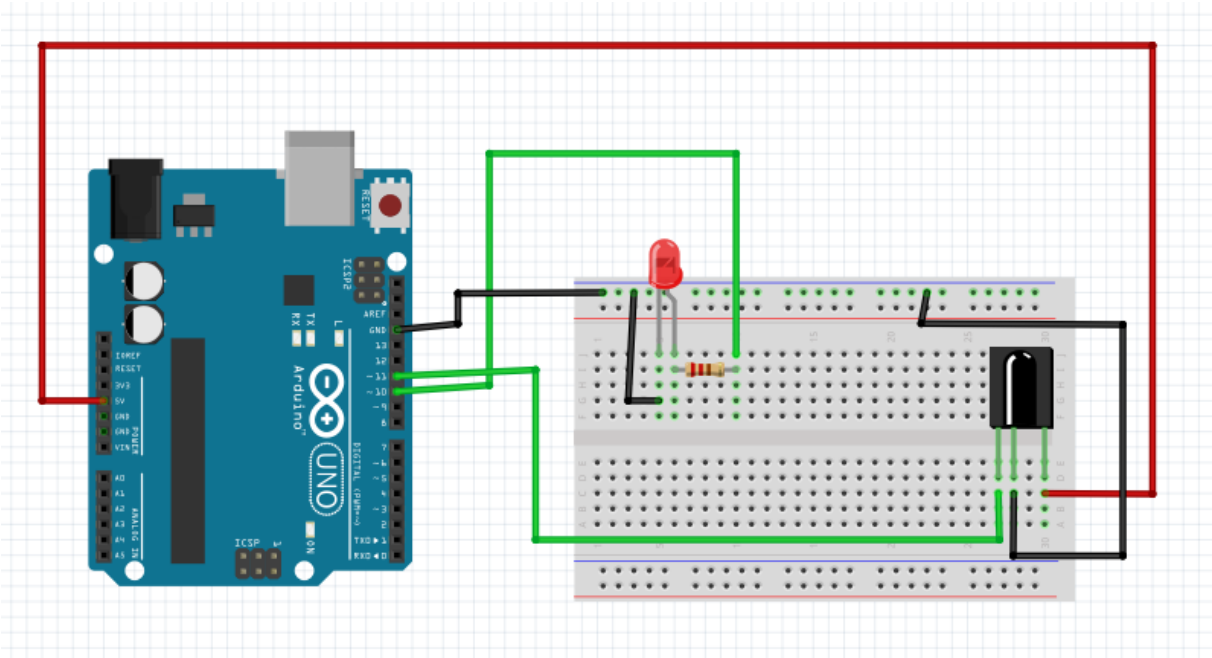
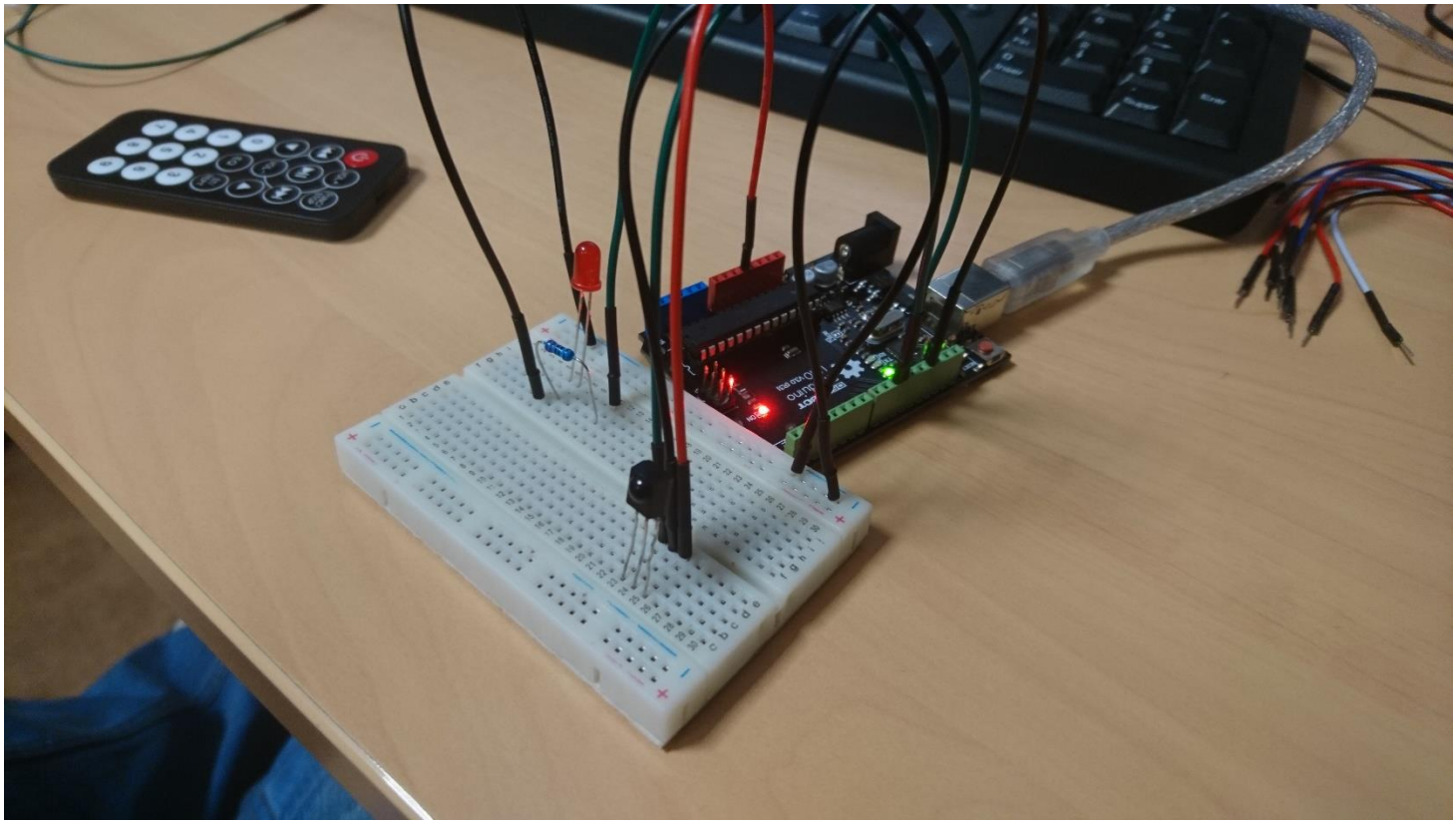
Projet n°14 : LED contrôlée par télécommande infrarouge

```
#include <IRremote.h> //Insère la librairie nécéssaire pour la télécommande (Il faudra peut-être la télécharger d'abord)
int RECV_PIN = 11; //Le port 11 correspond au capteur infrarouge
int ledPin = 10;  //Le port 10 est pour la LED
boolean ledState = LOW; //Variable correspondant a l'état de la LED
IRrecv irrecv(RECV_PIN);  //Définit RECV_PIN comme étant un capteur infrarouge
decode_results results; //Définit results comme résultats du capteur infrarouge

void setup() {
  Serial.begin(9600); //Démarre le port Série
  irrecv.enableIRIn();  //Démarre le capteur infrarouge
  pinMode(ledPin,OUTPUT); //Définit la LED en tant que sortie
}

void loop() {
  if (irrecv.decode(&results)){ //Vérifie que le capteur infrarouge reçoit des données
    Serial.println(results.value,HEX);  //Affiche la valeur hexadécimale du signal reçu dans le moniteur série
    if(results.value == 0xFD00FF){  //Valeur hexadécimale correspondant au bouton On/Off de la télécommande
      ledState = !ledState; //Inversion
      digitalWrite(ledPin, ledState); //Change l'état de la LED
    }
    irrecv.resume();  //Relance le capteur infrarouge
  }
}
```

Schéma



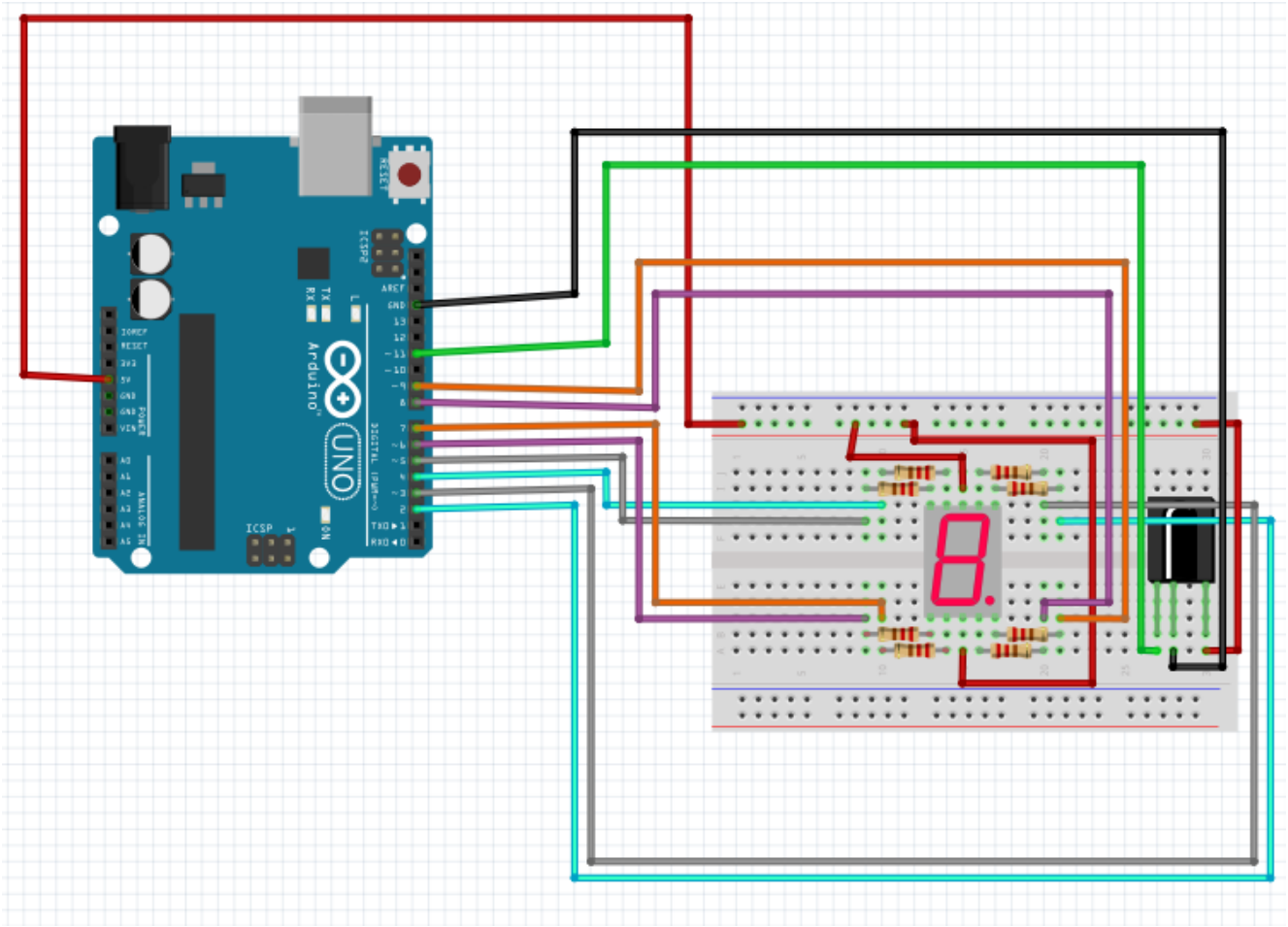
Résultat Final



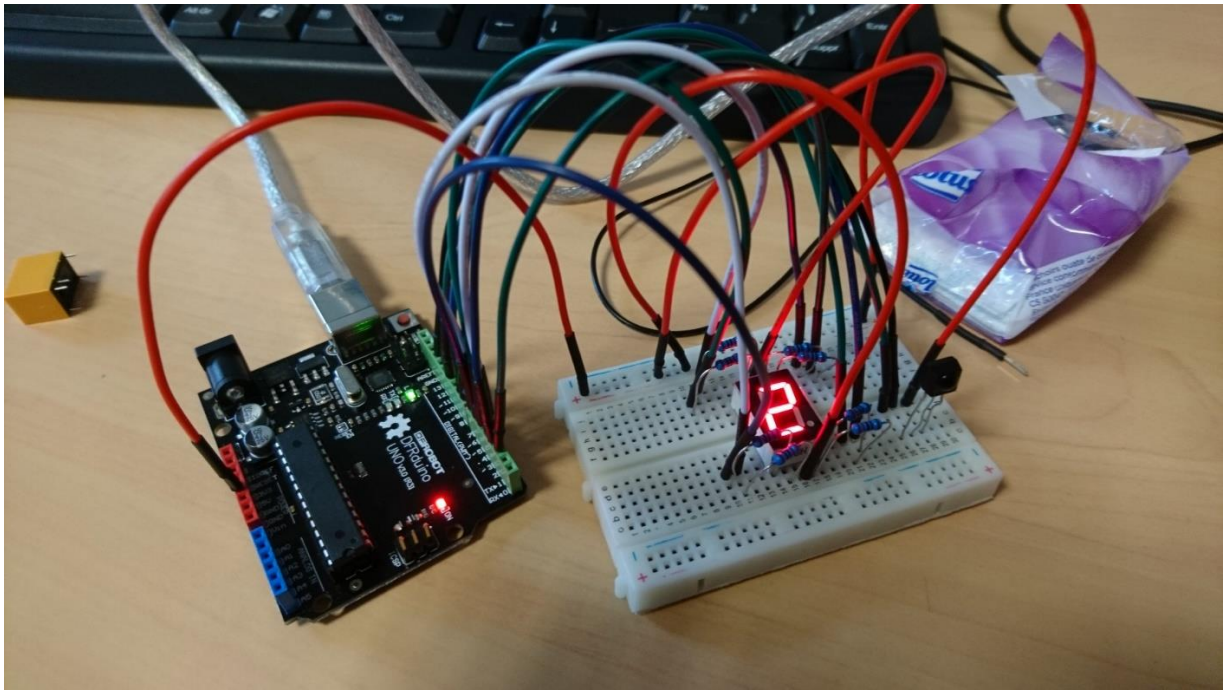
Projet n°15 : Affichage digital contrôlé par télécommande infrarouge

```
#include <IRremote.h> //Insère la librairie nécéssaire pour la télécommande
int RECV_PIN = 11;  //Le port 11 correspond au capteur infrarouge
IRrecv irrecv(RECV_PIN); //Définit RECV_PIN comme étant un capteur infrarouge
decode_results results; //Définit results comme résultats du capteur infrarouge
int currentNumber = 0;  //Crée la variable currentNumber
long codes[12]= //Codes pour la télécommande IR correspondant aux boutons 0 à 9, arrière et avant
{
  0xFD30CF,0xFD08F7,  //0,1
  0xFD8877,0xFD48B7,  //2,3
  0xFD28D7,0xFDA857,  //4,5
  0xFD6897,0xFD18E7,  //6,7
  0xFD9867,0xFD58A7,  //8,9
  0xFD20DF,0xFD609F,  //arrière,avant
};
int number[10][8] = //Tableau pour l'affichage digital
{
  {0,0,0,1,0,0,0,1},  //0 (upright,up,upleft,center,downleft,down,downright,dot)
  {0,1,1,1,1,1,0,1},  //1
  {0,0,1,0,0,0,1,1},  //2
  {0,0,1,0,1,0,0,1},  //3
  {0,1,0,0,1,1,0,1},  //4
  {1,0,0,0,1,0,0,1},  //5
  {1,0,0,0,0,0,0,1},  //6
  {0,0,1,1,1,1,0,1},  //7
  {0,0,0,0,0,0,0,1},  //8
  {0,0,0,0,1,0,0,1},  //9
};
void numberShow(int i) {  //Fonction qui active l'affichage digital
for(int pin = 2; pin <= 9 ; pin++){
  digitalWrite(pin, number[i][pin - 2]);
}
}
void setup(){
  Serial.begin(9600); //Démarre le port série
  irrecv.enableIRIn();  //Démarre le capteur infrarouge
  for(int pin = 2 ; pin <= 9 ; pin++){  //Sélectionne les ports 2 à 9 comme sorties
    pinMode(pin, OUTPUT);
    digitalWrite(pin, HIGH);
  }
}

void loop() {
  if (irrecv.decode(&results)) {  //Vérifie que le capteur infrarouge reçoit des données
    for(int i = 0; i <= 11; i++){
      if(results.value == codes[i]&& i <= 9){ //Vérifie que le bouton pressé est entre 0 et 9
        numberShow(i);
        currentNumber = i;
        Serial.println(i);
        break;
      }
      else if(results.value == codes[10]&& currentNumber != 0){ //Vérifie que le bouton pressé est arrière et que le nombre actuel n'est pas 0
        currentNumber--;  //Diminue le nombre affiché
        numberShow(currentNumber);
        Serial.println(currentNumber);
        break;
      }
      else if(results.value == codes[11]&& currentNumber != 9){ //Vérifie que le bouton pressé est avant et que le nombre actuel n'est pas 9
        currentNumber++;  //Augmente le nombre affiché
        numberShow(currentNumber);
        Serial.println(currentNumber);
        break;
      }
    }
  Serial.println(results.value, HEX); //Affiche la valeur hexadécimale du signal reçu dans le moniteur série
  irrecv.resume();  //Relance le capteur infrarouge
  }
}
```

Schéma



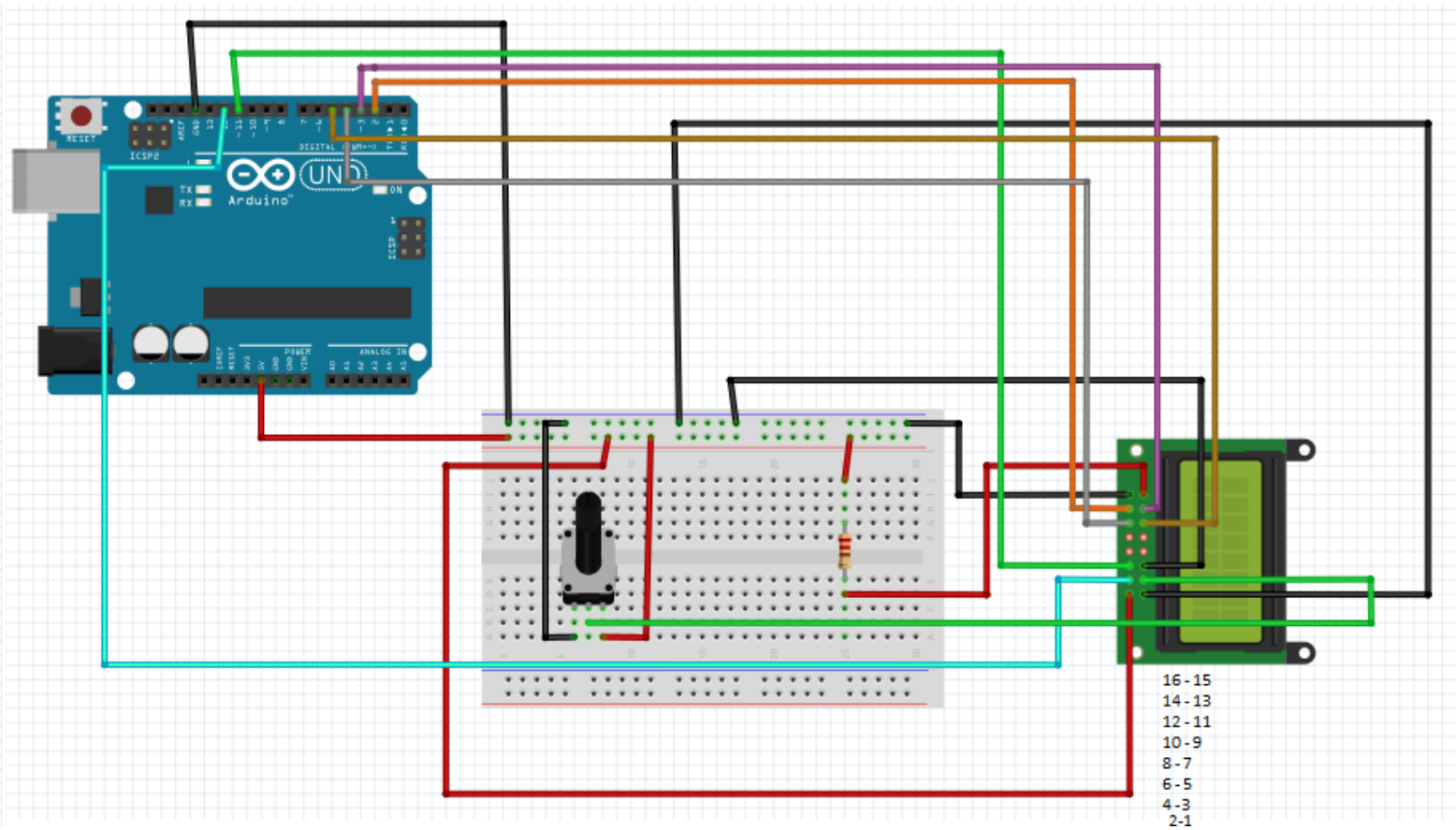
Résultat Final



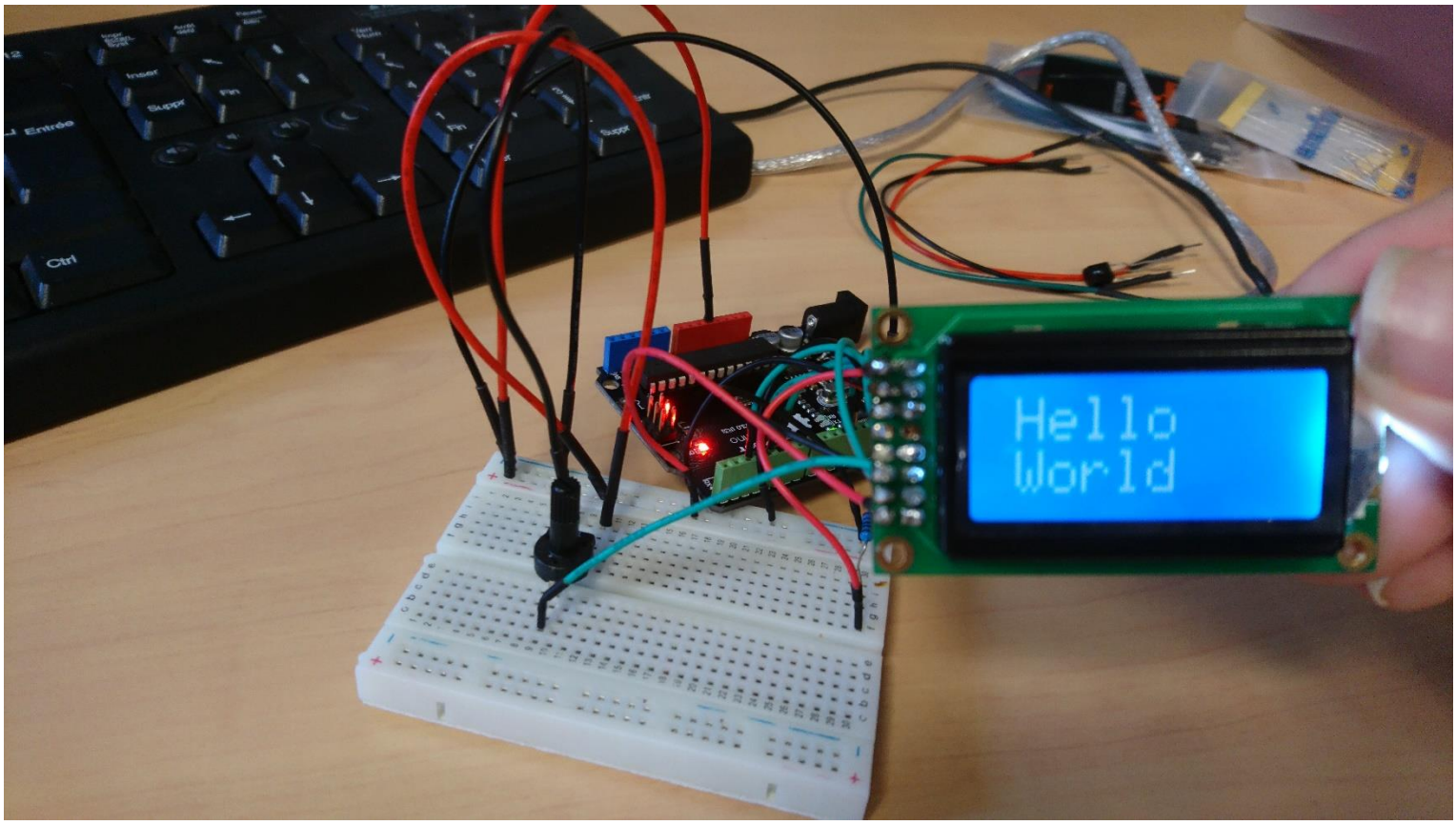
Projet n°16 : Ecran LCD

```
#include <LiquidCrystal.h>  //Bibliothèque nécessaire pour le LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  //Déclaration des ports du LCD

void setup() {
  lcd.begin(8,2); //Taille de l'écran LCD
  lcd.clear();  //Reset l'affichage du LCD
  lcd.print("Hello");  //Affiche le texte sur l'écran
  lcd.setCursor(0,1); //Passage à la ligne du dessous
  lcd.print("World");
}

void loop() {
}
```

Schéma



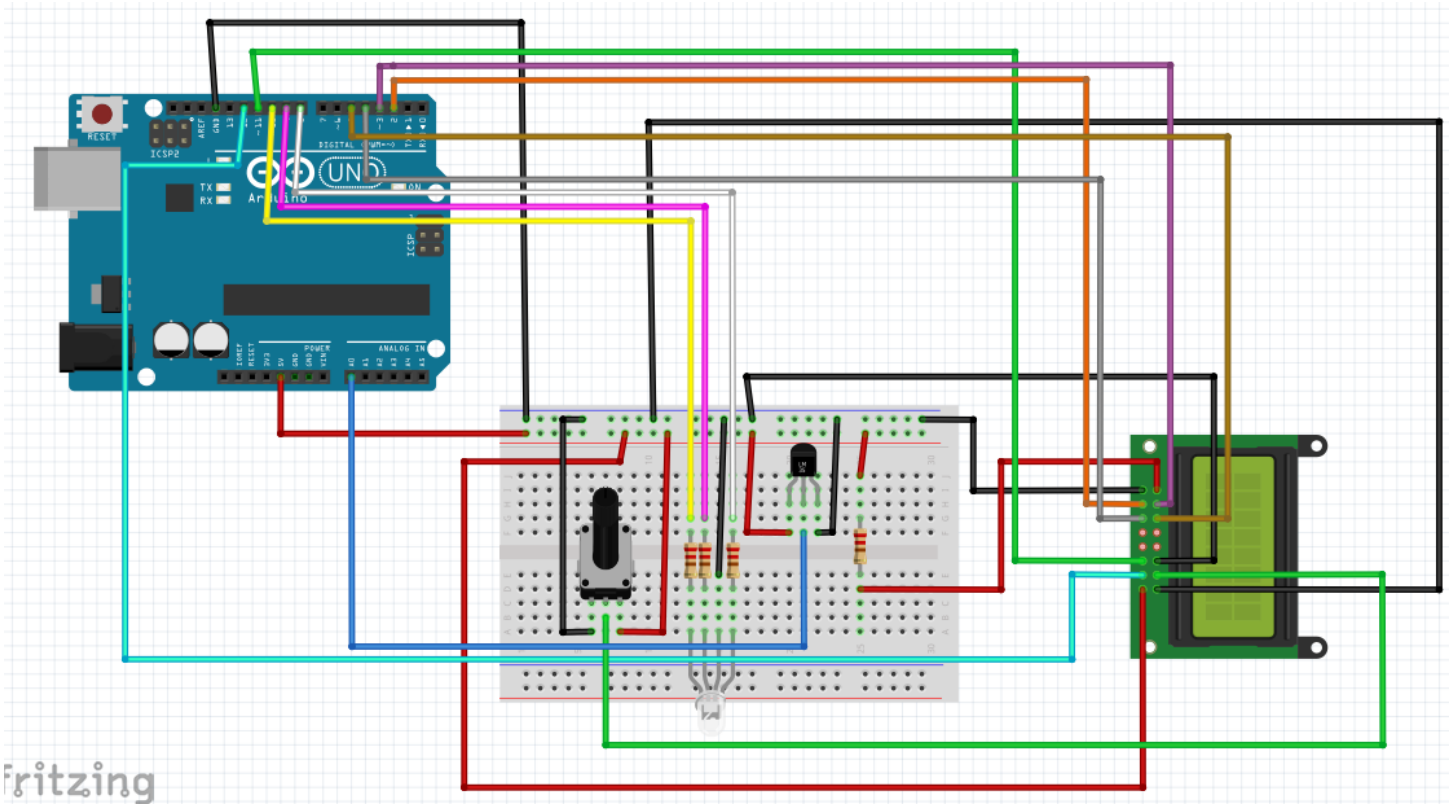
Résultat Final



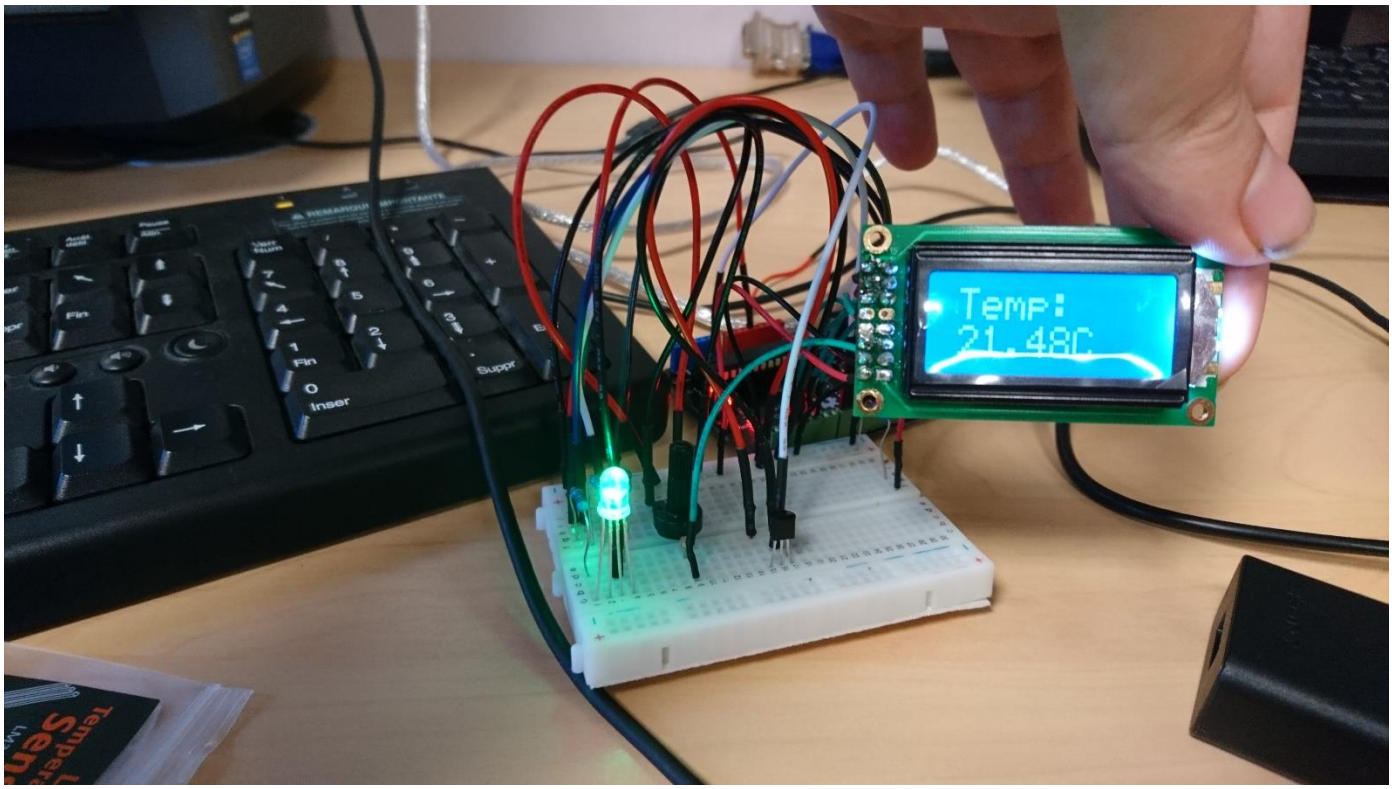
Projet n°17 : Ecran LCD lié à un capteur de température

```
#include <LiquidCrystal.h>  //Bibliothèque nécessaire pour le LCD
int redPin = 8, bluePin = 9, greenPin = 10; //Ports LED RGB
unsigned long TepTimer;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  //Déclaration des ports du LCD

void setup() {
  lcd.begin(8,2); //Taille de l'écran LCD
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  Serial.begin(9600);
}

void colorRGB(int red, int green, int blue) { //Fonction qui change la couleur de la LED
  analogWrite(redPin, constrain (red,0,255));
  analogWrite(greenPin, constrain (blue,0,255));
  analogWrite(bluePin, constrain (green,0,255));
}

void loop() {
  int val;  //Valeur du capteur LM35
  double data;  //Variable de la temperature
  val = analogRead(0);
  data = (double) val * (5/10.24);  //Conversion
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Temp:");
  lcd.setCursor(0,1);
  lcd.print(data);
  lcd.print("C");
  if (data < 20)  { //Si moins de 20°C, LED bleue
    colorRGB(0,0,100);
  }
  else if (data < 30) { //Sinon, si moins de 30°C LED verte
    colorRGB(0,100,0);
  }
  else {
    colorRGB(150,0,0);  //Sinon, LED rouge
  }
  delay (1000); //Temporise une seconde
}
```

Schéma



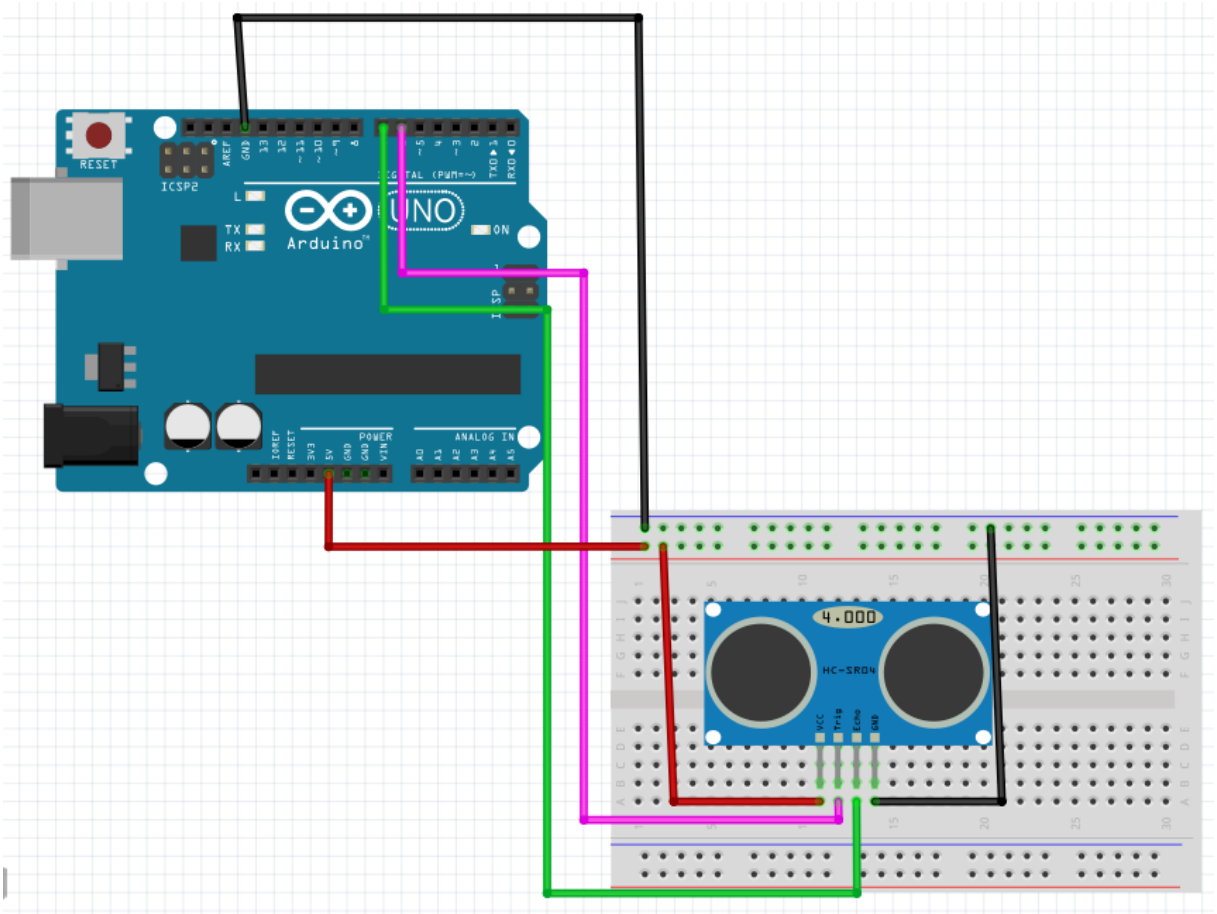
Résultat Final



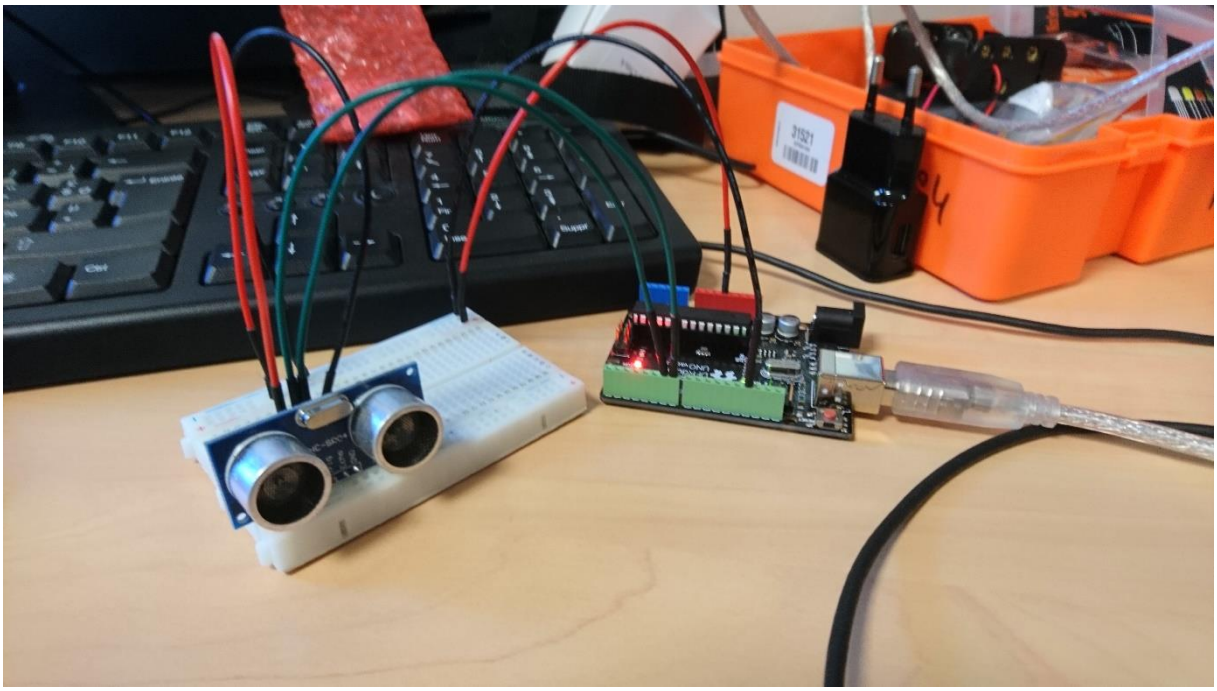
Projet n°18 : Capteur Ultrason

```
const int triggerPin = 6; //Broche Trig du capteur ultrason
const int echoPin = 7; //Broche Echo du capteur ultrason
long distance;

void setup() {
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  Serial.begin(9600);
}

void loop() {
  delay(500);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH); //Envoi du signal ultrason
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  distance = pulseIn(echoPin, HIGH) / 58; //Convertit la valeur sortant du capteur en cm
  if (distance == 0) {  //Reset la broche Echo si elle reste bloquée à 0
    delay (100);
    pinMode (echoPin, OUTPUT);
    digitalWrite (echoPin, LOW);
    delay (100);
    pinMode (echoPin, INPUT);
  }
  Serial.print("Distance:");
  Serial.print(distance);
  Serial.print("cm");
}
```

Schéma



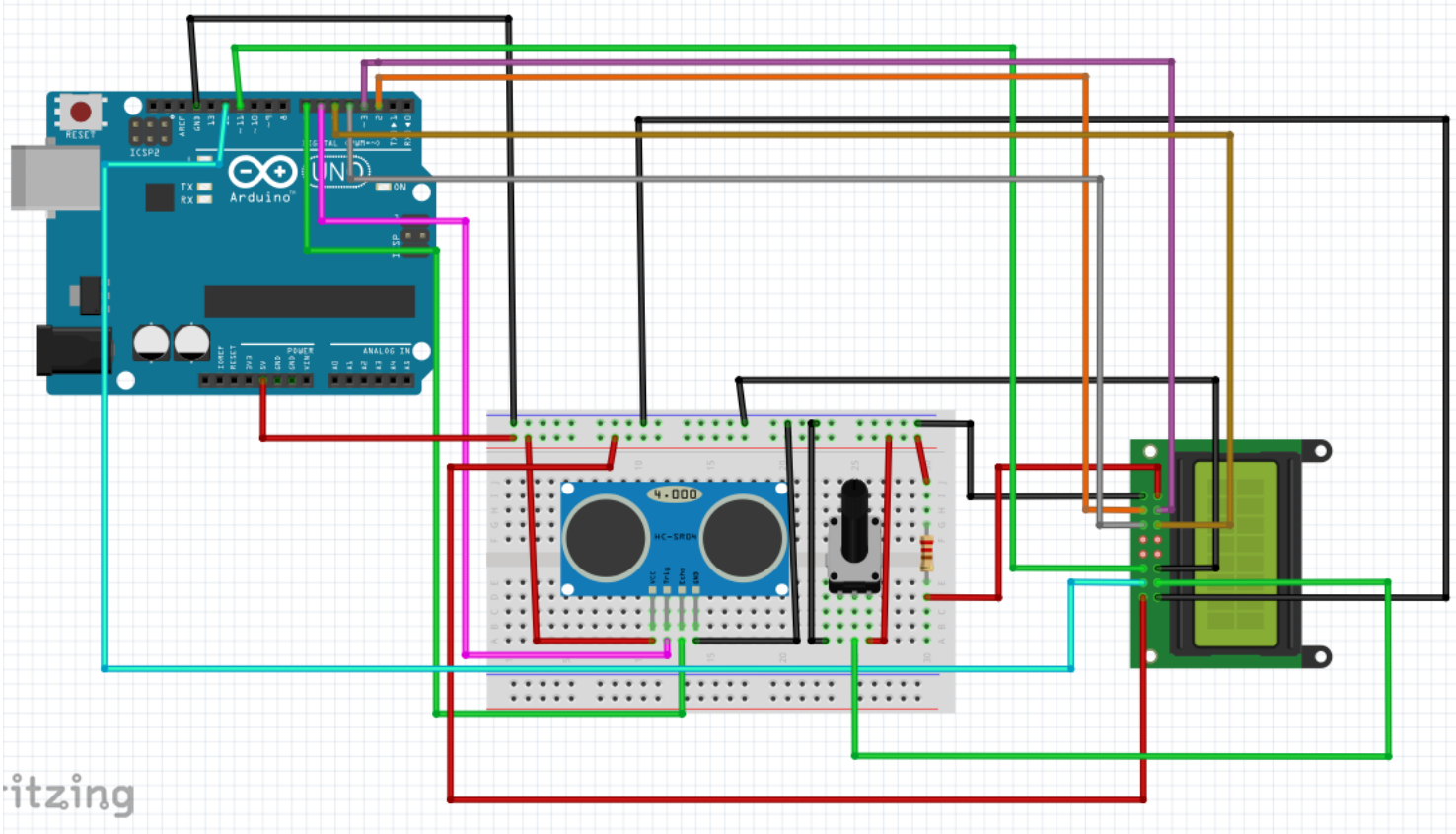
Résultat Final



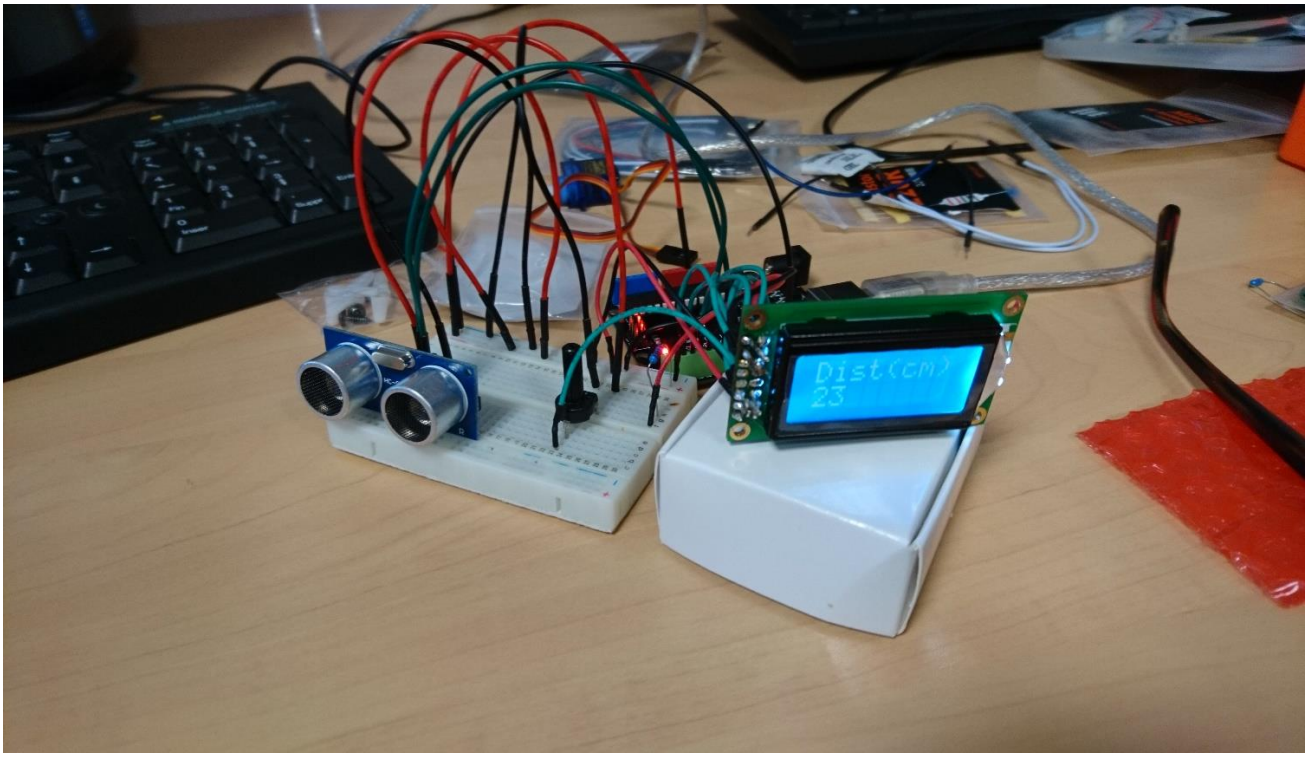
Project n°19 : Capteur ultrason lié à un écran LCD

```
#include <LiquidCrystal.h>  //Bibliothèque nécessaire pour le LCD
const int triggerPin = 6; //Broche Trig du capteur ultrason
const int echoPin = 7; //Broche Echo du capteur ultrason
long distance;
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);  //Déclaration des ports du LCD

void setup() {
  lcd.begin(8,2); //Taille de l'écran LCD
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
}

void loop() {
  delay(500);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  distance = pulseIn(echoPin, HIGH) / 58; //Convertit la valeur sortant du capteur en cm
  if (distance == 0) {  //Reset la broche Echo si elle reste bloquée à 0
    delay (100);
    pinMode (echoPin, OUTPUT);
    digitalWrite (echoPin, LOW);
    delay (100);
    pinMode (echoPin, INPUT);
  }
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Dist(cm):");
  lcd.setCursor(0,1);
  lcd.print(distance);
}
```

Schéma



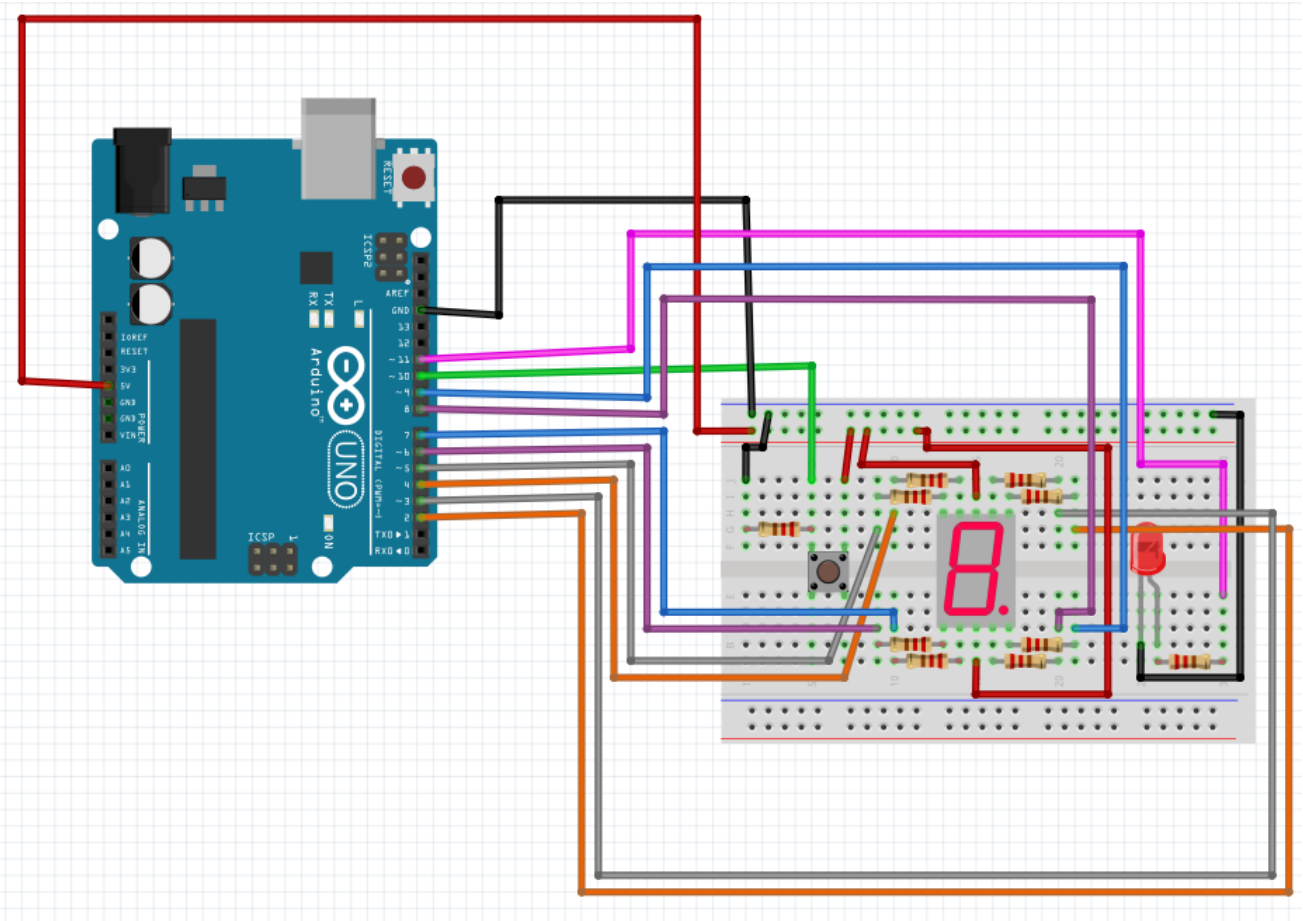
Résultat Final



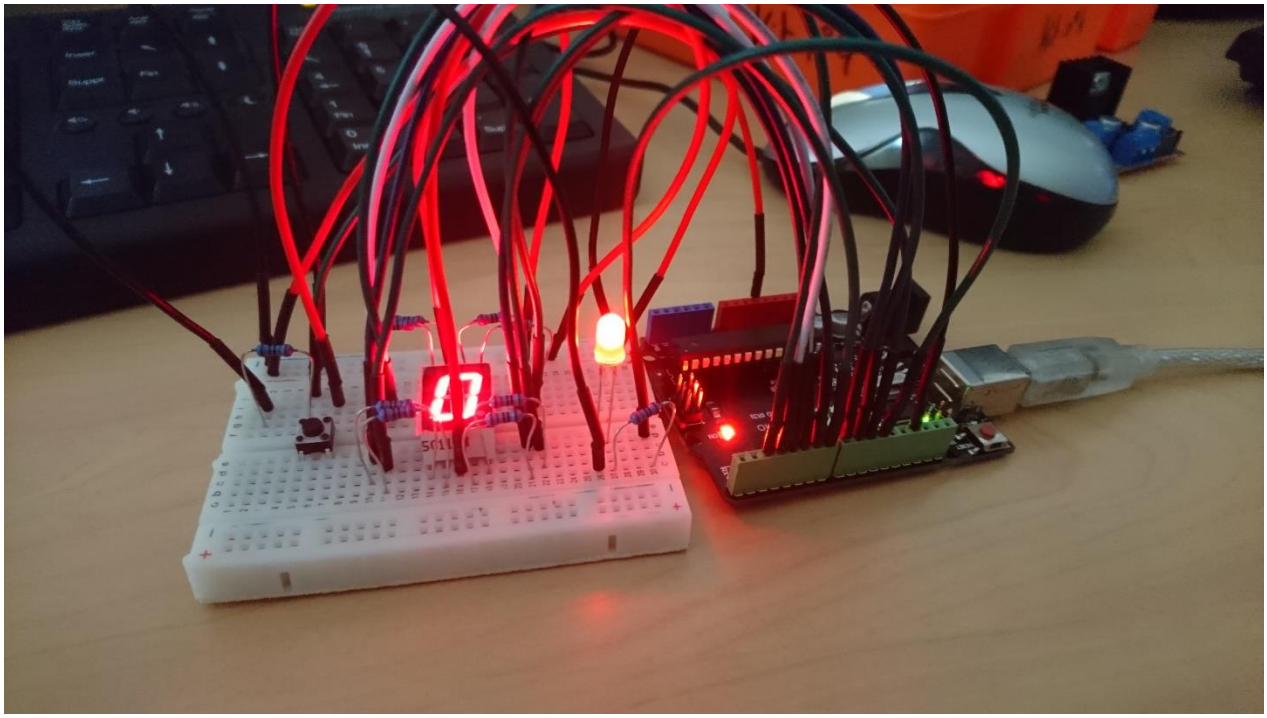
Projet n°20 : Décompteur

```
int currentNumber = 0;  //Crée la variable currentNumber
int button = 10;  //Port Bouton
int led = 11;
int number[10][8] = //Tableau pour l'affichage digital
{
  {0,0,0,1,0,0,0,1},  //0 (upright,up,upleft,center,downleft,down,downright,dot)
  {0,1,1,1,1,1,0,1},  //1
  {0,0,1,0,0,0,1,1},  //2
  {0,0,1,0,1,0,0,1},  //3
  {0,1,0,0,1,1,0,1},  //4
  {1,0,0,0,1,0,0,1},  //5
  {1,0,0,0,0,0,0,1},  //6
  {0,0,1,1,1,1,0,1},  //7
  {0,0,0,0,0,0,0,1},  //8
  {0,0,0,0,1,0,0,1},  //9
};

void numberShow(int i) {  //Fonction qui active l'affichage digital
for(int pin = 2; pin <= 9 ; pin++){
  digitalWrite(pin, number[i][pin - 2]);
  }
}

void countdown() {  //Fonction du dé 6 faces
  delay(1000);
  currentNumber--;
  numberShow(currentNumber);
}

void setup(){
  Serial.begin(9600); //Démarre le port série
  for(int pin = 2 ; pin <= 9 ; pin++){  //Sélectionne les ports 2 à 9 comme sorties
    pinMode(pin, OUTPUT);
    digitalWrite(pin, HIGH);
  }
  pinMode(led,OUTPUT);
}

void loop() {
  int state = digitalRead(button); //Lit l'état du bouton
  if (state == HIGH){ //Bouton activé
    currentNumber = 9;
    numberShow(currentNumber);
    digitalWrite(led, LOW);
    countdown();
    countdown();
    countdown();
    countdown();
    countdown();
    countdown();
    countdown();
    countdown();
    countdown();
    digitalWrite(led, HIGH);
  }
}
```

Schéma :



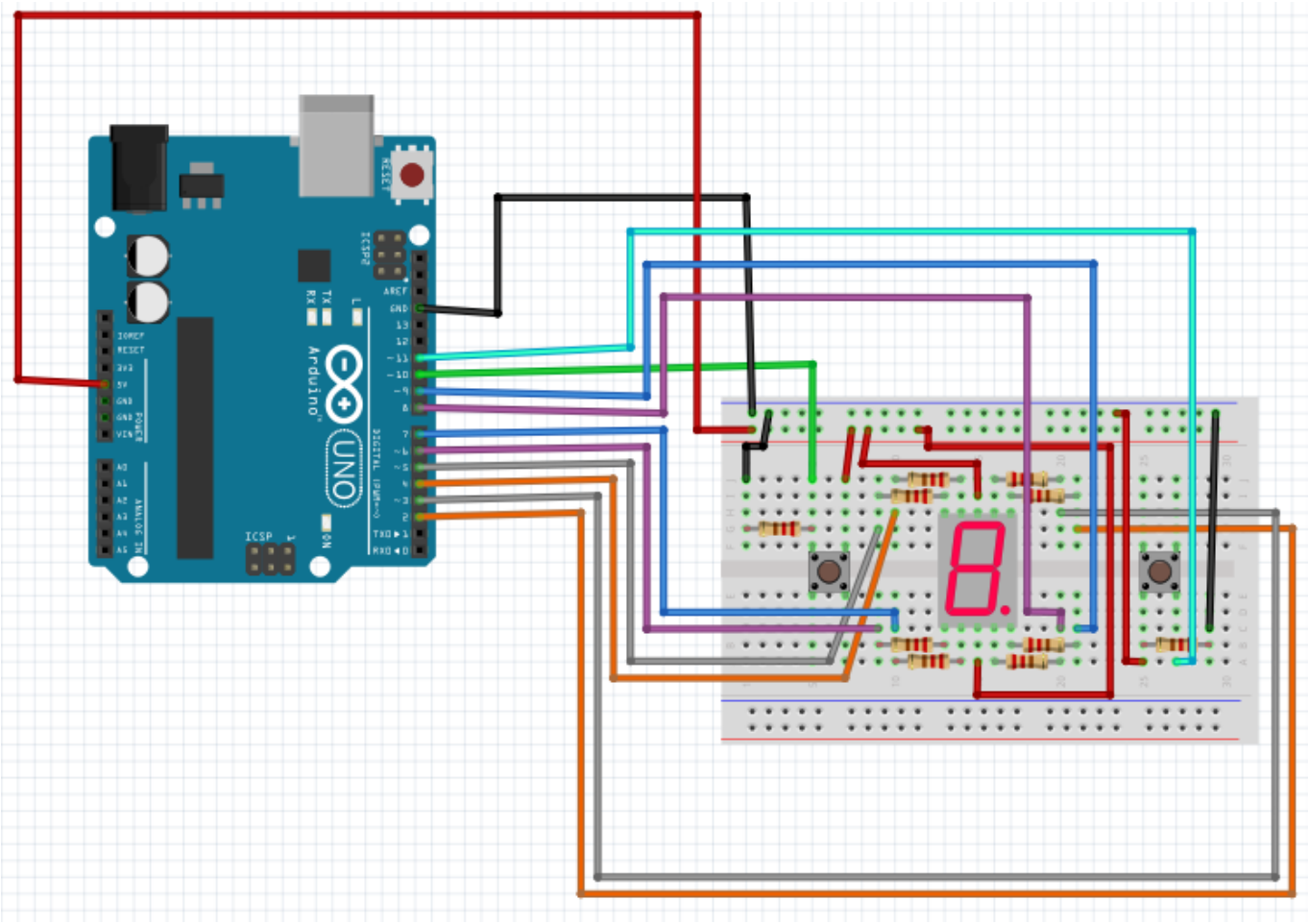
Résultat Final :



Projet n°21 : Dés

```
int currentNumber = 0;  //Crée la variable currentNumber
int button = 10;  //Port Bouton
int button2 = 11;
int dice;
int number[10][8] = //Tableau pour l'affichage digital
{
  {0,0,0,1,0,0,0,1},  //0 (upright,up,upleft,center,downleft,down,downright,dot)
  {0,1,1,1,1,1,0,1},  //1
  {0,0,1,0,0,0,1,1},  //2
  {0,0,1,0,1,0,0,1},  //3
  {0,1,0,0,1,1,0,1},  //4
  {1,0,0,0,1,0,0,1},  //5
  {1,0,0,0,0,0,0,1},  //6
  {0,0,1,1,1,1,0,1},  //7
  {0,0,0,0,0,0,0,1},  //8
  {0,0,0,0,1,0,0,1},  //9
};

void numberShow(int i) {  //Fonction qui active l'affichage digital
for(int pin = 2; pin <= 9 ; pin++){
  digitalWrite(pin, number[i][pin - 2]);
  }
}

void d6() {  //Fonction du dé 6 faces
  dice = random(0,6);
  delay(20);
  numberShow(dice);
  dice = random(0,6);
  dice++;
  numberShow(dice);
}

void d10() {  //Fonction du dé 10 faces
  dice = random(0,10);
  delay(20);
  numberShow(dice);
  dice = random(0,10);
  numberShow(dice);
}

void setup(){
  Serial.begin(9600); //Démarre le port série
  for(int pin = 2 ; pin <= 9 ; pin++){  //Sélectionne les ports 2 à 9 comme sorties
    pinMode(pin, OUTPUT);
    digitalWrite(pin, HIGH);
  }
}

void loop() {
  int state = digitalRead(button); //Lis l'état du bouton
  if (state == HIGH){ //Bouton activé
    d6();
  }
  int state2 = digitalRead(button2);
  if (state2 == HIGH){
    d10();
  }
}
```

Schéma



Résultat Final