



# **CFGS AUTOMATIZACIÓ I ROBOTICA INDUSTRIAL**

**M6 – UF2 : SENSORS AVANÇATS**

**VISIÓ BÀSICA AMB PIXY2 I ARDUINO**

**Alumne:** Jose Granados Diaz

**Professor:** Josep Vargas

**Curs:** 2021-2022

**Data:** Dimarts 20 de Maig del 2022

## ACTIVITAT 1

### VISIÓ BÀSICA AMB PIXY2 I ARDUINO

#### **Descripció**

L'activitat consistirà en la programació de la detecció de color i posició d'unes peces plàstiques utilitzant com a sensor una càmera Pixy2 que es pot parametritzar a través del software PixyMon, connectada a una placa Arduino per al port I2C. Els programes i llibreries necessàries les pots trobar a la següent pàgina:

<https://pixycam.com/downloads-pixy2/>

#### **Activitats**

Realitza la definició dels paràmetres a la càmera Pixy2 i el programa d'arduino per activar una sèrie de sortides a partir de la detecció de la càmera i utilitzant la maqueta d'acord als següents criteris:

Color:

- 4 sortides diferents d'acord al color de l'objecte a detectar

Posició

- 4 Sortida del color parpadejant si hi ha tres objectes del mateix color en línia

Els documents a presentar en pdf seran:

- Portada
- Enunciat de l'activitat
- Descripció de la configuració en la càmera Pixy2
- Programa d'arduino (enviar en format informàtic amb l'informe)

## ACTIVITAT 2

### VISIÓ PROFESSIONAL AMB SISTEMA KEYENCE I CONNEXIÓ DIRECTA A PLC

#### **Descripció**

L'activitat consistirà en la programació del sistema de detecció de color d'unes peces plàstiques utilitzant com a sensor un sistema professional Keyence CV que es pot parametritzar a través de la consola de programació del sistema connectat a un PLC Omron CP1L, la documentació la pots trobar al manual en el suro del Clickedu.

#### **Activitats**

##### **Color**

- Realitza la programació del sistema per tal que el sistema detecti els colors de les diferents peces (4 colors diferents), transfereixi els resultats a àrea de memòria del PLC i fes el programa en el PLC per tal d'activar 4 sortides.

##### **Àrea**

- Realitza la programació del sistema per tal que el sistema detecti l'àrea de la peça, en cas de posar una peça de qualsevol color però àrea superior a un valor que s'activi un altra sortida.

Els documents a presentar seran:

- Portada
- Enunciat de l'activitat
- Descripció de la configuració en el sistema Keyence CV
- Programa de PLC

## Índex

|  |           |
|--|-----------|
| <b>1. Càmera visió artificial Pixy .....</b> | <b>5</b>  |
| <b>-Descripció: .....</b>                    | <b>5</b>  |
| <b>-Configuració:.....</b>                   | <b>6</b>  |
| <b>2.Programació:.....</b>                   | <b>7</b>  |
| <b>3.Resultat i conclusions: .....</b>       | <b>10</b> |

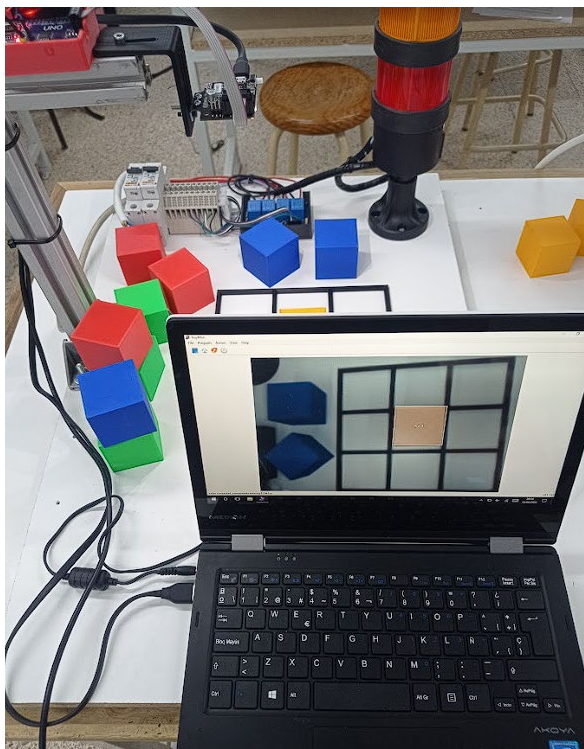
## 1. Càmera visió artificial Pixy

### - Descripció:

Pixy no és una càmera normal, sinó que és una càmera lliure i intel·ligent. Aquesta càmera no només es pot connectar a altres dispositius sense cap problema o simplement emmagatzema dades sinó que pot reconèixer colors d'imatges que grava, indicar-nos la posició, l'àrea entre altres coses...

Pixy ve acompanyat de PixyMon, un programari que és capaç de controlar i emmagatzemar les dades de la càmera Pixy perquè una vegada que la càmera s'apagui, l'ordinador torni a bolcar les dades i no perdi els colors assignats. A més, Pixy es pot connectar no només a un ordinador sinó també a plaques com Arduino o Raspberry Pi 2 que poden funcionar com una placa auxiliar i donar autonomia a la càmera Pixy.

En aquest cas nosaltres la programem en Arduino per poder aconseguir fer el joc 3 en ratlla amb una maqueta que té una reixa de posicions i amb blocs de plàstic quadrats de diferents colors impresos amb impressora 3D.



### - Configuració:

A l'hora de configurar la càmera Pixy connectem el cable USB al ordinadors i obrim el programa PixyMon. Una vegada obert anem a la pestanya superior Action i activem el flash de la càmera per aconseguir una millor visió en els blocs. Una vegada activat el flash assignem els colors. Per assignar els colors col·loquem un bloc de un color i fem clic en assignar en un dels 7 llocs de memòria. En aquest cas perquè el programa funcioni activant la llum correctament del bloc que detecta s'han de configurar de la següent manera:

- Verd: 1
- Vermell: 2
- Blau: 3
- Groc: 4



## 2. Programació:

```
#include <Pixy2.h>
```

```
Pixy2 pixy; // This is the main Pixy object
```

```
// Variables que us pot donar la llibreria de la pixy
```

```
// pixy.ccc.blocks[i].m_signature El número de color de l'objecte detectat (1-7 normalment)  
// pixy.ccc.blocks[i].m_x La ubicació x del centre de l'objecte detectat (0 a 316)  
// pixy.ccc.blocks[i].m_y La ubicació y del centre de l'objecte detectat (0 a 208)  
// pixy.ccc.blocks[i].m_width L'ample de l'objecte detectat (1 a 316)  
// pixy.ccc.blocks[i].m_height L'altura de l'objecte detectat (1 a 208)  
// pixy.ccc.blocks[i].m_angle L'angle de l'objecte detectat (-180 a 180).  
// pixy.ccc.blocks[i].m_index L'índex de seguiment del bloc.  
// pixy.ccc.blocks[i].m_age El número de fotogrames que s'ha rastrejat el bloc.  
// pixy.ccc.blocks[i].print() Una funció membre que imprimeix la informació de l'objecte detectat  
en el puerto serie
```

```
// Variables de colors dels blocs i de la seva posició en x i y
```

```
int colorBloque1;  
int colorBloque2;  
int colorBloque3;  
int colorBloque4;  
int x1;  
int x2;  
int x3;  
int y1;  
int y2;  
int y3;
```

```
// Declara unes variables per a el número de blocs que detecta la càmera (i) i per al color del  
bloc (s).
```

```
int i;  
int s;
```

```
void setup(){  
  Serial.begin(9600);  
  Serial.print("Starting...\n");
```

```
  pixy.init();
```

```
// Aquest bloc és per a declarar els pins com a sortides.
```

```
pinMode (3, OUTPUT);  
pinMode (4, OUTPUT);  
pinMode (5, OUTPUT);  
pinMode (6, OUTPUT);
```

// Aquest bloc és perquè els relés de la placa que hi ha connectada estan cablats a la inversa i  
així els posem tots  
// a nivell alt que vol dir les bombetes parades.

```
digitalWrite (3,HIGH);  
digitalWrite (4,HIGH);  
digitalWrite (5,HIGH);  
digitalWrite (6,HIGH);
```

```
}
```

//A través de la càmera Pixy nosaltres assignem los colores de los cubs de la següent manera:

//Verda: 1

//Vermell: 2

//Blau: 3

//Groc: 4

```
void loop(){
```

```
  pixy.ccc.getBlocks(); // Funció que carrega els blocs que hi han
```

```
  while (pixy.ccc.numBlocks >=3 ){ // Sempre que detecti mes de 3 blocs...
```

```
    pixy.ccc.getBlocks();
```

```
    colorBloque1 = pixy.ccc.blocks[0].m_signature; // Assignem les variables del color entrar si  
    entra un color blau assignarà un 3 a la variable colorBloque1.
```

```
    colorBloque2 = pixy.ccc.blocks[1].m_signature;
```

```
    colorBloque3 = pixy.ccc.blocks[2].m_signature;
```

```
    if ((colorBloque1 == colorBloque2) && (colorBloque1 == colorBloque3) && (colorBloque2 ==  
    colorBloque3)){ //Compara els colors dels blocs
```

```
      x1=pixy.ccc.blocks[1].m_x; //Es guarda la posició en x dels blocs introduïts
```

```
      x2=pixy.ccc.blocks[2].m_x;
```

```
      x3=pixy.ccc.blocks[3].m_x;
```

```
      y1=pixy.ccc.blocks[1].m_y; //Es guarda la posició en y dels blocs introduïts
```

```
      y2=pixy.ccc.blocks[2].m_y;
```

```
      y3=pixy.ccc.blocks[3].m_y;
```

```
    }
```



```
if (((x1 >=150) && (x1<=250)) && ((x2 >=80) && (x2<=220)) && ((x3 >=270) &&
(x3<=300)) or ((y1 >=60) && (y1<=200)) && ((y2 >=10) && (y2<=200)) && ((y3 >=160) &&
(y3<=200)) or ((x1 >=80) && (x1<=200)) && ((x2 >=100) && (x2<=260)) && ((x3 >=100) &&
(x3<=190)) and ((y1 >=20) && (y1<=150)) && ((y2 >=20) && (y2<=160)) && ((y3 >=20) &&
(y3<=150)) or ((x1 >=100) && (x1<=220)) && ((x2 >=60) && (x2<=260)) && ((x3 >=100) &&
(x3<=250)) && ((y1 >=50) && (y1<=200)) && ((y2 >=80) && (y2<=250)) && ((y3 >=30) &&
(y3<=100))) { // Compara la posició dels blocs per determinar si es 3 en ratlla dins d'un marge
de posició.
```

```
switch (colorBloque1) { //Si hi han 3 en ratlla amb el CASE fem que s'activi només la que
esta en 3 en ratlla i fem que s'aturi i s'activi la llum del color que detecta.
```

```
case 1: digitalWrite (3,LOW); delay (200); digitalWrite (3, HIGH); delay (200); break; // Per
exemple, per a la línia de case:1, si colorBloque1 és igual a 1 (el nombre l'hem assignat
mitjançant la càmera Pixy, en el nostre cas era el verd) s'activarà la sortida 3 que correspon a
la llum verda.
```

```
case 2: digitalWrite (4,LOW); delay (200); digitalWrite (4, HIGH); delay (200); break; //Els
LOW i HIGH seguits de delay serveixen perquè la corresponent llum s'encengui i s'apagui en
interval·ls de 1000 mil·lisegons (1 segon).
```

```
case 3: digitalWrite (5,LOW); delay (200); digitalWrite (5, HIGH); delay (200); break;
```

```
case 4: digitalWrite (6,LOW); delay (200); digitalWrite (6, HIGH); delay (200); break;
```

```
}
}
```

**Serial.print**(pixy.ccc.numBlocks); //Mitjançant aquestes successives línies es visualitza al nostre monitor els diferents paràmetres obtinguts amb les funcions que hem executat. Cal tenir en compte que la velocitat del nostre monitor ha d'estar posada a 115.200 bauds.

```
Serial.print(" "); // Això correspon a un espai.
Serial.print(colorBloque1);
Serial.print(" ");
Serial.print(colorBloque2);
Serial.print(" ");
Serial.print(colorBloque3);
Serial.print(" ");
Serial.print(x1);
Serial.print(" ");
Serial.print(x2);
Serial.print(" ");
Serial.print(x3);
Serial.print(" ");
Serial.print(y1);
Serial.print(" ");
Serial.print(y2);
Serial.print(" ");
Serial.print(y3);
Serial.println(); // Això correspon a un salt de línia.
delay(100);
}
}
```

### **3. Resultat i conclusions:**

Hem aconseguit ajustar els rangs de manera que les peces en posició L no detecti 3 en ratlla ni quan dues estan en vertical i una al mig. Al ajustar tant el rang a vegades li costa que detecti el 3 en ratlla depenen la posició en la que deixem el bloc. Però d'aquesta manera es molt mes estable i intenta marcar nomes 3 en ratlla quan ho es.

- Vídeo de la prova amb la programació comentada anteriorment amb blocs de diferents colors i posicions:

<https://youtu.be/CZPrAyV0vH4>

Un dels problemes detectat després de unes quantes proves es que si es mou la càmera la programació ja no serveix perquè els valors no entren dintre del rang. Si fem el rang molt mes ampli aquest detecta 3 en ratlla quan no ho es.

Si intentem ajustar el programa perquè detecti nomes 3 en ratlla quan ho es, la càmera també fa variar molt els valors en x i y, es molt difícil aconseguir ajustar amb un rang molt petit quan els valors varien molt.



La il·luminació en la càmera també es molt important ja que si no hi ha bona il·luminació la càmera no es capaç de reconèixer correctament l'àrea i la posició del bloc. El color també es molt important quan no hi ha molta il·luminació.

Una de les millores que es podria fer es fixar correctament la càmera que detectes correctament totes les peces de la graella, il·luminar l'espai amb il·luminació extra i fer una graella en la qual les peces hi entrin justes que no tinguin marge on les podem col·locar i que la separació entre aquestes sigui mes gran per evitar que la càmera els detecti junts. També es podrien fer unes peces amb poca alçada perquè la càmera nomes detecti la cara superior.