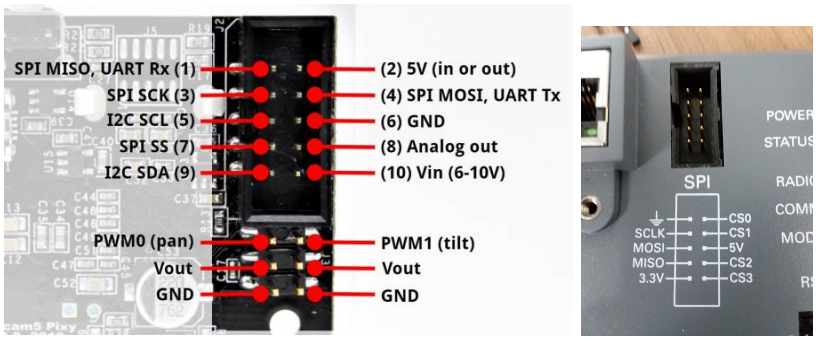
Pixy2 SPI Instructions

1.) Wiring: Connect the following pins from the Pixy to the roboRIO SPI Port

Pin 1	MISO
Pin 2	5V
Pin 3	SCK
Pin 4	MOSI
Pin 5	SCL
Pin 6	GND
Pin 7	CS0



Pixy2 Pinout on right with roboRIO on Left

2.) If starting a new project, two modifications need to be added to `build.gradle`.

a.) Under *repositories* add the following:

```
maven {
        url 'https://nexus.otake.pw/repository/maven-public/'
}
```

b.) Under *dependencies* add the following:

```
compile 'pw.otake.pseudoresonance:pixy2-java-api:1.3.4'
```

Note 1: For example code this has already been done

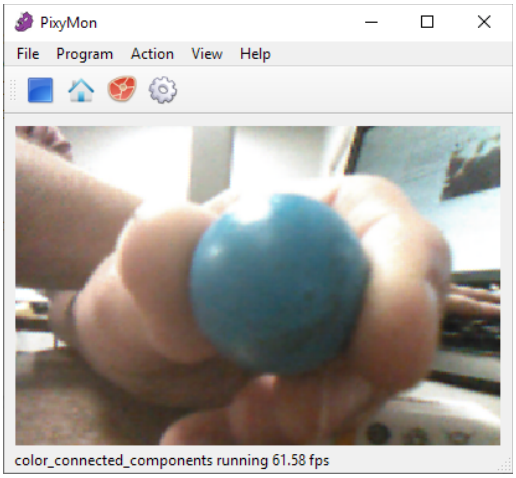
Note 2: Ensure that your first compile is with the computer online, as to allow Visual Studio to download the library. Once the library is downloaded, testing shows that offline compile is possible.

3.) Program the roboRIO. Either use the example project or base code of the example provided

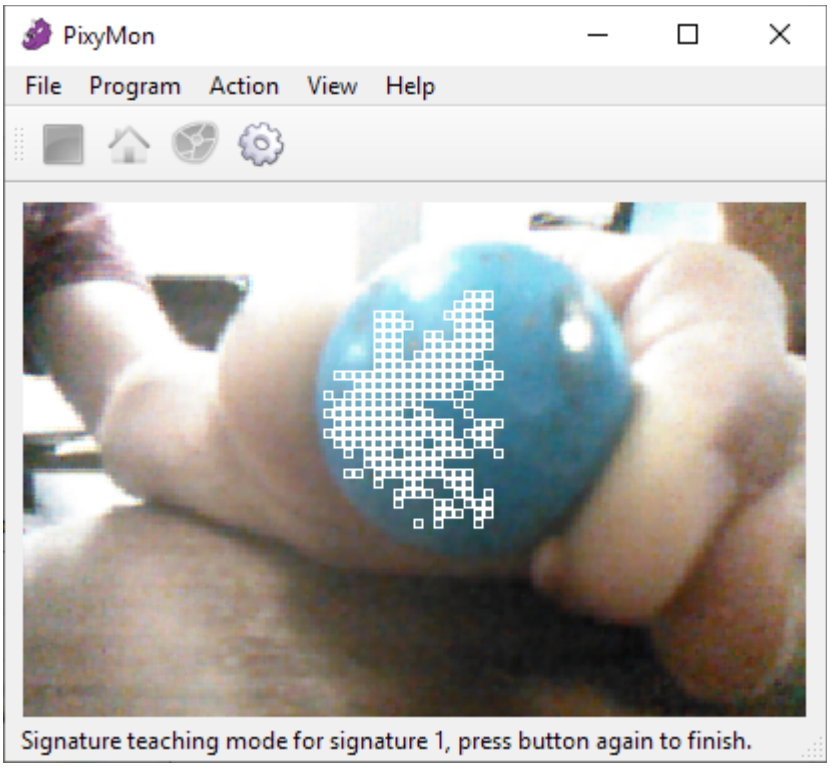
4.) Train the pixyCam

a. Open PixyMon v2

b. Place your target in the frame of the image



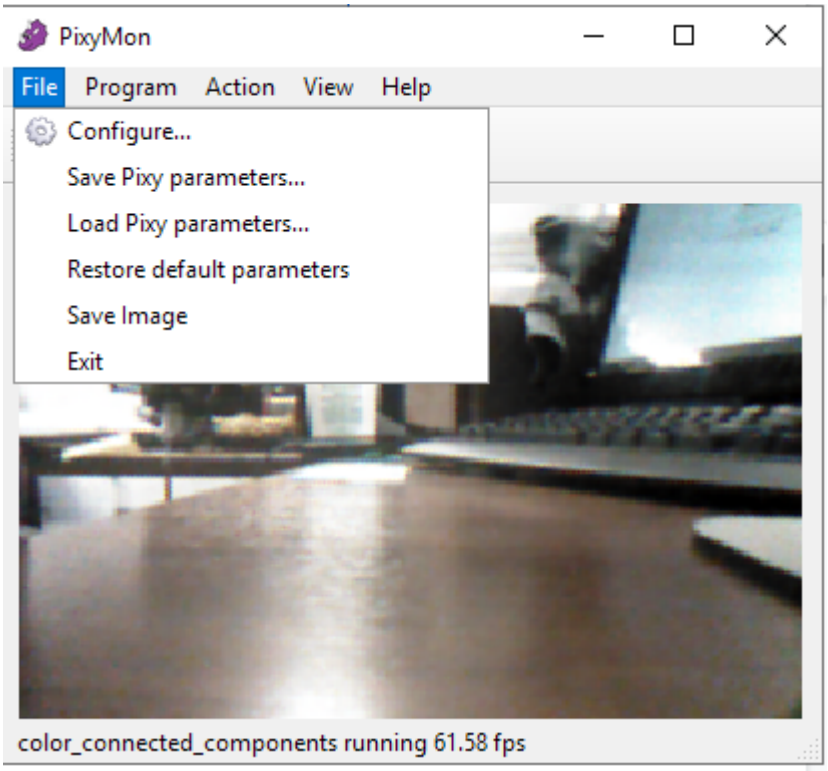
c. Hold the button for a second, and release. White squares should show up on the target



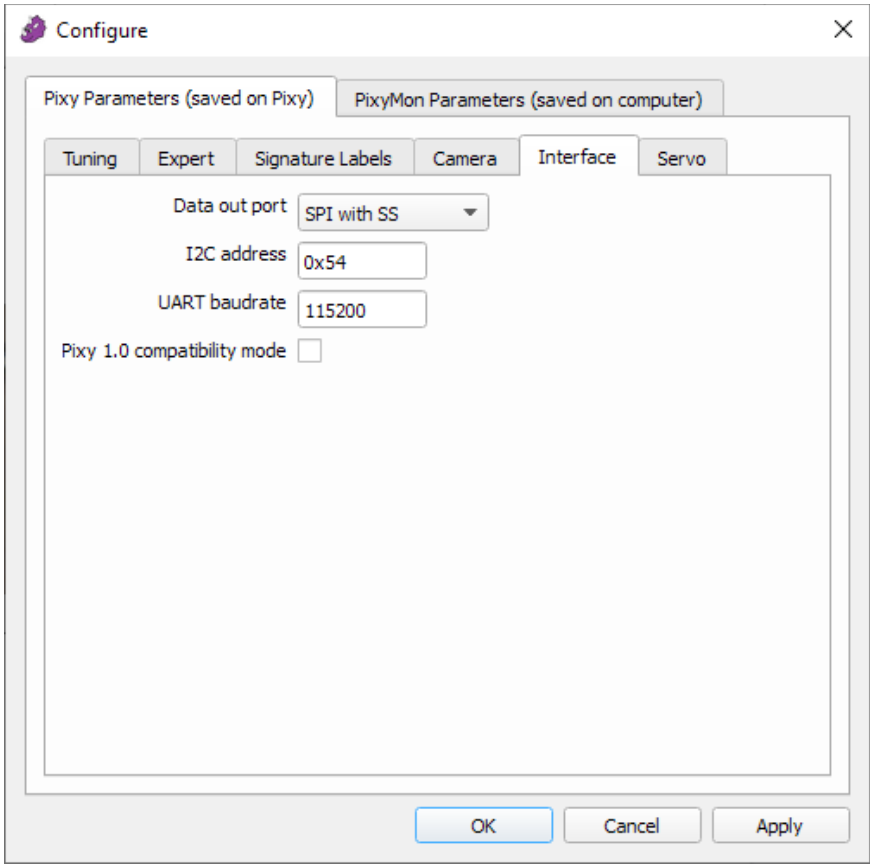
d. Press the button again. Your target is now trained

5.) Change the settings for the Pixy2 to be in SPI Mode

a.) Go to **File/Configure**



b.) Go to the **Interface** tab under **Pixy Parameters (Saved on Pixy)**



c.) Ensure that `Data out port` is “SPI with SS”

The roboRIO now should work with the Pixy2 over SPI.

Example Java: Pixy2 with SPI (ensure changes are made to build.gradle)

```
/*----------------------------------------------------------------------------*
/
/* Copyright (c) 2018 FIRST. All Rights Reserved.
*/
/* Open Source Software - may be modified and shared by FRC teams. The code
*/
/* must be accompanied by the FIRST BSD license file in the root directory of
*/
/* the project.
*/
/*----------------------------------------------------------------------------*
/

package frc.robot;

import java.util.ArrayList;

import edu.wpi.first.wpilibj.TimedRobot;
import edu.wpi.first.wpilibj.smartdashboard.SmartDashboard;
import io.github.pseudoresonance.pixy2api.*;
import io.github.pseudoresonance.pixy2api.Pixy2CCC.Block;

/**
* PIXY SPI EXAMPLE
* A few notes
* 1.) Two things need to be added to the build.gradle file. See the attached
tutorial document
* 2.) This code is for using Chip Select 0 (CS0) and the onboard SPI Port
* 3.) If you are using SPI mode, go into PixyMon and set the control mode to
SPI
*/
```

```
public class Robot extends TimedRobot {
 /**
  * This function is run when the robot is first started up and should be used
  * for any initialization code.
  */
 private Pixy2 pixycam;
 boolean isCamera = false;
 //private SPILink spi;
 int state=-1;
 @Override
 public void robotInit() {
   pixycam = Pixy2.createInstance(Pixy2.LinkType.SPI);
 }

 @Override
 public void teleopPeriodic() {
   if(!isCamera)
     state = pixycam.init(1); // if no camera present, try to initialize
   isCamera = state>=0;

   SmartDashboard.putBoolean("Camera", isCamera);   //publish if we are
connected
   pixycam.getCCC().getBlocks(false,255,255); //run getBlocks with arguments to
have the camera
                                              //acquire target data
   ArrayList<Block> blocks = pixycam.getCCC().getBlocks(); //assign the data to
an ArrayList for convinience
   if(blocks.size() > 0)
   {
     double xcoord = blocks.get(0).getX();       // x position of the largest
target
     double ycoord = blocks.get(0).getY();       // y position of the largest
target
     String data   = blocks.get(0).toString();   // string containing target
info
     SmartDashboard.putBoolean("present", true); // show there is a target
present
     SmartDashboard.putNumber("Xccord",xcoord);
     SmartDashboard.putNumber("Ycoord", ycoord);
     SmartDashboard.putString("Data", data );
   }
   else
     SmartDashboard.putBoolean("present", false);
   SmartDashboard.putNumber("size", blocks.size()); //push to dashboard how
```

```
 many targets are detected
  }


}
```