

Recherche pour :

CATÉGORIES

[Arduino](http://notepad.xavierdetourbet.com/category/arduino/)

(<http://notepad.xavierdetourbet.com/category/arduino/>)

[Divers](http://notepad.xavierdetourbet.com/category/non-classe/)

(<http://notepad.xavierdetourbet.com/category/non-classe/>)

[Imprimante 3d](http://notepad.xavierdetourbet.com/category/imprimante-3d/)

(<http://notepad.xavierdetourbet.com/category/imprimante-3d/>)

[Python](http://notepad.xavierdetourbet.com/category/python/)

(<http://notepad.xavierdetourbet.com/category/python/>)

[Raspberry](http://notepad.xavierdetourbet.com/category/raspberry/)

(<http://notepad.xavierdetourbet.com/category/raspberry/>)

[RSS](http://notepad.xavierdetourbet.com/category/rss/)

(<http://notepad.xavierdetourbet.com/category/rss/>)

Connecter une camera Pixart

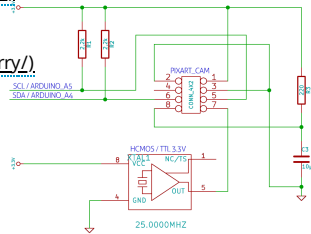
sur Arduino en protocole I2C

MAR 11, 2017

EN-PROTOCOLE-I2C/#RESPOND)

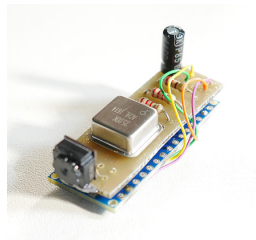
(<http://notepad.xavierdetourbet.com/connect-une-camera-pixart-sur-arduino-en-protocole-i2c/>)

Schéma de câblage



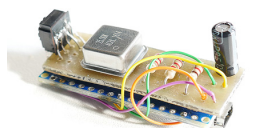
([/content/iwand/img/pixart_i2c_arduino.png](http://content.iwand/img/pixart_i2c_arduino.png))

Schéma de câblage



([/content/iwand/img/wii_camera_pcb_2.jpg](http://content.iwand/img/wii_camera_pcb_2.jpg))

Montage du PCB



([/content/iwand/img/wii_camera_pcb_1.jpg](http://content.iwand/img/wii_camera_pcb_1.jpg))

Arduino Nano sous le PCB

Note : Le fichier de routage du PCB n'est plus téléchargeable. Après montage du circuit, le soudage de la caméra a été une véritable galère (voir photo ci-dessus). Il est conseillé de fixer la caméra directement sur les pistes du PCB sans soudure par simple contact à l'instar des sockets de microprocesseur.

Attention : La caméra Pixart ne supporte pas le contact prolongé du fer à souder

Format des données ([Source](http://wiibrew.org/wiki/Wiimote#IR_Camera) (http://wiibrew.org/wiki/Wiimote#IR_Camera))

Le format définit la présence ou pas de certaines données, il en découle 3 modes de fonctionnement (*Basic*, *Extended* et *Full*) avec une organisation des informations différentes dans chacun d'eux.

La configuration du mode s'effectue en écrivant la valeur indiquée entre parenthèse dans le registre **0x33**.

Le code arduino fourni en bas de page traite les données en mode *Extended*, il est possible de rendre le script plus rapide en ignorant les données de sensibilité avec le mode *Basic*.

BASIC (0×01)								
Byte	7	6	5	4	3	2	1	0
0	X1<7:0>							
1	Y1<7:0>							
2	Y1<9:8>	X1<9:8>	Y2<9:8>	X2<9:8>				
3	X2<7:0>							
4	Y2<7:0>							

EXTENDED (0×03)								
Byte	7	6	5	4	3	2	1	0
0	X<7:0>							
1	Y<7:0>							
2	Y<9:8>	X<9:8>	S<3:0>					

FULL (0×05)								
Byte	7	6	5	4	3	2	1	0
0	X<7:0>							
1	Y<7:0>							
2	Y<9:8>	X<9:8>	S<3:0>					
3	0	X min<6:0>						
4	0	Y min<6:0>						
5	0	X max<6:0>						
6	0	Y max<6:0>						
7	0							
8	Intensity<7:0>							

Sensibilité

Cette partie est la plus délicate à traiter. Le fabricant [Pixart](http://www.pixart.com.tw/) (<http://www.pixart.com.tw/>) n'a pas rendu public la datasheet du capteur.
Les valeurs fournies par [wiibrew](http://wiibrew.org/wiki/Wiimote#IR_Camera) (http://wiibrew.org/wiki/Wiimote#IR_Camera) sont donc à traiter avec une grande réserve.

Toutefois, le développeur [Johnny Chung Lee](http://johnnylee.net/projects/wii/) (<http://johnnylee.net/projects/wii/>) fait ressortir à travers son travail le code source suivant dans le fichier `windowsxp/WiimoteTest/WiimoteLib/Wiimote.cs`

```
WriteData(REGISTER_IR_SENSITIVITY_1, 9, new byte[] { 0x00, 0x00, 0x00, 0x00, 0:
WriteData(REGISTER_IR_SENSITIVITY_2, 2, new byte[] { 0x40, 0x00 }));
```

Après quelques test il est confirmé que le code d'Inio propose le meilleur compromis.
En revanche le code fournis par Kestrel semble inutilisable, la sensibilité est bien augmentée au maximum mais le capteur est trop sensible aux lumières parasites... à éviter !

Block 1		Block 2	Notes
00 00 00 00 00 00 90 00	C0	40 00	Suggested by Marcan (/wiki/User:Marcan)
00 00 00 00 00 00 FF 00	0C	00 00	Suggested by Kestrel (max sensitivity)
00 00 00 00 00 00 90 00	41	40 00	Suggested by inio (high sensitivity)
02 00 00 71 01 00 64 00	fe	fd 05	Wii level 1
02 00 00 71 01 00 96 00	b4	b3 04	Wii level 2
02 00 00 71 01 00 aa 00	64	63 03	Wii level 3 (Suggested by Cliff)
02 00 00 71 01 00 c8 00	36	35 03	Wii level 4
07 00 00 71 01 00 72 00	20	1f 03	Wii level 5

Un autre facteur important pour optimiser la détection des blobs par le capteur Pixart est la led infrarouge.
Plusieurs facteurs sont à prendre en considération comme l'angle d'ouverture et la puissance.

Un facteur est toutefois plus critique que les autres, il s'agit de la longueur d'onde qui doit s'accorder au mieux avec le spectre de sensibilité du capteur, sa valeur est donnée à 940nm.

Les valeurs théoriques des datasheets n'étant pas toujours le reflet de la réalité, il est préférable d'effectuer quelques tests pour évaluer le rendements des leds.

Internet est encore d'une aide précieuse car l'artiste [Richard Scott](https://steimintern.wordpress.com/2011/11/23/ir-testing/) (<https://steimintern.wordpress.com/2011/11/23/ir-testing/>) a publié ses résultats de recherche en s'appuyant sur les tests donnés [ici](http://wiicanetouchgraphic.blogspot.fr/2009/03/wii-remote-ir-sensitivity.html) (<http://wiicanetouchgraphic.blogspot.fr/2009/03/wii-remote-ir-sensitivity.html>) pour se focaliser uniquement sur les meilleurs leds.

Un [autre article](https://steimintern.wordpress.com/2011/11/29/lenses/) (<https://steimintern.wordpress.com/2011/11/29/lenses/>) très intéressant traite l'utilisation des objectifs pour augmenter le champ de vision du capteur.

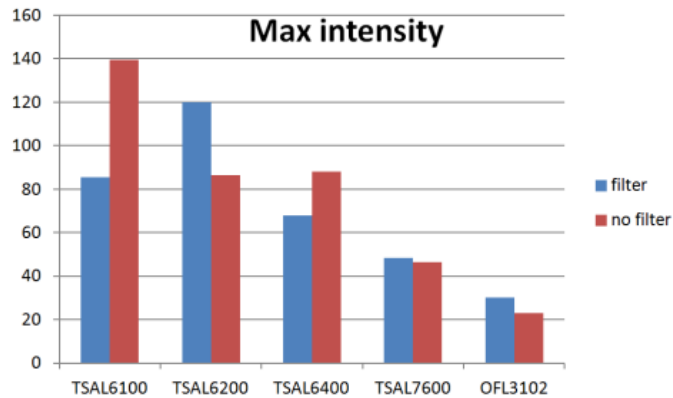


Fig. Comparatif de leds infrarouge 940nm

Note : Bien que les led de 850nm ne figure pas dans le tableau ci-dessus ces dernières restent envisageables. Elles seront toutefois perçues par le capteur avec deux fois moins d'intensité.

CODE ARDUINO / [Télécharger \(/content/iwand/code/camera_wii.pde\)](/content/iwand/code/camera_wii.pde)

```

/*****
* Arduino 1.6.7
* Initialization IR sensor code based on kako's work http://kako.com/neta/200
* and http://wiibrew.org/wiki/Wiimote\_\(http://wiibrew.org/wiki/Wiimote\)#IR\_Cam
* Modified by Xavier Detourbet http://notepad.xavierdetourbet.com/?p=1021 (ht
*
* ONLY FOR ONE BLOB & MAX BIT RATE I2C 800Khz
*****/

#include <Wire.h>

#define SENSOR_ADDRESS 0x58
byte data_buf[4];
int x, y, s;
byte setting1[9] = {0x00,0x00,0x00,0x00,0x00,0x00,0x90,0x00,0x41};
byte setting2[2] = {0x40,0x00};

void Write_2bytes(byte d1, byte d2)
{
    Wire.beginTransmission(SENSOR_ADDRESS);
    Wire.write(d1); Wire.write(d2);
    Wire.endTransmission();
}

void Write_Nbytes(byte d1, byte * ds, byte n) {
    Wire.beginTransmission(SENSOR_ADDRESS);
    Wire.write(d1);
    for ( byte i = 0; i < n; i++ ){
        Wire.write(ds[i]);
    }
    Wire.endTransmission();
}

void setup()
{
    Serial.begin(9600); // 9600 for processing
    Wire.begin();

    TWBR = 1; // max bit rate possible 800khz // IR_cam support the Fast Plus
    // To change the bit rate use TWBR = ((F_CPU / F_I2C) - 16) / 2
    // For example 400khz (Fast Mode) with 16Mhz oscillator => TWBR = ((1600000

    // IR sensor initialize
    Write_2bytes(0x30,0x08); delay(10); // begin config
    Write_Nbytes(0x00,setting1,9); delay(10); // sensibility 1
    Write_Nbytes(0x1A,setting2,2); delay(10); // sensibility 2
    Write_2bytes(0x33,0x03); delay(10); // data mode extended
}

void loop()
{
    Wire.beginTransmission(SENSOR_ADDRESS);
    Wire.write(0x36);
    Wire.endTransmission();
    Wire.requestFrom(SENSOR_ADDRESS, 4);

    for (byte i = 0 ; i < 4 ; i++){
        data_buf[i] = 0 ;
        data_buf[i] = Wire.read();
    }

    x = data_buf[1];
    y = data_buf[2];
    s = data_buf[3];
    x += (s & 0x30) <<4;
    y += (s & 0xC0) <<2;
    s &= 0x0F ;

    Serial.print( int(x) );
    Serial.print( "," );
    Serial.print( int(y) );
    Serial.print( "," ); // only if you use processing to visualize
    Serial.println( int(s) );
    //Serial.print((char)0); // required to finish XML data for XMLSocket proto
    delay(30); // only if you use processing to visualize
}

```

CODE PROCESSING / Télécharger (/content/iwand/code/camera_wii.ino)

```

import processing.serial.*;

int lf = 10;    // Linefeed in ASCII
String myString = null;
Serial myPort;  // The serial port

void setup() {
  // List all the available serial ports
  println(Serial.list());
  // Open the port you are using at the rate you want:
  myPort = new Serial(this, Serial.list()[10], 9600);
  myPort.clear();
  // Throw out the first reading, in case we started reading
  // in the middle of a string from the sender.
  myString = myPort.readStringUntil(lf);
  myString = null;
  size(1023,767);
  //frameRate(30);
}

void draw() {
  background(77);
  //while (myPort.available() > 0) {
  myString = myPort.readStringUntil(lf);
  if (myString != null) {
    int[] output = int (split(myString, ','));

    println(myString); // display the incoming string

    int x = output[0];
    int y = output[1];

    ellipseMode(RADIUS); // Set ellipseMode to RADIUS
    fill(255, 0, 0); // Set fill to white
    ellipse(x, y, 20, 20);
  }
}

```

Caméra DFRobot

La boutique [DFRobot](http://www.dfrobot.com/index.php?route=product/product&product_id=1088&search=camera+infrared&description=true#_V77c1PmLS9IDFRc) ([http://www.dfrobot.com/index.php?](http://www.dfrobot.com/index.php?route=product/product&product_id=1088&search=camera+infrared&description=true#_V77c1PmLS9IDFRc)

[route=product/product&product_id=1088&search=camera+infrared&description=true#_V77c1PmLS9IDFRc](http://www.dfrobot.com/index.php?route=product/product&product_id=1088&search=camera+infrared&description=true#_V77c1PmLS9IDFRc)

propose une caméra infrarouge identique à celle intégrée dans une manette wiimote.

Là où il était nécessaire de démonter le capteur de la manette pour y rajouter les résistances de pull-up du bus I2C ainsi que le résonateur 24Mhz, tout est intégré dans le module de DFRobot, un gain de temps énorme pour un prix aux alentours de 24\$.



Liens utiles :

- [Datasheet](http://www.dfrobot.com/wiki/index.php/Positioning_ir_camera) (http://www.dfrobot.com/wiki/index.php/Positioning_ir_camera)
- [Données techniques sur la caméra et le protocole](http://wiibrew.org/wiki/Wiimote#IR_Camera) (http://wiibrew.org/wiki/Wiimote#IR_Camera)

Câblage

- Rouge – VCC+
- Jaune – SDA
- Vert – SCL
- Noir – GND

Sur un arduino Nano la ligne SDA se trouve sur ADC4 et SCL en ADC5.
La tension de fonctionnement est de 3V3 à 5V.

So, what do you think ?

You must be [logged in](http://notepad.xavierdetourbet.com/wp-login.php?redirect_to=http%3A%2F%2Fnotepad.xavierdetourbet.com%2Fconnecter-une-camera-pixart-sur-arduino-en-protocole-i2c%2F) (http://notepad.xavierdetourbet.com/wp-login.php?redirect_to=http%3A%2F%2Fnotepad.xavierdetourbet.com%2Fconnecter-une-camera-pixart-sur-arduino-en-protocole-i2c%2F) to post a comment.