

## BARRIR - Code Arduino

/\*  
\*\*\*\*\*  
\*/

BARRIR v2. Barrière photo infra rouge

auth: Yves Le Chevalier - Maj le 06/01/2016

Déclencheur d'appareil photo ou de flash en mode open flash par une barrière infrarouge.

Usage destiné à la prise de vue d'insectes ou d'oiseaux et à la photo de gouttes.

Système à base d'Arduino UNO pilotant un appareil reflex (développé sur Nikon D7000 et flash SB700).

Barrière constituée d'une diode IR émettrice SFH4546 et en réception d'un phototransistor LTR4206

Fonctions gérées :

Mise en pause (fonctionnement normal mais sans déclenchement).

Ecran LCD pouvant être éteint ou allumé.

Réglages : Sensibilité de détection (seuil minimal de la variation de réception IR),

Délai avant déclenchement (intervalle entre coupure du rayon et déclenchement),

Temporisation entre deux déclenchements consécutifs,

Nombre de coupures avant déclenchement (pour photo de gouttes)

Allumage du laser de visée pour alignement des diodes et mise au point objectif.

Affichage : Mesure du rayonnement reçu (affichage permanent).

Variation de mesure du rayonnement ayant provoqué le déclenchement.

Nombre de coupures avant déclenchement (si plusieurs coupures demandées).

Nombre de déclenchement effectués (Remis à zéro à chaque modification des réglages)

Mode de pilotage choisi (Boitier ou flash).

Led jaune témoin de l'ouverture de l'obturateur du boitier

Led rouge témoin de la mise en pause (fonctionnement sans déclenchement).

Led verte témoin de la mise en fonctionnement avec déclenchements.

Pins utilisées : A0,A1,A4,A5,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D16,D17

Consommation : de 100 mA en mode Boitier + LCD off à 150 mA en mode flash + LCD on

\*/

```
#include <Wire.h>
```

```
#include <LCD.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7); // déclaration afficheur LCD série
```

```
#define LTR4206 0 // data du phototransistor LTR4206 en entrée sur pin A0
```

```
#define potreg 1 // potentiomètre réglages en entrée sur pin A1
```

```
// LCD : SDA sur pin A4 SCL sur pin A5
```

```
#define bpause 2 // bouton de mise en pause ou activation (interruption 0 sur pin D2)
```

```
#define swsensi 3 // switch de réglage sensibilité de détection
```

```
#define swtempo 4 // switch réglage temporisation entre déclenchements
```

```
#define swdelai 5 // switch réglage délai avant déclenchement
```

```
#define swnbeve 6 // switch réglage nb événements avant déclenchement
```

```
#define swactreg 7 // switch activation du potentiometre pour réglages
```

```
#define swpdv 8 // switch du mode prise de vues (par opposition à réglages)
```

```
#define swlcd 9 // switch réglage fonctionnement avec lcd ou sans LCD
```

```
#define tempause 10 // 2 leds témoins (en opposition) : (vert = action) ou (rouge = pause)
```

```
#define temobtu 11 // led jaune témoin état obturateur (allumé = ouvert)
```

```
#define swbofla 12 // switch pilotage boitier ou flash
```

```
#define pinflash 13 // commande déclenchement direct du flash (APN en mode bulb)
```

```
#define pinfocus 17 // commande déclenchement focus appareil photo (pin A3 = pin D17)
```

```
#define pinobtur 16 // commande déclenchement shoot appareil photo (pin A2 = pin D16)
```

```
int valpot = 0; // valeur lue sur potentiomètre pour réglages
```

```
int mesurir = 0; // valeur IR lue
```

```
int varmin = 15; // variation de mesure minimale indiquant une coupure du rayon
```

```
int varmes = 0; // variation d'illumination mesurée
```

```
long delai = 0; // delai avant déclenchement en millisec
```

```
int mesurpre = 0; // valeur de la mesure précédente
```

```
int nbcoupur = 0; // indic du nombre de coupures lues
```

```
int nbmincou = 1; // nb mini de coupures avant déclenchement
```

```
int nbdeclen = 0; // indic du nombre de déclenchements
```

```
long tempor = 0; // temporisation demandée entre 2 déclenchements (millisec)
```

```
long dttempo = 200; // temporisation minimale dû aux contraintes APN (millisec)
```



```

    if (digitalRead(swnbeve)) nbmincou = map(valpot, 0, 1023, 1, 20); // réglage nb coupures avant déclenchement
  }
  nbcoupur = 0;
  nbdeclen = 0;
  mesurpre = 0;
  Affichreg();
  indacti = false; // remise en pause si réglages
  digitalWrite(tempause, LOW); // Led témoin rouge allumée (en pause), verte éteinte
}
//-----
void Activpause() { // routine d'interruption quand appui sur bouton
  timbou = millis();
  if (timbou - timpre > 300) { // temporisation 1/2 sec anti-rebond du bouton
    indacti = !indacti; // inversion de l'indicateur d'activation
    if (!indacti) {
      digitalWrite(tempause, LOW); // Led témoin rouge (pause) allumée, verte éteinte
      Quitmodof ();
    } else digitalWrite(tempause, HIGH); // Led témoin verte (activité) allumée, rouge éteinte
    timpre = timbou;
  }
}
//-----
void Pilotapn() { // mode pilotage du boitier
  if (!indboit) { // première entrée dans la fonction
    indboit = true;
    Quitmodof ();
    Affichmode();
  }
  if (varmes > varmin) { // détection d'un coupure du rayon
    nbcoupur = nbcoupur + 1;
    if (nbcoupur == nbmincou) Declenapn(); // déclenchement si nb coupures mini atteint
    else mesurpre = mesurir; // stockage mesure précédente
    Affichcoup();
  }
  else mesurpre = mesurir; // stockage mesure précédente
  Affmesur();
  indreg = false; // on efface l'indicateur de mode Réglage
}
//-----
void Quitmodof () {
  if (indflash) { // si on vient de quitter le mode flash
    indflash = false;
    digitalWrite(pinobtur, HIGH); // fermeture obturateur resté ouvert en mode flash
    digitalWrite(pinfocus, HIGH); // extinction focus
    analogWrite(temobtu, 0); // led témoin obturateur fermé
  }
}
//-----
void Pilotflash() { // mode pilotage du flash en open flash
  if (!indflash && indacti) { // première entrée dans la fonction et activation
    indflash = true;
    indboit = false;
    Affichmode();
    digitalWrite(pinfocus, LOW); // allumage focus
    digitalWrite(pinobtur, LOW); // ouverture obturateur (APN en mode Bulb)
    analogWrite(temobtu, 10); // led témoin obturateur ouvert
  }
  if (varmes > varmin) { // détection d'un coupure du rayon
    nbcoupur = nbcoupur + 1;
    if (nbcoupur == nbmincou) Declenflash(); // déclenchement si nb coupures mini atteint
    else mesurpre = mesurir; // stockage mesure précédente
    Affichcoup();
  }
}

```

```

else mesurpre = mesurir; // stockage mesure précédente
Affmesur();
indreg = false; // on efface l'indicateur de mode Réglage
}
//-----
void Affichmode() { // affichage du mode choisi
  lcd.setCursor (13, 3);
  if (digitalRead(swbofla)) lcd.print("BOITIER");
  else lcd.print(" FLASH ");
}
//-----
void Affmesur() { // affichage permanent de la mesure du rayon
  if (digitalRead(swlcd)) lcd.setBacklight(HIGH); // couper le rétroéclairage LCD
  else lcd.setBacklight(LOW); // activer le rétroéclairage LCD
  lcd.setCursor (0, 0);
  lcd.print("M.");
  lcd.setCursor (2, 0);
  lcd.print(mesurir);
  lcd.setCursor (6, 0);
  if (!indacti) lcd.print(" PAUSE ");
  else lcd.print(" MARCHÉ ");
  Affichvar();
}
//-----
void Declenapn() {
  if (indacti) { // déclenchement seulement si en activité
    delay(delai); // attente du temps de delai avant déclenchement
    analogWrite(temobtu, 10); // led témoin obturateur ouvert
    digitalWrite(pinfocus, LOW); // allumage focus
    digitalWrite(pinobtur, LOW); // ouverture obturateur
    delay(200);
    digitalWrite(pinobtur, HIGH); // fermeture obturateur
    digitalWrite(pinfocus, HIGH); // extinction focus
    analogWrite(temobtu, 0); // led témoin obturateur fermé
    nbdeclen++; // +1 dans nb déclenchements
  }
  Comptages();
}
//-----
void Declenflash() {
  if (indacti) { // déclenchement seulement si en activité
    delay(delai); // attente du delai avant déclenchement
    digitalWrite(pinflash, LOW); // allumage flash
    delay(30);
    digitalWrite(pinflash, HIGH); // extinction flash
    digitalWrite(pinobtur, HIGH); // fermeture obturateur
    analogWrite(temobtu, 0); // led témoin obturateur fermé
    digitalWrite(pinfocus, HIGH); // extinction focus
    delay(170); // délai imposé par le boitier
    digitalWrite(pinfocus, LOW); // allumage focus
    digitalWrite(pinobtur, LOW); // ré-ouverture obturateur
    analogWrite(temobtu, 10); // led témoin obturateur ouvert
    nbdeclen++; // +1 dans nb déclenchements
  }
  Comptages();
}
//-----
void Comptages() {
  Affichlcd();
  delay(tempor + dtempo); // temporisation entre 2 déclenchements
  mesurpre = 0;
  nbcoupur = 0;
}

```

```

//-----
void Affichlcd() {
  lcd.clear();
  Affmesur();
  lcd.setCursor (15, 0);
  lcd.print("V.");
  lcd.print(varmes);
  lcd.setCursor (10, 2);
  lcd.print("Nb.Dec.");
  lcd.print(nbdeclen);
  Affichvar();
}
//-----
void Affichreg() {
  if (digitalRead(swsensi)) {
    lcd.setCursor (0, 1);
    lcd.print("V.min. ");
    lcd.setCursor (6, 1);
    lcd.print(varmin);
    lcd.setCursor (0, 2);
    lcd.print(" ");
    lcd.setCursor (0, 3);
    lcd.print(" ");
    lcd.setCursor (10, 1);
    lcd.print(" ");
    lcd.setCursor (10, 2);
    lcd.print(" ");
  }
  else if (digitalRead(swdelai)) {
    lcd.setCursor (0, 2);
    lcd.print("Delai. ");
    lcd.setCursor (6, 2);
    lcd.print(delai);
    lcd.setCursor (0, 1);
    lcd.print(" ");
    lcd.setCursor (0, 3);
    lcd.print(" ");
    lcd.setCursor (10, 1);
    lcd.print(" ");
    lcd.setCursor (10, 2);
    lcd.print(" ");
  }
  else if (digitalRead(swtempo)) {
    lcd.setCursor (0, 3);
    lcd.print("Tempo. ");
    lcd.setCursor (6, 3);
    lcd.print(tempor + dtempo);
    lcd.setCursor (0, 1);
    lcd.print(" ");
    lcd.setCursor (0, 2);
    lcd.print(" ");
    lcd.setCursor (10, 1);
    lcd.print(" ");
    lcd.setCursor (10, 2);
    lcd.print(" ");
  }
  else if (digitalRead(swnbeve)) {
    lcd.setCursor (10, 1);
    lcd.print("Coup. / ");
    lcd.setCursor (18, 1);
    lcd.print(nbmincou);
    lcd.setCursor (0, 1);
    lcd.print(" ");
  }
}

```

```

        lcd.setCursor (0, 2);
        lcd.print("    ");
        lcd.setCursor (0, 3);
        lcd.print("    ");
        lcd.setCursor (10, 2);
        lcd.print("    ");
    }
    else {
        lcd.setCursor (10, 2);
        lcd.print("Laser On");
        lcd.setCursor (0, 1);
        lcd.print("    ");
        lcd.setCursor (0, 2);
        lcd.print("    ");
        lcd.setCursor (0, 3);
        lcd.print("    ");
        lcd.setCursor (10, 1);
        lcd.print("    ");
    }
    Affichmode();
}
//-----
void Affichvar() {
    lcd.setCursor (0, 1);
    lcd.print("V.min.  ");
    lcd.setCursor (6, 1);
    lcd.print(varmin);
    lcd.setCursor (0, 2);
    lcd.print("Retar.  ");
    lcd.setCursor (6, 2);
    lcd.print(delai);
    Affichmode();
    lcd.setCursor (0, 3);
    lcd.print("Tempo.  ");
    lcd.setCursor (6, 3);
    lcd.print(tempor + dtempo);
}
//-----
void Affichcoup() {
    if (nbmincou > 1) {
        lcd.setCursor (10, 1);
        lcd.print("Coup. / ");
        lcd.setCursor (15, 1);
        lcd.print(nbcoupur);
        lcd.setCursor (18, 1);
        lcd.print(nbmincou);
    }
}
/*****/

```

=====