videos/mi_primera_pagina_web_nodejs_instalar_mon)

Liga de Mongoose. <http://mongoosejs.com/docs/>

### El uso de Motor de Transformación

En el curso mencionado por la antigüedad usa un motor de transformación llamado Jade, pero el problema suscitado es que se actualizo y se cambió de nombre. El nuevo nombre es Pug , sin embargo tiene lo mismo usos e instalación. Alguna modificación que se tuvo que hacer fue correspondiente a la direccionamiento y al envío de datos por POST. En las referencias del código añadidas se puede encontrar las soluciones. Pero también se agrega en los comentarios del post.

[https://codigofacilito.com/videos/mi\\_primera\\_pagina\\_web\\_nodejs\\_crear\\_productos](https://codigofacilito.com/videos/mi_primera_pagina_web_nodejs_crear_productos)

[https://codigofacilito.com/videos/mi\\_primera\\_pagina\\_web\\_nodejs\\_subir\\_imagenes](https://codigofacilito.com/videos/mi_primera_pagina_web_nodejs_subir_imagenes)



## PUG

“El proceso de renderizado general de Pug es simple. pug.compile () compilará el código fuente de Pug en una función de JavaScript que toma un objeto de datos (llamado "locales") como argumento. Llame a esa función resultante con sus datos, y ¡voilà !, devolverá una cadena de HTML procesada con sus datos.”

<https://pugjs.org/api/getting-started.html>

## Sensor MQ135

### Data Sheet

<https://www.olimex.com/Products/Components/Sensors/Gas/SNS-MQ135/resources/SNS-MQ135.pdf>

## CARACTERISTICAS

Amplio alcance de detección Respuesta rápida y alta sensibilidad

Estable y larga vida Circuito de accionamiento simple

## SOLICITUD

Se utilizan en equipos de control de calidad del aire para edificios / oficinas, son adecuados para detectar

de NH<sub>3</sub>, NO<sub>x</sub>, alcohol, benceno, humo, CO<sub>2</sub>, etc.

En el proyecto se analizó en específico el GAS BUTANO, de los encendedores comunes. Curva característica

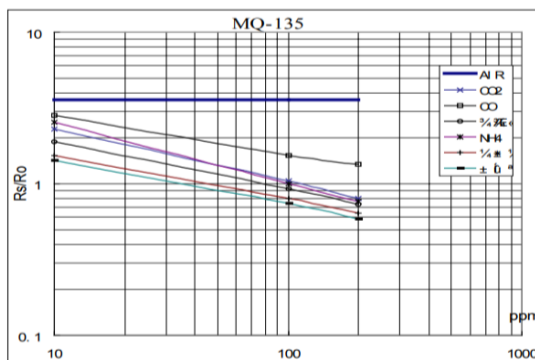


Fig.3 is shows the typical sensitivity characteristics of the MQ-135 for several gases. in their: Temp: 20 Humidity: 65% O<sub>2</sub> concentration 21% RL=20kΩ R<sub>0</sub>: sensor resistance at 100ppm of NH<sub>3</sub> in the clean air. R<sub>s</sub>: sensor resistance at various concentrations of gases.

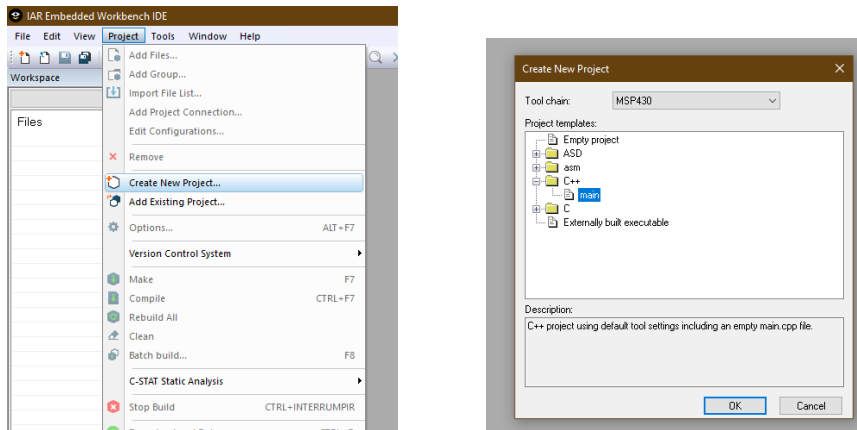
## Programación

### IAR System

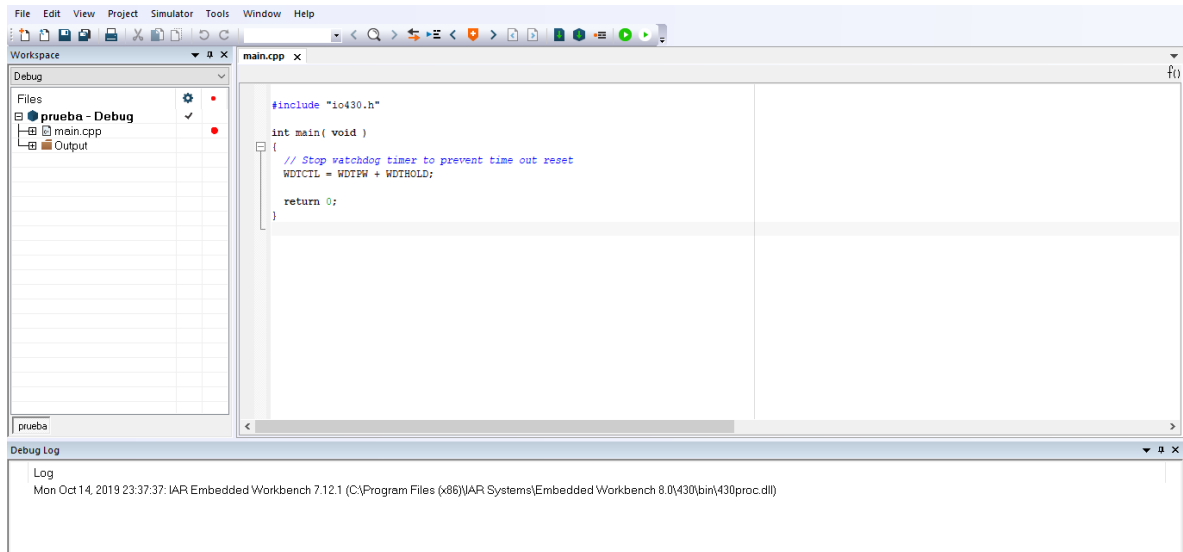
Una vez instalado el programa configuraremos primero el IAR y reconozca nuestro **M430G2553** DE NUESTRA LaunchPad MSP430.

- Damos en NEW FILE, en la ventana de FILE en la parte superior izquierda.
- Desplegara un proyecto en blanco, así que configuraremos el IAR

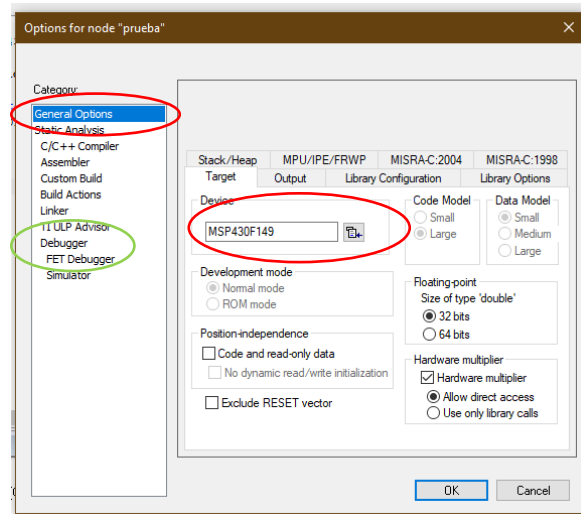
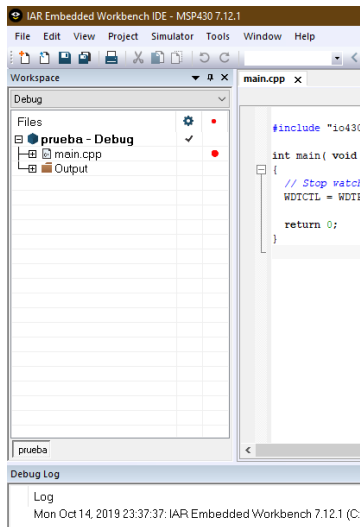




Tras seleccionar el Main de C++, le asignamos un nombre en alguna dirección de la computadora, se nos creara un entorno vacio pero ya con el "WorkSpace" realizado, por lo tanto es recomendable guardar dicho WorkSpace, dando clic en File-> Save Workspace. (recomendado en la misma ruta)



Ahora configuremos el entorno específico por lo cual accederemos a dar con el Boton secundario del ratón, sobre los archivos "prueba.debug" en el menu de la izquierda. Despues damos clic en **opcion** desplegando el siguiente menú



Como se puede apreciar en el menú **General option** podemos acceder a que dispositivo usaremos ahí buscaremos **M430G2553**

Después procederemos en el menú **Debugger** a verificar que este la opción en Driver de **FET DEBUGGER**.

Listo nuestro programa ya está habilitado para nuestro microcontrolador.

## Programación

### Texas Instruments

Para programar solamente se explicara mediante sentencias que hace cada parte, debido a que es un tema sencillo.

```
#include "io430.h"

int valores[9]={10,18,30,45,85,200,450,700};
volatile int f=0;
volatile int y=0;
volatile int m,i,z,c,n;
volatile float x=222;
char receiver=0;
int memory[10];

int actualizar(void)
{
  ADC10CTL0 |= ENC+ADC10SC; //Aqui activa el convertidor
  y=ADC10MEM;
  f=y/2;
  AJUSTE();
}

int AJUSTE(void)
{
  if(f>valores[0] && f<=valores[1])
  {
    x=(valores[1]-valores[0]);
    y=y-x;
    x=(100/x);
    y=y*x;
    if(f>valores[1] && f<=valores[2])
  }
```

Declaración de variables

Llamada de convertidor ADC, por método de programación orientada a objetos

Inicio de programa, con declaración de las interfaces, como habilitación Serial a 9600 Baup

Con los pines entre otros datos.

```

int main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDCTL = WDTPW + WDTHOLD;

    //DCO Trabajando a 1MHz aprox.
    BCSCTL1 = CALDC1_1MHZ;
    DCOCTL = CALDCO_1MHZ;

    //Configurando ADC
    ADC10CTL0 = SREF_1+ADC10SHT_3+REFON+ADC10ON+REF2_5V; //Configuracion de Pag 553 registro uno
    ADC10CTL1 = INCH_5+CONSEQ_0; // Canal de entrada 5 (un solo canal )
    ADC10AE0 |= BIT5;

    //*****Para cambiar los valores de BAUD , por cambio de los valores de freq interna del micro
    pagina 424*****

    P1SEL = BIT1 + BIT2 ; // Configurando las terminales P1.1 y P1.2
    P1SEL2 = BIT1 + BIT2 ; // como parte del sistema UART
    P1DIR &= ~BIT1; // P1.1 <-- RXD
    P1DIR |= BIT2; // P1.2 --> TXD
    UCA0CTL1 |= UCSSEL_2; // SMCLK
    //-----9600bps-----
    UCA0BR0 = 104; // 1MHz 9600
    UCA0BR1 = 0; // 1MHz 9600
    UCA0MCTL = UCBRS0; // Modulation UCBRSx = 1

```

```

// Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCIB0RX_VECTOR
__interrupt void USCIB0_ISR(void)
{
    receiver=UCA0RXBUF;
    // UCA0TXBUF=receiver; // CARGAR EL VALOR AL BUFFER DE TRANSMISION
    //while (!(IFG2&UCA0TXIFG)) // BANDERA QUE ESPERA A QU ACABE DE RECIBIR LOS DAT
    //{

    if(receiver=='s' || receiver=='S')
    {
        actualizar();
        send();
        P2OUT ^=BIT1;// apagar o prender
    }
}

```

Parte de  
habilitación de  
interrupción

```

#include "io430.h"

#include "CON_DEC_SERIAL.h"

char buffer; // variable de Rx,Tx

volatile int s,val;

int uno(void)
{
    ADC10CTL0 =
    SREF_0+ADC10SHT_3+REFON+ADC10ON;
    //Configuracion de Pag 553 registro uno

    ADC10CTL1 = INCH_3+CONSEQ_0;
    // Canal de entrada 5 (un solo canal )

    ADC10AE0 |= BIT3;

    ADC10CTL0 |= ENC+ADC10SC; //Aqui
    activa el convertidor

    __delay_cycles(190);

    val=ADC10MEM;

    s=val;
}

```

```

}

int dos(void)
{
    ADC10CTL0 =
    SREF_0+ADC10SHT_3+REFON+ADC10ON;
    //Configuracion de Pag 553 registro uno

    ADC10CTL1 = INCH_4+CONSEQ_0;
    // Canal de entrada 5 (un solo canal )

    ADC10AE0 |= BIT4;

    ADC10CTL0 |= ENC+ADC10SC; //Aqui
    activa el convertidor

    __delay_cycles(190);

    val=ADC10MEM;

    s=val;
}

int tres(void)
{
    ADC10CTL0 =
    SREF_0+ADC10SHT_3+REFON+ADC10ON;
    //Configuracion de Pag 553 registro uno

    ADC10CTL1 = INCH_5+CONSEQ_0;
    // Canal de entrada 5 (un solo canal )
}

```

```

ADC10AE0 |= BIT5;

ADC10CTL0 |= ENC+ADC10SC; //Aqui
activa el convertidor

__delay_cycles(190);

val=ADC10MEM;

s=val;
}

int cuatro(void)
{
    ADC10CTL0 =
    SREF_0+ADC10SHT_3+REFON+ADC10ON;
    //Configuracion de Pag 553 registro uno

    ADC10CTL1 = INCH_6+CONSEQ_0;
    // Canal de entrada 5 (un solo canal )

    ADC10AE0 |= BIT6;

    ADC10CTL0 |= ENC+ADC10SC; //Aqui
    activa el convertidor

    __delay_cycles(190);

    val=ADC10MEM;

    s=val;
}

```

|

```
}

int main( void )
{
    // Stop watchdog timer to prevent time
    out reset

    WDTCTL = WDTPW + WDTHOLD;

    BCSCCTL1 = CALBC1_12MHZ;

    DCOCTL = CALDCO_12MHZ;

    // CONFIGURACIONES DE PINES

    P2SEL |= (BIT6 | BIT7); ///SALIDAS

    P2SEL2 &= ~(BIT6 | BIT7); ///SALIDAS

    P1DIR |= BIT0;      // P1.0 (LED) como
    salida digital

    /*****Para cambiar
    los valores de BAUD , por cambio de los
    valores de freq interna del micro

    pagina 424*/

    P1SEL = BIT1 + BIT2 ;      //
    Configurando las terminales P1.1 y P1.2

    P1SEL2 = BIT1 + BIT2 ;      // como
    parte del sistema UART

    P1DIR &= ~BIT1;      // P1.1 <--
    RXD

    P1DIR |= BIT2;      // P1.2 -->
    TXD

    UCA0CTL1 |= UCSSEL_2;      //
    SMCLK

    //-----9600bps-----12 MHz

    UCA0BR0 = 1250;      //
    12MHz/9600=1250 (see User's Guide)

    UCA0BR1 = 0x04;      // 1250 =
    0x4e2

    UCA0MCTL = UCBRS_0+UCBRF_0;      //
    Modulation UCBRSx=0, UCBRFx=0

    UCA0CTL1 &= ~UCSWRST;      //
    **Initialize USCI state machine**

    IE2 |= UCA0RXIE;      // Enable
    USCI_A0 RX interrupt

    __bis_SR_register(GIE);

    while (1)

    {

    }

    return 0;
}

// Echo back RXed character, confirm TX
buffer is ready first

#pragma vector=USCIAB0RX_VECTOR

__interrupt void USCI0RX_ISR(void)
{
    buffer= UCA0RXBUF;    /// LEER EL
    VALOR AL BUFFER DE TRANSMISION

    P1OUT^=BIT0;

    if(buffer == '1' )

    uno();

    if(buffer == '2' )

    dos();

    if(buffer == '3' )

    tres();

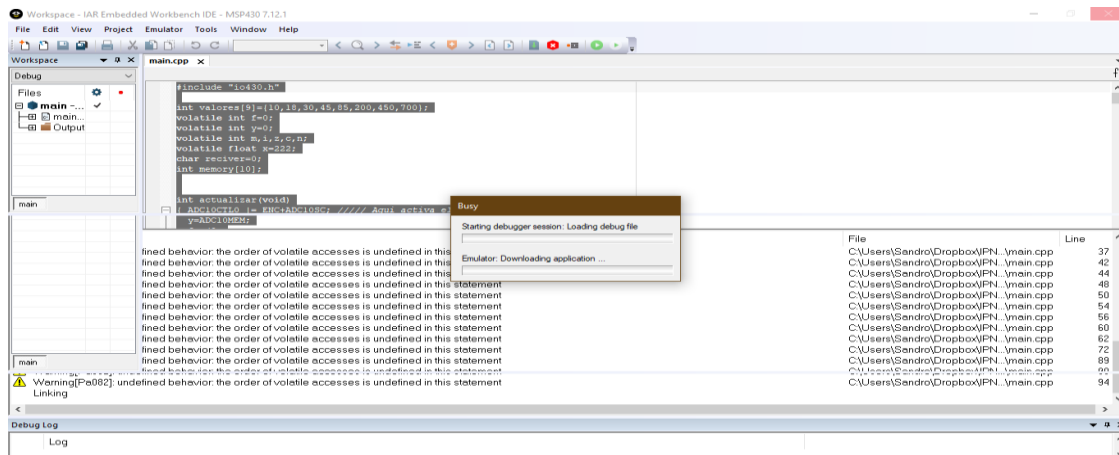
    if(buffer == '4' )

    cuatro();

    Send_Num(s);
}
```

Iniciar programa y comprobar

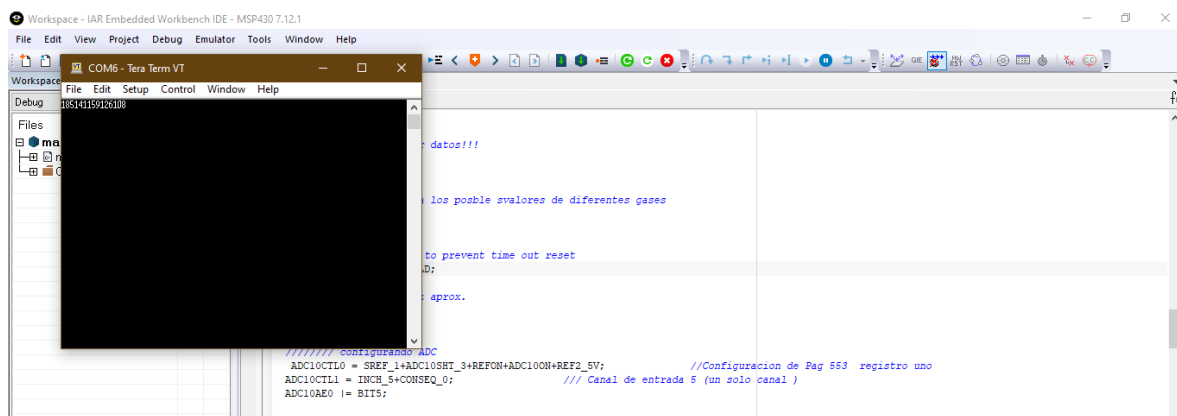
Ahora en la parte superior debe estar un botón tipo Play , le damos clic y el programa se enlaza con la Lauchpad previamente conectada, procede a cargar el programa.



Damos F5, o en la parte inferior un play pero en blanco y el programa debería estar corriendo.

Para comprobarlo se busca un lector de puertos COM, por ejemplo TeraTerm y vemos que datos esta recibiendo en el bus.

Podemos forzar la entra de datos mandando por ese puerto COM, el numero "1" , "2", "3" y "4", ahí activamos la lectura la entrada analógica.



## Json

La programación se realizó en **Json** compilado por **Node Js**, es a través del procesador de texto **Sublime Text 3**. Sin embargo en un reporte sería muy tedioso describir a detalle toda la programación porque es un trabajo demasiado largo de escribir, por ello lo que recomiendo que se vean video Tutoriales para que den una idea de cada función a detalle.

Pero a continuación se describirá el programa a ejecutar y algunas notas en cada sección resaltadas para referencias Online a las cuales dirigirse. Sí desea Clonar el archivo la liga es.

[https://github.com/ricsandro/IOT\\_Raspberry\\_ESIMEZAC\\_ICE.git](https://github.com/ricsandro/IOT_Raspberry_ESIMEZAC_ICE.git)

### Aquí son las variables instaladas por método npm

```
var express = require('express');
var mongoose = require('mongoose');
var bodyParser = require('body-parser');
var methodOverride = require('method-override');
var app = express();
```

### Aquí seleccionamos que tipo de base de datos usamos, el primero es la base de datos local para pruebas como se muestran en los videotutoriales, y el otro es la versión web obtenida en 1 minuto 22

```
//mongoose.connect('mongodb://localhost/prueba');
mongoose.connect('mongodb+srv://ricsando:.....Base
de datos-.....=majority');
```

### seleccionamos el numebre de tipo de codificación "json", y los motores de compilación para la interpretación de librerías

```
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended:true}));
app.set('view engine','pug');
app.set('port',process.env.PORT||8080)
app.use(express.static("public"));
app.use(methodOverride('_method'));
```

### Está función es para dar de alta en la base de datos un archivo, solo se necesita usar 1 vez, así que tras llenar la base "serial" con todos los nombres la "comentamos" para no usarla

```
app.get('/',function(sol,resp){
/*var data={
  serial:'s2',
  name:'.',
  valor:'0',
  actuador:'n',
}
var dates = new prog(data);
dates.save(function(err){
  console.log(dates);
});*/
```

```
    resp.render('index')
```

```
}}
```

### El render a las paginas web

```
app.get('/acerca',function(sol,resp){
    resp.render('acerca')
```

```
}}
```

### Solo son variables que se declaran para ser llenadas con los datos recibidos

```
var programSchema=
{
    serial: String,
    name: String,
    valor: String,
    actuador: String,
};
```

### Declaramos la variable de uso para heredar datos a mongoose

```
var prog= mongoose.model('prog',programSchema);
```

```
//impresion de Datos
```

```
{
```

```
1
```

<https://www.youtube.com/watch?v=yKWQKO4XU5I&t=1760s>

```
app.get('/sensado/general',function(sol,resp)
{
    var
id=sol.params.id;

    prog.findOne({'serial':'s1'||'s2'||'s3'||'s4'},function
(error,dates)
        {console.log(dates);

resp.render('sensado/general',{user:dates});
        });
});
app.get('/sensado/uno',function(sol,resp)
{

    prog.findOne({'serial':'s1'},function(error,dates)
        {console.log(dates);

resp.render('sensado/uno',{user:dates});
        });
});
app.get('/sensado/dos',function(sol,resp)
{

    prog.findOne({'serial':'s2'},function(error,dates)
        {console.log(dates);

resp.render('sensado/dos',{user:dates});
        });
});
app.get('/sensado/tres',function(sol,resp)
{

    prog.findOne({'serial':'s3'},function(error,dates)
        {console.log(dates);

resp.render('sensado/tres',{user:dates});
        });
});
app.get('/sensado/cuatro',function(sol,resp)
{

    prog.findOne({'serial':'s4'},function(error,dates)
        {console.log(dates);

resp.render('sensado/cuatro',{user:dates});
        });
});
}
```

### Parte medular de la IOT, en esta dirección web lo que recibimos es un código Json con estructura especifica que nuestro programa entenderá, en la variable cah será decodificado y llenado. Para su posterior actualización de la base de datos con update

```
////variable
```

```
{ var i=0;
```

```
    app.post("/catch/",function(sol,resp)
```

```
{
```

```
    var cah={
```

```
        serial: sol.body.serial,
```

```
        name: sol.body.name,
```

```
        valor: sol.body.valor,
```

```

    sol.body.actuador,
    };
    console.log('Datos
Recibidos');
    console.log(cah);

    prog.findOne({'serial':
sol.body.serial},function(err,idlog)
    {
        var tempo={
            name:cah.name,
            valor:cah.valor,
            actuador:cah.actuador,
        };

        prog.update({'serial':sol.body.serial},tempo,function(rellando){});

        prog.find({'serial':
sol.body.serial},function(erre,idloga){

            if(erre){console.log(erre);}

            else{console.log('datos Resguardados');}

        });

        console.log(err);
    });

    resp.render('catch/index',{data:cah});
});

}

app.get('/catch/index',function(sol,resp)
{resp.render('catch/index',{data:'s'});});

Si llegásemos a cometer el error de declarar una
variables con el mismo nombre ejemplo .. 2 archivos
guardados como "serial:s1" a la hora de buscar el
nombre, el programa puede entrar en conflicto por lo
cual adjunto esta función para borrar de la base de
datos los datos no deseados
//eliminar
var id;
{
app.get('/delete',function(sol,resp)
{

```

```

    prog.findOne({'serial':'s2'},function(error,dates)
    { id=dates._id;

    prog.remove({'_id':id},function(err)

    {if(err){console.log(err);}

    else{console.log('valor eliminado');}

    });
    });
}

Como acciones IOT, podemos entender que son lo
que el otro dispositivo (receptor) con conectarse a la
página web, podrá leer e interpretar lo que se este
mandando
//// acciones
var accion1='n';
var accion2='n';
var accion3='n';
var accion4='n';
{
    app.get( 'respuno', function ( req, res ) {
    prog.findOne({'serial':'s1'},function(error,dates){
        if(error){console.log(error);}
        else{//console.log('valores

a contestar');

        //console.log(dates.actuador);

        accion1=dates.actuador

        });
    res.send( accion1 );
    } );

    app.get( 'respdos', function ( req, res ) {
    prog.findOne({'serial':'s2'},function(error,dates){
        if(error){console.log(error);}
        else{//console.log('valores

a contestar');

        //console.log(dates.actuador);

        accion2=dates.actuador

        });
    res.send( accion2 );
    } );

    app.get( 'resptres', function ( req, res ) {
    prog.findOne({'serial':'s3'},function(error,dates){
        if(error){console.log(error);}
        else{//console.log('valores

a contestar');

        //console.log(dates.actuador);

        accion3=dates.actuador

        });
    res.send( accion3 );
    } );

    app.get( 'respcuatro', function ( req, res ) {
    prog.findOne({'serial':'s4'},function(error,dates){
        if(error){console.log(error);}

```



```

        else{//console.log('valores
a contestar');

        //console.log(dates.actuador);

        accion4=dates.actuador

        });
        res.send( accion4 );
    } );
}

```

*Aquí son los render's de las visualizaciones de los html que se diseñaron para ello solo se dejan como muestra pero necesitan sustentar*

```

///Renders
{
    app.get("/sensado/general",function(sol,resp)
    {resp.render('sensado/general',{fail:""});});

    app.get("/sensado/uno",function(sol,resp)
    {resp.render('sensado/uno',{fail:""});});

    app.get("/sensado/dos",function(sol,resp)
    {resp.render('sensado/dos',{fail:""});});

    app.get("/sensado/tres",function(sol,resp)
    {resp.render('sensado/tres',{fail:""});});
}

```

```

app.get("/sensado/cuatro",function(sol,resp)
{resp.render('sensado/cuatro',{fail:""});});

```

```

app.get("/cuenta/index",function(sol,resp)
{resp.render('cuenta/index');});

```

```

app.get("/cuenta/signin",function(sol,resp)
{resp.render('cuenta/signin',{fail:""});});

```

```

app.get("/admin/index",function(sol,resp)
{resp.render('admin/index');});
app.get("/admin/form",function(sol,resp)
{resp.render('admin/form');});
app.get('/cuenta/ingresar',function(sol,resp){
resp.render('cuenta/ingresar')});

```

```

app.get('/user/index',function(sol,resp)
{
resp.render('user/index',{user:loginuser});});

```

}  
*Aquí es donde ejecuta el app*

```

app.listen(process.env.PORT || 8080,() =>
console.log('fine'))

```

Para descargar el programa como prueba se dejara algún ejemplo, pero se dejara la página de prueba subida a servidores Heroku.

<https://sensoresonlinesandro.herokuapp.com/>

Para poder subir se necesite de un video tutorial, a pesar de que se programó en otro programa (VS), se pudo tomar como referencia el método. Además dejare el package.json que es después de mucho tiempo se pudo alzar la pagina web.

<https://www.youtube.com/watch?v=nm7gWyZvdVU&t=861s>

*Nota: el gitignore, no se inserto.*

*2.- Las versiones de todos los motores probablemente se actualicen a la realización de este doc.*

Package.json

```

{
  "name": "paginaweb",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node app.js"
  },
  "keywords": [],

```

```

  "author": "",
  "license": "ISC",
  "dependencies": {
    "express": "^4.17.1",
    "method-override": "^3.0.0",
    "mongoose": "^5.8.1",
    "pug": "^2.0.4"
  },
  "devDependencies": {},
  "description": ""
}

```

```
|
```

```
}
```

## Python

Como ya se describió python es una programa de compilación que se un entorno muy agradable, para los siguientes programas solo se necesita tener conexión a Internet y tener conectado al puerto Serial el dispositivo, en este caso nuestra MSP430

### Transmision datos a internet. py

```
import os
import glob
import time
from urllib.parse import urlencode
from urllib.request import Request, urlopen
import requests
import serial
import sys
rx=0
act='T'

url = 'http://sensoresonlinesandro.herokuapp.com/catch' #
Set destination URL here
respuno=
'http://sensoresonlinesandro.herokuapp.com/respdos'
data = {'serial': 's2', 'name': 'E.Hall', 'valor': rx, 'actuador': act} #
Set POST fields here
request = Request(url, urlencode(data).encode())

practis = Request(respuno)

ser= serial.Serial(port='/dev/ttyUSB0',
#ser= serial.Serial(port='/dev/ttyACM0',
    baudrate=9600,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_ONE,
    bytesize=serial.EIGHTBITS,
    timeout=1
)

def send():
```

```
    data = {'serial': 's2', 'name': 'E.Hall', 'valor': rx, 'actuador': act}
# Set POST fields here
    request = Request(url, urlencode(data).encode())
    json = urlopen(request).read().decode()
def rea():
    lectura = urlopen(practis).read().decode()
    print(lectura)

lectura='1'
while True:
    rea()
    time.sleep(2)
    ser.write(lectura.encode('ascii'))
    try:
        rx=ser.readline()
        rx=rx.decode("utf-8", "strict")
        rx=int(rx)
        rx=(rx*100/1023)
        rx=int(rx)
        print('El valor leído del sensor es')
        print(rx)
        if(rx>60):
            act='t';
        else:
            act='f'
        send()
    except KeyboardInterrupt: # Keyboard CTRL c, Shut down
serial and close
        ser.close()
        sys.exit(1)

except:
    print('sin datos')
```

### Recepción de acción online .py

```
from urllib.parse import urlencode
from urllib.request import Request, urlopen
import requests
import time

respuno= 'http://sensoresonlinesandro.herokuapp.com/respdos'

practis = Request(respuno)

while True:
    lectura = urlopen(practis).read().decode()
    print(lectura)
    time.sleep(2)
```

## Trabajos citados

Foundation, R. P. (20 de 01 de 2020). *Raspberrypi*. Obtenido de <https://www.raspberrypi.org/>

MOCQ, F. (2016, Enero). *Raspberry Pi 2 utilizetodo el potencial de su nano-ordenador*. Barcelona: ENI.

Rossum, G. V. (Septiembre de 2009). *docs.python.org.ar*. Obtenido de <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>