

---

# Diario de Desarrollo

**Práctica Profesional Supervisada**

---

ALONSO, Alvaro — BADALONI, Maximiliano

112358.fn@gmail.com — maxi.badaloni@gmail.com

Comienzo 1 Septiembre, 2014



# Contents

<b>Lunes, 1 Septiembre 2014</b>	<b>1</b>
1 Sistema de gestión de versiones: Git . . . . .	1
2 Selección de sensor de temperatura . . . . .	1
3 Pre selección de PIC 18Fxx . . . . .	2
<b>Martes, 2 Septiembre 2014</b>	<b>3</b>
1 Topología de un nodo esclavo . . . . .	3
2 Compra de componentes . . . . .	3
<b>Miércoles, 3 Septiembre 2014</b>	<b>5</b>
1 Esquema de conexión del LM35 (Sensor de temperatura) . . . . .	5
2 Prueba del sensor LM35 . . . . .	6
3 Prueba del regulador LM317 . . . . .	6
4 Prueba del Diagrama de la Fig.2 . . . . .	7
5 Realización del esquemático en EagleCAD . . . . .	7
<b>Jueves 4, Septiembre</b>	<b>8</b>
1 Compra de los elementos faltantes . . . . .	8
2 Prueba del Diagrama de la Fig.2 . . . . .	8
3 Prueba del Diagrama nuevo . . . . .	9
4 Nueva versión del esquemático en EagleCAD . . . . .	9
<b>Viernes 5, Septiembre 2014</b>	<b>10</b>
1 Diseño del PCB para el sensor LM35 . . . . .	10
<b>Lunes, 8 Septiembre 2014</b>	<b>11</b>
1 Selección de un microcontrolador . . . . .	11
2 Selecccion de un entorno de desarrollo . . . . .	11
<b>Martes, 9 Septiembre 2014</b>	<b>13</b>
1 Fuente de 12 y 5 V . . . . .	13
2 Entorno de desarrollo . . . . .	13
3 Programacion del PIC16F84A . . . . .	13
<b>Miercoles, 10 Septiembre 2014</b>	<b>14</b>
1 Comunicacion UART . . . . .	14
0.0.1 SMS . . . . .	15

## *Contents*

0.0.2	Dudas . . . . .	15
0.0.3	Corriente . . . . .	15

# Lunes, 1 Septiembre 2014

## 1 Sistema de gestión de versiones: Git

Puesta en marcha del sistema de gestión de versiones Git: <http://git-scm.com/>. Ver también: [http://en.wikipedia.org/wiki/Git\\_%28software%29](http://en.wikipedia.org/wiki/Git_%28software%29)

Para poder trabajar de manera colaborativa se utiliza el servidor remoto de github <https://github.com/>. Este servicio es gratis para repositorios que permanezcan abiertos a todo el publico. También, presenta el beneficio de plantear un marco para poder compartir desarrollos y códigos de manera libre. En particular en este desarrollo se opto por una licencia GPL v2.

El esquema del repositorio es el siguiente:

```
Sensor-Temperatura/ ..... Directorio Raiz
├── DailyLog/ ... Este directorio contiene
│   │           el MLTeX del diario de
│   │           desarrollo(*.tex ).
│   ├── dailyLog.tex ..... Archivo principal
│   ├── Sep
│   │   ├── 1-4.tex ..... Archivo Semanal
│   │   └── Oct
│   │       ├── 1-4.tex
│   │       └── Nov
│   │           └── 1-4.tex
│   └── Doc ... En este directorio
│       │       se encuentra toda la
│       │       documentación utilizada
│       │       durante el desarrollo.
│       ├── Datasheets
│       └── Lista de Componentes
└── Design ... Directorio destinado
    │         a los diseños de
    │         esquemáticos y PCB.
    ├── LM35
    └── LM35-Remote.sch ..... Esquemático
```

## 2 Selección de sensor de temperatura

Para la selección del sensor se tuvo en cuenta principalmente el material disponible. De esta manera las posibilidades de un sensor de temperatura que se pueden encontrar en

el mercado local se reducen a:

- LM335
- LM35
- Termistores comunes (1 10K)

Las hojas de datos(Datasheets) de los componentes LM335 y LM35 se encuentran en el directorio *Doc/*. Por otro lado los termistores son dejados como segundas opciones debido a que no son lineales como los dos anteriores, lo que representa una mayor complejidad en el ajuste de la curva y una posible perdida de precisión. Ver:

<http://es.wikipedia.org/wiki/Termistor#Introducci.C3.B3n>

Ambos sensores son analógicos con salidas en voltaje.

Sensor	Factor de escala	Rango de operación
LM335	$10mV/^{\circ}K$	$-40$ a $100^{\circ}C$
LM35	$10mV/^{\circ}C$	$-55$ a $150^{\circ}C$

Table 1: Comparación de los sensores LM335 y LM35

Como podemos observar de la Tabla 1 el sensor *LM35* presenta beneficios en cuanto al rango y no es necesaria la conversión de  $^{\circ}K$  a  $^{\circ}C$  no requiriendo la sustracción de una gran constante de voltaje. Por tales motivos se opta por utilizar el *LM35* como sensor para el proyecto.

### 3 Pre selección de PIC 18Fxx

Se desea tener un sistema en el que un nodo central(beagleBone) se comunique con nodos esclavos los cuales deben ser capaces de medir la temperatura de 3 a 6 puntos diferentes. La comunicación entre nodos es digital y es el nodo esclavo el encargado de la conversión analógica/digital y de la transmisión al nodo central.

Estas razones nos llevan a elegir un micro-controlador con los suficientes canales ADC para poder realizar la conversión y un UART para la transmisión digital de los datos. Los PIC de la familia 18Fxx presentan características que los hacen interesantes para este proyecto.

# Martes, 2 Septiembre 2014

## 1 Topologia de un nodo esclavo

Para cumplir con las siguientes condiciones del proyecto:

- Múltiples puntos de sensado
- Distancias variables entre los puntos y el nodo central.
- Numero variable de puntos.

Se opta por utilizar una topologia como la que se observa en la Fig.1

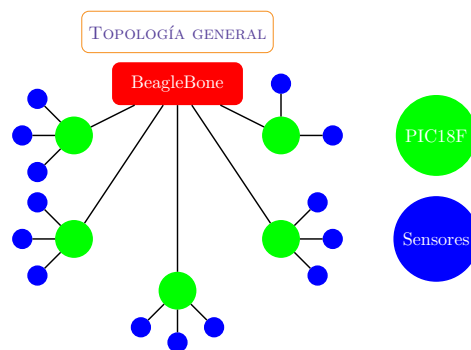


Figure 1: Topologia general del sistema

De esta manera el nodo esclavo es capaz de sensar un numero  $x$  de puntos, cercanos al mismo y transmitirlos de manera digital al nodo central, de manera que la distancia no sea un factor tan influyente. Además, este sistema permite la utilización de un bus de comunicación, haciendo que el mismo pueda ser escalable, adaptándose fácilmente a un mayor numero de puntos de medida.

## 2 Compra de componentes

Se compraron los siguientes elementos en Electrónica Mendoza.

*Martes, 2 Septiembre 2014*

<b>Cantidad</b>	<b>Descripción</b>	<b>Importe</b>
6	Cable UTP	36
10	1n4148	2
2	18F2550	193
15	Resistencia	3.75
50	Resistencia	6
1	18F4550	101.96
5	LM35	120
3	LM317	17.10
9	Potenciometros	42.38
2	LM385	30.60
10	Capacitor	4.10

Table 2: Compras del día 2 de Septiembre

Lo que hace un total de \$557.61



# Miércoles, 3 Septiembre 2014

## 1 Esquema de conexión del LM35 (Sensor de temperatura)

Debido a que los sensores se pueden encontrar a una distancia considerable del nodo esclavo, se debe diseñar una manera de poder llevar la señal analógica hasta el microcontrolador. Para ello se opta por un esquema como el que se observa en la Fig.2

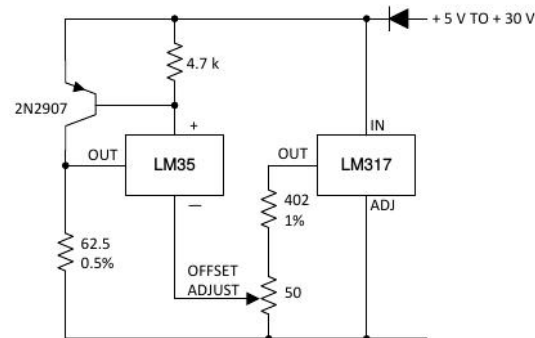


Figure 2: Esquema de conexión del LM35 (4 a 20 mA)

Con este tipo de conexión convertimos nuestra señal de tensión a una de corriente de 4 a 20mA. Lo que nos asegura poder transmitir durante mayores distancias. Detallamos a continuación el funcionamiento de cada una de las partes del circuito:

- **LM35:**  
Tiene a su salida una tensión proporcional a la temperatura. De manera que en *OUT* tendremos  $10mV/^{\circ}C$ . Al pasar por la resistencia se convierte en una corriente proporcional a  $I_{out} = \frac{V_{out}}{R_{62.5}}$ . Como el negativo está conectado con una resistencia a un regulador de tensión nos permite ajustar la escala.
- **LM317:**  
En su salida la tensión regulada que cambia el offset del LM35. Y la corriente se suma a la del LM35.
- **Transistor PNP:**  
(Ver: [http://es.wikipedia.org/wiki/Transistor\\_de\\_uni%C3%B3n\\_bipolar](http://es.wikipedia.org/wiki/Transistor_de_uni%C3%B3n_bipolar)) La resistencia de 4.7K permite que la corriente pase por el transistor en vez de por LM35 de esta manera es el transistor el que la entrega cuando  $I \geq \frac{V_{0.6}}{4.7K}$

Miércoles, 3 Septiembre 2014

Una vez que tenemos nuestra señal de 4 a 20mA para ser leída por el ADC usamos una resistencia de  $50\Omega$  y una  $V_{ref}$  para poder cambiar la escala en el PIC

## 2 Prueba del sensor LM35

Para poder verificar el correcto funcionamiento del integrado realizamos el circuito de la Fig.3

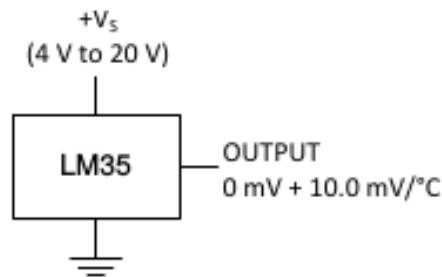


Figure 3: Sensor de temperatura básico

Mediante un tester medimos el valor de tensión, el cual nos entrega como resultado  $230mV$  equivalente a  $23^{\circ}C$ . Esto es coherente y proseguimos con los ensayos sobre este sensor.

## 3 Prueba del regulador LM317

Se realiza el esquema de la Fig.4 para probar si el controlador responde la manera esperada. Mediante el potenciómetro se cambian los valores de tensión de salida del controlador. Alimentado con  $12v$  se consiguen valores desde  $2v$  a  $11.5v$ . De esta manera se ha verificado que el controlador funciona.

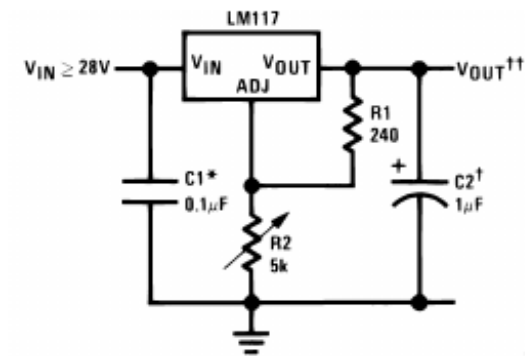


Figure 4: Regulador de tensión típico

## 4 Prueba del Diagrama de la Fig.2

No se pudo realizar por falta del transistor PNP 2N2907

## 5 Realización del esquemático en EagleCAD

Se realizo en el software EagleCAD el diagrama de la Fig.2 junto con la primera versión del PCB.

# Jueves 4, Septiembre

## 1 Compra de los elementos faltantes

Debido a que el día 3 de Septiembre no se pudo realizar los ensayos por falta de material, el jueves a la mañana se realizo la compra de los materiales necesarios. Además, se sumaron los materiales para

- La adaptación de una fuente de computadora como fuente de tensión para los ensayos
- El desarrollo de los circuitos impresos.

Cantidad	Descripción	Importe
4	Banana Hembra	6
4	Banana Macho	12
4	Cocodrilo	10
1	Transferencia	35
1	Placa Fibra	58
20	Resistencia	2.40
2	2N2907	9.08
3	2N3906	3.66
2	LM35	48
4	Unipolar	7
1	Jac	3.86
1	Plug	5

Table 3: Compras del día 4 de Septiembre

Lo que hace un total de \$200

## 2 Prueba del Diagrama de la Fig.2

Tras realizar los ensayos del circuito de la Fig.2 se observa que

- Los valores de las resistencias deben ser muy específicos para que el sistema funcione entre los 4 a 20mA. Sobre todo la de  $62.5\Omega$  que tiene su valor de
$$62.5\Omega = \frac{1V(Rango\ Sensor)}{16mA(Rango\ Senal)}$$

- Al utilizar el regulador *LM317* obtenemos el desfazaje de escala a los  $4mA$  pero agregamos complejidad en el ajuste de las resistencias y calor al sistema de medición.

Para poder minimizar el numero de resitencias a ajustar y la complejidad del sistema se opta por no utilizar el *LM317* y conectar directo el *LM35*. Esto nos da una escala que arranca  $0mA$  para  $0^{\circ}C$ . Luego si utilizamos una resistencia de valor comercial al 0.5% de lamina de metal para que sea mas precisa y menos sensible al calor, tenemos la escala  $20mA(Rango\ Senal) = \frac{1V(Rango\ Sensor)}{(50\Omega)}$ . Obteniendo un diagrama como el que se ve en la Fig.5

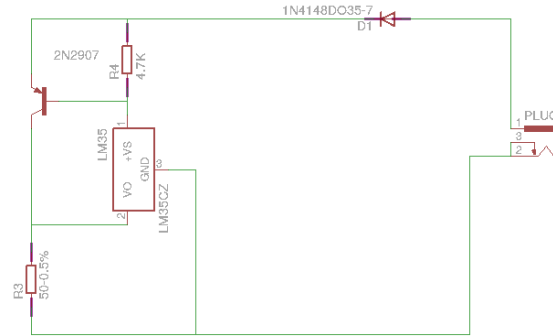


Figure 5: Diagrama de conexión

Finalmente, obtenemos un sistema con menos componentes y que se ajusta por software pero que al mismo tiempo es mas repetible.

### 3 Prueba del Diagrama nuevo

Utilizando el nuevo esquemático y las resistencias con las que contamos son:

- $46.1\Omega$  para la salida del *LM35*
- $217\Omega$  para transformar de corriente a tensión nuevamente.

Los resultados que obtenemos son  $V = 1.19v$  lo cual significa que circula una corriente  $I = \frac{1.19v}{217\Omega} = 5.5mA$  por lo que el *LM35* esta entregando  $V = 5.5mA \cdot 46.1 = 0.25v$  lo cual es coherente con la temperatura del lugar.

### 4 Nueva versión del esquemático en EagleCAD

Se elimina el uso del *LM317* y se conecta por plug en ves de utilizar pines.

# Viernes 5, Septiembre 2014

## 1 Diseño del PCB para el sensor LM35

Tras haber modificado y testeado el circuito para el sensor LM35 se diseña mediante *eagleCAD* el circuito impreso. Se puede ver en la Fig.6

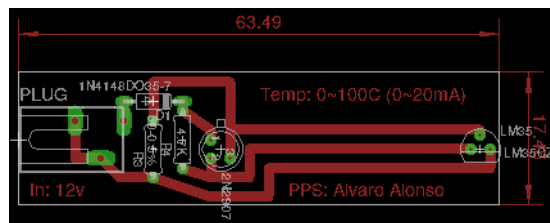


Figure 6: PCB del sensor LM35

# Lunes, 8 Septiembre 2014

## 1 Selección de un microcontrolador

Si bien la semana pasada se había optado por utilizar un microcontrolador de la familia 18Fxx resulta de interés el disminuir los costos del sistema por esta razón se contempla la posibilidad de usar algunos de los siguientes *PICs*

Patas	Antiguos	Nuevos
28	16F873	16F883
28	16F876	16F886
40	16F874	16F884
40	16F877	16F887

Table 4: Opciones de *PIC16F*

Dentro de la tabla<sup>4</sup> el *PIC16F884* se encuentra disponible en el mercado local. Esto lo posiciona por sobre los otros a la hora de la selección. Se comienza el análisis con este último PIC.

La información correspondiente al *PIC16F884* se puede obtener de

<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en026564>

## 2 Seleccion de un entorno de desarrollo

Buscamos encontrar un entorno de desarrollo que sea libre para poder ser usado sin restricciones. Se plantean así las siguientes soluciones

- Netbeans
- Eclipse

Sobre este último se cuenta con experiencia en el desarrollo de aplicaciones embebidas y en C. Por tal razón es que resulta más interesante poder encontrar una solución en este entorno.

El plugin más difundido para poder realizar esta tarea es PIC C Builder:

<http://kr3l.wordpress.com/2008/11/02/using-eclipse-for-pic-development/>

<http://www.chiefdelphi.com/forums/showthread.php?t=35571>

<http://sourceforge.net/projects/piccbuilder/>

<https://github.com/ecdpalma/piccbuilder>

*Lunes, 8 Septiembre 2014*

<http://marketplace.eclipse.org/content/pic-c-builder-eclipse#.VA3KIWD0qis>

Por otro lado, existe el IDE oficial de microchip llamado MPLAB X IDE:

[http://www.microchip.com/pagehandler/en\\_us/family/mplabx/](http://www.microchip.com/pagehandler/en_us/family/mplabx/)

el cual esta basado en NetBeans IDE from Oracle. Por ser ambos un proyecto abierto resulta muy interesante.

<http://microchip.wikidot.com/install:mplabx-lin64>



# Martes, 9 Septiembre 2014

## 1 Fuente de 12 y 5 V

Reutilización de una fuente de Pc como fuente de tensión para los ensayos. Las tensiones que son necesarias para las pruebas son 12v para el sensor de temperatura y 5v para el PIC.

Para trabajar de manera mas cómoda sobre una protoboard se colocan sobre la fuente conectores banana y un interruptor. Mediante cables con conectores banana en un extremo y cocodrilo en el otro podemos utilizar la fuente en los ensayos de manera segura.

## 2 Entorno de desarrollo

Tras haber ensayado las soluciones de entorno de desarrollo nombradas anteriormente y no haber obtenido buenos resultados se opta por seguir otro camino. Mediante el software PIKlab de linux.

<https://www.youtube.com/watch?v=Nj7LzBHzdkk>

[http://fabrice.sincere.pagesperso-orange.fr/cm\\_electronique/projet\\_pic/0clignotant/clignotant.htm](http://fabrice.sincere.pagesperso-orange.fr/cm_electronique/projet_pic/0clignotant/clignotant.htm)

Programas usados:

- Piklab: IDE <http://sourceforge.net/p/piklab/wiki/Home/>
- sdcc: Compilador.
- pk2cmd: Programador por línea de comandos <http://mcuee.blogspot.com.ar/2009/04/pickit-2-under-arch-linux-with-pk2cmd.html>. MUY IMPORTANTE: COPIAR PK2DeviceFile.dat de /usr/share/pk2/PK2DeviceFile.dat a /usr/bin/.

## 3 Programación del PIC16F84A

Para comprobar el funcionamiento de la plataforma realizamos un programa que se encargue de encender/apagar un LED. <http://www.micahcarrick.com/pic-c-programming-linux.html>

# Miercoles, 10 Septiembre 2014

## 1 Comunicacion UART

[http://www.stmental.net/~dfoster/obd/rs232\\_test.html](http://www.stmental.net/~dfoster/obd/rs232_test.html)

<http://www.piclist.com/techref/microchip/16F/628/UARTtest.htm>

Ver tambien: [http://www.voti.nl/blink/index\\_1.html](http://www.voti.nl/blink/index_1.html)

---

# Notas

---

Precios a tener en cuenta:

- Bateria de Gel: Disponibles en Electronica Mendoza  
Precio: \$270  
Se plantea la necesidad de utilizar baterias en caso emergencia?
- Placas de Fibra para PCB:  
Precio:  $20 \times 20 = \$75$  y  $15 \times 15 = \$39$

## 0.0.1 SMS

La empresa CEM(Ver: <http://www.cemsrl.com.ar/>) cuenta con un modulo GSM para alarmas domiciliarias G100: <http://www.cemsrl.com.ar/productos/productos.php?marca=11&producto=223>

Resulta interesante poder adaptar este modulo para ser utilizado mediante la beaglebone.

## 0.0.2 Dudas

- Class-B Safety Software: Resulta de interés implementarlo??: <http://www.microchip.com/pagehandler/en-us/technology/homeAppliance/classbsafetysoftware.html>

## 0.0.3 Corriente

Para poder tener una lectura de la corriente se puede utilizar el sensor *LTS 6 – NP* de LEM. Consultar disponibilidad en el mercado local.loca