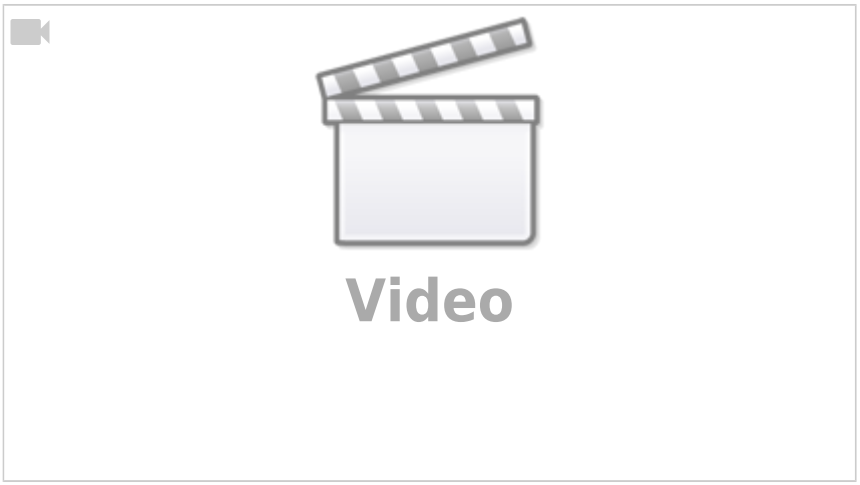
Sensor de humedad en la tierra

Conceptos básicos de la humedad del suelo



Sensor de humedad resistivo

- Basados en este sitio http://www.gardenbot.org/howTo/soilMoisture/
- ¿Tres pines o un pin del MCU? Con dos pines digitales que actuan de inversores del sentido de la corriente más el pin análogo se evita la rápida galvanización de las puntas de prueba en la tierra. Para que la galvanización no altera las lecturas y tenemos cantidad de pines limitados en el ESP-12

Sensor de humedad capacitivo

Basado en los diseños encontrados en el siguiente enlace http://zerocharactersleft.blogspot.com.co/2011/11/pcb-as-capacitive-soil-moisture-sensor.html y adaptándolo a materiales locales se ha construído un sensor para el jardín de las delicias versión unloquer. A continuación se muestra el proceso:

Fix Me! Describir los pasos

Fix Me! Agregar los videos de demostración

20160127_175959.ogv 20160130_225152.ogv 20160130_230025.ogv

opciones comerciales - http://vegetronix.com/

Software

- No funciona en arduino para el ESP8266 la librería FreqCounter por como son manejadas las interrupciones :S
- Ya se por lo menos qué es lo que hay que hacer para implementar una alternativa a la librería FreqMeasure que no funciona para el ESP.
    1. Necesito tomar una muestra de la señal por un determinado periodo de tiempo.
    2. Contar las interrupciones en el flanco de caída de la señal cuadrada que arroja el sensor
    3. Medir el tiempo entre cada interrupción (para detectar el período)
    4. Sacar el promedio de los periodos e invertirlo para obtener la frecuencia de la señal

Código que no funciona. La implementación del manejo de las interrupciones esta mal. Adaptado de acá http://openenergymonitor.org/emon/node/123

```
//Number of pulses, used to measure energy.
long pulseCount = 0;

//Used to measure power.
unsigned long pulseTime,lastTime;

//power and energy
double freq,muestraPromedio,muestraTemporal;

void setup()
{
Serial.begin(115200);

// interrupt attached to = pin5
pinMode(5, INPUT_PULLUP);
attachInterrupt(5, onPulse, FALLING);
}

void loop()
```

```
{

}

// The interrupt routine
void onPulse()
{


//disable_interrupt()
//pulseCounter
if(pulseCount++ < 4) {// Número de pulso de la muestra
  //Calculate power
  //used to measure time between pulses.
  lastTime = pulseTime;
  pulseTime = micros();
  muestraTemporal = freq;
  freq = pulseTime - lastTime;

  muestraPromedio = (muestraTemporal + freq)/2;
} else {
  pulseCount = 0;
  //Print the values.
  Serial.println(muestraPromedio);

}
//enableInterrup()

}
```

Manejo de Interrupciones

https://en.wikipedia.org/wiki/Interrupt_handler
https://www.arduino.cc/en/Reference/AttachInterrupt
https://www.sparkfun.com/tutorials/326

https://www.youtube.com/watch?v=DGPEB2Q83E8&nohtml5=False
https://www.youtube.com/watch?v=D_7ciW_TCac https://www.youtube.com/watch?v=RiYmLy8yUAY

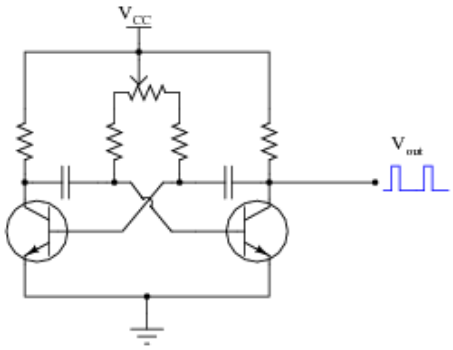
Portando FreqMeasure para el ESP8266

- La librería verifica unas definiciones para en cada plataforma definir adecuadamente el tipo de timer y flags que va a usar en https://github.com/unloquer/FreqMeasure/blob/master/util/FreqMeasureCapture.h#L69, se agrega una para que pase la compilación. El compilador usa el macro -DESP8266 que proviene del archivo https://github.com/esp8266/Arduino/blob/master/platform.txt en la variable build.extra_flags

Investigación pwm basado en variación de capacitancia

https://www.allaboutcircuits.com/worksheets/signal-modulation/

The *oscillator* circuit in this diagram generates a square wave with an adjustable duty cycle:



A student desires to use this circuit as the basis for a *pulse-width modulation* (PWM) power controller, to vary the amount of power delivered to a DC load. Since the oscillator circuit is built to produce weak signals and not deliver power directly to a load, the student adds a power MOSFET to switch heavy load currents:

TODO

- Probar sming https://github.com/SmingHub/Sming
- Hay varios ejemplos que están en c++ plano. Sming es para poderlos probar
- Probar platforma.io http://platformio.org/#!/

Referentes

Uso de los timers de ESP → https://github.com/DINGGLABS/ESP8266/blob/master/WiFi-Button/WiFi-Button.ino
arduino-frequency-counterduty-cycle-meter → https://www.electronicsblog.net/arduino-frequency-counterduty-cycle-meter/
FreqCounter → http://www.pjrc.com/teensy/td_libs_FreqMeasure.html
Timer en el ESP8266 → http://www.switchdoc.com/2015/10/iot-esp8266-timer-tutorial-arduino-ide/
Contador de frecuencia con arduino (usa interrupciones - ISR) → http://www.chemcool.co.za/p/603366/arduino-frequency-counter-6-mhz
Ver como implementan los timers → https://github.com/NEBULAIR/esponics/blob/master/esponics-01/esponics-01.ino
Tachometer - ESP8266 implementation → https://github.com/eadf/esp8266_tachometer Contador de pulsos - http://openenergymonitor.org/emon/buildingblocks/introduction-to-pulse-counting → código: http://openenergymonitor.org/emon/node/123 (Por este lado va, ¿cómo se deshabilitan las interrupciones?)
https://github.com/esp8266/source-code-examples/blob/master/interrupt_example/Makefile

https://github.com/esp8266/Arduino/blob/ea302aab05480ad36c076b618abb642b1eb1893b/hardware/esp8266com/esp8266/libraries/Servo/src/esp8266/ServoTimers.h Sensor comercial de DFrobot:

- http://www.didacticaselectronicas.com/index.php/sensores/humedad/sensor-de-humedad-de-suelo-capacitivo-detail
- https://www.dfrobot.com/wiki/index.php?title=Capacitive_Soil_Moisture_Sensor_SKU:SEN0193

From:
https://wiki.unloquer.org/ -

Permanent link:
https://wiki.unloquer.org/proyectos/jardin_delicias/tecnologicos/sensores/humedad_tierra?rev=1489275981

Last update: **2017/03/11 23:46**