

ATELIER :
RAPPORT TP
ACTIVITES PRELIMINAIRES

TP : Initiation Arduino

L'objectif du TP est de s'initier à l'utilisation de la carte arduino et son fonctionnement. Pour ce faire on a eu accès à diverses activités ; nous permettant de nous (re) familiariser avec les capteurs, les entrées/sorties analogiques & digitales, le port série, et quelques applications d'électronique analogique de support.

2.Clignotement d'une LED

2.1 Modification du programme et mesure

On modifie le programme de sorte à ce que la LED s'allume pendant 100ms et soit éteinte pendant 900ms pour avoir une période de 1 s.

```
void setup() {  
    pinMode(11, OUTPUT);           // initialize digital pin 11 as  
    //an output  
}  
  
// the loop function runs over and over again forever  
void loop() {  
    digitalWrite(11, HIGH);       // turn the LED on (HIGH is the voltage  
    //level)  
    delay(100);                   // wait for 100 ms  
    digitalWrite(11, LOW);        // turn the LED off by making the  
    //voltage LOW  
    delay(900);                   // wait for 900 ms  
}
```

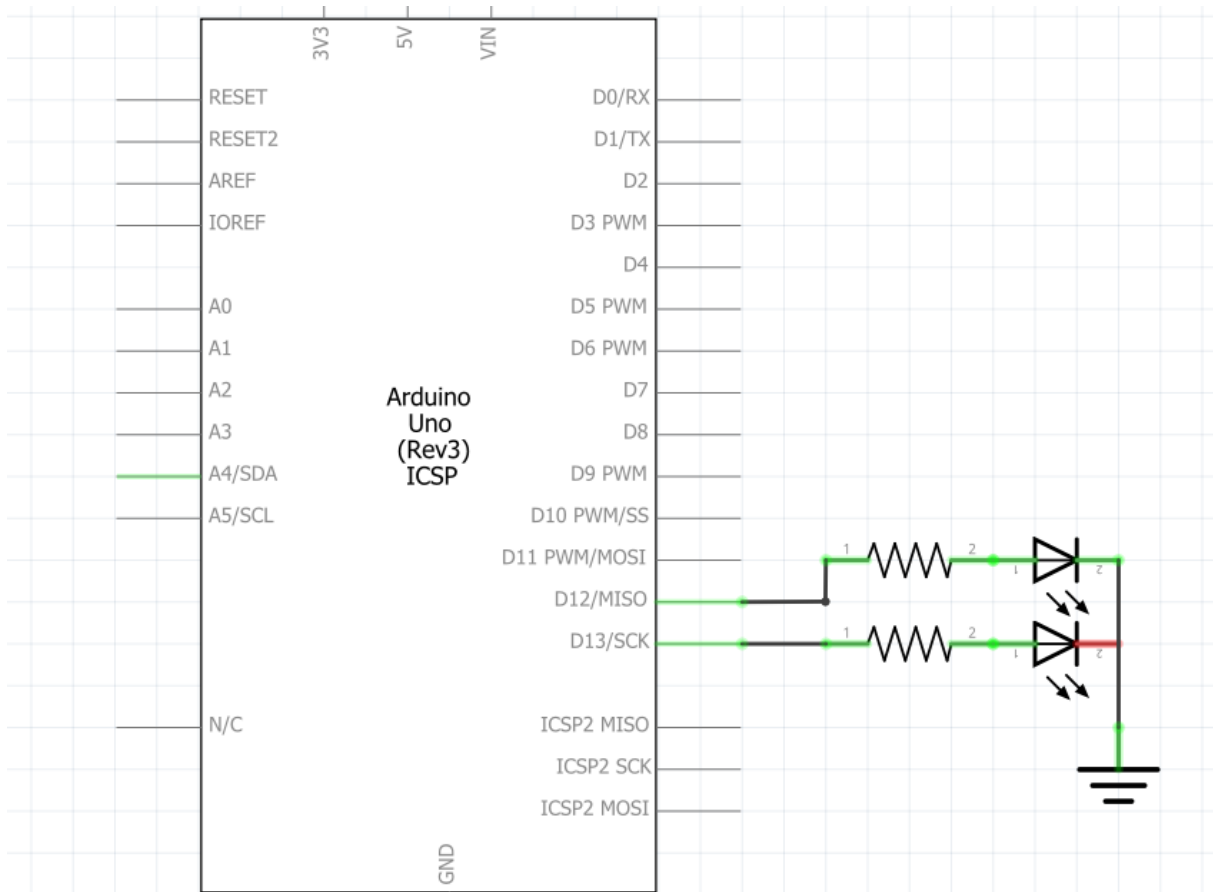
On branche la sonde à la sortie 11 pour relever la tension.



TDS 2012C - 09:44:30 22/01/2020

On relève une tension maximale de 5V.

2.2 Connection des LEDs



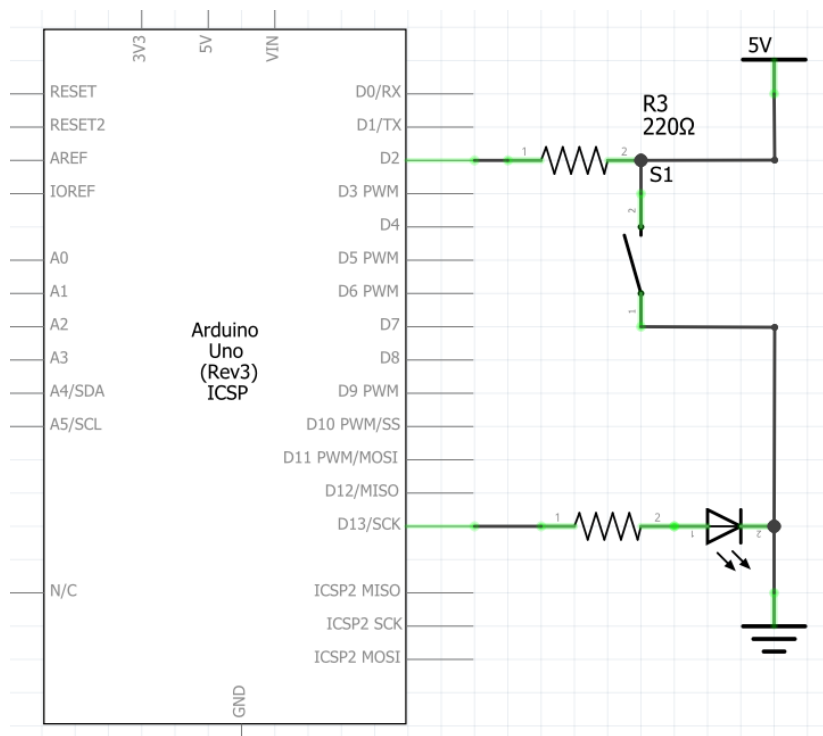
On utilise la relation $U=R \cdot I$ donc $R=U/I$, avec $I=10 \text{ mA}$ et $U=5\text{V}$:

On a $R=500 \Omega$.

Le programme pour allumer les deux LED en opposition de phase :

```
void setup() {  
    pinMode(13, OUTPUT);    // initialize digital pin 13 as an output  
    pinMode(12, OUTPUT);    // initialize digital pin 12 as an output  
}  
  
void loop() {  
    digitalWrite(13, HIGH) ;    //turn the LED on (HIGH is the  
    //voltage level)  
    digitalWrite(12, LOW);    //turn the LED off (LOW is the voltage  
    //level)  
    delay(500);                // wait for 500 ms  
    digitalWrite(13, LOW);    //turn the LED off (LOW is the voltage  
    //level)  
    digitalWrite(12, HIGH);    //turn the LED on (HIGH is the  
    //voltage level)  
    delay(500);                // wait for 500 ms  
}
```

3. Utilisation d'un interrupteur switch



On modifie le programme précédent en ajoutant un bouton :

```
void setup() {  
    pinMode(13, OUTPUT);    // initialize digital pin 13 as an output  
    pinMode(12, OUTPUT);    // initialize digital pin 12 as an output  
    pinMode(2, INPUT);      // initialize digital pin 2 as an input  
}
```

```
void loop() {  
    if(digitalRead(2)==1){  
        blink_opp_phase();  
    }    // if you switch on the button , blink_opp_phase() is  
    //executed  
    else{  
        turnoffLED();  
    }    // if you switch off the button, turnoffLED() is executed  
}
```

```
void turnoffLED(){  
    digitalWrite(13, LOW);    //turn the LED off (LOW is the  
    //voltage level)  
    digitalWrite(12, LOW);    //turn the LED off (LOW is the  
    //voltage level)  
}
```

```
void blink_opp_phase(){  
    digitalWrite(13, HIGH);    //turn the LED on (HIGH is the  
    //voltage level)  
    digitalWrite(12, LOW);    //turn the LED off (LOW is the  
    //voltage level)  
    delay(500);    // wait for 500 ms  
    digitalWrite(13, LOW);    //turn the LED off (LOW is the  
    //voltage level)  
    digitalWrite(12, HIGH);    //turn the LED on (HIGH is the  
    //voltage level)
```

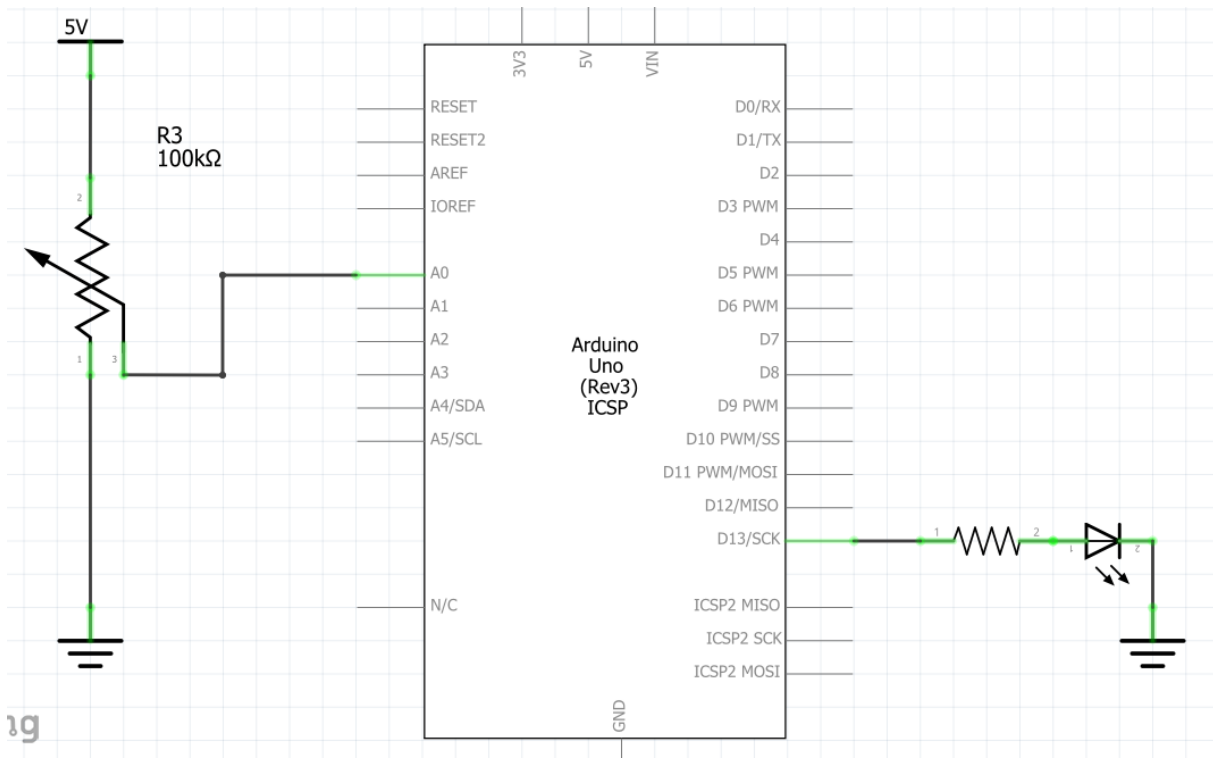
```

    delay(500);           // wait for 500 ms
}

```

4. Utilisation de l'entrée analogique

4.1. Variation de la fréquence d'allumage d'une LED en fonction de la tension d'entrée



On écrit le programme suivant :

```

void setup() {
    pinMode(13, OUTPUT);           // initialize digital pin 13 as an
    //output

    Serial.begin(9600);           // initialize serial communication at
    //9600 bits per second
}

// the loop routine runs over and over again forever:
void loop() {
    int sensorValue = analogRead(A0);           //getting the voltage
    //sensorValue from the pin A0

    Serial.println(sensorValue);           //print sensorValue

```

```

    digitalWrite(13, HIGH);          //turn the LED on (HIGH is the
//voltage level)

    delay(sensorValue/2);           // delay in between reads for
//stability

    digitalWrite(13, LOW);          //turn the LED off (LOW is the
//voltage level)

    delay(sensorValue/2);           // delay
}

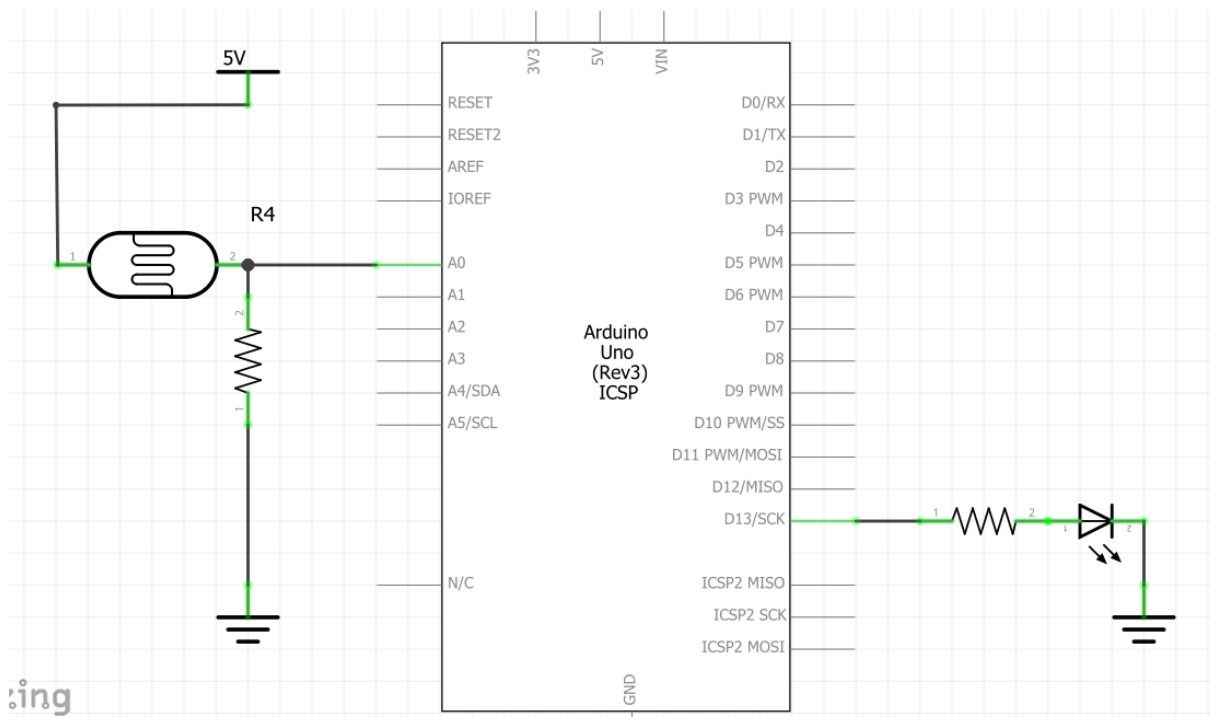
```

4.2. Mesure de l'éclairement

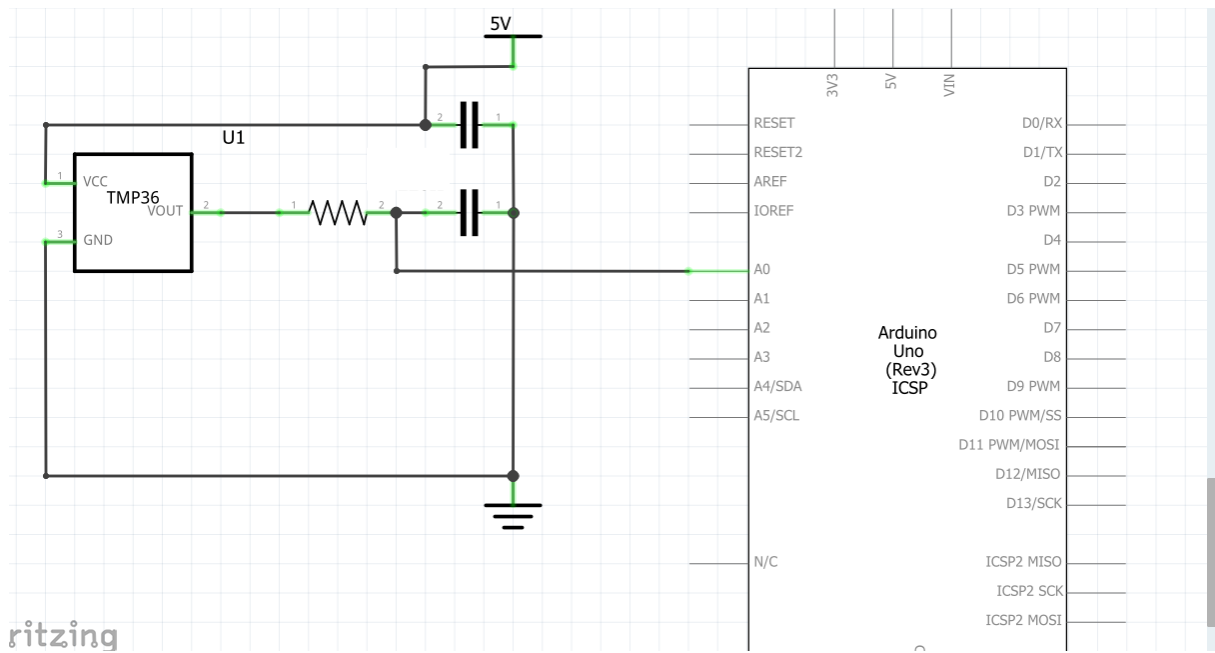
Le capteur fonctionne un peu comme l'œil humain, plus l'intensité lumineuse est forte plus sa résistance diminue.

La tension maximale d'utilisation du capteur est entre 50V et 400 V. La sensibilité est la relation entre la lumière incidente et la réponse du capteur, dans notre cas c'est la relation entre la lumière incidente et la résistance du capteur.

On utilise le montage suivant pour détecter le passage d'un objet :



4.3. Mesure de la température



Le capteur utilise la même propriété qu'une diode : la tension change à un rythme connu en fonction de la température. La tension de sortie est de 500mV pour une température nulle. Le capteur a une sensibilité de $\pm 1^\circ$.

On écrit le programme suivant :

```
int sensorPin = A1;    //the analog pin the TMP36's Vout (sense) pin
                       //is connected to A1

void setup()
{
  pinMode(sensorPin, INPUT);    //initialize sensorPin as an input
  Serial.begin(9600);          // initialize serial communication
  //at 9600 bits per second
}

void loop()
{
  int reading = analogRead(sensorPin);    //getting the voltage
  //reading from the temperature sensor

  float voltage = reading * 5.0;
```



```

voltage /= 1024.0;      // converting reading to voltage

Serial.print(reading); Serial.println(" volts");      // print out
//the voltage

float temperatureC = (voltage - 0.5) * 100 ; //converting from 10
//mv per degree wit 500 mV offset
//to degrees ((voltage - 500mV) times 100)

Serial.print(temperatureC); Serial.println(" degrees C"); // print
//the temperature

delay(1000);          //wait a second
}

```

Pour déterminer la valeur de R , on utilise la fonction de transfert Vs/Ve :

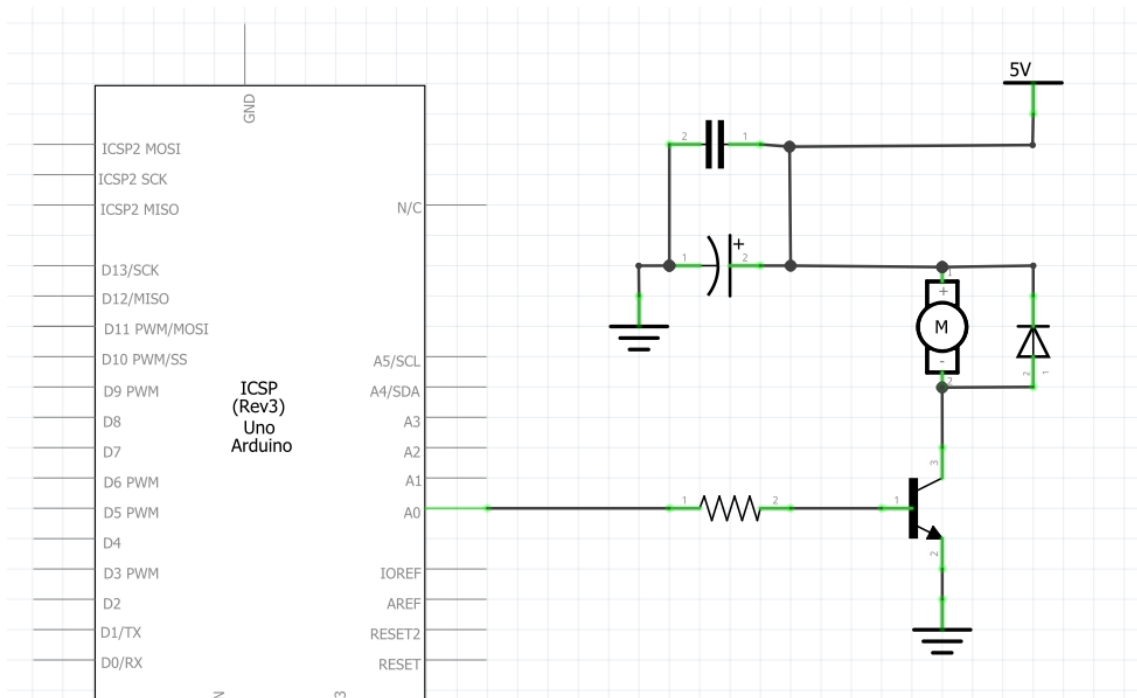
$$V_s/V_e = (1/jC\omega) / (R + (1/jC\omega)) = 1 / (1 + jRC\omega) = 1 / (1 + (\omega/\omega_c)) \quad \text{avec } \omega_c = 1/RC$$

Or $\omega_c = 2 * \pi * f_c$ avec $f_c = 10$ Hz, donc $R = 1 / (2 * \pi * f_c * C)$ avec $C = 100$ nF.

On retrouve donc $R = 160$ k Ω .

La sortie est plus stable car les composants ajoutés ont filtré les hautes fréquences.

5. Commande d'un moteur à courant continu en PWM



Pour déterminer R :

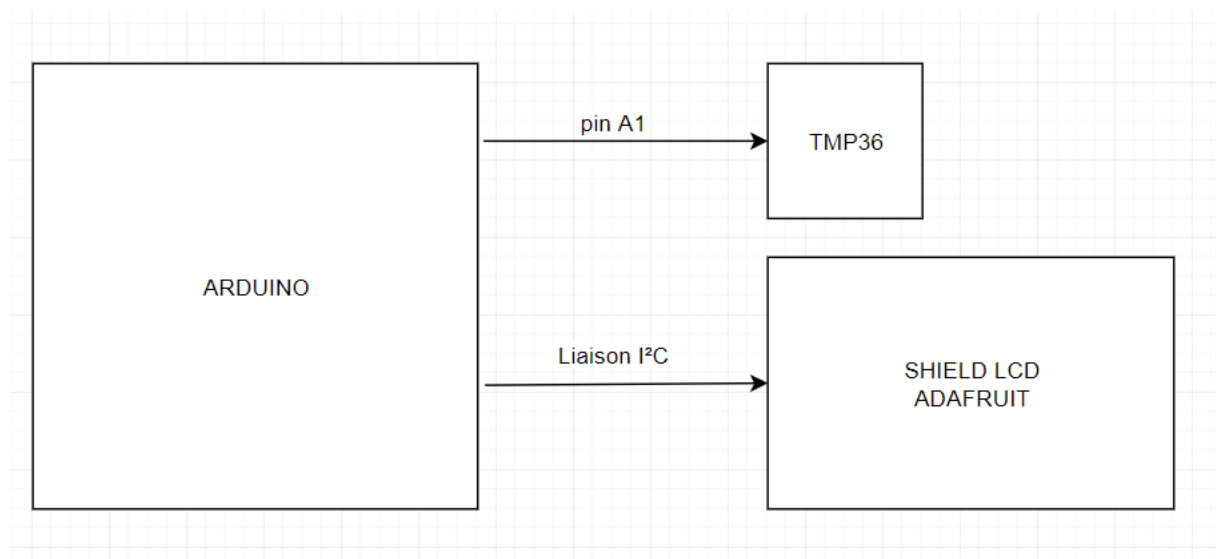
$$5 = R \cdot I_b + V_{be} \Leftrightarrow R = (5 - V_{be}) / I_b \Leftrightarrow R = \beta \cdot ((5 - V_{be}) / I_c) \Leftrightarrow R = 100 \cdot ((5 - 0.6) / 0.5) \Leftrightarrow R = 820 \Omega$$

La diode permet d'éviter un reflux de tension vers le collecteur.

Les condensateurs réduisent le couplage entre le signal et l'alimentation avec le découplage, une éventuelle pointe de courant ne provoquerait pas de chute de tension. Les capacités emmagasinent de l'énergie pour « boucher les trous ».

L'arduino délivre une tension de 5.1V, la tension Vce du transistor est environ égale à 7 V.

6. Affichage d'une grandeur sur un afficheur LCD



Pour afficher la température sur un écran LCD, on utilise le code suivant :

```
int sensorPin = A1;    //the analog pin the TMP36's Vout (sense) pin
//is connected to A1

void setup() {
    Serial.begin(9600);    // initialize serial communication at 9600
//bits per second

    lcd.begin(16, 2);    //intialise the LCD with 16 columns and 2
//lines

    pinMode(sensorPin, INPUT);    //initialize sensorPin as an input.

    // Print a message to the LCD. We track how long it takes since
    // this library has been optimized a bit
```

```

    lcd.setBacklight(WHITE);
}

void loop() {
    lcd.setCursor(0,0);    // set the LCD on the first column of the
    //first line

    int read1 = analogRead(sensorPin);    //getting the voltage
    //read1 from the temperature sensor

    delay(100);            //wait 100 ms

    int read2 = analogRead(sensorPin);    //getting the voltage read2
    //from the temperature sensor

    delay(100);            //wait 100 ms

    int read3 = analogRead(sensorPin);    //getting the voltage read3
    //from the temperature sensor

    int read = (read1 + read2 + read3) / 3;    // read is equal to
    //the average of read1, read2 and read3

    float tens = read*5.0;

    tens /= 1024.0;        // converting read to voltage

    lcd.print("TENS: "); lcd.print(read); lcd.println("mV    ");
    // print the voltage on the first column of
    //the first line of the lcd

    float temp = (tens -0.5)*100;    //converting from 10 mv per
    //degree wit 500 mV offset

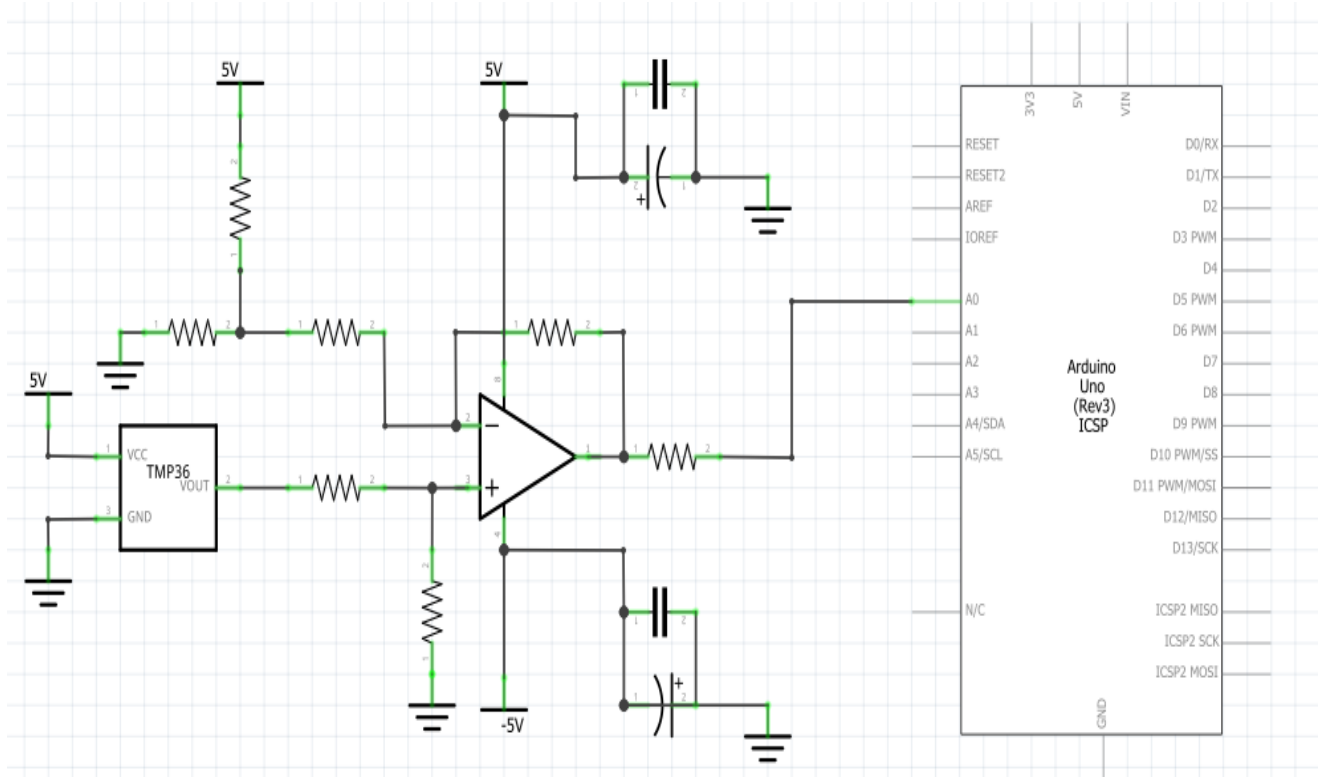
    lcd.setCursor(0,1);    // set the LCD on the first column of the
    //second line

    lcd.print("TEMP: "); lcd.print(temp); lcd.print(" degC");
    //print the voltage on the first column of
    //the second line of the lcd

    delay(1000);        //wait a second
}

```

7. Utilisation d'un AOP



On établit l'équation :

$$V_{out} = VR_4 * (-R_2/R_1) + V_{Tmp36} * (R_2/(R_1+R_2)) * (1+(R_2/R_1))$$

$$V_{out} = VR_4 * (-R_2/R_1) + V_{Tmp36} * (R_2/(R_1+R_2)) * ((R_1+R_2)/R_1)$$

$$V_{out} = VR_4 * (-R_2/R_1) + V_{Tmp36} * (R_2/R_1)$$

$$V_{out} = (R_2/R_1) * (V_{Tmp36} - VR_4)$$

Avec $VR_4 = 5 * (R_4/(R_3+R_4)) = 0.75$ et $R_2/R_1 = 10$

On trouve $V_{Tmp36} = 0.1 * V_{out} + 0.75$

On utilise cette équation pour notre programme :

```
int sensorPin = A1;        //the analog pin the TMP36's Vout (sense)
                             //pin is connected to A1
```

```

void setup()
{
  pinMode(sensorPin, INPUT);      //initialize sensorPin as an input
  Serial.begin(9600);             // initialize serial communication at
  //9600 bits per second
}

void loop()
{
  int reading = analogRead(sensorPin); //getting the voltage
  //reading from the AOP

  float voltage = reading * 5.0;
  voltage /= 1024.0;              // converting reading to voltage

  float reading2 = (0.1*voltage)+0.75; //converting the voltage of
  // AOP to find the voltage of the
  // temperature sensor

  float temperatureC = (reading2-0.5)*100; //converting from 10 mv
  //per degree wit 500 mV offset

  Serial.print(temperatureC); Serial.println(" degrees C"); // print
  //the temperature

  delay(1000);                    //wait a second
}

```

Nous pouvons conclure que ce TP nous a permis de se remémorer et de se familiariser à nouveau avec le fonctionnement de la carte arduino et des différents composants.