

Déclenchement d'une caméra Basler

1. Introduction

Ce document explique comment utiliser une [caméra Basler](#) en déclenchement matériel au moyen d'une carte Arduino. L'acquisition de chaque image est déclenchée par l'arduino, ce qui permet d'effectuer des mesures physiques synchrones avec la vidéo, avec l'arduino lui-même ou avec une carte d'acquisition déclenchée par l'arduino. Le déclenchement matériel permet aussi de synchroniser plusieurs caméras Basler.

2. Câblage de la caméra

Les caméras Basler Ace (USB 3.0) possèdent un connecteur à 6 broches permettant d'effectuer un déclenchement matériel de l'acquisition des images.

The 6-pin connector is used to access the physical input and output lines on the camera. The pin numbering for the 6-pin connector is as shown in Figure 39.

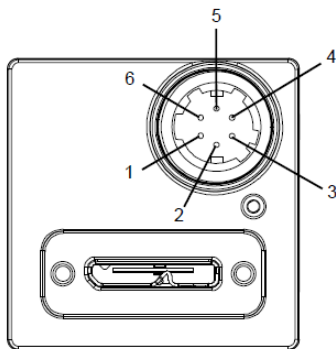


Fig. 39: Pin Numbering for the 6-pin Connector

The pin assignments and designations for the 6-pin connector are shown in Table 20.

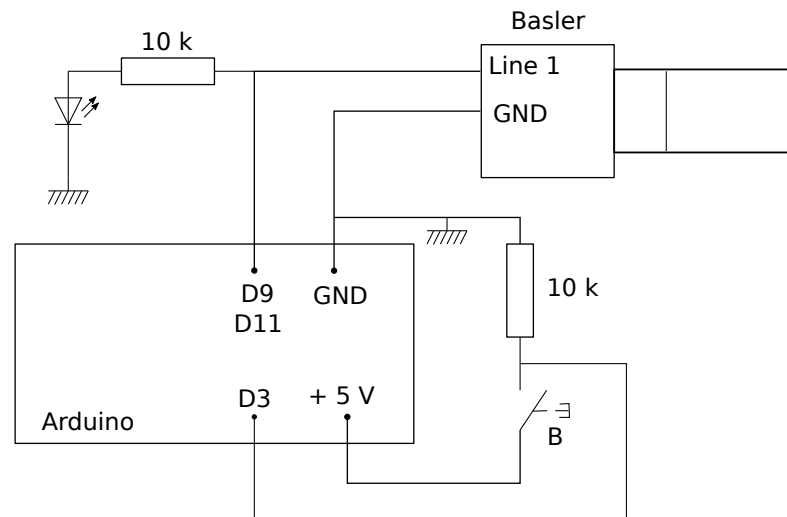
Pin	Designation	Function
1	Line 3	Direct-coupled General Purpose I/O (GPIO)
2	Line 1	Opto-isolated I/O IN
3	Line 4	Direct-coupled General Purpose I/O (GPIO)
4	Line 2	Opto-isolated I/O OUT
5	-	Ground for opto-isolated I/O
6	-	Ground for direct-coupled GPIO

Table 20: Pin Assignments for the 6-pin Connector and Related Designations

Nous utilisons les broches 2 et 5 : Line 1 (entrée opto-isolée) et masse associée. On peut aussi utiliser les broches 1 et 6 (Line 3 et masse).

3. Programme Arduino

Voici le branchement de la caméra sur l'arduino (UNO, NANO ou MEGA).



Le programme arduino génère des impulsions sur la sortie D9 (Arduino UNO ou NANO) ou sur la sortie D11 (Arduino MEGA) afin de déclencher les acquisitions des images par l'entrée Line 1 de la caméra. Le bouton poussoir (B) permet de déclencher manuellement le début de l'acquisition par un front montant sur l'entrée D3. Une connexion de cette entrée à la voie Synchro EXT de la carte d'acquisition Sysam SP5 permet éventuellement de déclencher une conversion A/N au même instant.

Il est possible de compter les impulsions (macro `COMPTEUR true`) afin de stopper la génération lorsqu'un nombre d'impulsions (définie par `NFRAME`) est atteint. Si le nombre d'images à acquérir n'est pas fixé à l'avance, on écrit `COMPTEUR false`; dans ce cas, la génération est stoppée par un second appuie sur le bouton. Il faudra bien sûr penser à faire ce second appuie avant de lancer une nouvelle acquisition. La LED permet de savoir si une émission d'impulsions est en cours.

Par exemple, les valeurs suivantes permettent de générer 1000 impulsions avec une période de 10 ms, soit une vidéo à 100 images par secondes durant 10 secondes. Il faut bien sûr que la caméra utilisée puisse fournir le taux d'acquisition demandé au format d'image choisi. Nous utilisons le modèle acA1300-200uc, qui peut fournir 200 images par secondes en pleine résolution (1280 par 1024) et plus à une résolution moindre.

[declenchement-camera.ino](#)

```
#define LINE1 9 // arduino NANO
#define START_BUTTON 3
#define PERIOD 10000 // période en microsecondes
#define NFRAME 1000 // nombre d'images
#define COMPTEUR true
```

```
uint16_t diviseur[6] = {0,1,8,64,256,1024};
uint32_t icr;
```

```
volatile uint32_t compteur;
bool actif;
```

Les impulsions sont générées par le Timer 1, configuré par la fonction ci-dessous :

```
void init_pwm_timer1(uint32_t period) {
    char clockBits;
    cli();
    TCCR1A = 0;
    TCCR1A |= (1 << COM1A1); //Clear OC1A on compare match when upcounting, set OC1A on
    TCCR1A |= (1 << COM1B1);
#ifdef defined(__AVR_ATmega2560__) || defined(__AVR_ATmega32U4__)
    TCCR1A |= (1 << COM1C1);
#endif
    TCCR1B = 1 << WGM13; // phase and frequency correct pwm mode, top = ICR1
    int d = 1;
    icr = (F_CPU/1000000*period/2);
    while ((icr>0xFFFF)&&(d<5)) { // choix du diviseur d'horloge
        d++;
        icr = ((F_CPU/1000000)*period/2/diviseur[d]);
        //Serial.println(d);
        //Serial.println(icr);
    }
    clockBits = d;
    ICR1 = icr; // valeur maximale du compteur
    TIMSK1 = 1 << TOIE1; // overflow interrupt enable
    TCNT1 = 0; // mise à zéro du compteur
    OCR1A = icr*0.5;

    TCCR1B |= clockBits; // déclenchement du compteur

    sei(); // activation des interruptions
}
```

Le gestionnaire d'interruption (déclenché par le Timer), compte les impulsions et stoppe la génération lorsque le nombre NFRAME est atteint.

```
ISR(TIMER1_OVF_vect) {
    if (COMPTEUR) {
        compteur +=1;
        if (compteur == NFRAME) {
            OCR1A = 0;
        }
    }
}
```

La fonction setup configure les entrées et sorties puis attribue une fonction pour gérer l'interruption déclenchée par le front montant sur l'entrée START_BUTTON (bouton poussoir).

```
void setup() {
    Serial.begin(115200);
```

```
    compteur = 0;
    pinMode(LINE1, OUTPUT);
    pinMode(START_BUTTON, INPUT);
    digitalWrite(LINE1, LOW);
    attachInterrupt(digitalPinToInterrupt(START_BUTTON), start, RISING);
    actif = false;
}

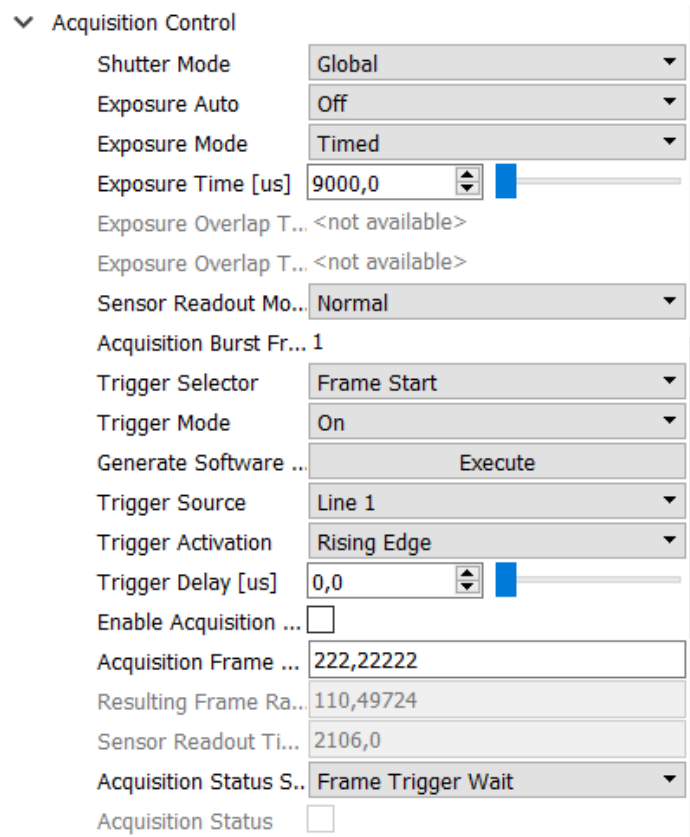
void start() {
    if (actif) {
        OCR1A = 0;
        delay(2000);
        actif = false;
    }
    else {
        compteur = 0;
        init_pwm_timer1(PERIOD);
        delay(2000);
        actif=true;
    }
}

void loop() {
}
```

4. Configuration de la caméra

La configuration se fait avec le logiciel Pylon Viewer.

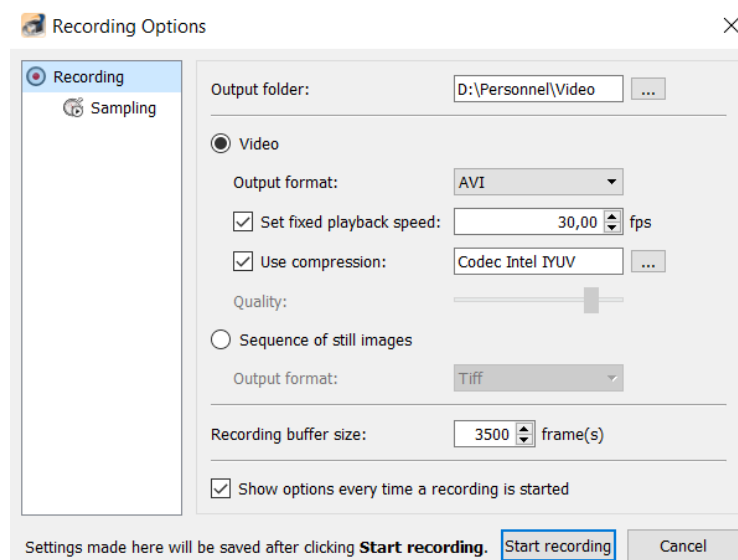
La configuration du déclenchement se fait dans l'onglet Acquisition Control :



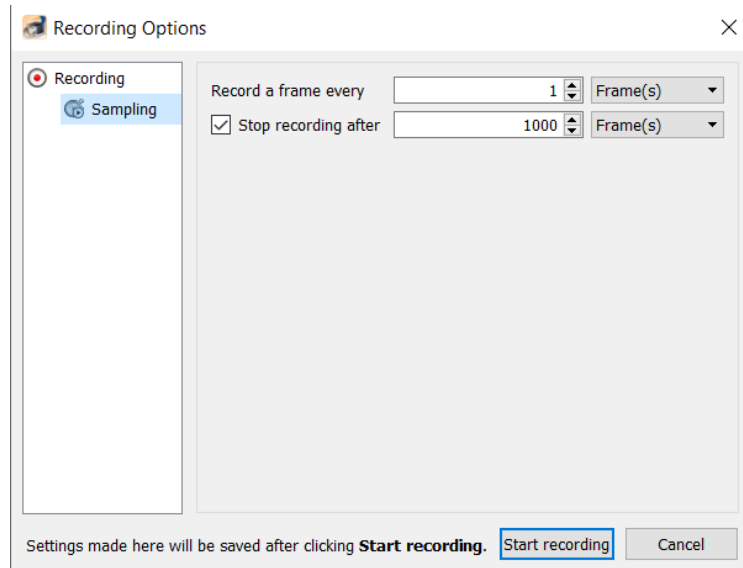
Le temps d'exposition doit être inférieur à la période d'échantillonnage. Les options à sélectionner sont :

- ▷ Trigger Selector : Frame Start
- ▷ Trigger Mode : ON
- ▷ Trigger Source : Line 1

Lorsque ces paramètres sont sélectionnés, l'affichage de l'image en temps réel ne se fait plus car la caméra attend le signal de déclenchement. Il faut appuyer sur le bouton poussoir pour déclencher l'acquisition des images (NFRAMES images). Pour enregistrer un fichier vidéo, cliquer sur le bouton Record, la fenêtre suivante s'ouvre :



Choisir le dossier de destination, la compression Intel IYUV (compression légère) et la taille du tampon, qui doit être supérieure au nombre total d'images. Cliquer ensuite sur Sampling et la fenêtre suivante apparaît :



Entrer le nombre d'images de la vidéo, en principe égal au nombre d'impulsions (NFRAME) envoyées par le programme arduino (ou en tout cas inférieur à ce nombre) puis cliquer sur Start Recording. L'enregistrement démarre lorsqu'on appuie sur le bouton poussoir.