



CNRLAB

CIM&ROBOTICS LABORATORY

로봇공학입문설계

14주차 로봇 팔(3)

로봇공학과

□ Robot Arm

Step 1. 영점 조절

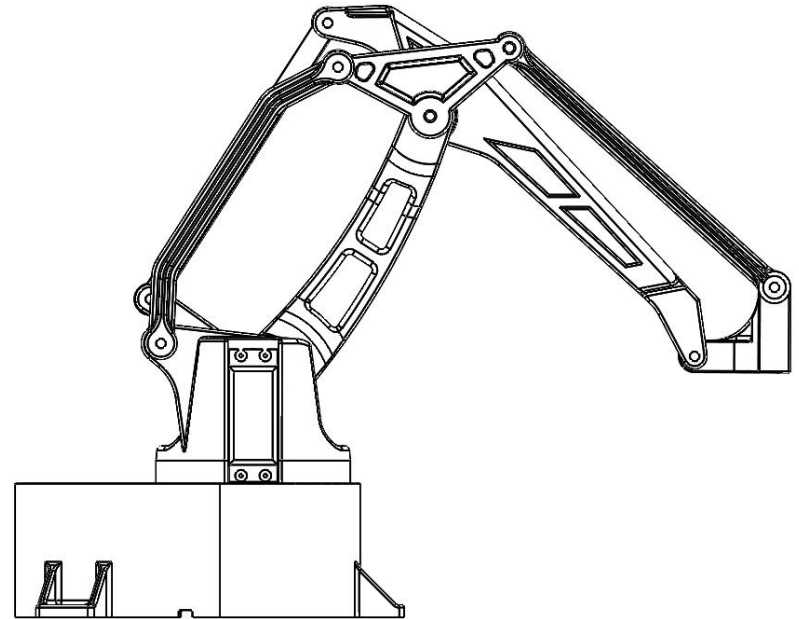
Step 2. 각도 변환

Step 3. 제한범위 설정

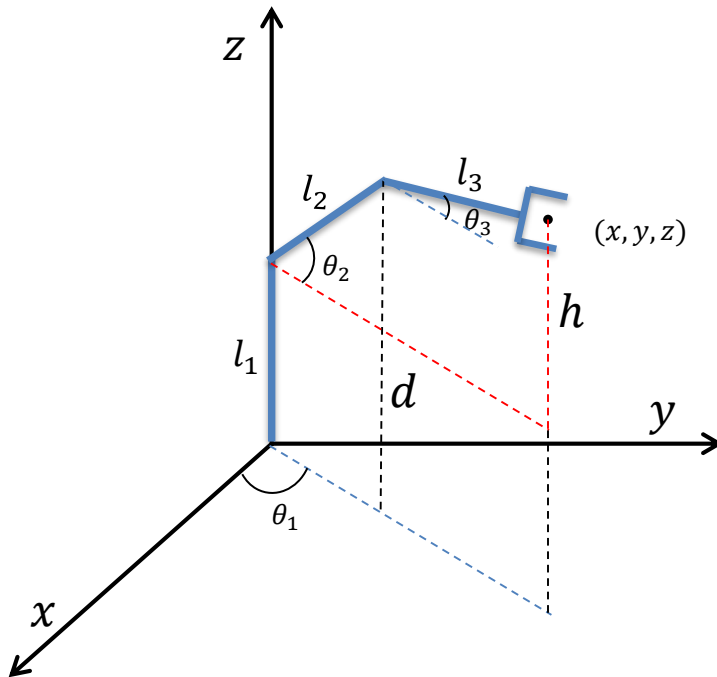
Step 4. Forward Kinematics

Step 5. Inverse Kinematics

Step 6. Point-to-Point Trajectory



□ Forward Kinematics of 3 link arm (Review)



Given $\theta_1, \theta_2, \theta_3$

Find x, y, z

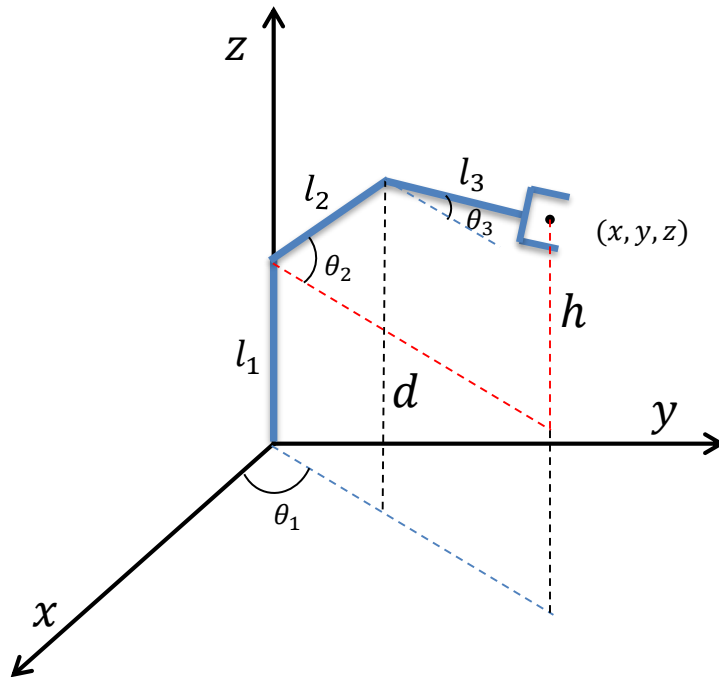
$$d = l_2 \cos \theta_2 + l_3 \cos \theta_3$$

$$h = l_2 \sin \theta_2 + l_3 \sin \theta_3$$

$$\therefore \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d \cos \theta_1 \\ d \sin \theta_1 \\ l_1 + h \end{bmatrix} = \begin{bmatrix} l_2 c_1 c_2 + l_3 c_1 c_3 \\ l_2 s_1 c_2 + l_3 s_1 c_3 \\ l_1 + l_2 s_2 + l_3 s_3 \end{bmatrix}$$

$$(c_2 = \cos \theta_2, s_2 = \sin \theta_2)$$

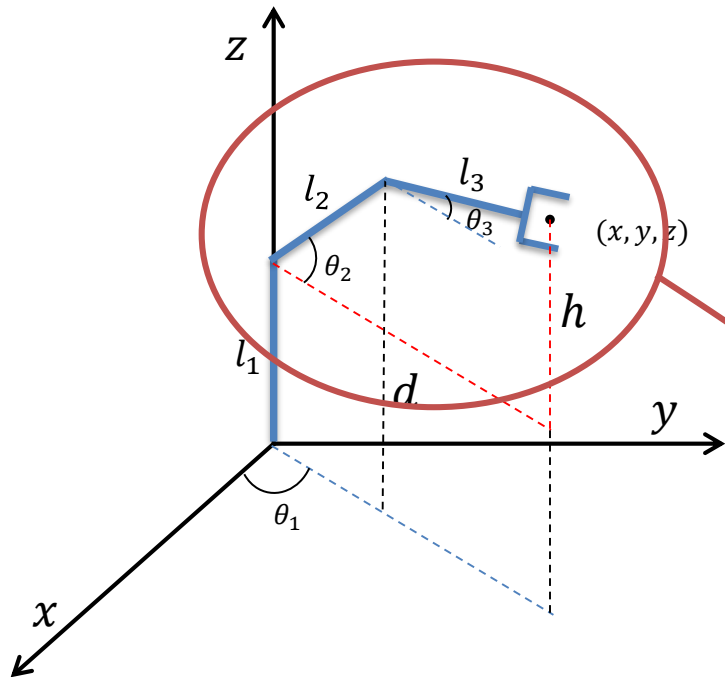
□ Inverse Kinematics of 3 link arm



Given x, y, z

Find $\theta_1, \theta_2, \theta_3$

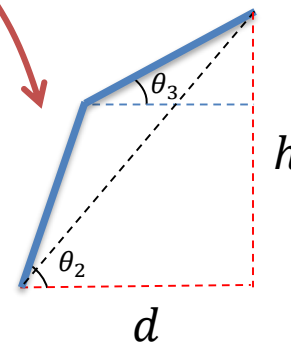
□ Inverse Kinematics of 3 link arm



Given x, y, z

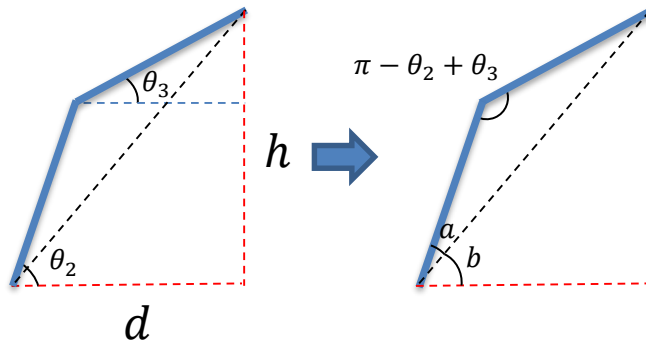
Find $\theta_1, \theta_2, \theta_3$

$$\theta_1 = \text{atan}\left(\frac{y}{x}\right), \quad -90^\circ \leq \theta_1 \leq 90^\circ$$



Find θ_2, θ_3

□ Inverse Kinematics of 3 link arm



$$d = \sqrt{x^2 + y^2}, \quad h = z - l_1$$

$$\theta_2 = a + b$$

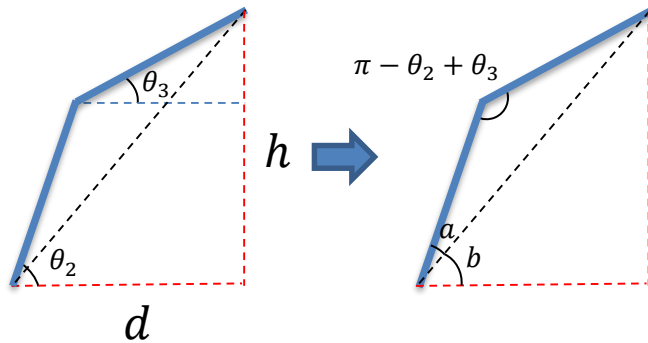
$$l_3^2 = l_2^2 + d^2 + h^2 - 2l_2\sqrt{d^2 + h^2}\cos(a)$$

$$a = \arccos\left(\frac{d^2 + h^2 + l_2^2 - l_3^2}{2l_2\sqrt{d^2 + h^2}}\right), \quad 0^\circ \leq a \leq 90^\circ$$

$$b = \operatorname{atan}\left(\frac{h}{d}\right), \quad -90^\circ \leq b \leq 90^\circ$$

$$\therefore \theta_2 = \arccos\left(\frac{l_3^2 - d^2 - h^2 - l_2^2}{2l_2\sqrt{d^2 + h^2}}\right) + \operatorname{atan}\left(\frac{h}{d}\right)$$

□ Inverse Kinematics of 3 link arm



$$d = \sqrt{x^2 + y^2}, \quad h = z - l_1$$

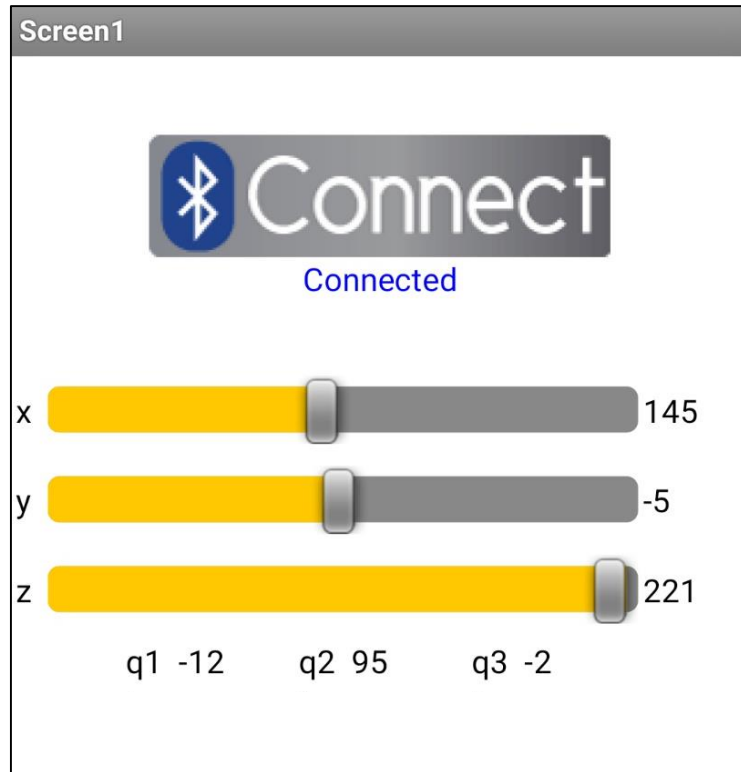
$$d^2 + h^2 = l_2^2 + l_3^2 - 2l_2l_3\cos(\pi - \theta_2 + \theta_3)$$

$$\theta_2 - \theta_3 = \arccos\left(\frac{d^2 + h^2 - l_2^2 - l_3^2}{2l_2l_3}\right)$$

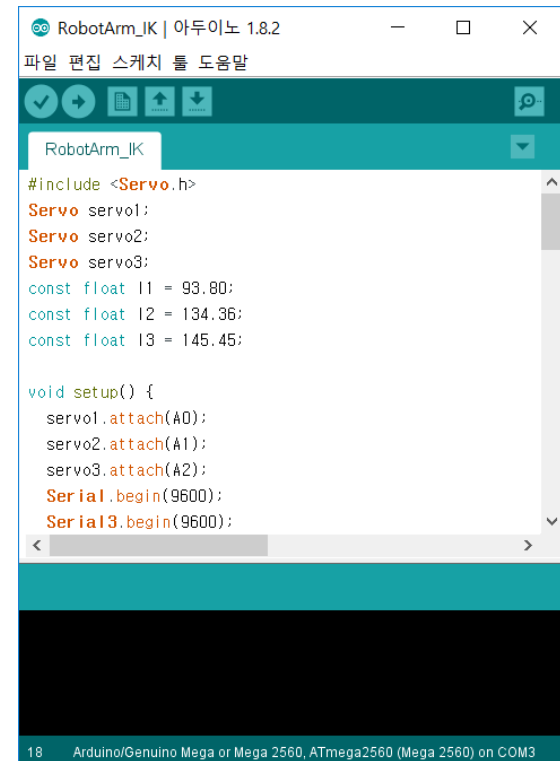
$$\theta_2 = \arccos\left(\frac{l_3^2 - d^2 - h^2 - l_2^2}{2l_2\sqrt{d^2 + h^2}}\right) + \operatorname{atan}\left(\frac{h}{d}\right)$$

$$\therefore \theta_3 = \arccos\left(\frac{l_3^2 - d^2 - h^2 - l_2^2}{2l_2\sqrt{d^2 + h^2}}\right) + \operatorname{atan}\left(\frac{h}{d}\right) - \arccos\left(\frac{d^2 + h^2 - l_2^2 - l_3^2}{2l_2l_3}\right)$$

□ Step5. Inverse Kinematics



RobotArm_IK.aia



RobotArm_IK.ino

□ Step5. Inverse Kinematics

```
#include <Servo.h>
Servo servo1;
Servo servo2;
Servo servo3;
const float l1 = 93.80;
const float l2 = 134.36;
const float l3 = 145.45;

void setup() {
  servo1.attach(A0);
  servo2.attach(A1);
  servo3.attach(A2);
  Serial.begin(9600);
  Serial3.begin(9600);
}
```

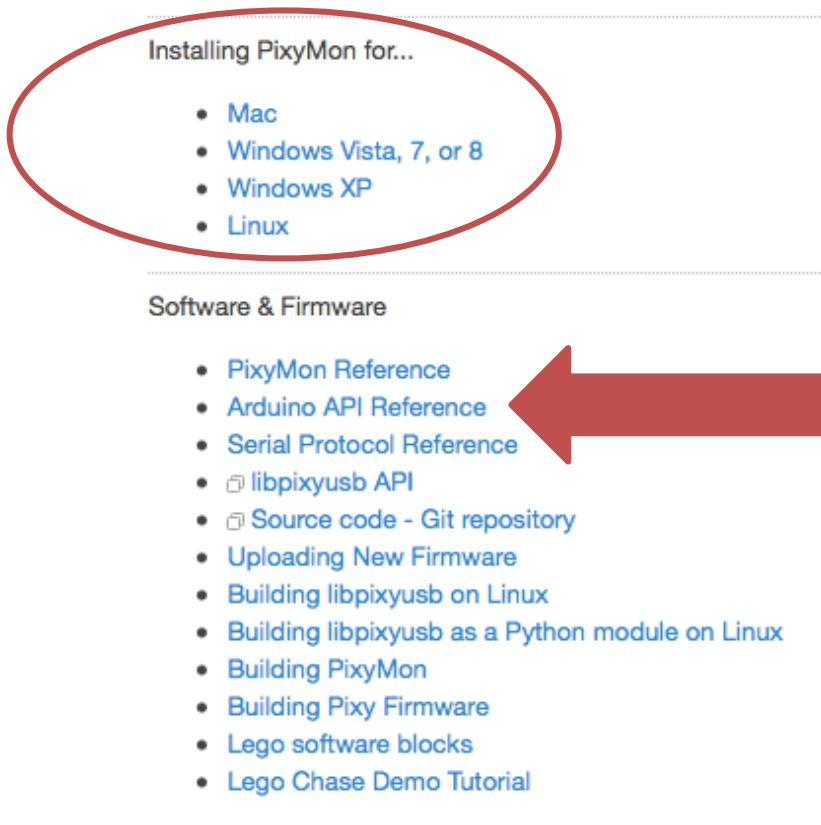
```
float cos_deg(int deg){
  float rad = deg/180.0*PI;
  return cos(rad);
}
float sin_deg(int deg){
  float rad = deg/180.0*PI;
  return sin(rad);
}
float atan_deg(float val){
  return atan(val)*180.0/PI;
}
float acos_deg(float val){
  return acos(val)*180.0/PI;
}
bool isWorkspace(int ang1, int ang2, int ang3){
  if(ang1>=-30 && ang1<=30 && ang2>=0 && ang2<=140 &&
  ang3>=-90 && ang3<=13 && ang2-ang3>=36 && ang2-ang3<=145) return 1;
  else return 0;
}
```

```
void loop() {  
  if(Serial3.available()){  
    float x = Serial3.parseInt();  
    float y = Serial3.parseInt();  
    float z = Serial3.parseInt();  
  
    if(Serial3.read() == '\n'){  
      // Inverse Kinematics  
      float d = sqrt(x*x+y*y);  
      float h = z-l1;  
      float ang1 = atan_deg(y/x);  
      float ang2 = acos_deg((d*d+h*h+l2*l2-l3*l3)  
                           /(2*l2*sqrt(d*d+h*h)))+atan_deg(h/d);  
      float ang3 = ang2-acos_deg((d*d+h*h-l2*l2-l3*l3)  
                                /(2*l2*l3));  
  
      if(isWorkspace(ang1, ang2, ang3)==1){  
        // 변환  
        float s_val1 = 180-((ang1*50.0/22.0)+90);  
        float s_val2 = 180-ang2;  
        float s_val3 = 180-(-ang3+45);  
        servo1.write(s_val1);  
        servo2.write(s_val2);  
        servo3.write(s_val3);  
      }  
    }  
  }  
}
```

```
    // Send Message  
    String Message = "";  
    Message += "M.";  
    Message.concat((int)ang1);  
    Message += ".";  
    Message.concat((int)ang2);  
    Message += ".";  
    Message.concat((int)ang3);  
    Message += ".";  
    Serial3.print(Message);  
  }  
  else{  
    Serial.println("Out of Workspace");  
  }  
}  
}  
}
```

❑ CMUCAM5 프로그램(Pixymon) 설치

- <http://cmucam.org/projects/cmucam5/wiki>

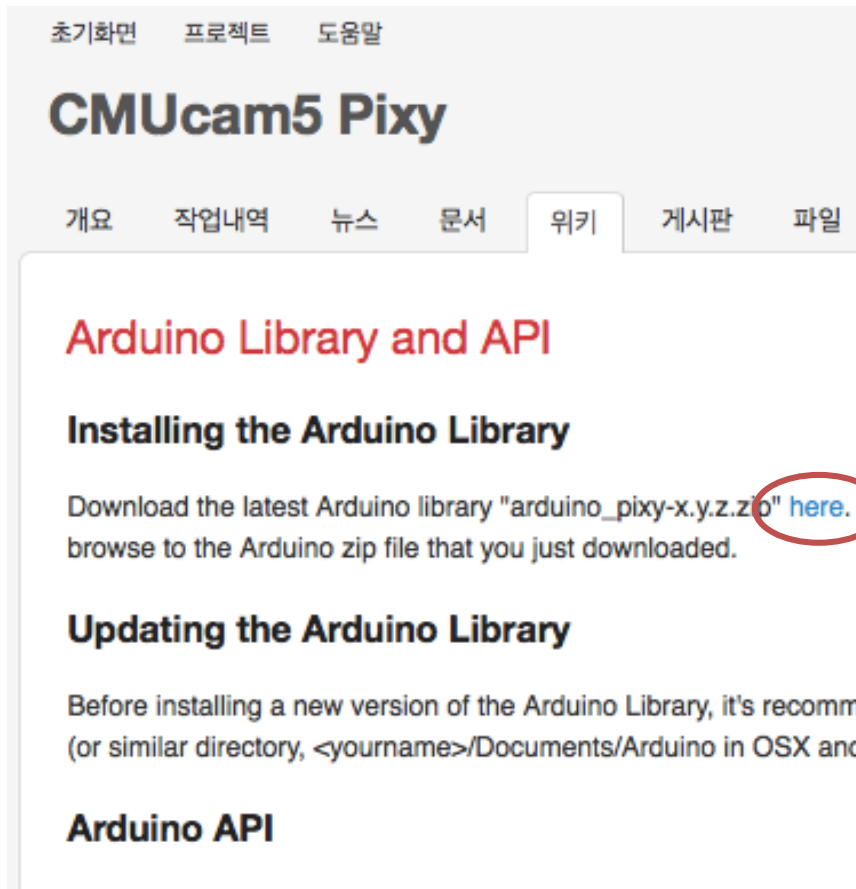


위 사이트에서 제일 아래로 내려가면 좌측과 같은 창이 있다.

자신의 운영체제에 맞는 파일을 다운받아 설치한다.

그 다음 화살표가 가르키는 링크(Arduino API Reference)를 클릭한다.

❑ CMUCAM5 프로그램(Pixymon) 설치

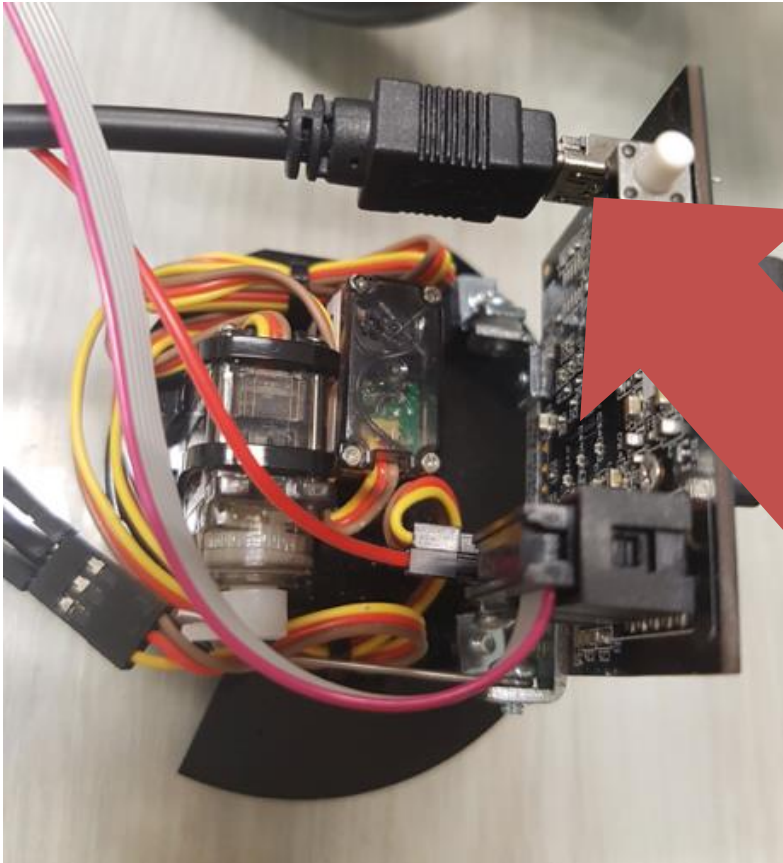


The screenshot shows the CMUcam5 Pixy website. At the top, there are navigation links: 초기화면, 프로젝트, and 도움말. Below this is the title CMUcam5 Pixy. A secondary navigation bar includes: 개요, 작업내역, 뉴스, 문서, 위키, 게시판, and 파일. The main content area has a heading **Arduino Library and API** in red. Underneath is the subheading **Installing the Arduino Library**. The text below reads: "Download the latest Arduino library "arduino_pixy-x.y.z.zip" [here](#). E" where the word "here" is circled in red. Below this is another subheading **Updating the Arduino Library** followed by text: "Before installing a new version of the Arduino Library, it's recomme (or similar directory, <yourname>/Documents/Arduino in OSX and". At the bottom of the visible section is the heading **Arduino API**.

베너를 타고 들어오면 "arduino_pixy-x,y,z,zip" 이라적혀있는 글 옆에 파란색으로 써진 here 을 누르면 라이브러리가 다운된다.

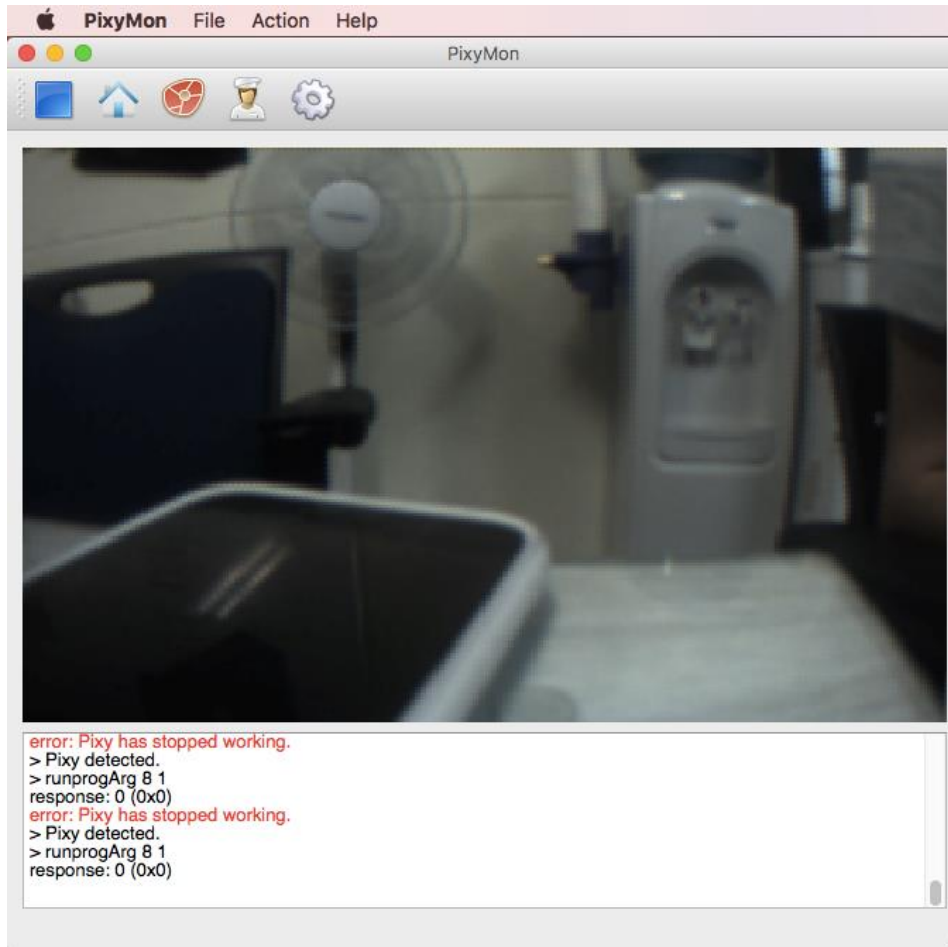
라이브러리 다운후 아두이노 라이브러리 zip파일 추가를 이용하여 라이브러리를 설치해준다.

□ CMUCAM5프로그램 소개



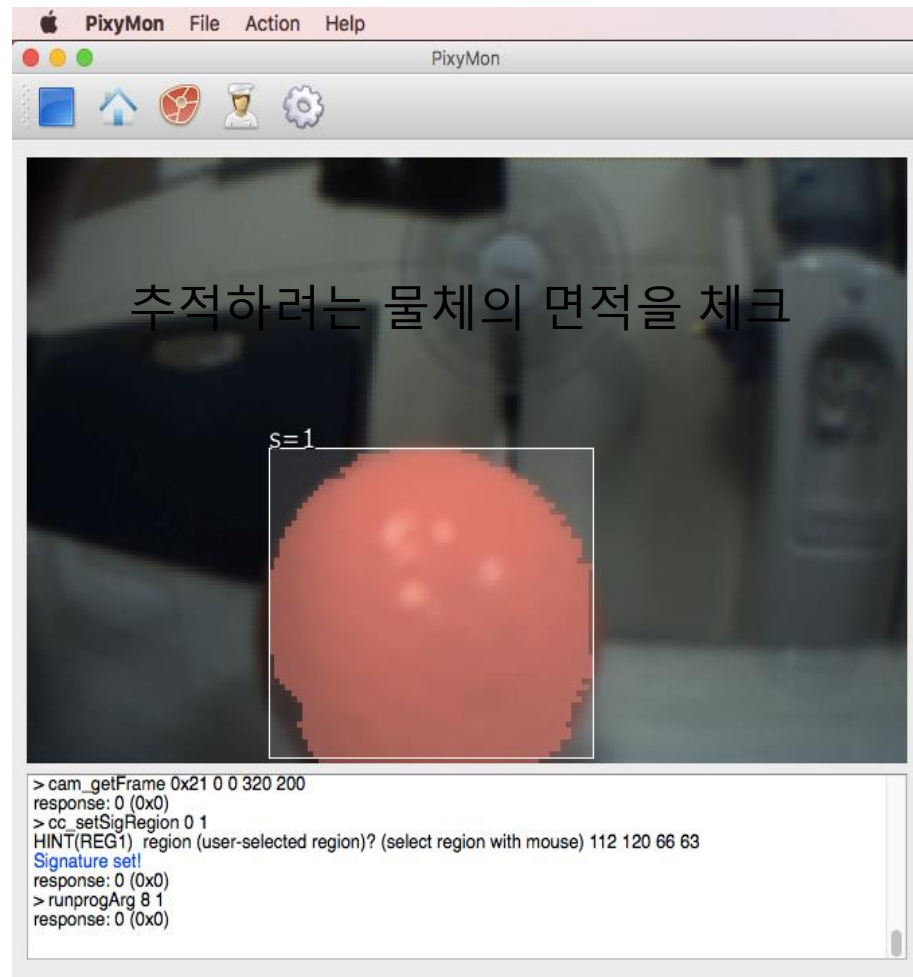
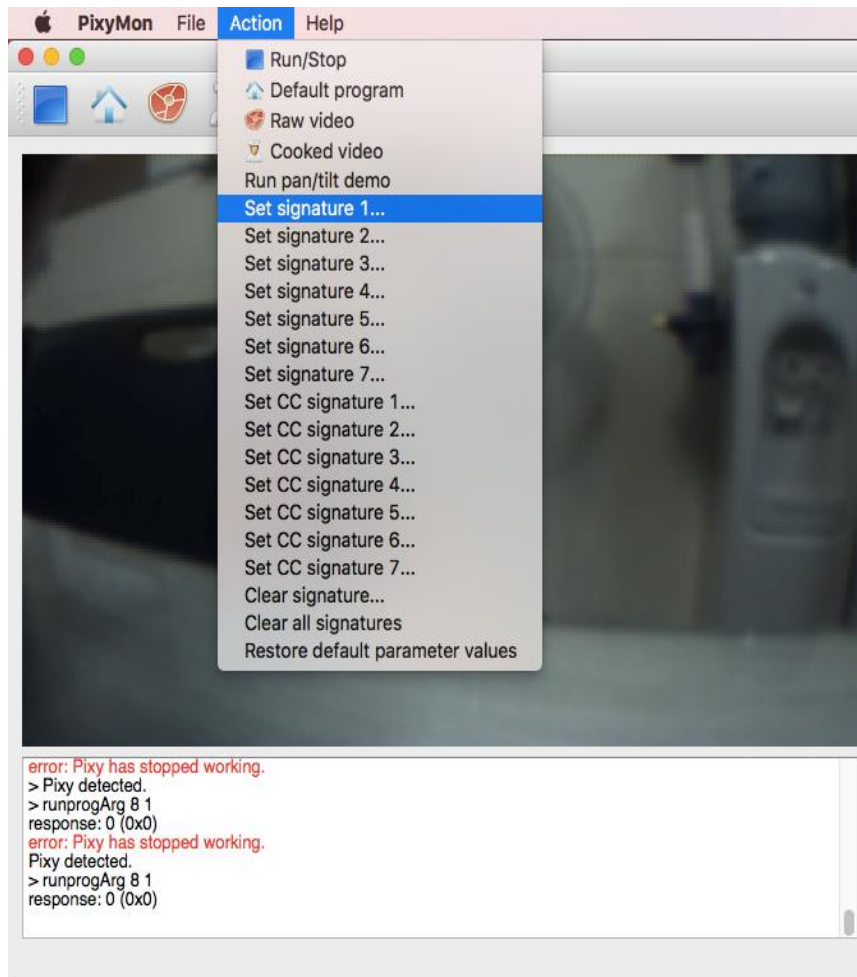
Cam 뒷부분에 usb호스트에
케이블을 꼽고 컴퓨터와 연결한다.

CMUCAM5프로그램 소개

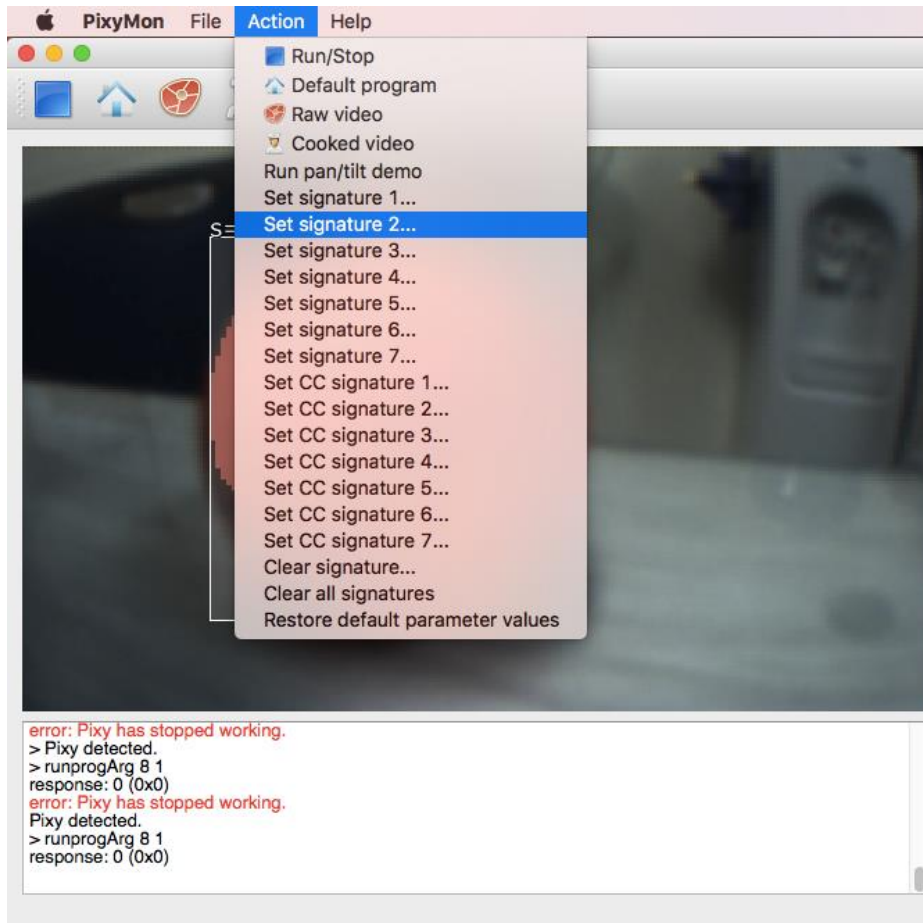


컴퓨터와 cam을 연결한후
프로그램을 설치한뒤 처음 화면을
실행하면 이와같은 화면이 나오면
연결이 잘 되었다는 뜻이다.

CMUCAM5 물건 트레킹

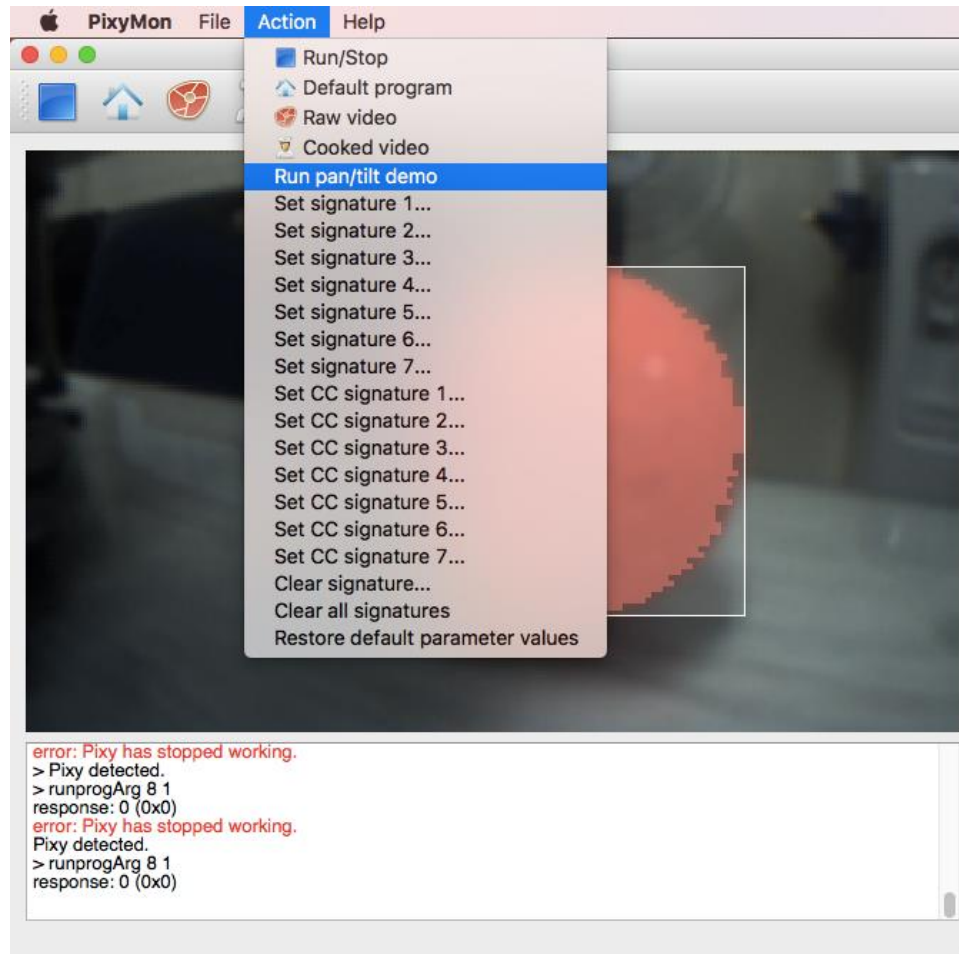


CMUCAM5 물건 트레킹

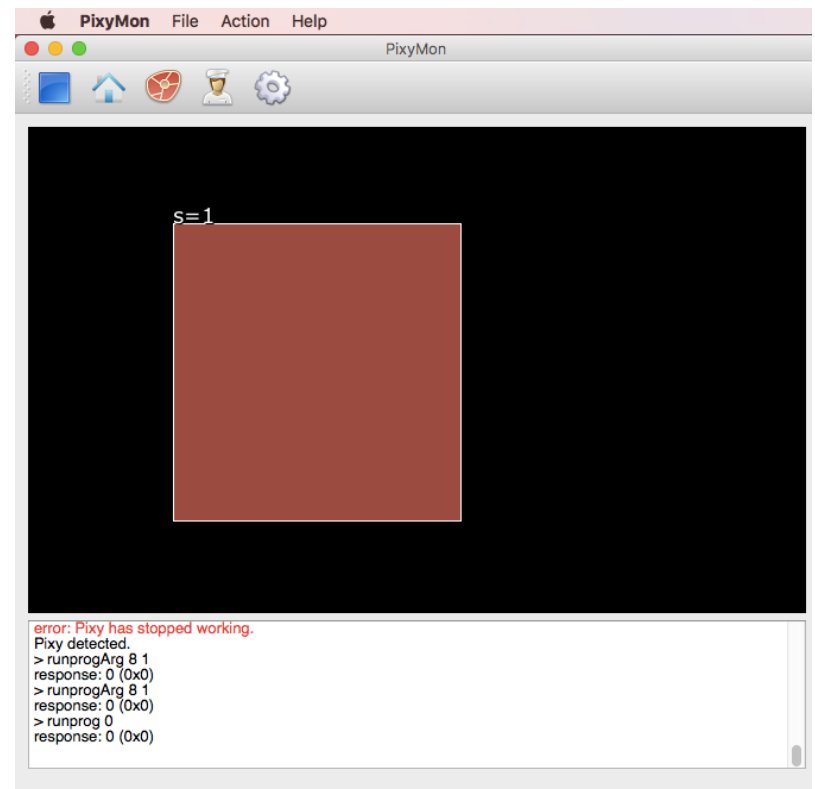


이때 다른 set을 누르고 하면
또다른 색상값을 인식 시킬수
있다.

CMUCAM5 물건 트레킹

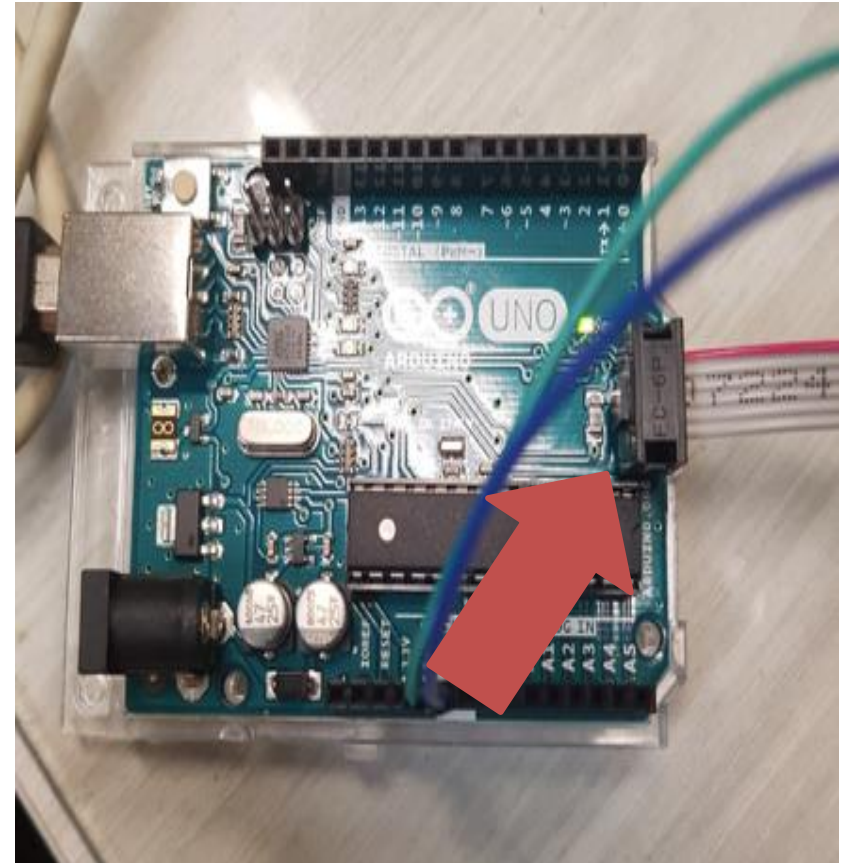
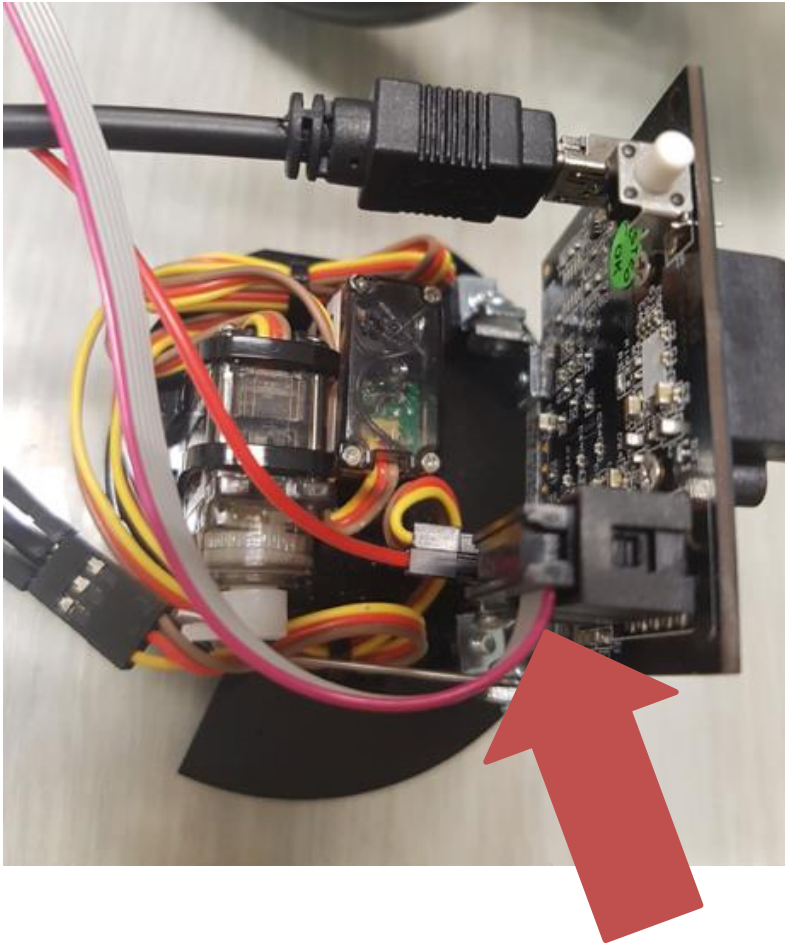


트레킹을 실행하는 버튼을 누르면 트레킹이 실행된다.



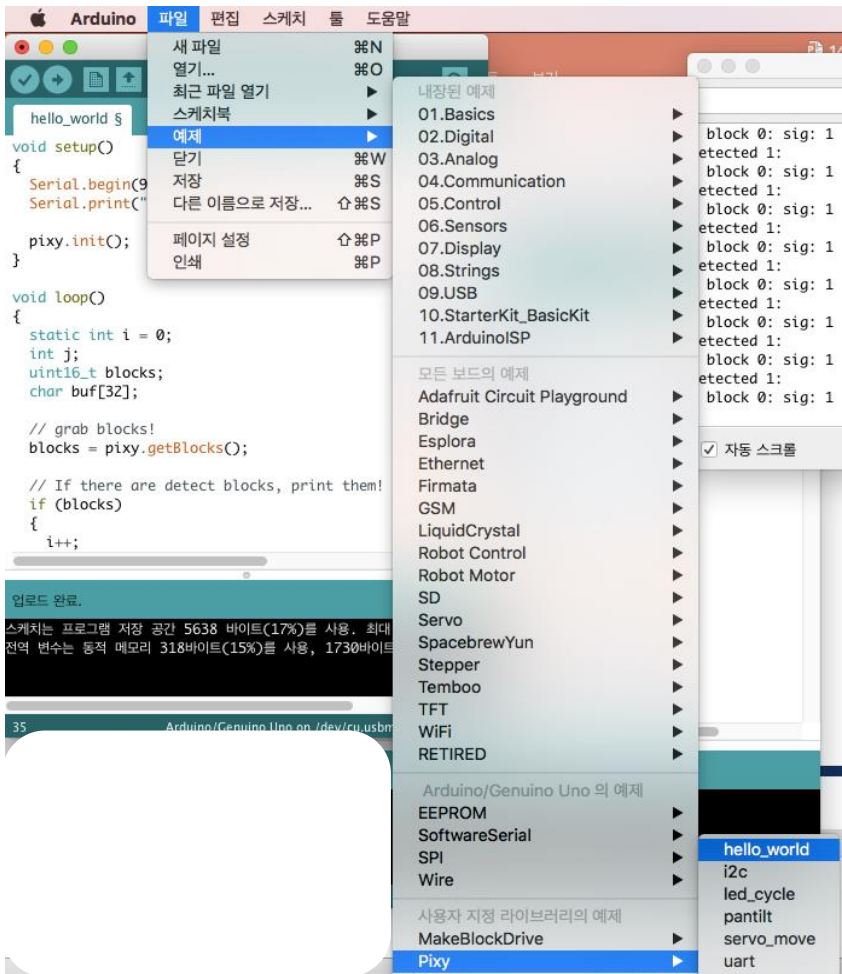
CMUCAM5

□ CMUCAM5 아두이노 신호 받기



❑ CMUCAM5 아두이노 신호 받기

파일 > 예제 > Pixy > hello_world



□ CMUCAM5 아두이노 신호 받기

```
#include <SPI.h>
#include <Pixy.h>

// This is the main Pixy object
Pixy pixy;

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

    pixy.init();
}

void loop()
{
    static int i = 0;
    int j;
    uint16_t blocks;
    char buf[32];

    // grab blocks!
    blocks = pixy.getBlocks();
```

```
    if (blocks)
    {
        panError = X_CENTER-pixy.blocks[0].x;
        tiltError = pixy.blocks[0].y-Y_CENTER;

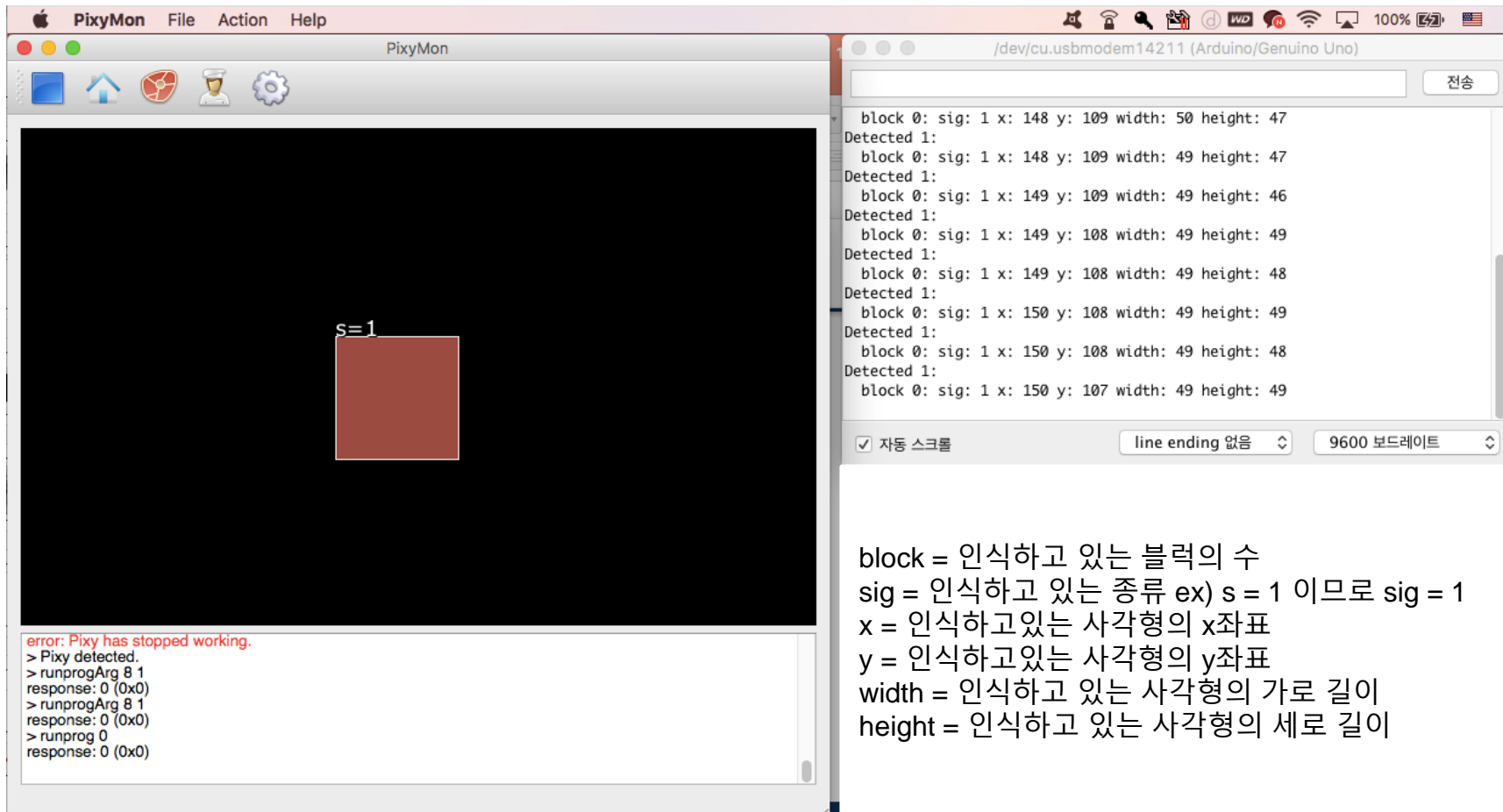
        panLoop.update(panError);
        tiltLoop.update(tiltError);

        pixy.setServos(panLoop.m_pos, tiltLoop.m_pos);

        i++;

        // do this (print) every 50 frames because printing every
        // frame would bog down the Arduino
        if (i%50==0)
        {
            sprintf(buf, "Detected %d:\n", blocks);
            Serial.print(buf);
            for (j=0; j<blocks; j++)
            {
                sprintf(buf, "  block %d: ", j);
                Serial.print(buf);
                pixy.blocks[j].print();
            }
        }
    }
}
```

CMUCAM5 아두이노 신호 받기



The screenshot displays two windows. The left window, titled 'PixyMon', shows a black camera feed with a single red square object detected. Above the square, the text 's=1' is visible. Below the feed, a console window shows the following text:

```
error: Pixy has stopped working.  
> Pixy detected.  
> runprogArg 8 1  
response: 0 (0x0)  
> runprogArg 8 1  
response: 0 (0x0)  
> runprog 0  
response: 0 (0x0)
```

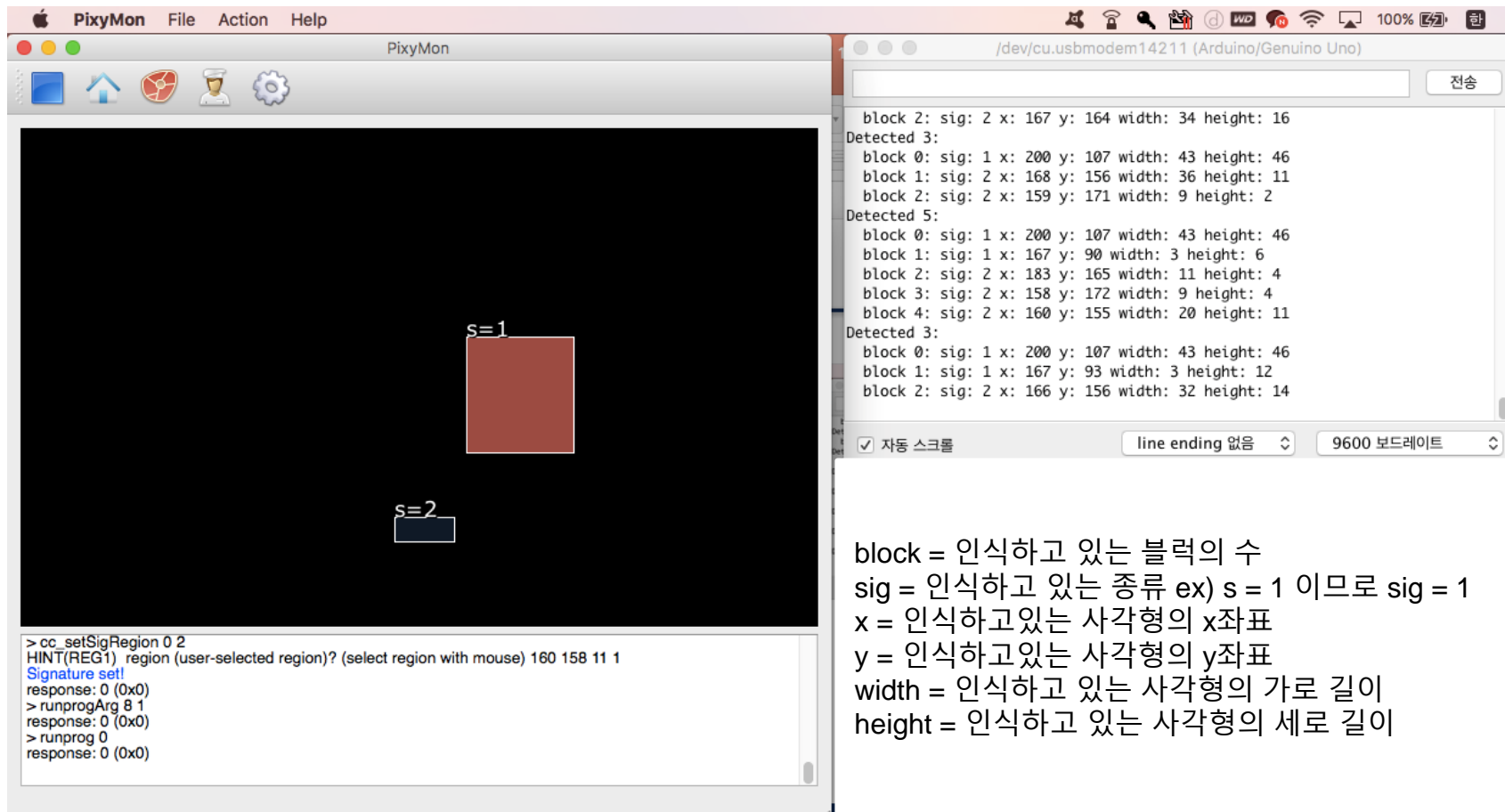
The right window, titled '/dev/cu.usbmodem14211 (Arduino/Genuino Uno)', shows a serial monitor with the following data:

```
block 0: sig: 1 x: 148 y: 109 width: 50 height: 47  
Detected 1:  
block 0: sig: 1 x: 148 y: 109 width: 49 height: 47  
Detected 1:  
block 0: sig: 1 x: 149 y: 109 width: 49 height: 46  
Detected 1:  
block 0: sig: 1 x: 149 y: 108 width: 49 height: 49  
Detected 1:  
block 0: sig: 1 x: 149 y: 108 width: 49 height: 48  
Detected 1:  
block 0: sig: 1 x: 150 y: 108 width: 49 height: 49  
Detected 1:  
block 0: sig: 1 x: 150 y: 108 width: 49 height: 48  
Detected 1:  
block 0: sig: 1 x: 150 y: 107 width: 49 height: 49
```

At the bottom of the serial monitor, there are checkboxes for '자동 스크롤' (checked) and 'line ending 없음', and a baud rate dropdown set to '9600 보드레이트'.

block = 인식하고 있는 블록의 수
sig = 인식하고 있는 종류 ex) s = 1 이므로 sig = 1
x = 인식하고있는 사각형의 x좌표
y = 인식하고있는 사각형의 y좌표
width = 인식하고 있는 사각형의 가로 길이
height = 인식하고 있는 사각형의 세로 길이

CMUCAM5 아두이노 신호 받기



The screenshot shows the PixyMon software interface on a Mac. The main window displays a black background with two detected blocks: a red square labeled 's=1' and a dark blue square labeled 's=2'. The bottom terminal window shows the following commands and responses:

```
> cc_setSigRegion 0 2
HINT(REG1) region (user-selected region)? (select region with mouse) 160 158 11 1
Signature set!
response: 0 (0x0)
> runprogArg 8 1
response: 0 (0x0)
> runprog 0
response: 0 (0x0)
```

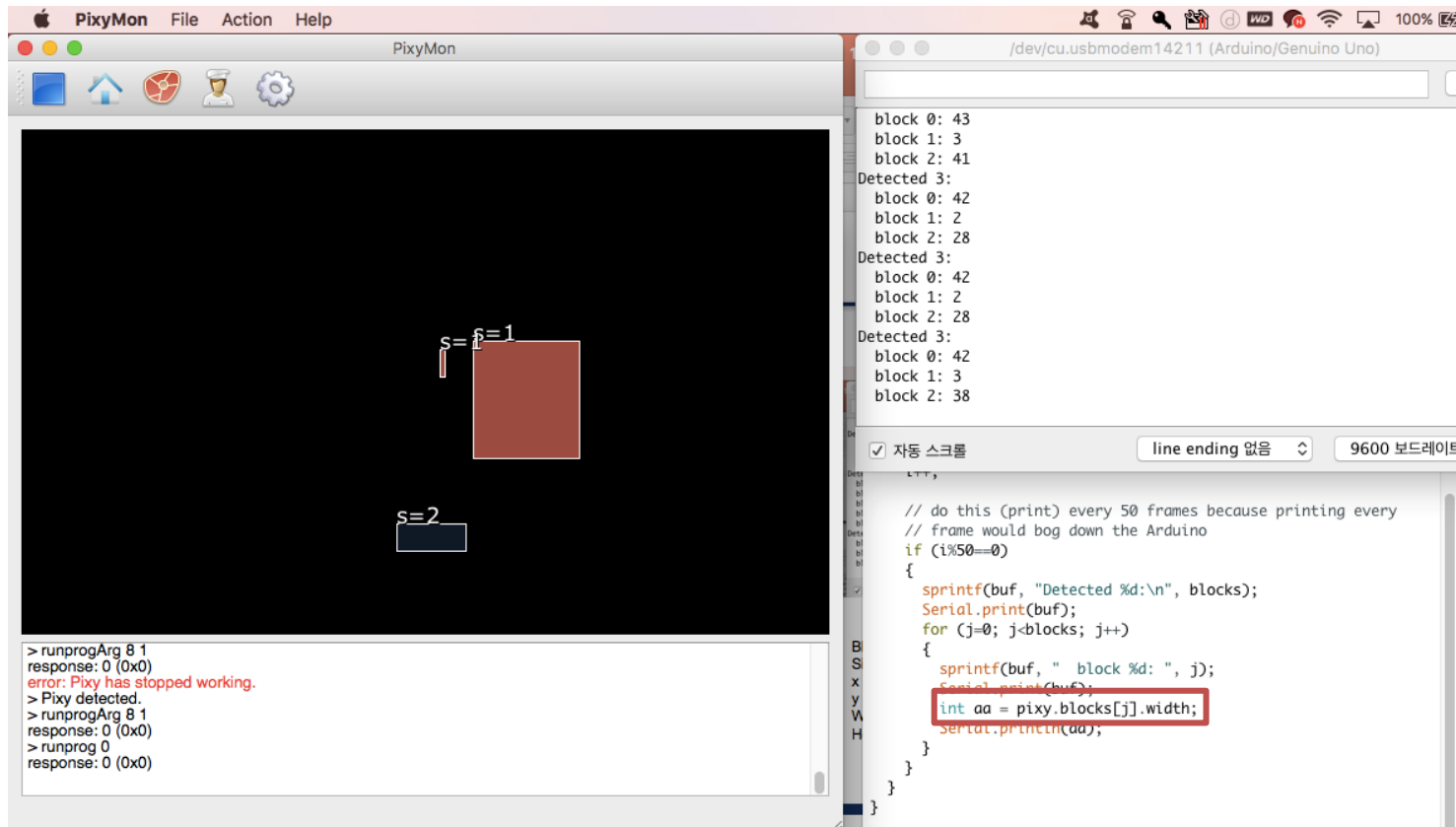
The right-hand window shows the detected blocks and their coordinates:

```
block 2: sig: 2 x: 167 y: 164 width: 34 height: 16
Detected 3:
block 0: sig: 1 x: 200 y: 107 width: 43 height: 46
block 1: sig: 2 x: 168 y: 156 width: 36 height: 11
block 2: sig: 2 x: 159 y: 171 width: 9 height: 2
Detected 5:
block 0: sig: 1 x: 200 y: 107 width: 43 height: 46
block 1: sig: 1 x: 167 y: 90 width: 3 height: 6
block 2: sig: 2 x: 183 y: 165 width: 11 height: 4
block 3: sig: 2 x: 158 y: 172 width: 9 height: 4
block 4: sig: 2 x: 160 y: 155 width: 20 height: 11
Detected 3:
block 0: sig: 1 x: 200 y: 107 width: 43 height: 46
block 1: sig: 1 x: 167 y: 93 width: 3 height: 12
block 2: sig: 2 x: 166 y: 156 width: 32 height: 14
```

block = 인식하고 있는 블록의 수
 sig = 인식하고 있는 종류 ex) s = 1 이므로 sig = 1
 x = 인식하고 있는 사각형의 x좌표
 y = 인식하고 있는 사각형의 y좌표
 width = 인식하고 있는 사각형의 가로 길이
 height = 인식하고 있는 사각형의 세로 길이

❑ CMUCAM5 아두이노 신호 받기

○ 아두이노에서 원하는 값만 받고 싶을 때

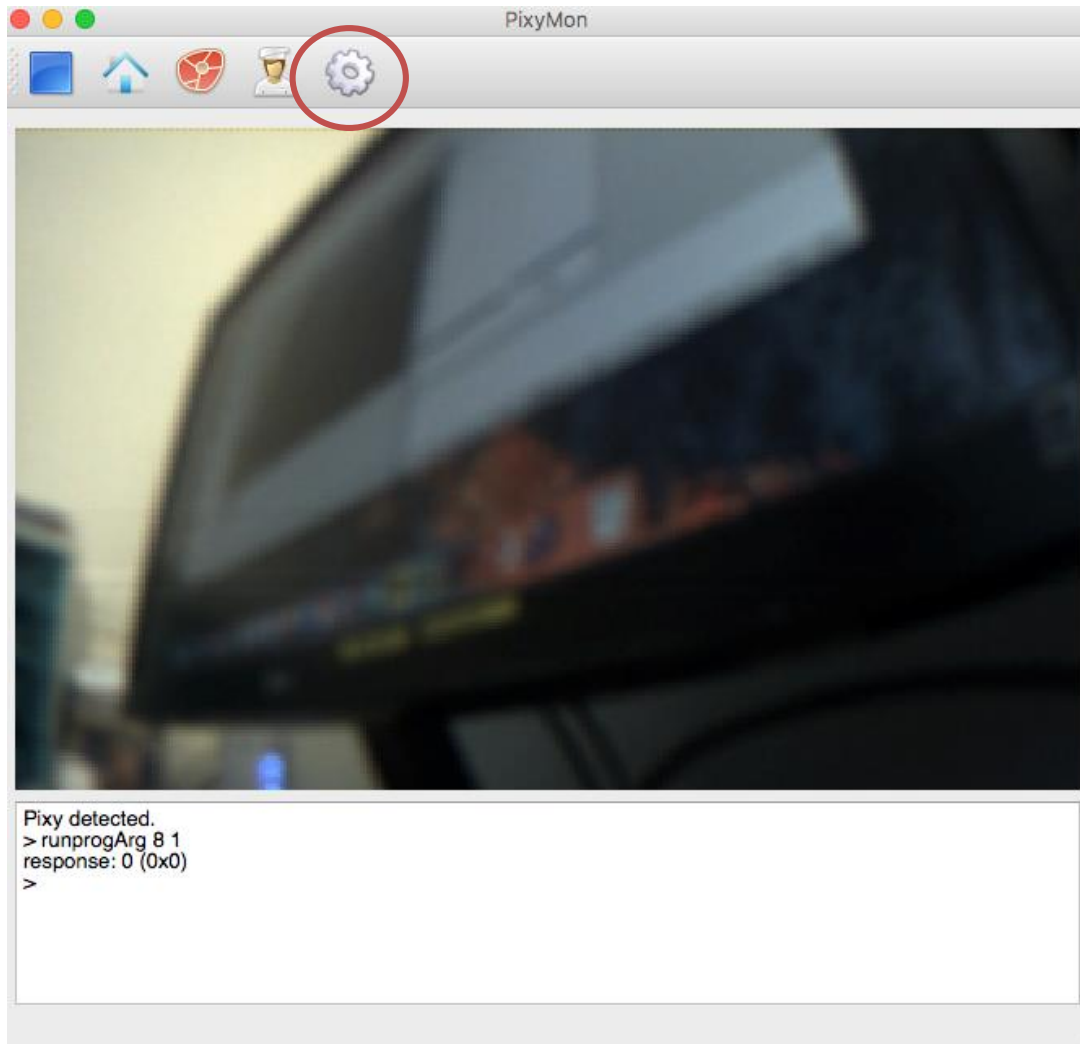


□ CMUCAM5 아두이노 신호 받기

○ 아두이노에서 원하는 값만 받고 싶을 때

- ❓ 받고자 하는 값의 이름을 plxy(카메라 변수 이름).block[블록 번호].(받고자 하는 것의 이름)넣고 다른 값에 저장한뒤에 출력하면 된다.(블록 번호는 출력되지 않는다.)
- ❓ x, y는 각 사각형의 중심의 위치를 나타낸다.
- ❓ 인식의 순서는 sig, x, y순서대로 작은 값부터 출력된다.

CMUCAM5



동그라미친 부분을
아이콘을 클릭하면 상세
설정이 가능하다.

Configure

Pixy Parameters (saved on Pixy) | PixyMon Parameters (saved on computer)

Signature Tuning | Signature Labels | Expert | Blocks | Interface | Camera | Servo | Pan/tilt Demo

Signature 1 range: 3.000000

Signature 2 range: 3.000000

Signature 3 range: 3.000000

Signature 4 range: 3.000000

Signature 5 range: 3.000000

Signature 6 range: 3.000000

Signature 7 range: 3.000000

Min brightness: 0.100000

Camera Brightness: 80

각 세팅된 색상값의 범위를 설정할 수 있다. (값을 늘리면 좀더 넓은 폭의 색이 인식되지만 노이즈가 발생할 가능성이 커진다. 반대로 줄이게 되면 노이즈는 줄어들지만 색상을 잘 인식 못하게 되는 현상이 일어날 수도 있다.)
적정 세팅을 하는 것이 중요하며 각 색상마다 따로따로 세팅이 가능하다.

Apply Cancel OK

