**Matériel, schéma électronique, code**

I. Matériel :

Carte ArduinoTM Uno



2 Capteurs barrière infrarouge 3mm

Afficheur Alphanumérique LCD 2X16 Rétroéclairé Bleu



Alimentation 9V 660mA Jack 2,1mm

**Ou**

Câble alimentation pour pile 9V



Breadboard - 830 contacts

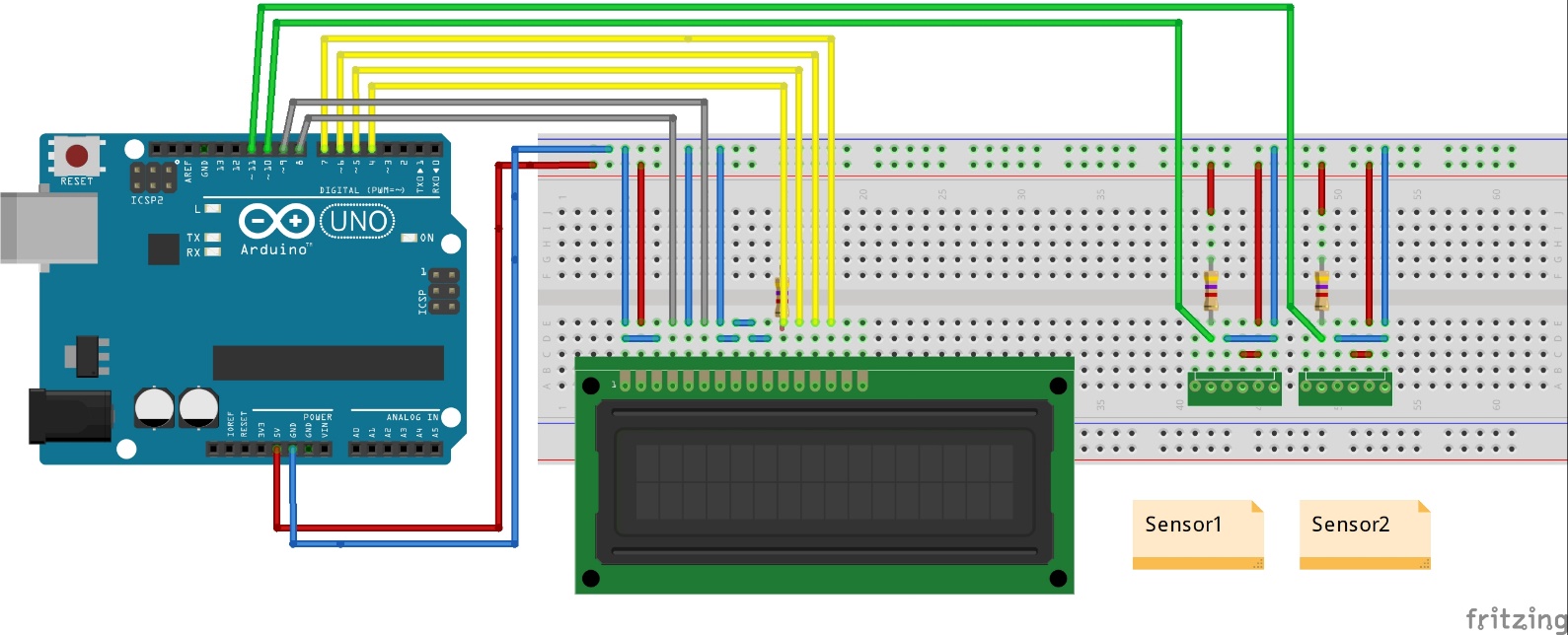


Kit de 70 wires pour prototypage



Câble USB type A-B

II. Schéma des connexions :



III. Code Microcontrôleur :

// Librairies utilisées

#include <LiquidCrystal.h>

// A modifier par l'élève

#define DISTANCE 0.150F // Distance entre les barrières optiques en mètre

#define MASSE 0.258F // Masse du mobile en kg

// Variables globales

const byte ledPin = 13; // LED

const byte irSensor1 = 2; // Sensor infrarouge démarrage du timer Pin #6

const byte irSensor2 = 3; // Sensor infrarouge arrêt du timer Pin #7

volatile bool sensor1 = LOW;

volatile bool sensor2 = LOW;

unsigned long start\_time = 0;

unsigned long delta\_time = 0;

float vitesse = 0.0f;

float energie = 0.0f;

// Définition LCD 2 Lignes 16 Caractères

const int rs = 8, en = 9, d4 = 4, d5 = 5, d6 = 6, d7 = 7;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {

Serial.begin(9600); // Configuration port serie pour le terminal

lcd.begin(16, 2); // Configuration LCD

lcd.clear(); // Efface LCD

lcd.setCursor(0, 0); // Position curseur ligne 0 colonne 0

lcd.print("HELLO "); // Message de démo

delay(800); // Attendre 3000ms

for(int x=0; x<16; x++)

{

lcd.scrollDisplayRight(); //message déroulant vers la droite première ligne

delay(250);

}

lcd.clear(); // Efface LCD

lcd.setCursor(0,1); // Position curseur ligne 0 colonne 0

lcd.print("HELLO "); // Message de démo

delay(150); // Attendre 3000ms

for(int x=0; x<16; x++)

{

lcd.scrollDisplayRight(); //message déroulant vers la droite deuxième ligne

delay(150);

}

lcd.clear(); // Efface LCD

lcd.setCursor(0, 0); // Position curseur ligne 0 colonne 0

lcd.print(" ENERGIE "); // Message de démo

lcd.setCursor(0,1); // Positionne curseur ligne 1 colonne 0

lcd.print(" CINETIQUE "); // Message de démo (suite)

delay(3000); // Attendre 3000ms

lcd.clear(); // Efface LCD

lcd.setCursor(0, 0); // Position curseur ligne 0 colonne 0

lcd.print(" DISTANCE (m) "); // Affiche titre Distance

lcd.setCursor(0,1); // Positionne curseur ligne 1 colonne 0

lcd.print(String(DISTANCE,3)); // Affiche Distance en m

delay(3000); // Attendre 2000ms

lcd.clear();

String mss;

mss = String(MASSE,3);

lcd.setCursor(0, 0); // Position curseur ligne 0 colonne 0

lcd.print(" MASSE (kg) "); // Affiche titre Masse

lcd.setCursor(0,1); // Positionne curseur ligne 1 colonne 0

lcd.print(mss); // Affiche Masse en kg

delay(3000); // Attendre 2000ms

lcd.clear(); // Efface LCD

lcd.setCursor(0, 0); // Position curseur ligne 0 colonne 0

lcd.print(" READY "); // Affiche READY L’application va démarrer

delay(2000); // Attendre 2000ms

pinMode(ledPin, OUTPUT); // Configure pin direction LED en Output

pinMode(irSensor1, INPUT); // Configure pin direction sensor1 en Input

pinMode(irSensor2, INPUT); // Configure pin direction sensor2 en Input

// Définition des interruptions sur pin #2 et #3 sur front descendant

attachInterrupt(digitalPinToInterrupt(irSensor1), start\_Timer, FALLING);

attachInterrupt(digitalPinToInterrupt(irSensor2), stop\_Timer, FALLING);

}

void loop() {

// Si on détecte l'objet en entrée ?

if (sensor1 == HIGH) {

// On capture le temps de départ en on l'affiche sur le terminal COM

start\_time = millis();

Serial.print("Start\_Time= ");

Serial.println(start\_time);

// RAZ flag sensor1

sensor1 = LOW;

// Allume LED

digitalWrite(ledPin, HIGH);

}

// Si on détecte l'objet en sortie ?

if (sensor2 == HIGH) {

// On capture la différence entre les 2 détections des sensors 1 et 2 en ms

delta\_time = millis() - start\_time;

// Désactive les interruptions

noInterrupts();

// RAZ flag sensor2

sensor2 = LOW;

// Eteindre LED

digitalWrite(ledPin, LOW);

// La vitesse peut être calculée en m/s

vitesse = (float)(DISTANCE / (float)delta\_time \* 1000);

// L’énergie peut être calculée en Joule

energie = (float)(MASSE \* pow(vitesse, 2) / 2);

String msg;

String lcdDisp;

// On convertit la vitesse (nb flottant) en chaine de caractères 2 décimales

msg = String(vitesse, 3);

// On concatène avec les unités dans une autre chaine pour le LCD

lcdDisp = String("V= "+msg+" m/s ");

// Affiche la vitesse m/s sur le LCD 1er Ligne

lcd.clear();

lcd.setCursor(0, 0);

lcd.print(lcdDisp);

// On convertit l’énergie (nb flottant) en chaine de caractères 2 décimales

msg = String(energie, 4);

// On concatène avec les unités dans une autre chaine pour le LCD

lcdDisp = String("Ec= "+msg+" J ");

// Affiche l’énergie en Joule sur le LCD 2e Ligne

lcd.setCursor(0, 1);

lcd.print(lcdDisp);

// On attend 2secondes

delay(2000);

// Active les interruptions et on recommence

interrupts();

}

}

// Fonction d’interruption sensor1

// A chaque détection du sensor1 cette fonction est exécutée

void start\_Timer() {

sensor1 = HIGH;

sensor2 = LOW;

}

// Fonction d'interruption sensor2

// A chaque détection du sensor2 cette fonction est exécutée

void stop\_Timer() {

sensor1 = LOW;

sensor2 = HIGH;

}