CI&CEQ
Chemical Industry & Chemical Engineering Quarterly

Available online at
Association of the Chemical Engineers of Serbia *AChE*
www.ache.org.rs/CICEQ

Chem. Ind. Chem. Eng. Q. 30 (2) S3–S6 (2024) CI&CEQ

Supplementary material to

SYSTEM DEVELOPMENT FOR MONITORING THE PRODUCTION PROCESS OF FREEZE-DRYING SAMPLES: A SIMPLE AND LOW-COST APPROACH

Vanessa Dal-Bó[1], Heitor Otacílio Nogueira Altino[1,2*], José Teixeira Freire[1]

[1]Federal University of São Carlos, Department of Chemical Engineering, Drying Center of Pastes, Suspensions and Seeds, Rod. Washington Luiz, km 235, P.O. Box 676, 13565-905, São Carlos, São Paulo, Brazil
[2]Federal University of Uberlandia, Faculty of Chemical Engineering, Av. João Naves de Ávila, 2121, Block 1K, 38408-100 Uberlândia, MG, Brazil

Chem. Ind. Chem. Eng. Q. 30 (2) 111–122 (2024)

S.1. Main code for data acquisition

```
// Main code for data acquisition
// SetPoint Temperature
  float Tset = 40.0;

// Libraries
 //DS18b20
   #include <OneWire.h>
   #include <DallasTemperature.h>
 //BME280
   #include <Wire.h>
   #include "cactus_io_BME280_I2C.h"
 //Hx711
   #include "HX711.h"

// Constants and Pins
 //DS18b20
   #define ONE_WIRE_BUS_PIN 10
 //MPX5500DP
   int sensorValue = 5;
   int Feed = 33;
 //SSR
   int In_Rele = 53;
 //Hx711
   HX711 cell(3, 2);
   double A, B, val1;
 //PushBotton
   const int switchPin = 4;
   const int ledPin = 8;
   int switchState = 0;
```

* Email: heitor.engquimica@gmail.com

```
  //Variáveis de tempo
    long sampleint =3000;
    long previousMillis =0;
// Objects
  //DS18b20
    OneWire oneWire(ONE_WIRE_BUS_PIN);
    DallasTemperature sensors(&oneWire);
    DeviceAddress Probe01 = { 0x28, 0x61, 0x64, 0x12, 0x3F, 0xF7, 0x1D, 0xCD };
    DeviceAddress Probe02 = { 0x28, 0x61, 0x64, 0x12, 0x3C, 0x74, 0x87, 0x3C };
    DeviceAddress Probe03 = { 0x28, 0x61, 0x64, 0x12, 0x3C, 0x75, 0xB6, 0x18 };
    DeviceAddress Probe04 = { 0x28, 0xFF, 0xC2, 0x8E, 0xA6, 0x16, 0x03, 0xA6 };
  //BME280
    BME280_I2C bme2(0x76);  // I2C using address 0x76

//Setup
void setup()
{
  Serial.begin(9600);

  //BME280
  if (!bme2.begin()) {
    Serial.println("Could not find a Second BME280 sensor, check wiring!");
    while (1);}

  //DS18b20
    sensors.begin();
    sensors.setResolution(Probe01, 10);
    sensors.setResolution(Probe02, 10);
    sensors.setResolution(Probe03, 10);
    sensors.setResolution(Probe04, 10);

  //MPX5500DP
    pinMode(Feed, OUTPUT);
    digitalWrite(Feed, HIGH);

  //SSR
    pinMode(In_Rele, OUTPUT);

  //Hx711
    A = -0.0001649626;
    B = 1359.5214135927044+0.2;
    val1 = 0.0;

  //PushBotton
    pinMode(ledPin, OUTPUT);
    pinMode(switchPin, INPUT);

  // Data required for Excel
    Serial.println("CLEARDATA");
    Serial.println("LABEL, Hora, P Amb. (kPa), HR (%), T Amb. (*C), P Vacuo (kPa), P cuba (kPa), Massa (g), T1 (*C),
T2 (*C), T3 (*C), T4 (*C)");
 }



//Main loop
void loop(){
  // Global variables
    double P0, HR, Ta, P1, P2, sensorValueAux, val1, T1;
          P0=HR=Ta=P1=P2=sensorValueAux=val1=T1=0;

  // Time count
    unsigned long currentMillis = millis();
      if (currentMillis - previousMillis > sampleint){
```

```
        previousMillis = currentMillis;

   for(int index =0; index < 3; index++){
     //BME280:
       bme2.readSensor();
       P0 = bme2.getPressure_MB()+P0;
       HR = bme2.getHumidity()+HR;
       Ta = bme2.getTemperature_C()+Ta;
     //MPX5500DP:
       sensorValueAux = analogRead(sensorValue);
       P1 = 0.54722*(sensorValueAux)-19.80068+P1;
       P2 = bme2.getPressure_MB()/10-(0.54722*(sensorValueAux) - 19.80068)+P2;
     //Hx711:
       val1 = A*(cell.read())+B+val1;// most recent reading
}

  //BME280:
    Serial.print("DATA, TIME,");
    Serial.print(P0/30, 3);
    Serial.print(",");
    Serial.print(HR/3, 3);
    Serial.print(",");
    Serial.print(Ta/3, 3);
    Serial.print(",");

  //MPX5500DP:
    Serial.print(P1/3, 3);
    Serial.print(",");
    Serial.print(P2/3, 3);
    Serial.print(",");

  //Hx711:
    Serial.print(val1/3, 2);
    Serial.print(",");

  //DS18b20:
    sensors.requestTemperatures();
    T1=sensors.getTempCByIndex(2);

  //SSR:
    if (T1 <= Tset){
    digitalWrite(In_Rele, HIGH);
    } else if (T1 > Tset){
    digitalWrite(In_Rele, LOW);}

  //PushBotton:
    switchState = digitalRead(switchPin);
    if (switchState == LOW){
       digitalWrite(ledPin, HIGH);
       B = B - val1/3;
    } else if ((switchState == HIGH)){
       digitalWrite(ledPin, LOW);
    }
}
  //DS18b20:
    sensors.requestTemperatures();
    printTemperature1(Probe01);
    Serial.print(",");
    printTemperature2(Probe02);
    Serial.print(",");
    printTemperature3(Probe03);
    Serial.print(",");
    printTemperature4(Probe04);
```

```
}//End of main loop

//DS18b20:
  void printTemperature1(DeviceAddress deviceAddress1){
  float tempC1 = sensors.getTempC(deviceAddress1);
  Serial.print(tempC1);}

  void printTemperature2(DeviceAddress deviceAddress2){
  float tempC2 = sensors.getTempC(deviceAddress2);
  Serial.print(tempC2);}

  void printTemperature3(DeviceAddress deviceAddress3){
  float tempC3 = sensors.getTempC(deviceAddress3);
  Serial.print(tempC3);}

  void printTemperature4(DeviceAddress deviceAddress4){
  float tempC4 = sensors.getTempC(deviceAddress4);
    Serial.println(tempC4);}
```

S.2. Supplementary Data

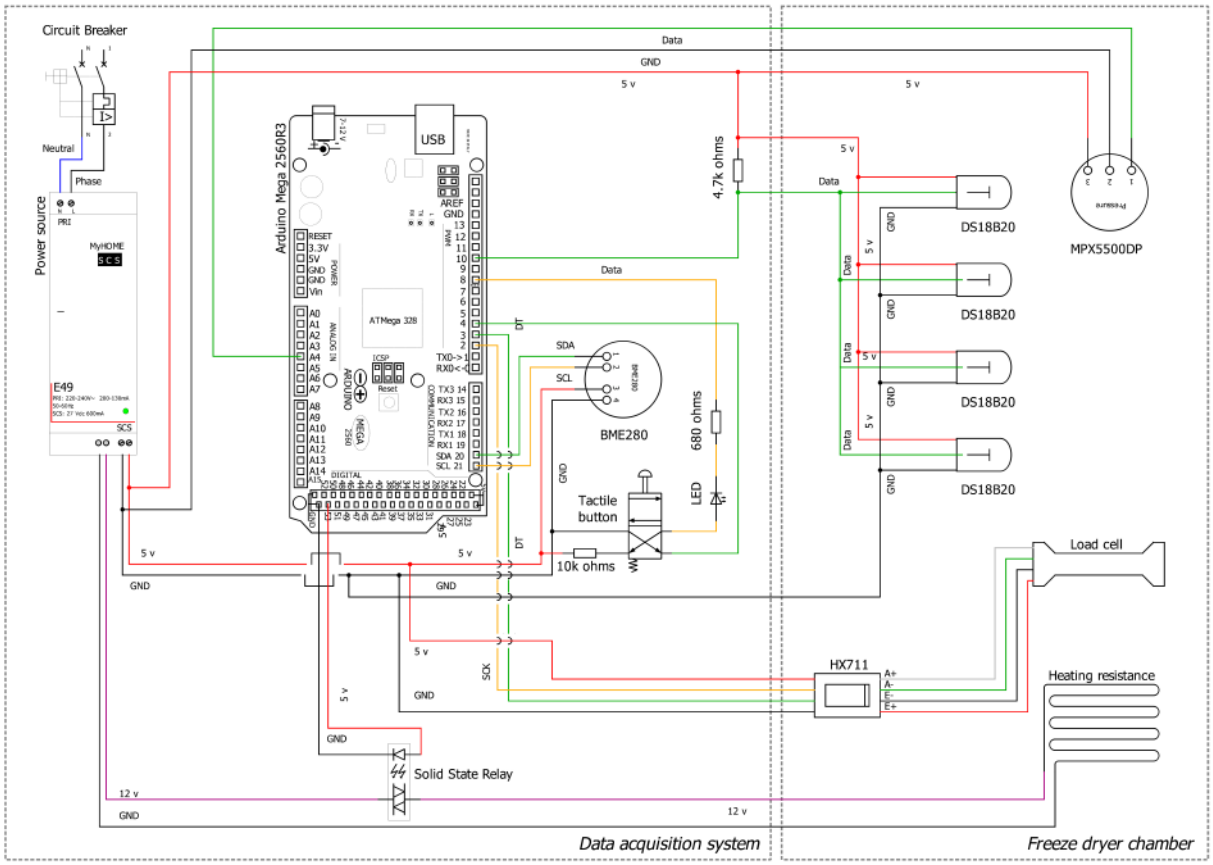

Figure S1. Electrical diagram of the data acquisition system developed.