****

**TREBALL FINAL DE GRAU**

|  |
| --- |
| **TÍTOL DEL TFG: SISTEMA SENSOR**  **TITULACIÓ: Grau en Enginyeria de Sistemes de Telecomunicació**  **AUTOR: Alejandro García Moreno**  **DIRECTOR: Francesc Josep Sanxis**  **DATA: 23 de setembre del 2016** |

Índice

[1. Objetivos 3](#_Toc454823787)

[2. Plan de diseño del sistema sensor 4](#_Toc454823788)

[2.1 Diagrama de estados (FSM) 4](#_Toc454823789)

[2.2 Registros de configuración del PIC18F 5](#_Toc454823790)

[2.3 Timers y ADC 5](#_Toc454823791)

[2.4 Comunicación con sensores digitales(I2C y otros protocolos) 6](#_Toc454823792)

[2.5 Comunicación serial(UART) con Labview 6](#_Toc454823793)

[2.6 Aplicación Labview 6](#_Toc454823794)

[3. Análisis de las señales a captar con los sensores 7](#_Toc454823795)

[3.1 Temperatura ambiente 7](#_Toc454823796)

[3.2 Humedad relativa 7](#_Toc454823797)

[3.3 Humedad de la tierra 7](#_Toc454823798)

[3.4 Intensidad lumínica 8](#_Toc454823799)

[4. ESTructura del programa principal(ide microchip) 9](#_Toc454823800)

[6. SENSORES: 10](#_Toc454823801)

[6.1 temp 10](#_Toc454823802)

[6.2 lum 10](#_Toc454823803)

[6.3 hum 10](#_Toc454823804)

[6.4 analog temp 10](#_Toc454823805)

[6.5 ter hum. 10](#_Toc454823806)

[6.6 I2c, 1 wire analog read, timers, sleep mode. 10](#_Toc454823807)

[7. Simulacion con proteus. paso a paso 11](#_Toc454823808)

[8. implementación real en una placa de pruebas 12](#_Toc454823809)

[9. Trasmison serial(USB o BLuetooth) 13](#_Toc454823810)

[10. Aplicacion LABview paso a paso 14](#_Toc454823811)

[11. PCB eagle. 15](#_Toc454823812)

[12. mejoras: 16](#_Toc454823813)

[12.1 mejoras en la alimentacion de los sensores: 16](#_Toc454823814)

[12.2 mejoras a la hora de escoger el modo de transmision de los datos wireless 16](#_Toc454823815)

[12.3 mejora a la hora de realizar el energy harvesting. 16](#_Toc454823816)

[13. 18](#_Toc454823817)

# Objetivos

El objetivo del proyecto es realizar un sistema de control de riego y lumínico para un invernadero. Para realizar dicho control, el sistema deberá de medir con sensores diferentes paramentos como la temperatura ambiente, la humedad relativa del aire, la humedad de la tierra y la intensidad lumínica. Una vez medidos estos valores, el sistema sensor deberá enviar, vía Wireless, esos datos a nuestra aplicación de control de Labview para ser mostrados por pantalla y poder realizar un estudio de estos parámetros en base al tiempo.

El sistema sensor se tendrá que realizar con un microcontrolador PIC18F de bajo consumo para que el sistema pueda ser autónomo con poca energía de alimentación. El PIC18F será programado en C usando el estilo de programación FSM. Los primeros programas se simularán en el simulador virtual Proteus para poder detectar bugs de una manera más eficaz. Una vez realizadas las simulaciones, el programa será cargado en nuestro PIC físico en nuestra placa de pruebas donde comprobaremos que todo lo simulado funciona.

En la elección de sensores, se intentará trabajar con sensores analógicos y digitales indistintamente. Respecto los sensores con interface digital, será importante realizar un estudio de sus protocolos de comunicación.

Finalmente, después de realizar el prototipo final en nuestra placa de pruebas, se diseñará con Eagle la correspondiente placa de circuito impreso(PCB) de nuestro sistema sensor y ,a poder ser, también se hará el mecanizado de la misma.

# Plan de diseño del sistema sensor

El sistema sensor se implementará con el microcontrolador PIC18F28K20.



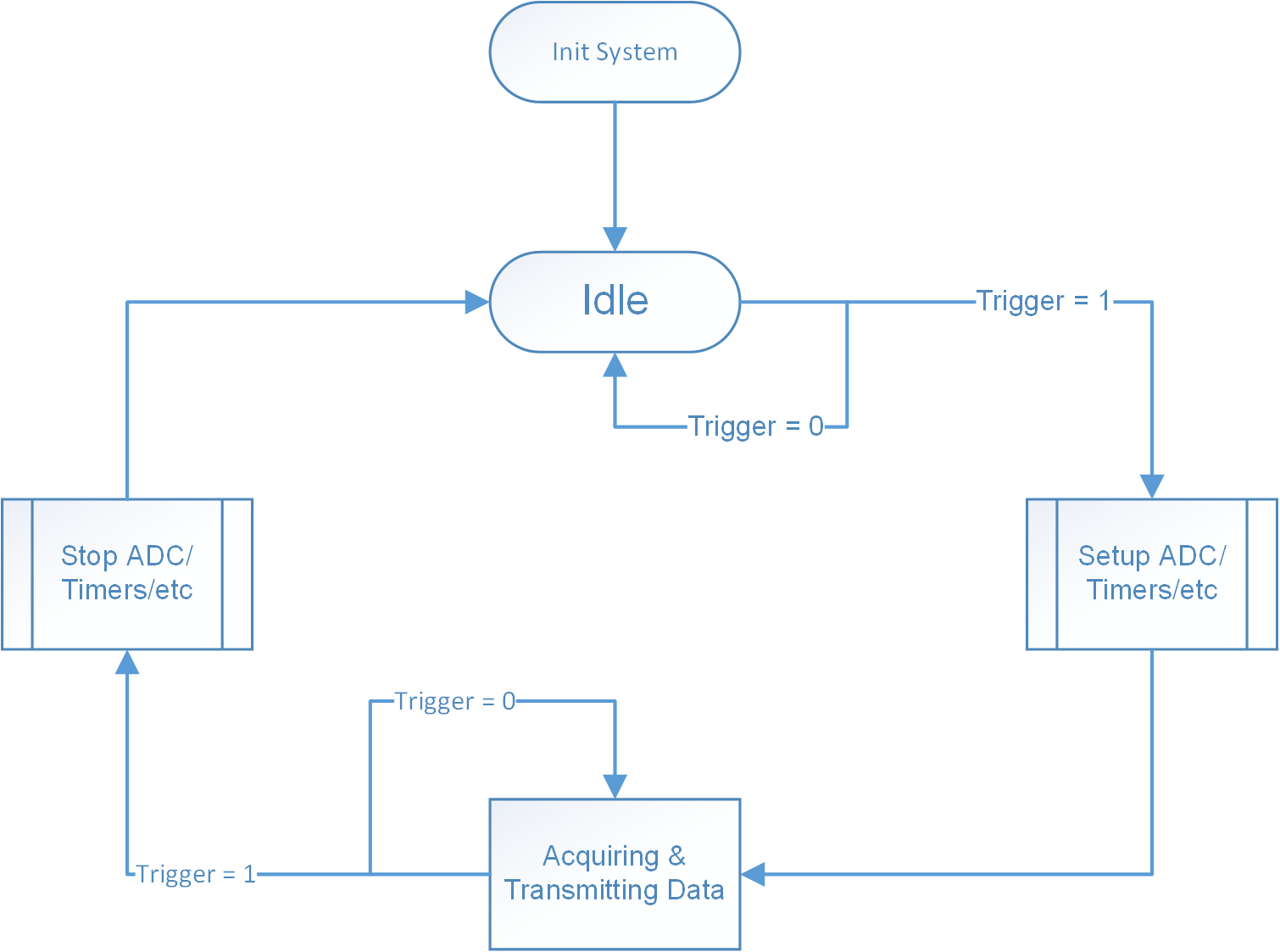
Características principales del microcontrolador son:

* Es un microcontrolador de 28 pines.
* Reloj interno de 16MHz/32.756KHz.
* Comunicación UART, SPI, I2C y PWM.
* 12 entradas al ADC-10bits.
* Alimentación 1.8V-3.6V.
* Sleep mode.

El sistema deberá de ser autónomo y de bajo consumo. De este modo, el sistema tendrá que ponerse a trabajar durante un periodo corto de tiempo en el cual estará tomando medidas y transmitiendo los datos a nuestra aplicación de *Labview.* Una vez se hayan envidado todos los datos, el microcontrolador se pondrá en modo Sleep durante un periodo largo de tiempo.

## Diagrama de estados (FSM)

El programa en C se programará con el estilo FSM(finite state machine) en el cual seguiremos el patrón del siguiente diagrama de estados.



Estados(ejemplo):

1. Idle
2. Setup ADC/Timers/etc
3. Acquiring & Transmiting Data
4. Stop ADC/Timers/etc

En Init System, configuraremos todo lo relacionado con los registros de configuración. Después, empezaremos en modo Idle y cambiaremos de estado cuando la variable trigger cambie a 1. Cuando queramos hacer una tarea en un estado, siempre tendremos el estado de Setup y Stop para hacer setup y reset de los registros necesarios para realizar dicha tarea. La tarea se dará por finalizada cuando la variable trigger sea = 1 y se pasará al siguiente estado tal y cómo mostramos en el diagrama.

## Registros de configuración del PIC18F

Configuración del oscilador interno.

Configuración del baudrate de la comunicación UART

## Timers y ADC

Queremos hacer que el microcontrolador entre en modo Sleep cuando acabe de adquirir y transmitir los datos. Con lo cual, tendremos que configurar un Timer para que cuente el tiempo que pasa mientras estamos en modo Sleep y que despierte al microcontrolador.

Cuando tomemos muestras con el ADC tomaremos 4 muestras en 1 segundo y haremos la media de las 4 muestras para obtener una muestra filtrada. Dicho esto, configuraremos otro Timer para que nos fije la frecuencia de muestreo en 4Hz.

[Cuando se activa el modo Sleep, se activa el *Watchdog Timer* que funciona con el oscilador interno de 32.756KHz y consume menos que el de16MHz.]

## Comunicación con sensores digitales(I2C y otros protocolos)

Realizaremos el estudio del protocolo I2C que nos permite comunicarnos con muchos sensores utilizando solamente 2 pines de PIC. Un sensor con interfaz digital I2C tiene por lo general una dirección de 7bits asignada o con una dirección I2C configurable. Para la adquisición de los datos, el microcontrolador llama al sensor por su dirección y éste contesta. [Anexo]

Trabajaremos con otro tipo de sensores digitales que tienen otros protocolos de comunicación no estandarizados y que el fabricante explica cómo implementarlos en tu microcontrolador. [1 Wire Comunication Anexos]

## Comunicación serial(UART) con Labview

Una vez configurada la comunicación serial del microcontrolador con un baudrate a 9600, conectaremos al microcontrolador un modulo UART-Bluetooth[Anexo] para poder comunicarnos de manera inalámbrica con un PC. Labview recibirá los datos por un puerto COM virtual a través de la conexión bluetooth.



## Aplicación Labview

Una vez recibamos los datos en el aplicación, los iremos mostrando por pantalla en tiempo real. Además podremos mandarle al microcontrolador comandos para que se ponga en modo Sleep o empiece a tomar medidas.

# Análisis de las señales a captar con los sensores

El teorema de Nyquist nos dice que para muestrear una señal analógica, sin perder información de la misma, tenemos que usar una frecuencia de muestreo el doble de su ancho de banda o superior.

Las señales que tenemos que medir tienen poca variancia en el tiempo por lo que podemos escoger frecuencias de muestreo muy bajas.

## Temperatura ambiente

Si nos ponemos en un caso extremo, nos podemos situar en el desierto del Sahara donde se una gran oscilación de temperatura entre el día y la noche. En los días de más oscilación se puede llegar a haber diferencias entre máximas y mínimas de 20-30ºC.[Referencia]

Queremos medir la temperatura con una precisión de 0,1ºC y queremos saber el máximo periodo de muestreo que podemos escoger. Definiremos el ancho de banda de la señal como el diferencial de la temperatura(∆T) entre la presión de lectura(p) y todo entre el diferencial del tiempo(∆t). Y por último, entre que se alcanza la máxima temperatura y la mínima transcurre en media unas 12h.

∆T=30ºC; p=0.1ºC; ∆t=12h;

Por lo tanto, si no queremos perder información de la señal de la temperatura no podemos muestrear con un periodo superior a 72s.

## Humedad relativa

La humedad relativa depende de la temperatura ambiente y la cantidad moléculas de H2O que hay en el aire.

La señal de la humedad relativa se obtendrá con un porcentaje y una precisión de 1%. Para esta señal no hará falta un estudio para obtener la frecuencia de muestro mínima porque tiene mucha menos variancia y se optará por una frecuencia de muestro única.

## Humedad de la tierra

La humedad de la tierra es la cantidad de moléculas de agua que hay por m2 de tierra pero con un sensor eso no lo podemos medir. Lo que sí que podemos hacer es medir la conductividad de la tierra. Por lo tanto, mediremos la conductividad de la tierra con un sensor que nos dé una salida en forma de voltaje para que nuestro ADC capte la señal.

Para esta señal no hará falta un estudio para obtener la frecuencia de muestro mínima porque tiene mucha menos variancia que la temperatura ambiente y se optará por una frecuencia de muestro única.

## Intensidad lumínica

La unidad del sistema internacional de la intensidad lumínica es la candela(Iv, símbolo cd). Esta unidad de medida es proporcional a lumens por estereorradianes(Iv=lm/sr). Pero la unidad que nos será más cómoda para trabajar será la iluminancia(Ev, símbolo lux), en la cual Ev es proporcional a lumens por metro cuadrado(Ev =lm/m2).

Un LDR es una resistencia que varia su resistividad según la luz que capte. Un fabricante de LDRs nos suele proporcionar la relación resistividad-iluminancia(Ω/lux), con lo cual obtener la iluminancia es relativamente sencillo midiendo la resistividad.

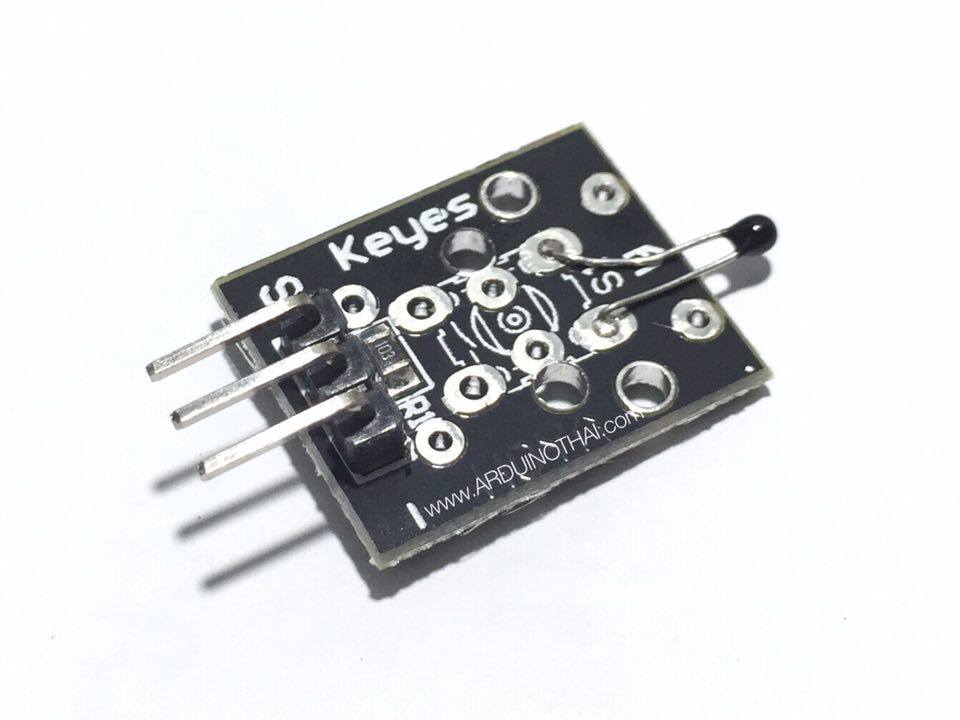
Más adelante explicaremos cómo detectaremos esta señal y qué estrategia seguiremos para su muestreo.

# SENSORES

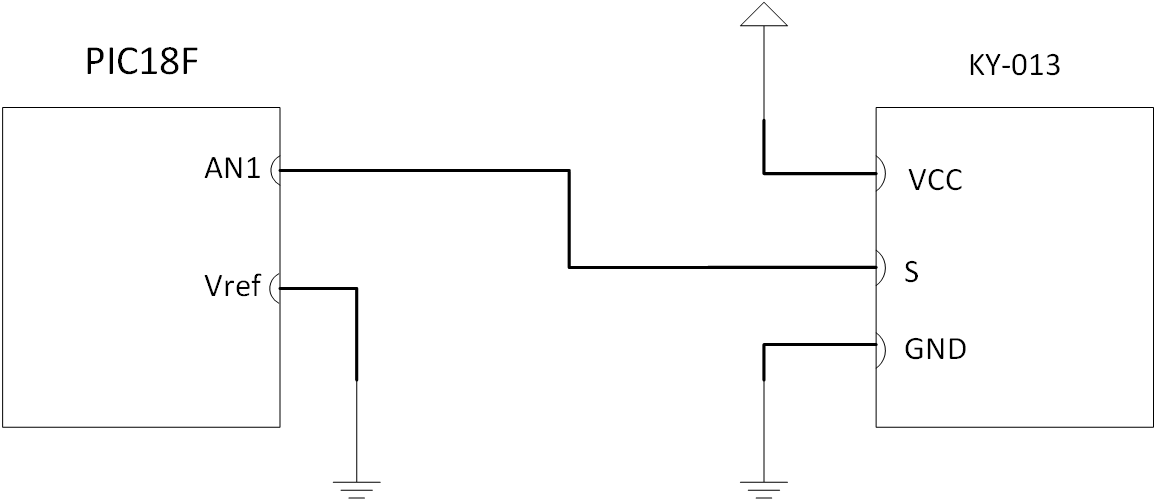
Para tomar muestras de las diferentes señales que tenemos que medir utilizaremos tanto sensores con interface de salida analógica como digital, con el fin de estudiar diferentes tipos de sensores y protocolos de comunicación.

## Temperatura Analog KY-013

El KY-013 es un termistor NTC, con lo cual su resistencia varía respecto la temperatura siguiendo un comportamiento exponencial. El fabricante nos proporciona la fórmula que nos convierte el código leído por el ADC y Temperatura.[Anexo][foto sensor]



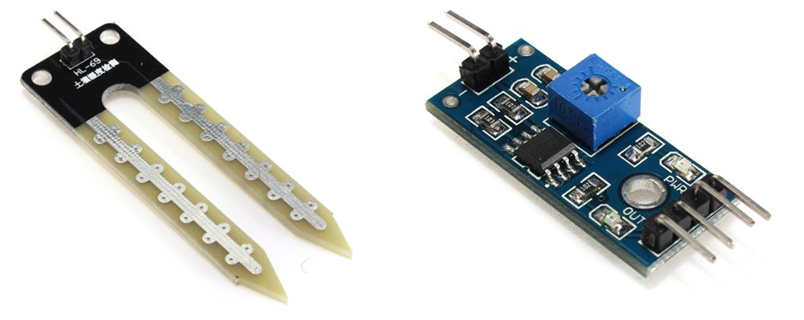
Además el fabricante nos explica cómo conectar el sensor a nuestro MCU. El KY-013 tiene 3 pines: 1 para ser conectado a VCC, 1 para conectar a GND y 1 para conectar la salida analógica(S) a una entrada del ADC(AN1).[figura]



(figura)

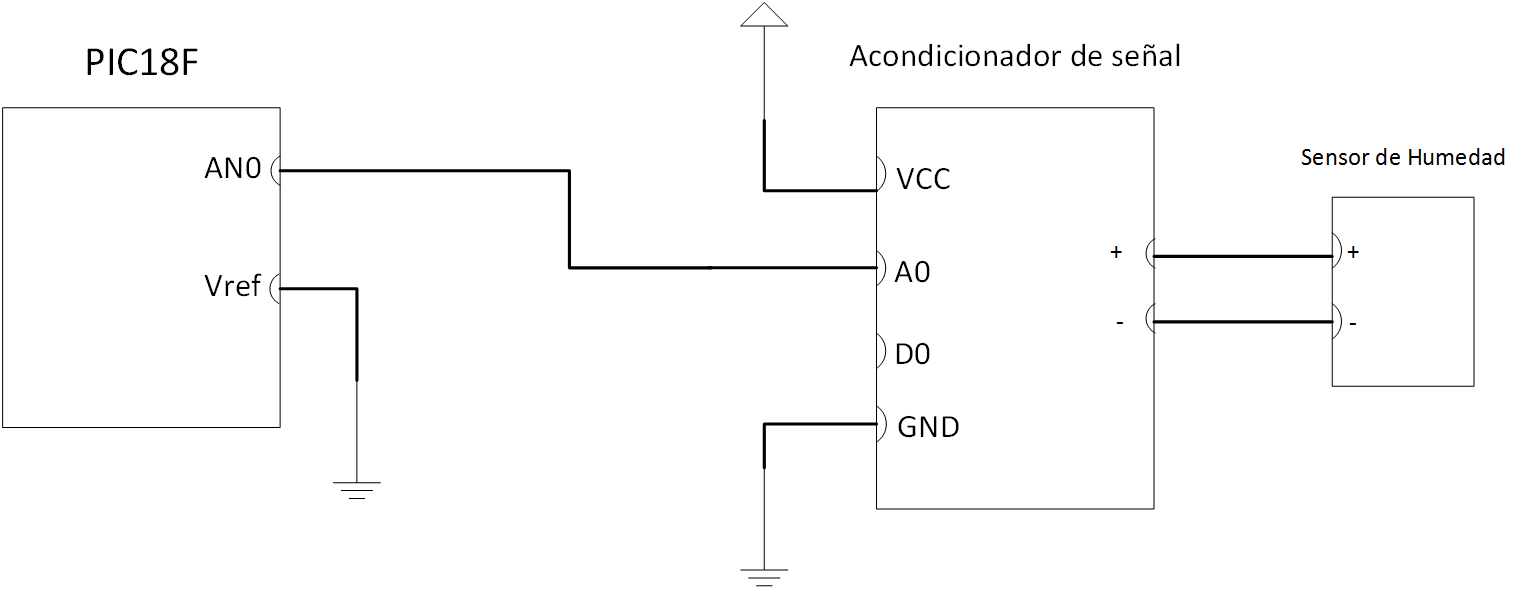
## Humedad Tierra Analog

El HTA es un sensor que detecta la conductancia de la tierra. El HTA tiene integrado un acondicionador de señal para que podamos medir la humedad de la tierra desde Vref hasta VCC con el ADC. Además el sensor tiene una salida digital, que no utilizaremos, que se pone en on/off cuando la tensión de salida del A0 es VCC/2.[Anexo][foto sensor]



(izquierda sensor, derecha acondicionador)

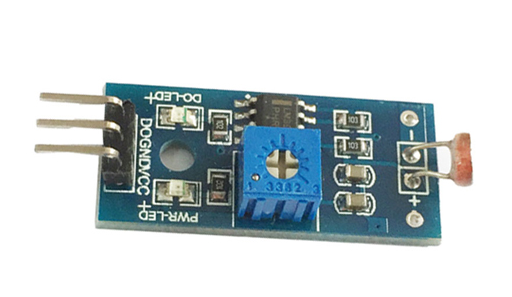
En este caso, conectaremos la salida A0 a la entrada del ADC AN0 y el resto de conexiones tal y cómo se muestra en la figura 1[lea Anexo].



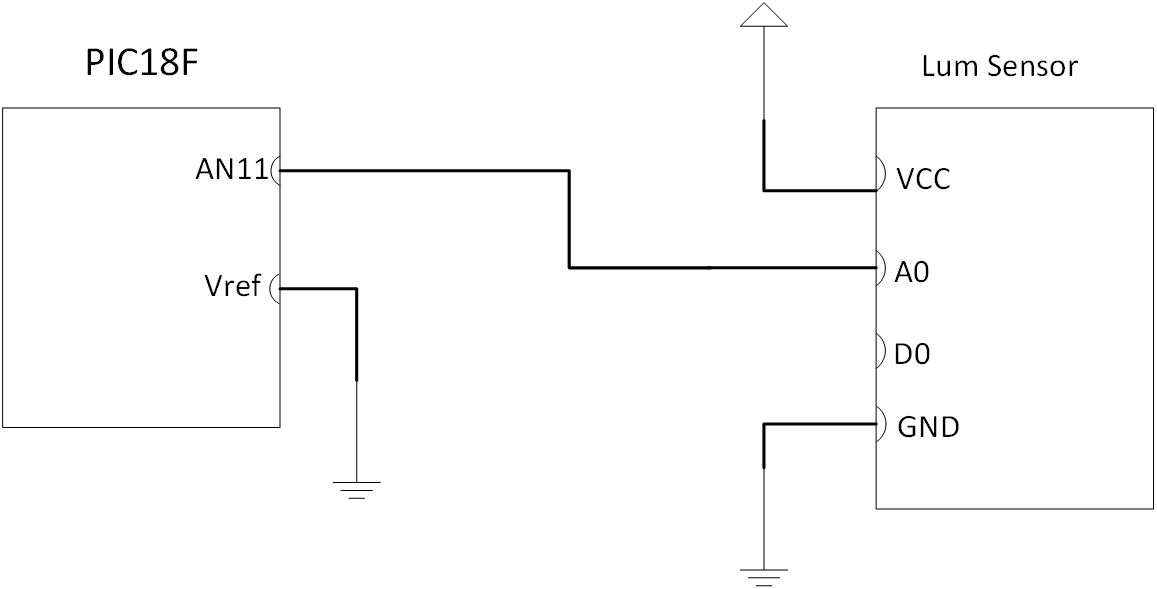
(figura1)

## Iluminancia Analog

El LDR es un sensor que detecta la Iluminancia que capta en su superficie fotosensible y hace variar su resistencia. El sistema sensor del LDR tiene integrado, igual que el HTA, un acondicionador de señal para que podamos medir la humedad de la tierra desde Vref hasta VCC con el ADC. Además el sensor tiene una salida digital, que no utilizaremos, que se pone en on/off cuando la tensión de salida del A0 es VCC/2.[Anexo][foto sensor]



En este caso, conectaremos la salida A0 a la entrada del ADC AN11 y el resto de conexiones tal y cómo se muestra en la figura 1[lea Anexo].

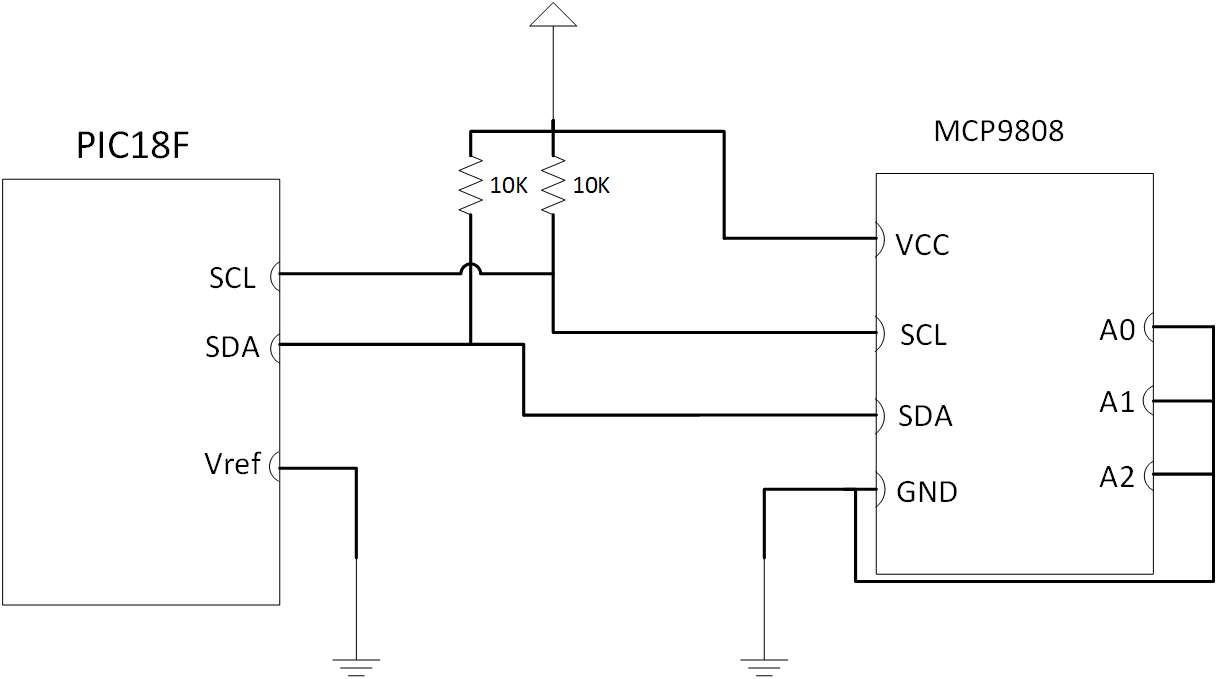


## Temperatura Digital I2C MCP9808

El MCP9808 es un sensor de temperatura con interface digital que utiliza el protocolo de comunicaciones I2C por lo tanto tendremos que configurar la MCU como Master porque el sensor será Slave. [Anexo I2C].

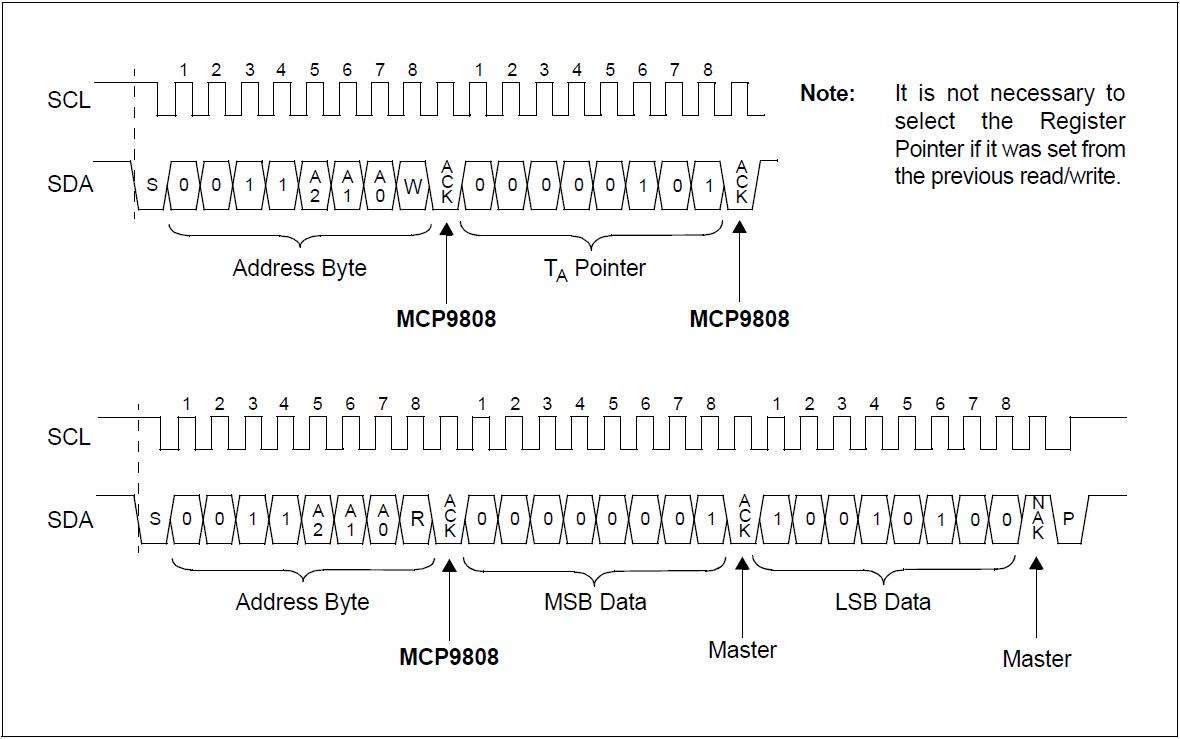


El MCP9808 tiene esta dirección I2C 0011XXX donde los últimos 3 bits de la dirección son configurables por vía externa. En el esquema de conexión, los puertos de A0, A1 y A2 están conectados a masa para que sean '0' y por lo tanto la dirección I2C sea 0011000.

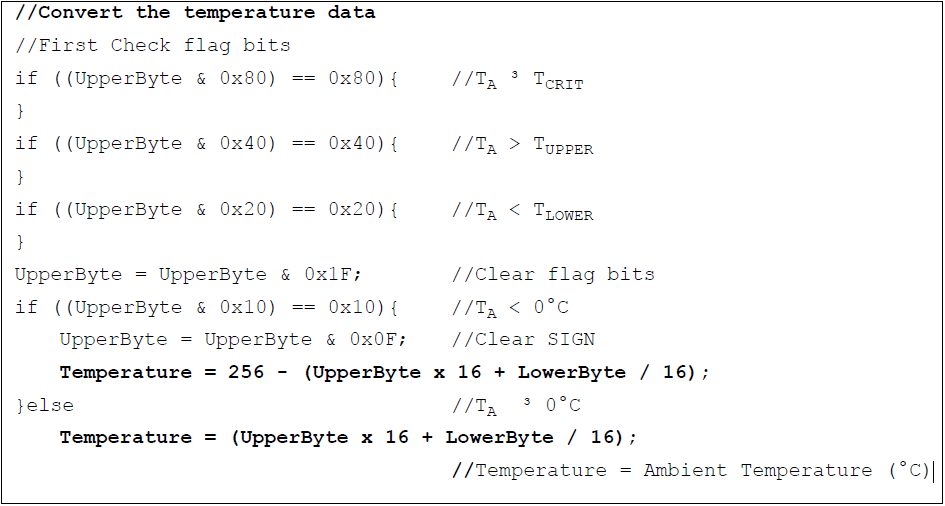


(Esquema de conexión con la MCU)

Durante la comunicación I2C[Anexo], deberemos de enviar el comando de 8bits '00000101' al sensor para que nos responda proporcionándonos la temperatura.



Finalmente el MCP9808 nos responderá con 2 bytes: un MSB y LSB. Entonces, gracias a la fórmula que nos proporciona el fabricante podremos traducir el código a 'ºC'. Sin embargo en el datasheet del fabricante nos proporciona el código C que tenemos que utilizar para hacer la conversión.

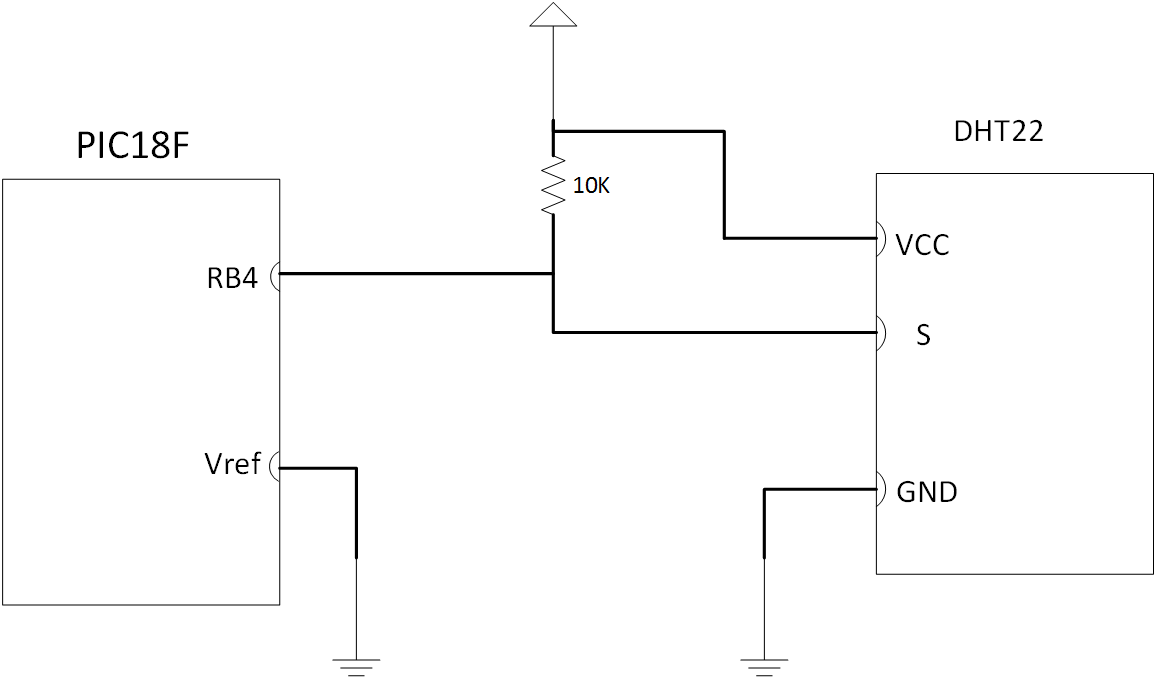


## Temperatura y Humedad Relativa Digital OneWire Protocol DHT22

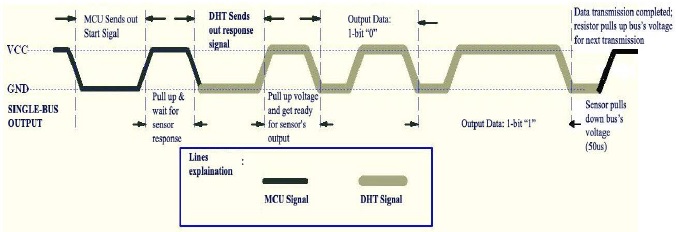
El DHT22 es un sensor de temperatura y humedad relativa con interface digital. La comunicación de realizará utilizando sólo un pin del MCU y se realizará de forma asíncrona porque no tenemos un clock como en la comunicación I2C.



El sensor utiliza un protocolo propio en el cual se nos explica en el datasheet[Anexo]. Además nos muestra cómo debemos conectar el DHT22 con el MCU.



Es un protocolo sencillo en el cual tendremos que utilizar el Timer2 para contar la duración de los pulsos porque según la duración del ellos, un pulso significará que se ha emitido un '0' o '1'[Anexo].



En el transcurso de la transmisión, el sensor enviará un total de 5 Bytes: 2Bytes de Temperatura, 2Bytes de Humedad Relativa y 1Byte para el Checksum. Una vez recibamos los datos, no tendremos que utilizar una fórmula para descifrar que valor hemos recibido porque en el MSB tenemos su valores enteros y en su LSB sus decimales.

# Estructura del programa principal(Ide microchip)

## Estados

1. Idle
2. Setup ADC (Ground Humidity)
3. Acquiring & Transmitting ADC (Ground Humidity)
4. Stop ADC (Ground Humidity)
5. Setup ADC (Analog Temp)
6. Acquiring & Transmitting ADC (Analog Temp)
7. Stop ADC (Analog Temp)
8. Setup OneWire (DHT22)
9. Acquiring & Transmitting OneWire (DHT22)
10. Stop OneWire (DHT22)
11. Setup I2C (Digital Sensor)
12. Acquiring & Transmitting I2C (Digital Sensor)
13. Stop I2C (Digital Sensor)
14. Setup ADC (Analog LDR)
15. Acquiring & Transmitting ADC (Analog LDR)
16. Stop ADC (Analog LDR)

## Diagrama de estados.png

# Simulación con Proteus. paso a paso

Primera simulación con un Timer y realizando una lectura analógica

# implementación real en una placa de pruebas

# Trasmison serial(USB o BLuetooth)

# Aplicacion LABview paso a paso

# PCB eagle.

# mejoras:

## mejoras en la alimentacion de los sensores:

## mejoras a la hora de escoger el modo de transmision de los datos wireless

## mejora a la hora de realizar el energy harvesting.

# 

Objetivo:

descripcion

TEMAS

ESTructura del programa principal(ide microchip)

SENSORES:

temp

lum

hum

analog temp

ter hum.

I2c, 1 wire analog read, timers, sleep mode.

Simulacion con proteus. paso a paso

implementacion real en una placa de pruebas

Trasmison serial(USB o BLuetooth)

Aplicacion LABview paso a paso

PCB eagle.

mejoras:

mejoras en la alimentacion de los sensores:

mejoras a la hora de escoger el modo de transmision de los datos wireless

mejora a la hora de realizar el energy harvesting.