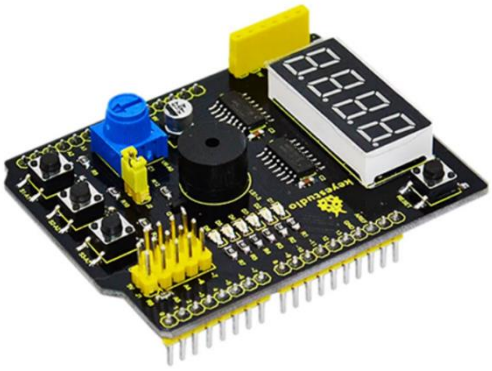
Exemple de programme pour le shield d'expérimentation 2 GT0037

Présentation :



Shield à but didactique prévu pour l'apprentissage de la programmation avec un microcontrôleur compatible Arduino Uno ou Mega2560 (non inclus).

Il vous permet d'obtenir un retour immédiat de ce que vous avez programmé (LEDs, sons, etc).

Ce shield comporte :

- 1 x afficheur 4 digits à 7 segments
- 6 x LEDs CMS raccordées en parallèle
- 1 x potentiomètre ajustable 10 K connecté à une entrée analogique
- 3 x boutons-poussoirs
- 1 x buzzer piezo
- 1 x connecteur avec interface UART (Rx, Tx, Vcc et GND)

Matériel nécessaire :

- 1 x carte compatible Uno ou Mega2560
- 1 x shield d'expérimentation 2 GT0037
- 1 x ordinateur avec l'IDE Arduino installé et configuré
- 1 x cordon USB pour la liaison entre la carte Arduino et l'ordinateur

---

Connexion avec une carte compatible Arduino Uno :

Téléchargement direct du programme (via Google Drive).

Ou, exemple de programme à copier/coller dans l'IDE Arduino :

```
// par défaut, l'afficheur donne la valeur analogique du potentiomètre
// en appuyant sur le bouton 1, vous afficherez "123" sur l'afficheur
// en appuyant sur le bouton 2, le buzzer sonnera
// en appuyant sur le bouton 3, les LEDs CMS feront un chenillard

// définition des broches des 74HC595
int latchPin = 4;  //ST_CP
int clockPin = 5;  //SH_CP
int dataPin = 2;   //DS

// définition des entrées des boutons-poussoirs
int key1 = A1;
int key2 = A2;
int key3 = A3;

// définition de la broche D3 pour le buzzer
int buzzer = 3;

// définition des broches digitales des LEDS CMS
int led1 = 13;
int led2 = 12;
int led3 = 11;
int led4 = 10;
int led5 = 9;
int led6 = 8;


int dat_wei[4] = { 0x01, 0x02, 0x04, 0x08 };  // afficheur à 7 segments
int dat_duan[10] = { 0xc0, 0xf9, 0xa4, 0xb0, 0x99, 0x92, 0x82, 0xf8, 0x80, 0x90 };  // les
// afficheurs donneront des chiffres de 0 à 9
char i = 0;
void setup() {
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);

  pinMode(key1, INPUT);
  pinMode(key2, INPUT);
  pinMode(key3, INPUT);

  pinMode(buzzer, OUTPUT);
```

```
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  pinMode(led6, OUTPUT);
  for (char i = 8; i < 14; i++)
    digitalWrite(i, HIGH);
}

void loop() {
  if (digitalRead(key1) == LOW)
    SMG();  // test des afficheurs
  if (digitalRead(key2) == LOW)
    buzzer_();  // test du buzzer
  if (digitalRead(key3) == LOW)
    led_display();  // test des LEDs CMS
  if (digitalRead(key1) == HIGH & digitalRead(key2) == HIGH & digitalRead(key3) == HIGH)
    analog();  // test de la sortie analogique
}

void SMG(void) {
  digitalWrite(latchPin, LOW);  // remise à zéro de l'afficheur
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  digitalWrite(latchPin, HIGH);

  while (1) {
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, MSBFIRST, dat_duan[i]);
    shiftOut(dataPin, clockPin, MSBFIRST, dat_wei[i]);
    digitalWrite(latchPin, HIGH);
    i++;
    if (i == 4) { i = 0; }
    if (digitalRead(key1) == HIGH) {
      digitalWrite(latchPin, LOW);  // remise à zéro de l'afficheur
      shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
      shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
      digitalWrite(latchPin, HIGH);
      break;
    }
  }
}

void buzzer_(void) {
  char i;

  digitalWrite(latchPin, LOW);  // remise à zéro de l'afficheur
```

```
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  digitalWrite(latchPin, HIGH);

  while (1) {
    for (i = 0; i < 80; i++)  // fréquence de la sortie sonore du buzzer
    {
      digitalWrite(buzzer, LOW);   // buzzer actif
      delay(1);                    // delai de 1 ms
      digitalWrite(buzzer, HIGH);  // buzzer inactif
      delay(1);                    // delai de 1 ms
    }
    for (i = 0; i < 100; i++)  // fréquence de la sortie sonore du buzzer
    {
      digitalWrite(buzzer, LOW);   // buzzer actif
      digitalWrite(buzzer, HIGH);  // buzzer inactif
      delay(2);                    // delai de 2 ms
    }
    if (digitalRead(key2) == HIGH) {
      digitalWrite(latchPin, LOW);  // remise à zéro de l'afficheur
      shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
      shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
      digitalWrite(latchPin, HIGH);
      break;
    }
  }
}

void led_display() {
  digitalWrite(latchPin, LOW);  // remise à zéro de l'afficheur
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x00);
  digitalWrite(latchPin, HIGH);

  while (1) {
    digitalWrite(led1, LOW);
    delay(100);
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
    delay(100);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, LOW);
    delay(100);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, LOW);
    delay(100);
    digitalWrite(led4, HIGH);
    digitalWrite(led5, LOW);
    delay(100);
```

```
    digitalWrite(led5, HIGH);
    digitalWrite(led6, LOW);
    delay(100);
    digitalWrite(led6, HIGH);
    if (digitalRead(key3) == HIGH) {
      break;
    }
  }
}


void analog() {
  int val, qian, bai, shi, ge;
  val = analogRead(A0);
  qian = val / 1000;
  bai = val % 1000;
  bai = bai / 100;
  shi = val % 100;
  shi = shi / 10;
  ge = val % 10;
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, dat_duan[qian]);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x01);
  digitalWrite(latchPin, HIGH);
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, dat_duan[bai]);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x02);
  digitalWrite(latchPin, HIGH);
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, dat_duan[shi]);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x04);
  digitalWrite(latchPin, HIGH);
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, MSBFIRST, dat_duan[ge]);
  shiftOut(dataPin, clockPin, MSBFIRST, 0x08);
  digitalWrite(latchPin, HIGH);
}
```

GO TRONIC

ROBOTIQUE ET COMPOSANTS ÉLECTRONIQUES

Si vous rencontrez des problèmes, merci de nous contacter par courriel à :

sav@gotronic.fr