Comp2403 Camera Setup

C.B.Price October 2021

Purpose	To configure the PixyCam to respond to red objects.
Files Required	**Pixymon** exe, available on the webpages.
ILO Contribution	LOs 1,2,3
Send to Me	nix
Homework	Read chapters 1 & 2

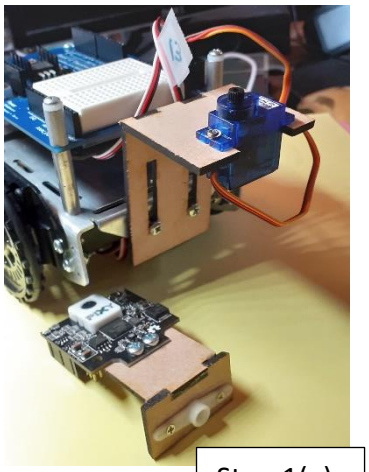
Activities

Protocol:
(1) Treat the kit like eggs (not hard-boiled)
(2) When you upload a sketch to the Arduino, make sure the power switch is off (position 0).
(3) When starting a sketch, hold down reset then turn the power on (switch position 2)
(4) Do not drive the turret with the Pixymon cable attached.

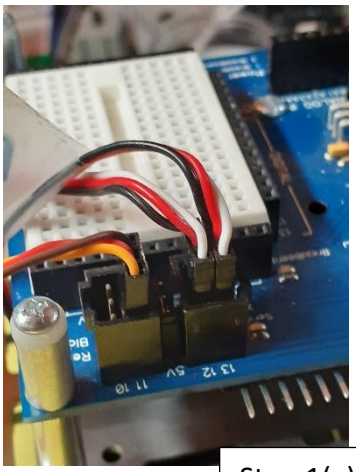
---

1 Mounting the PixyCam Turret.

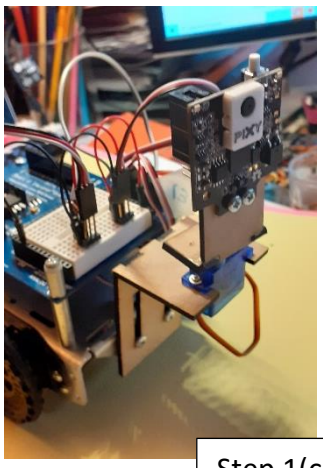
(a) Mount the bracket with the servo onto the front of the robot. Do not attach the PixyCam yet. Plug the servo wire into the connector labelled **11** and make sure you have the plug in the right way – check the colors.



Step 1(a)



Step 1(a)



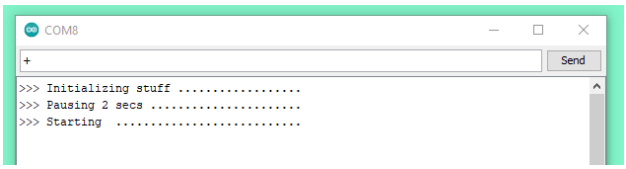
Step 1(c)

(b) Now we must centre the servo, so the PixyCam will face forward. First run the sketch **CBP_3402_R_Turret_Tweak** and the motor will turn to 90 degrees, the forward position.

(c) Now carefully mount the cam onto the motor; there are teeth which will engage. The cam will not point exactly forwards. Screw the cam onto the motor, carefully. Yes, I said “carefully”.

(d) Now we shall apply an **offset** to get the camera exactly pointing forward. With the sketch **CBP_3402_R_Turret_Tweak** running, open up the serial monitor; you will get a screen like this

---



Now in the input box (at the top, left of "Send") type either **+** and enter or **–** and enter. This will make the turret rotate in steps of 1 degree. Continue until the cam is facing forwards

Note down the value of 'offset' *reported in the monitor.*

(e) Now press ***** in the monitor, the cam will rotate to 180 degrees, then to 0 degrees, then back to 90 degrees. ***Remember*** the position of the cam for these angles!

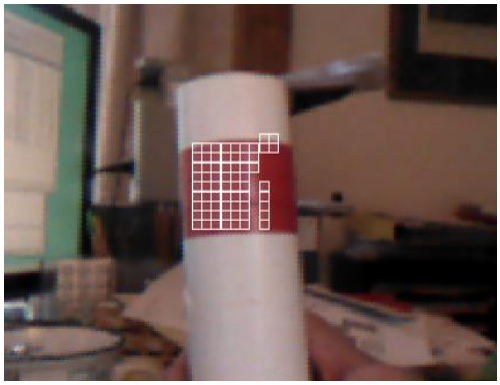
---

2 Tweaking PixyCam to recognize red object

Here we shall investigate what the PixyCam sees, and how to tweak it to recognize red objects in its field of view.

(a) Download and install **Pixymon** exe from the webpages. Fire this up.
(b) Carefully connect the cam to the PC through the USB cable – you should see what the cam sees
(c) Now do the following steps

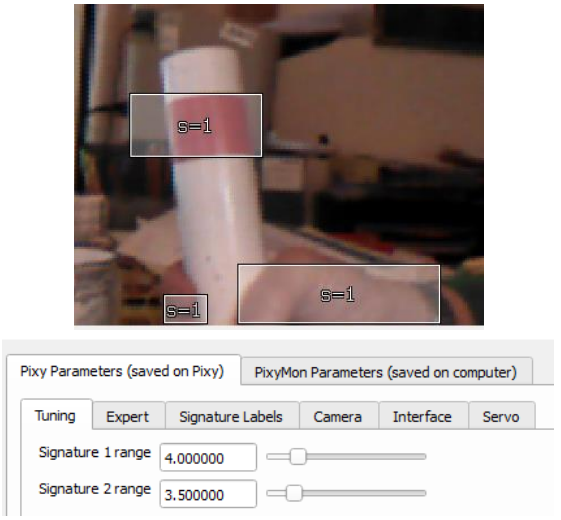
- Wait till all the cam's LEDs have stopped flashing
- Hold down the white button – the LEDs will cycle through colors. Release the button when you get a red (after around 10 secs)
- Move the red object (on the cylinder) in the cam's field of view. You should see something similar to the image below

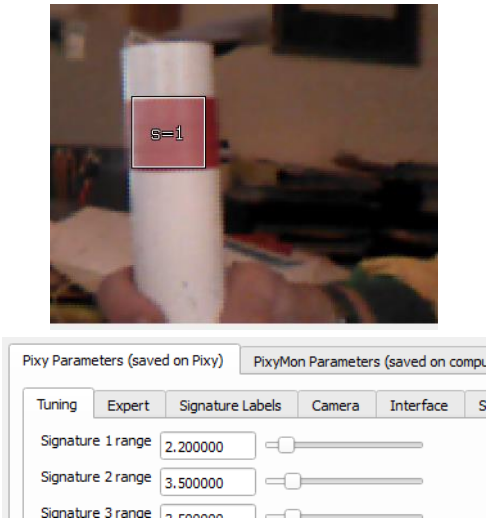


- Move the object so that the grid covers as much of the red part as possible
- Press and release the white button. You should get something similar to the image below

- You can see that the cam has segmented and labelled the red component. Now move the object around and you should see the cam tracks it easily.
- Maybe you need to repeat the above steps, until you are happy that the cam can detect the object at small and large distances

(d) You may find you're getting *false positives* and *false negatives*. To tweak the detection bring up the Configure dialogue box **File > Configure** then under **Pixy Parameters** select the **Tuning** tab. Move slider 1 to the left if you want it to less inclusive (false positives) and to the right if you want it to be more inclusive (false negatives). Look at the example below.





(e) Now keep the object at the centre of the cam's field of view and move it near and far. Estimate the minimum and maximum distances of detection.

(f) Now choose an intermediate distance. Now move the object from side to side. Estimate the field of view of the camera in degrees.

3 Looking at the PixyCam Code

> ***A note to coders:*** *The Pixy2 library code* ****.h*** *files can be found here. E:\Arduino\portable\sketchbook\libraries\Pixy2*
> *Look at Pixy2CCC.h for connected color components. It's OO.*

(a) Load up the sketch **CBP_3402_R_Camera_Investigate**. Start by reading the code. Look for the following

- Where the servos (wheels and turret) are declared and where the turret is attached to its Arduino pin
- Where the pixy object is declared and its variables too
- Look in the **setup()** function and look for all lines of code relating to the **pixy** object. Make sure you understand what these do (if not, follow the Wiki link on the website)

(b) Look for the line **offset = 0;** and replace the **0** with the offset you found above.

(c) Run the sketch and open up the Serial Monitor. Place the red object in the cam's field of view and move it around. You should see information about its position (x,y) on the Monitor. Work out where the origin of the x-y axes is.

(d) Add code to get the width and height of the image (use the **m_width** and **m_height** fields of the

blocks object to do this, see below) and print it to the Serial Monitor (how the **cprintf** function works is explained below below). Use the same syntax for the (x,y) position.

```
uint16_t m_signature;
uint16_t m_x;
uint16_t m_y;
uint16_t m_width;
uint16_t m_height;
int16_t m_angle;
uint8_t m_index;
uint8_t m_age;
```

I've created a c-equivalent printf function to replace the cumbersome Arduino printing to the Serial Monitor. Here's how it works:

There are two parts to **cprintf;** the first is a *string* which is printed, the second is a list of parameters. The string contains a placeholder for each parameter, this always starts with a **%.** If your parameter is an **int** then the placeholder is **%d**, if your parameter is a **float** then the placeholder is **%f**. See the tab **CBP_3402_Helper1** for other possibilities.

Here's some examples

Code	**Possible output**
`cprintf("Result is %d\n",intVar);`	Result is 69
`cprintf("Matt has %d large coconuts\n",nrNuts)`	Matt has 2 large coconuts
`cprintf("Robot: x = %f, y = %f\n",xPos,yPos);`	Robot: x = 30.2, y = 29.3
`cprintf("Blob[%d] has width %d\n",index,width).`	Blob[1] has width 44

(e) Make sure the width and height change as you move the object around. Do the values agree with what you expect? Or can you agree with what the value tell you?

4 [Optional] Recognizing Green or Yellow Blocks Sort it!