

Projet-ScIn MOOREV_ Trigger-Caméra Modulation Fréquence

INTRODUCTION

Le but de notre projet ARE est de déclencher une caméra Go Pro à l'aide d'une carte électronique Arduino dans le but de récolter des données. Les données qui seront les causes du déclenchement pourront être le pH de l'eau, la température etc. Cela appelle aussi à une adaptation de la caméra au milieu avec un processus d'éclairage la nuit, un potentiel processus de détection des espèces. Le cœur du sujet résidera donc dans le fait de déclencher la caméra sous l'action de certaines conditions nécessaires. Le dessein final serait de mettre le système dans des conditions réelles en milieu extérieur, ce pendant plusieurs jours, afin d'en observer les résultats.

Objectif : Intégration d'une carte mesures de pH + température et d'une Arduino pour la programmation du déclenchement d'une caméra Go Pro autonome.

Il s'agit d'interfaçage de l'Arduino avec les cartes de mesures de pH SENSE0161 (ou ADCM355), avec enregistrement des mesures sur carte microSD, pour pouvoir réveiller la caméra sur un seuil de mesure de température ou de pH.

Matériel fourni :

- 1 Arduino Nano
- 1 carte support microSD
- 2 Cartes mesure SENSE0161 DFRobot avec électrodes pH
- 2 cartes mesures ADUCM355 WATER QUALITY SENSOR (T°C + pH ou REDOX(ORP))
- 2 sondes T°C long. 30 cm - thermistance (Thermocoax)

Code pour déterminer la température

```
/*  
 * Created by ArduinoGetStarted.com  
 *  
 * This example code is in the public domain  
 *  
 * Tutorial page: https://arduinogetstarted.com/tutorials/arduino-temperature-sensor  
 */
```

```

#include <OneWire.h>
#include <DallasTemperature.h>

const int SENSOR_PIN = 13; // Arduino pin connected to DS18B20 sensor's DQ pin

OneWire oneWire(SENSOR_PIN);    // setup a oneWire instance
DallasTemperature tempSensor(&oneWire); // pass oneWire to DallasTemperature library

float tempCelsius;  // temperature in Celsius
float tempFahrenheit; // temperature in Fahrenheit

void setup()
{
  Serial.begin(9600); // initialize serial
  tempSensor.begin(); // initialize the sensor
}

void loop()
{
  tempSensor.requestTemperatures();    // send the command to get temperatures
  tempCelsius = tempSensor.getTempCByIndex(0); // read temperature in Celsius
  tempFahrenheit = tempCelsius * 9 / 5 + 32; // convert Celsius to Fahrenheit

  Serial.print("Temperature: ");
  Serial.print(tempCelsius);  // print the temperature in Celsius
  Serial.print("°C");
  Serial.print(" ~ ");    // separator between Celsius and Fahrenheit
  Serial.print(tempFahrenheit); // print the temperature in Fahrenheit
  Serial.println("°F");

  delay(500);
}

```

Code de déclenchement de caméra

```

#include <OneWire.h>

// Définir la broche de données pour le capteur DS18B20
#define ONE_WIRE_BUS 2

```

```
// Initialiser l'objet OneWire
OneWire oneWire(ONE_WIRE_BUS);

void setup() {
  // Ouvrir une communication série à 9600 bauds
  Serial.begin(9600);
}

void loop() {
  // Initialiser le capteur
  byte addr[8];
  if (!oneWire.search(addr)) {
    // Si aucun capteur n'est détecté, attendre
    delay(1000);
    return;
  }

  // Vérifier que le capteur est bien un DS18B20
  if (addr[0] != 0x28) {
    // Si ce n'est pas un DS18B20, attendre
    delay(1000);
    return;
  }

  // Initialiser la communication avec le capteur
  oneWire.reset();
  oneWire.select(addr);
  oneWire.write(0x44); // Lancer la conversion de température

  // Attendre la fin de la conversion
  delay(1000);

  // Lire la température
  oneWire.reset();
  oneWire.select(addr);
  oneWire.write(0xBE); // Demander la lecture de la température
  byte data[9];
  for (byte i = 0; i < 9; i++) {
    data[i] = oneWire.read();
  }
}
```

```

}

int temp_raw = (data[1] << 8) | data[0];
float temp_celsius = (float) temp_raw / 16.0;

// Afficher la température
Serial.print("Température: ");
Serial.print(temp_celsius);
Serial.println(" °C");

// Attendre avant de recommencer
delay(1000);
}

```

Code pour avoir le pH

```

/*
# This sample code is used to test the pH meter V1.0.
# Editor : YouYou
# Ver   : 1.0
# Product: analog pH meter
# SKU   : SEN0161
*/

#define SensorPin A0      //pH meter Analog output to Arduino Analog Input 0
#define Offset 0.00       //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40    //times of collection
int pHArray[ArrayLenth]; //Store the average value of the sensor feedback
int pHArrayIndex=0;
void setup(void)
{
    pinMode(LED,OUTPUT);
    Serial.begin(9600);
    Serial.println("pH meter experiment!"); //Test the serial monitor
}
void loop(void)
{
    static unsigned long samplingTime = millis();
    static unsigned long printTime = millis();

```

```

static float pHValue,voltage;
if(millis()-samplingTime > samplingInterval)
{
    pHArray[pHArrayIndex++] = analogRead(SensorPin);
    if(pHArrayIndex == ArrayLenth) pHArrayIndex = 0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime = millis();
}

if(millis() - printTime > printInterval) //Every 800 milliseconds, print a numerical, convert the state of the LED
indicator
{
    Serial.print("Voltage:");
    Serial.print(voltage,2);
    Serial.print("  pH value: ");
    Serial.println(pHValue,2);
    digitalWrite(LED,digitalRead(LED)^1);
    printTime = millis();
}
}

double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
        return 0;
    }
    if(number<5){ //less than 5, calculated directly statistics
        for(i=0;i<number;i++){
            amount+=arr[i];
        }
        avg = amount/number;
        return avg;
    }else{
        if(arr[0]<arr[1]){
            min = arr[0];max=arr[1];
        }
        else{

```

```

    min=arr[1];max=arr[0];
}
for(i=2;i<number;i++){
    if(arr[i]<min){
        amount+=min;    //arr<min
        min=arr[i];
    }else {
        if(arr[i]>max){
            amount+=max;    //arr>max
            max=arr[i];
        }else{
            amount+=arr[i]; //min<=arr<=max
        }
    } //if
} //for
avg = (double)amount/(number-2);
} //if
return avg;
}

```

Code de déclenchement avec pH

```

// Inclure la bibliothèque pour la communication série
#include <SoftwareSerial.h>

// Définir les broches de communication série
SoftwareSerial pHSerial(2, 3); // RX, TX

// Définir la broche de commande de la caméra
const int cameraPin = 4;

// Définir les constantes pour la calibration du capteur de pH
const float pH7Voltage = 2.5;
const float pH4Voltage = 1.98;
const float pH10Voltage = 3.07;

// Fonction pour convertir la tension en pH
float convertToPH(float voltage) {
    float pH = 7 - (voltage - pH7Voltage) / (pH7Voltage - pH4Voltage) * 3;
    return pH;
}

```

```

}

void setup() {
  // Ouvrir la communication série pour le débogage
  Serial.begin(9600);
  // Configurer la communication série pour le capteur de pH
  pHSerial.begin(9600);
  // Définir la broche de commande de la caméra en mode sortie
  pinMode(cameraPin, OUTPUT);
}

void loop() {
  // Lire la tension du capteur de pH
  pHSerial.write('R'); // Envoyer la commande pour lire la tension
  delay(1000); // Attendre la réponse
  while (pHSerial.available() < 4) {} // Attendre la réponse complète
  float voltage = (float)pHSerial.parseFloat() / 1000.0; // Convertir la réponse en voltage
  // Convertir la tension en pH
  float pH = convertToPH(voltage);
  // Afficher les résultats
  Serial.print("Voltage = ");
  Serial.print(voltage, 2);
  Serial.print(" V, pH = ");
  Serial.print(pH, 2);
  Serial.println();
  // Déclencher la caméra si le pH est inférieur à 7
  if (pH < 7) {
    digitalWrite(cameraPin, HIGH);
    delay(500);
    digitalWrite(cameraPin, LOW);
  }
  // Attendre une seconde avant de recommencer
  delay(1000);
}

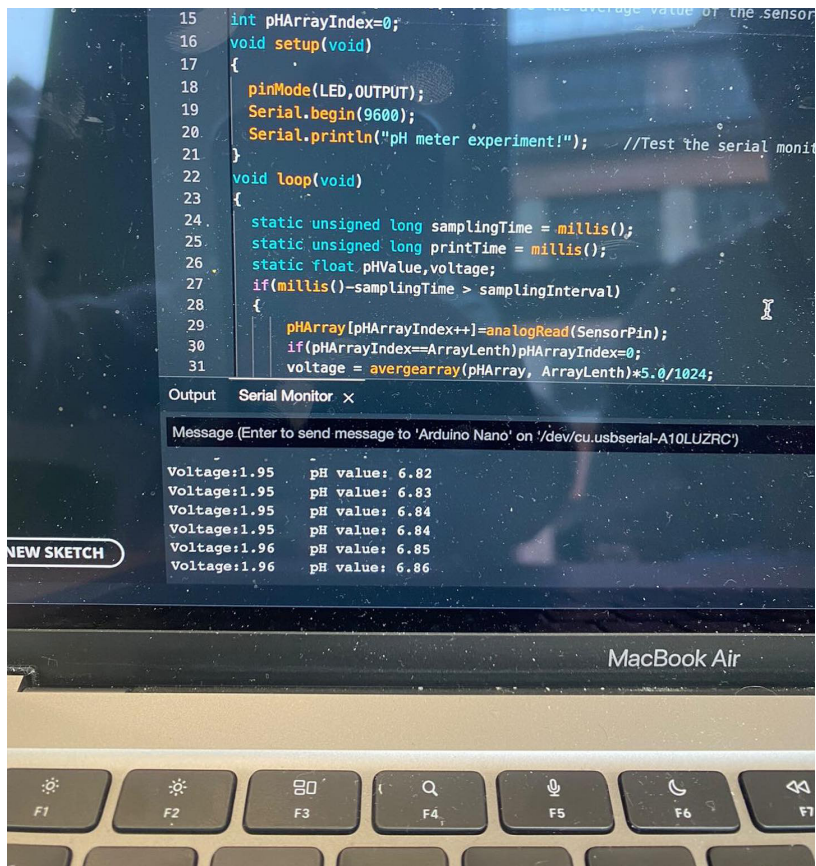
```

⚠ Les codes ci-dessus restent à tester, aucune manipulation n'a été faite (On n'a pas eu accès à notre boîte ouin ouin).]

Mardi 11 Avril

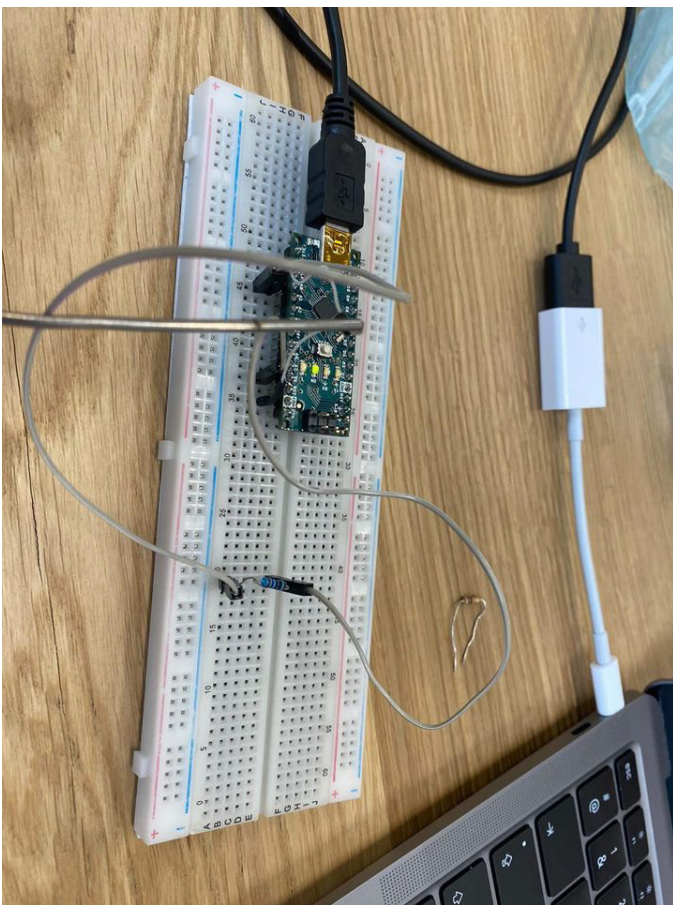
Nous avons eu accès à notre boîte et avons put effectuer quelques manipulations.

Nous avons testé le code pour la mesure du pH que l'on à trouvé sur le wiki de dfrobot; à l'aide d'une sonde de pH, de notre carte Arduino Nano, d'un Mac et de quelques branchements.



Comme les images ci-dessus l'indique, le code a fonctionner et la sonde est en capacité de prélever le pH d'un liquide. (L'eau étant prélevée du robinet il est normal que le pH ne soit pas exactement équivalent à 7, mais qu'il s'en approche)

Nous avons aussi testé le code de prélèvement de température écrit précédemment, le code semble bien fonctionné, on reçoit bien des valeurs de Température dans le Serial Monitor cependant les valeurs ne semblent pas cohérentes probablement dû à un problème de conversion



BIBLIOGRAPHIE

1. *Analog Input | Arduino Documentation.* <https://docs.arduino.cc/built-in-examples/analog/AnalogInput>. Consulté le 30 mars 2023.
2. *analogRead() - Arduino Reference.* <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>. Consulté le 30 mars 2023.
3. *Nano | Arduino Documentation.* <https://docs.arduino.cc/hardware/nano>. Consulté le 30 mars 2023.

4. *SPI - Arduino Reference.*

<https://www.arduino.cc/reference/en/language/functions/communication/spi/>. Consulté le 30 mars 2023.

5. *WiFi101 - Arduino Reference.* <https://www.arduino.cc/reference/en/libraries/wifi101/>. Consulté le 30 mars 2023.

6. « Arduino - Temperature Sensor | Arduino Tutorial ». *Arduino Getting Started*, <https://arduinogetstarted.com/tutorials/arduino-temperature-sensor>. Consulté le 30 mars 2023.

7. *DS18B20 Temperature Sensor For Arduino - DFRobot Wiki.*

https://wiki.dfrobot.com/Gravity_DS18B20_Temperature_Sensor_Arduino_Compatible_V2_SKU_DFR0024. Consulté le 30 mars 2023.

8. SKU SEN0161 PH Meter. https://wiki.dfrobot.com/PH_meter_SKU_SEN0161_. Consulté le 7 avril 2023.

Revision #9

Created 29 March 2023 08:58:43 by Le Bris Nadine

Updated 19 July 2023 10:33:55 by Si Mohammed Yaniss