

Rapport de la Séance numéro 3 : 6 janvier

- Module température

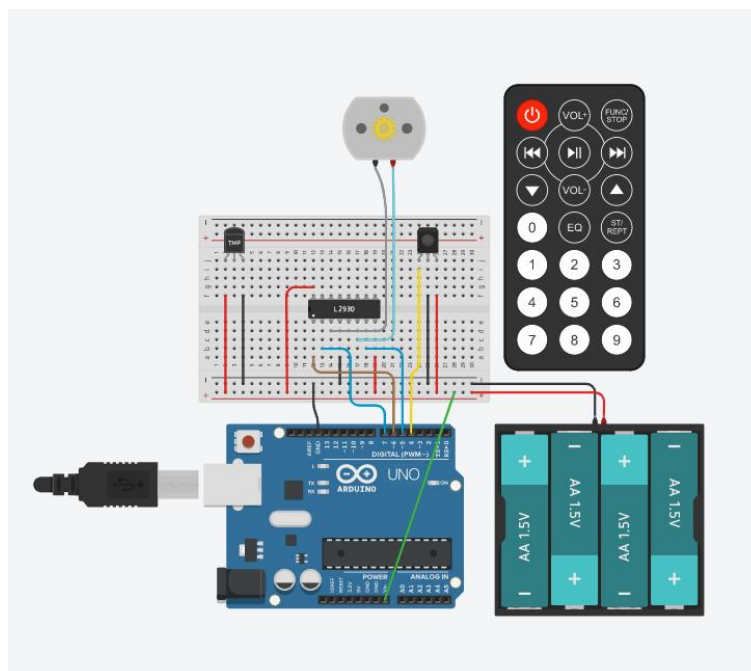
Aujourd'hui j'ai écrit le programme permettant de lire la température de la pièce. J'ai enregistré avec mon téléphone les sons MP3 qui seront lu quand l'utilisateur demandera la température.

- Module volets

J'ai complètement retravaillé mon programme volets et j'ai décidé de représenter le montage sur Tinkercard pour pouvoir avoir une trace du montage beaucoup plus lisible.

- Télécommande

J'ai réalisé le code de la télécommande IR avec son capteur sur Tinkercard pour finaliser la partie volet+télécommande.



Actuellement mon programme ressemble à ca :



```
volets_  
#include <IRremote.h>  
  
int motor1_enablePin = 5; //pwm  
int motor1_in1Pin = 4;  
int motor1_in2Pin = 6;  
int TempPin = 0;  
long IR;  
char TempOut[30];  
char tmp[5];  
long now = 0;  
long nw2 = 0;  
long TT1 = 10000; // timer temperature  
long TT2 = 5000; // timer moteur1  
  
IRrecv irrecv(7); //pin rcpt IR  
  
decode_results results;  
  
void setup()  
{  
  Serial.begin(9600); //initialisation communication série  
  Serial.println("Reception IR");  
  irrecv.enableIRIn(); // Activation IR  
  
  //on initialise les pins du moteur 1  
  pinMode(motor1_in1Pin, OUTPUT);  
  pinMode(motor1_in2Pin, OUTPUT);  
  pinMode(motor1_enablePin, OUTPUT);  
}  
  
void loop()  
{  
  if (irrecv.decode(&results)) // est ce qu'on a recut le signal IR?  
  {  
    Serial.println(results.value);  
    IR = results.value;  
    irrecv.resume();  
  }  
  // telecommande  
  if (IR > 0) {  
    switch (IR)  
    {  
      case 16753245:  
        Serial.println("STOP");  
        stopMotor();  
        break;  
    }  
  }  
}
```

```

        case 16769055:
            Serial.println("DOWN");
            SetMotor1(100, true);
            sprintf(nw2, "%d", millis());
            Serial.println(nw2);
            break;

        case 16748655:
            Serial.println("UP");
            SetMotor1(100, false);
            sprintf(nw2, "%d", millis());
            break;
    } // sw
    IR = 0;
}

// conversion temperature avec thermistance
int tempReading = analogRead(TempPin);
float tempVolts = tempReading * 5.0 / 1024.0;
float tempC = (tempVolts - 0.5) * 10.0;
dtostrf(tempC, 4, 3, tmp);

// afficher temperature toutes les 10 sec
if (millis() > now + TT1) {
    sprintf(TempOut, "Temperature : %s °c", tmp);
    Serial.println(TempOut);
    Serial.println(nw2);
    now = millis();
}

// pour stopper le moteur au bout de 5 sec
if ( nw2 > 100 and ( millis() > nw2 + TT2)) {
    stopMotor();
    Serial.println(nw2);
    sprintf(nw2, "%d", 0);
}

}

//Fonction qui set le moteur1
void SetMotor1(int speed, boolean reverse)
{
    analogWrite(motor1_enablePin, speed);
    digitalWrite(motor1_in1Pin, ! reverse);
    digitalWrite(motor1_in2Pin, reverse);
    Serial.println("Motor on");
}

void stopMotor()
{
    Serial.println("Stop Motor");
    digitalWrite(motor1_in1Pin, false);
    digitalWrite(motor1_in2Pin, false);
}

```