Universidad del País Vasco Euskal Herriko Unibertsitatea

GIPUZKOAKO INGENIARITZA ESKOLA
ESCUELA DE INGENIERÍA DE GIPUZKOA

GIPUZKOAKO INGENIARITZA ESKOLA

ESCUELA DE INGENIERÍA DE GIPUZKOA

GRADU AMAIERAKO LANA/ TRABAJO FIN DE GRADO

Industria Elektronikaren eta Automatikaren Ingeniaritzako Gradua

Data: 2018/07/02

[WoodSmart: Egurraren hezetasuna neurtzen duen haririk gabeko sentsore erresistiboa]

[Eranskinak]

Ikaslearen izen eta abizenak: Elene Beitia Loinaz

Zuzendariaren izen eta abizenak: Nora Barroso Moreno

Zuzendarikidearen izen eta abizenak: Tknika

Laburpena

Atal hau memoriako edo proiektuko beste dokumentuak garatu, azaldu edo justifikatzen dituzten dokumentuek osatzen dute.

Aurkibidea

1. Hezetasuna kalkulatzeko programa

```
//DEFINIZIOAK//

#include <Wire.h>
#include <SparkFunBME280.h>
#include <Adafruit_ADS1015.h>
#include "QuickStats.h"

#define MY_RFM69_NEW_DRIVER
#define MY_RADIO_RFM69

#include <MySensors.h>

#define MY_DEBUG

#define A0 3
#define A1 4
#define A2 5
#define EN 6
#define ANALOG_POWER 7
#define I2C_POWER 8

Adafruit_ADS1115 ads;
QuickStats stats; //Mediak kalkulatzeko liburutegia
BME280 bme280;

MyMessage msg(0, V_HUM);

// Erresistentzien informazioa //
const int n_resistances = 5;
const int range_outputs[n_resistances] = {5, 6, 7, 3, 4}; //Arduinoko pinen zenbakia
const float range_resistances[n_resistances] = {0.0995, 1.498, 10, 100, 1000}; //
Erresistentzien balioa Mohm-etan
const float resistances_max_deviation[n_resistances] = {0.5, 0.5, 0.5, 0.5, 0.5}; //
Erresistetzia bakoitzean onartzen den desbideratze maximoa(%)
const long resistances_delay[n_resistances] = {0,5,5,15,30}; // Berrelikadurako
erresistentzien arabera kalkuluak egiteko itxaro behar den denbora
int16_t range_index = 0;//Momentu bakoitzean ze erresistentzia aukeratuta dagoen
//////////////////////////////////////

// Atzerapenen informazioa //
const int change_range_delay = 100;
const int individual_measurement_delay = 0;
const int cycle_measurement_delay = 0;
unsigned long moisture_measurement_delay = 10000;
//////////////////////////////////
```

```
const float final_resistance_max_deviation = 0.50; //Irteeraren desbideratze
maximoa(%)
const float max_deviation = 0.1;//Irteeraren desbideratze
const int moisture_n_measurements = 5;
const int power_n_measurements = 10;
const int output_n_measurements = 50;
const float a = -0.046368; // a parametroaren balioa paraleloan neurtzean
const float b = 1.07042; // b parametroaren balioa paraleloan neurtzean
float power_voltage_values[power_n_measurements];
float output_voltage_values[output_n_measurements];
float resistance_values[moisture_n_measurements];
float moisture_values[moisture_n_measurements];
float power_voltage = 0.0; float output_voltage = 0.0;
int16_t power_index = 0; int16_t output_index = 0;
int16_t moisture_index = -1;
unsigned long cycles_start_time = 0;
unsigned long complete_measurement_start_time = 0;
int measurement_finished = 1;
float ambient_temperature;

//FUNTZIOAK//

void set_analog_power(int value) {
 digitalWrite(ANALOG_POWER, value);//Switcha itzali edo pizteko.
 digitalWrite(I2C_POWER, value); // Ilc eta tenperatura sensorearea piztu edo
itzaltzeko.
}

void turn_off_comunication_pins() {

 digitalWrite(A0, LOW);
 digitalWrite(A1, LOW);
 digitalWrite(A2, LOW);
 digitalWrite(EN, LOW);
}

int get_current_range() {

 return range_index; //Bektorearen posizioa.
}

float get_current_range_value() {

 return range_resistances[get_current_range()]; //Berrelikadurako erresistentziaren
balioa.
}

float get_next_range_value() {
```

```
 float coeficient = 1.5;  // Mugatik gertu dagoen balio bat neurtzen ari denean, bi rangoren artean programa konmutatzen egotea ekiditeko erabiltzen den koefizientea.

 if (range_index == (n_resistances-1)) {//Erresistentzia azken balioan dagoen konprobatzen du.
   return range_resistances[range_index]*coeficient;
 }

 return range_resistances[range_index+1]*coeficient;
}


void set_range(int index) {// Indexa eguneratu eta muxaren irteera aldatzen ditu
 if (index < 0 or index > n_resistances) {//Rangotik kanpo ematen duen balio bat pasatzen bazaio zero jartzen zaio.
   set_range(0);
 }

 range_index = index; //Parametro bezela pasatako balioa aldagai globalan gorde.
 set_output(range_outputs[index]);//Muxari pasatzeko bere pin berria zein den.
}


int set_next_range() {//Zuzenean ze rangoan zauden jakin gabe hurrengo rangoan jarri.

 if (range_index == (n_resistances-1)) {
   return 0;
 } else {
   range_index++;
 }

 set_output(range_outputs[range_index]);//Muxaren balioa aldatzeko
 return 1;
}


int set_previous_range() {{//Zuzenean ze rangoan zauden jakin gabe rangoan jaitsi.

 if (range_index == 0) {
   return 0;
 } else {
   range_index--;//rangoz jaisteko.
 }

 set_output(range_outputs[range_index]);
 return 1;
}
```

```
void set_output(int number) {//Muxaren irteera aldatu.

//Arazoa: Arduinoko programan eta datasheet-ean pinak 1-etik 8-ra definituta daude baina muxaren pinen izenak S1etik S8ra doazen harren hauek hautatzeko 0-tik 7-ra dijoa, beheko taulan azaltzen den bezala 0=S1 izanik e.a. Ondorioz dagoen zenbakiari -1 egin behar zaio.

 Number // A2 A1 A0  => bitarrean ->Muxaren Pina
 1          //  0   0   0 =>      0        -> S1
 2          //  0   0   1 =>      1        -> S2
 ...        //      ...
 8          //  1   1   1 =>      7        -> S8

 number--;

 if (number > 7 or number < 0) {
  Serial.println(F("We just admit numbers until 7"));
  return;
 }

 int A0_state, A1_state, A2_state;

 A0_state = number % 2;//Zatiketaren ondarra
 number = number / 2;
 A1_state = number % 2;
 number = number / 2;
 A2_state = number % 2;

 digitalWrite(A0, A0_state);//Bakoitzarai bere balioa
 digitalWrite(A1, A1_state);
 digitalWrite(A2, A2_state);
 digitalWrite(EN, HIGH);
}

void print_report() {
 Serial.print(F("[Report] Inverter resistance: ")); Serial.print(get_current_range_value());
Serial.print(F("Mohm"));
 Serial.print(F(", feeding measures: ")); Serial.print(power_n_measurements);
 Serial.print(F(", output measures: ")); Serial.print(output_n_measurements);
 Serial.print(F(", number of cycles: ")); Serial.println(moisture_n_measurements);
}

void sleep_new(unsigned long time) {
 digitalWrite(SDA,0);
 digitalWrite(SCL,0);
 sleep(time);
 digitalWrite(SDA,1);
 digitalWrite(SCL,1);
```

```
}

void before() {
 #ifdef MY_DEBUG //MY_DEBUG =0/1 Izan daiteke. 1 komoentarioak agertzen dira eta 0 dakagunean ez dira komentarioak ikusten.
   Serial.println(F("we started the program, establishing connection with the radio..."));
 #endif
}

void presentation() {
 // Esketxeko informazioa “gateway” eta “Controller”-ari bidaltzen die.
 sendSketchInfo("WoodSmart", "1.0);
 present(0, S_HUM);
}

void setup() {

 #ifdef MY_DEBUG
   Serial.println(F("We start the measurements"));
 #endif
 pinMode(A0, OUTPUT);
 pinMode(A1, OUTPUT);
 pinMode(A2, OUTPUT);
 pinMode(EN, OUTPUT);
 pinMode(ANALOG_POWER, OUTPUT);
 pinMode(I2C_POWER, OUTPUT);

 set_analog_power(1);//Arduino piztean beste guztia pizteko.
 set_range(3);//Berrelikaduran zein erresistetziatik hasi jakiteko. 3.posizioan dagoena

 Wire.begin();//bme280 liburutegiak eskatzen digu.
 bme280.setI2CAddress(0x76);//i2c helbidea paraleloan jarri daitezkenez zeini bidali jakin behar duelako.
 ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V  0.1875mV (adc1115-aren kasuan)
 if(bme280.beginI2C() == false) Serial.println(F("Temperature sensor no connected"));//Liburutegia hasieratu.
 ads.begin();

}

//DEFINIZIOAK//

void loop() {

 //Honera prozesu guztia bukatzean sartzen da.
 if (moisture_index == moisture_n_measurements) {//Moisture_index neurketetan non gauden, moisture_n_measurements zenbat neurketa egin behar ditun
```

```
float final_moisture_value
=stats.average(moisture_values,moisture_n_measurements); //Prozesuko neurketa
guztiak gordetzen dituen betorea.
float final_moisture_stdev = stats.stdev(moisture_values,moisture_n_measurements);
  float final_resistance_value =
stats.average(resistance_values,moisture_n_measurements);
 float final_resistance_stdev =
stats.stdev(resistance_values,moisture_n_measurements);
   float final_resistance_stdev_percentage =
(100*final_resistance_stdev)/final_resistance_value;  //Resistantziaren desbiazioaren
portzentaia

  if (final_resistance_value > get_next_range_value()) {//Neurtu den balioa hurrengo
ragoan neurtu daitekeen konprobatu. Honela bada hurrego rangora pasa.
    if (set_next_range()) {
     Serial.println(F("We rise in rank"));
     power_index = 0;
     output_index = 0;
     moisture_index = -1;
     return;
    }
  }

  if ((0.0 <= final_resistance_stdev_percentage) && (final_resistance_stdev_percentage
<= resistances_max_deviation[get_current_range()]) && (0.0 <= final_moisture_value)
&& (final_moisture_value <= 100.00)) {
    Serial.println(F("-------------------------------------"));
    Serial.print(F("----- The average of humidity is: ")); Serial.print(final_moisture_value,
1); Serial.print(F("%"));
    Serial.print(F(" (Deviation: ")); Serial.print(final_moisture_stdev, 5); Serial.print(F(")"));
    Serial.print(F(" (Resistance: ")); Serial.print(final_resistance_value, 4); Serial.print(F("
Mohm"));
    Serial.print(F(" -> Deviation: ")); Serial.print(final_resistance_stdev_percentage, 2);
Serial.print(F(")"));
    Serial.print(F(" (Cycle time: ")); Serial.print((millis() - cycles_start_time)/1000.0, 2);
Serial.println(F(" seg) -----"));
    float complete_measurement_time = ((millis() -
complete_measurement_start_time)/1000.0) + resistances_delay[range_index];
    Serial.print(F("----- Total measurement time: "));
Serial.print(complete_measurement_time, 2); Serial.print(F(" seg."));
    Serial.print(F(" (Subtracting stabilization time: "));
Serial.print(complete_measurement_time - resistances_delay[range_index]);
Serial.println(" seg) -----");
    Serial.println(F("-------------------------------------"));
    unsigned long ahora = millis();
    send(msg.set(final_moisture_value, 1));
    moisture_index = -1;
    measurement_finished = 1;
```

```
    int r = get_current_range();
    turn_off_comunication_pins();
    set_analog_power(0);
    Serial.print(F("Time to send the data by radio and turn off everything: "));
Serial.println(millis()-ahora);
    Serial.print(F("We have turned off the analog circuit electronics and go to sleep ("));
Serial.print(moisture_measurement_delay); Serial.println(F(" milliseconds)"));
    sleep_new(moisture_measurement_delay);
    set_range(r);

   } else {
    Serial.print(F("Invalid measure (")); Serial.print(final_resistance_value);
Serial.print(F(" Mohm)"));
    Serial.print(", we repeat the process(Deviation: ");
Serial.print(final_resistance_stdev_percentage); Serial.println(F("%)"));
    moisture_index = 0;
   }
  }

  if (moisture_index == 0) {
   power_index = 0;
   output_index = 0;
   cycles_start_time = millis();//Ordua apuntatzen du. Ondoren prozesua bukatzean
zenbat tardatu duen jakiteko.
   print_report();
  }

  if (moisture_index == -1) {//-1 itxaron zirkuitua estabilizatu arte,sleep bitartez.
   set_analog_power(1);
   if (measurement_finished == 1)
    complete_measurement_start_time = millis();
    measurement_finished = 0;
   Serial.print(F("We are waiting for the signal to stabilize ("));
Serial.print(resistances_delay[range_index]); Serial.println(F(" seconds)"));
   sleep_new(resistances_delay[range_index]*1000.0);
   moisture_index++;
   return;
  }

  //5.neurketan dena ondo dagoenean dena itzaltzen da. Horregatik sentsorea berriro
piztean hasieraketa prozesua egin behar da.

  bme280.setI2CAddress(0x76);
  if(bme280.beginI2C() == false) Serial.println(F("Temperature sensor not connected"));
  ambient_temperature = bme280.readTempC();
  Serial.print(" Temperature: ");
  Serial.println(ambient_temperature, 2);
```

```
 //Elikaduraren neurketak //
  float power_bits, power_avg, power_stdev = 0.0;

  while (power_index < power_n_measurements) { //Ziklo bakoitzean 10 aldiz neurtzen
da zirkuituaren elikadura.
    power_bits = ads.readADC_Differential_2_3();//adc-aren bit kopurua ematen du.
    power_voltage = (power_bits * 0.1875) / 1000.0;//bit horiek tentsioan bihurtzeko
TWOTRHIDS erabiltzen denez * 0,1875.
    power_voltage_values[power_index] = power_voltage;//Bektorean sartzeko
    power_index++;//Bektorearen posizioa aldatu
    sleep(individual_measurement_delay);//Programa honen kasuan 0 beraz ez da sleep
egiten.
  }

  power_avg = stats.average(power_voltage_values,power_n_measurements);
  power_stdev = stats.stdev(power_voltage_values,power_n_measurements);


  // Operazionalaren irteerako neurketak //
  float output_bits, output_avg, output_stdev = 0.0;

  while (output_index < output_n_measurements) {//Ziklo bakoitzean 50 aldiz neurtzen
da operazionalaren irteerako balioa.
    output_bits = ads.readADC_Differential_0_1();
    output_voltage = (output_bits * 0.1875) / 1000.0;
    output_voltage_values[output_index] = output_voltage;
    output_index++;
    sleep(individual_measurement_delay);
  }

  output_avg = stats.average(output_voltage_values,output_n_measurements);
  output_stdev = stats.stdev(output_voltage_values,output_n_measurements);

  #ifdef MY_DEBUG
    Serial.print(F("Input voltage in the ADC: ")); Serial.print(output_avg, 5);
Serial.print(F("V"));
    Serial.print(F(" (Deviation: ")); Serial.print(output_stdev, 5); Serial.println(F(")"));
  #endif


  output_avg=-output_avg;//adc-an sartzen dena operazionalaren irteerakoaren
aurkakoa da, horregatik *-1.

  if (output_stdev / abs(output_avg) > max_deviation) {//Desbideratzea>desbideratze
maximoa den kasuetan agertuko da bakarrik.
    Serial.println(F("¡The output voltage of the operational oscillates a lot... ¿Do we do
something?"));
  }
```

```
 #ifdef MY_DEBUG
  Serial.print(F("Output voltage in the operational: ")); Serial.print(output_avg, 5);
Serial.print(F("V"));
  Serial.print(F(" (Deviation: ")); Serial.print(output_stdev, 5); Serial.println(F(")"));
 #endif

 if (abs(output_avg) < 0.020) {//Berrelikadurako erresistentzia oso txikia den kasuan.
  Serial.print(F("-- Very low output voltage (")); Serial.print(abs(output_avg), 4);
Serial.println(F("V), we try to rise in rank"));
  if (set_next_range()) {
   power_index = 0;
   output_index = 0;
   moisture_index = -1;
   return;//Looparen gora joaten da.
  }
 }

 if (abs(output_avg) > 4.0) {//Berrelikadurako erresistentzia oso altua den kasuan.
  Serial.print(F("-- High output voltage, we try get a lower rank; "));
Serial.println(output_avg, 4);
  if (set_previous_range()) {
   power_index = 0;
   output_index = 0;
   moisture_index = -1;
   return;
  }
 }


 #ifdef MY_DEBUG
  Serial.print(F(" (Deviation: ")); Serial.print(power_stdev, 5); Serial.println(F(")"));
 #endif

//Programa honeraino iritsi bada, dena ondo neurtua dago eta desbideratze
txikiarekin. //Orduan egurraren hezetasuna kalkulatzen da.
//Kodigo zati hau 5 aldiz errepikatuko da.
 float wood_resistance = 0.0;
 wood_resistance = -
(power_avg/output_avg)*get_current_range_value();//Erresistentzia kalkulatzeko
 resistance_values[moisture_index] = wood_resistance;//5 neurketa horietatik zein
posiziotan gauden jakiteko.

 #ifdef MY_DEBUG
  Serial.print(F("Wood resistance: ")); Serial.print(wood_resistance, 5); Serial.println(F("
Mohm"));
 #endif
```

```
 //Egurraren hezetasuna kalkulatu eta gorde//
 float wood_moisture;
 float wood_moisture_correction;
 wood_moisture = ((log10(log10(wood_resistance) + 1) - b) / a);
 wood_moisture_correction =-
(0.00147*ambient_temperature*log(10)+log(exp(a*wood_moisture*log(10)+b*log(10))+1)
-1.075*log(10))/(log(10)*(0.000158*ambient_temperature+0.0262));//log=ln
 moisture_values[moisture_index] = wood_moisture_correction;//Hezetasunaren
balioa bektorean gorde.

 #ifdef MY_DEBUG
   Serial.print(F("Wood humidity (%): ")); Serial.print(wood_moisture, 5);
Serial.print(F("% - ")); Serial.print(wood_moisture_correction, 5); Serial.println(F("%"));
 #endif


 //Indexa reseteatu eta eguneratu//
 power_index = 0;
 output_index = 0;
 moisture_index++;

 sleep(cycle_measurement_delay);

}
```