

Connecting Pixy to mBot

By Wilmer Arellano

Updated 10-24-2018

[Download the Examples Here](#)



- You are free:
 - to Share — to copy, distribute and transmit the work
- Under the following conditions:
 - Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
 - Noncommercial — You may not use this work for commercial purposes.
 - No Derivative Works — You may not alter, transform, or build upon this work.



- With the understanding that:
 - Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
 - Other Rights — In no way are any of the following rights affected by the license:
 - Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
 - The author's moral rights;
 - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
 - Notice — For any reuse or distribution, you must make clear to others the license terms of this work.

Start Here

- https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:pixy_regular_quick_start
- Install PixyMon

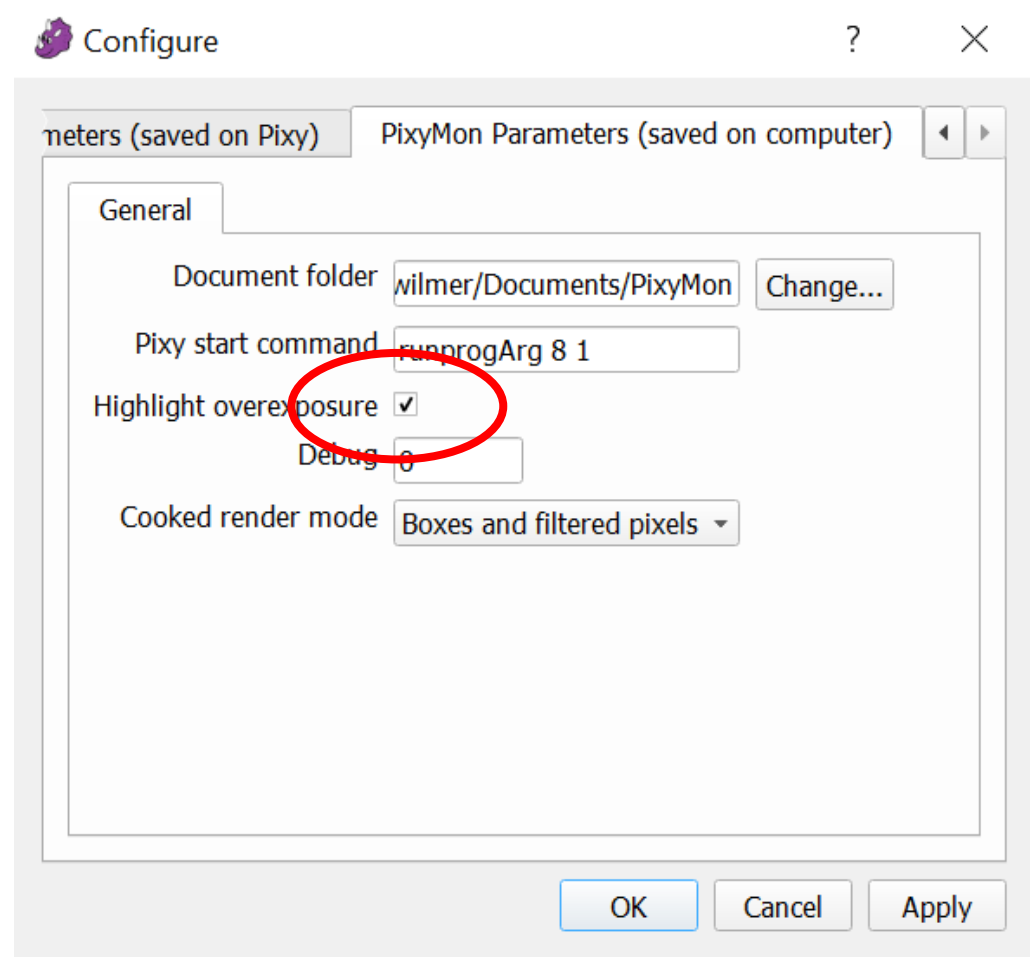
Next

- Go to:
- https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:teach_pixy_an_object_2
- Watch the video.
- Learn about the two methods to capture signature.
- Read the content about white balance and Signature Tuning.

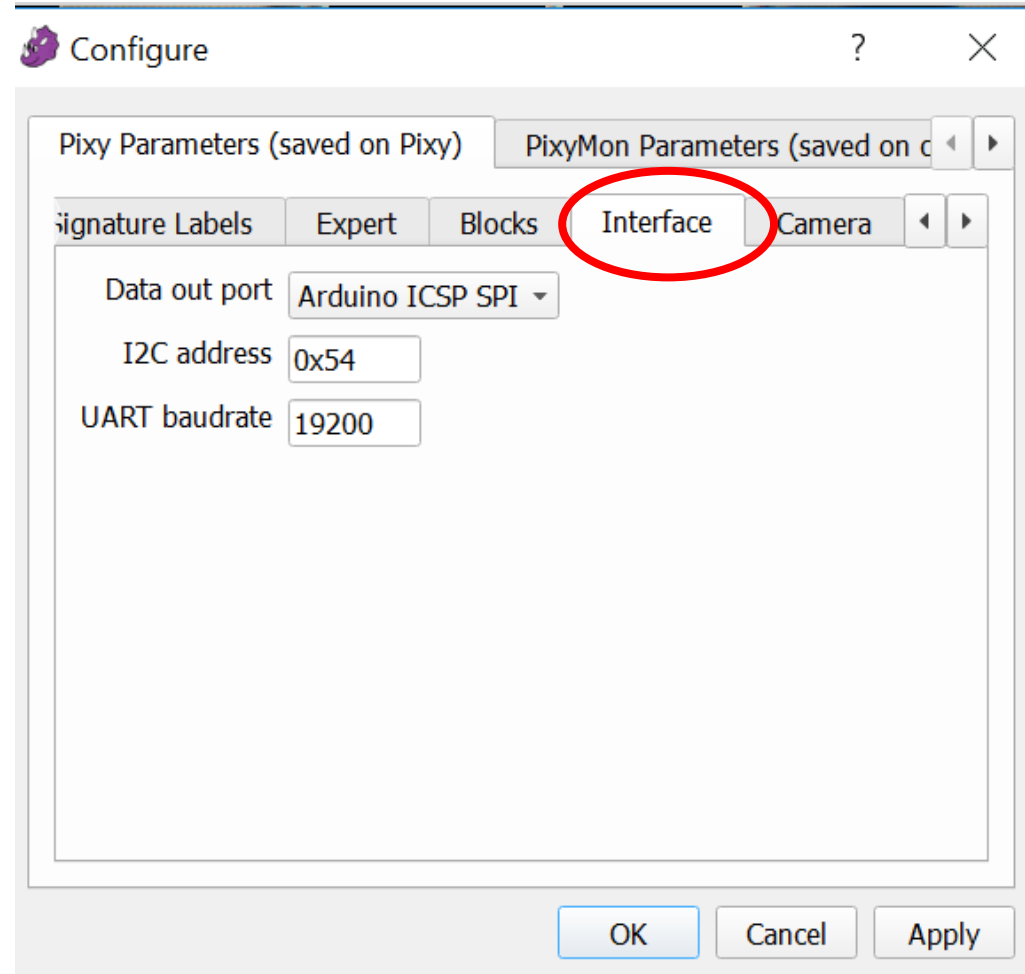
Improving Detection Accuracy

- Learn about ways to improve signature capture at:
- https://docs.pixycam.com/wiki/doku.php?id=wiki:v1:some_tips_on_generating_color_signatures_2
- Read about overexposure highlighting, minimum brightness, and signature teach threshold.

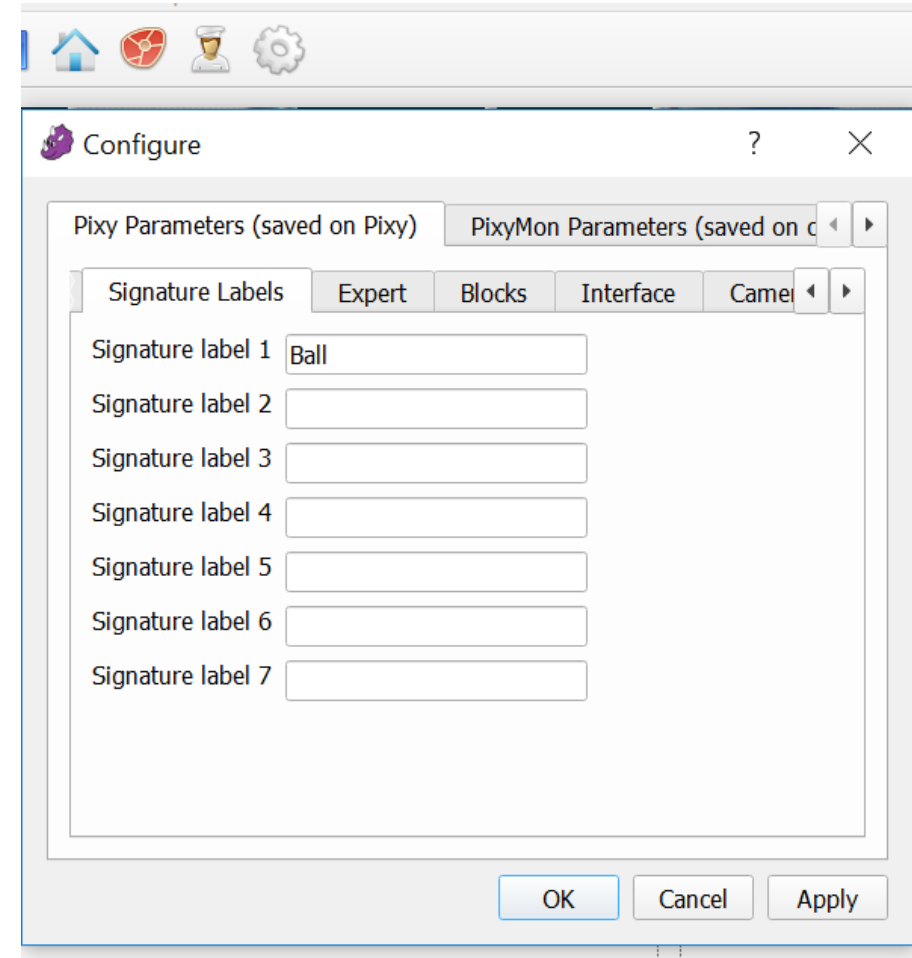
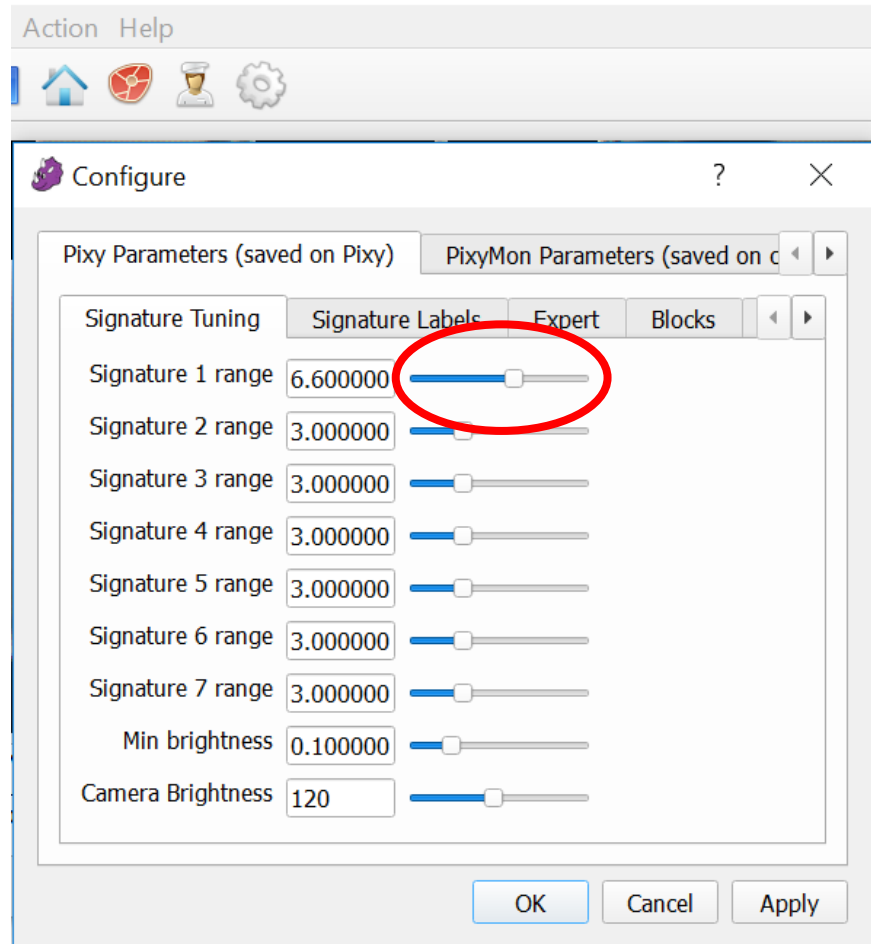
Configuration Details



Configuration Details



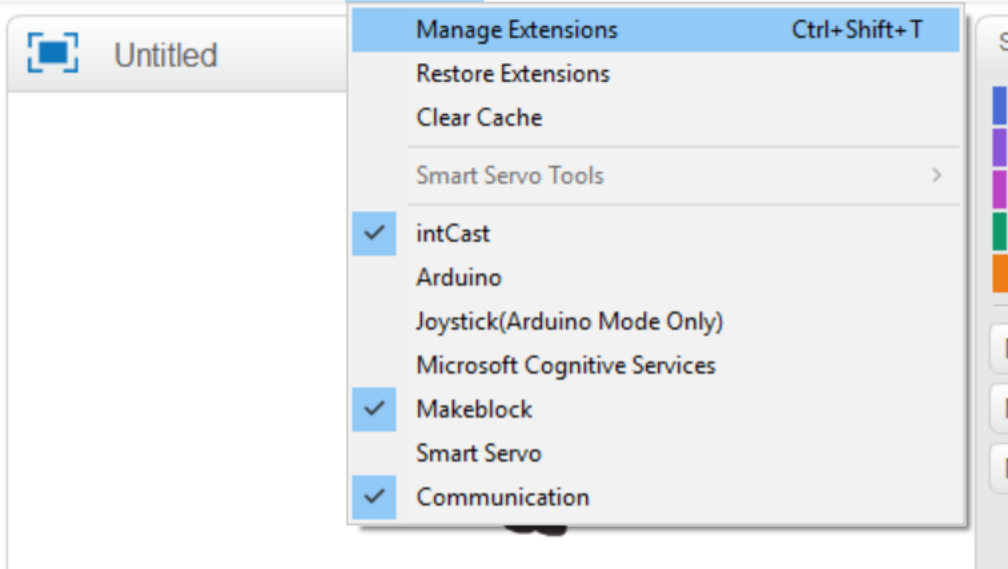
Configuration Details



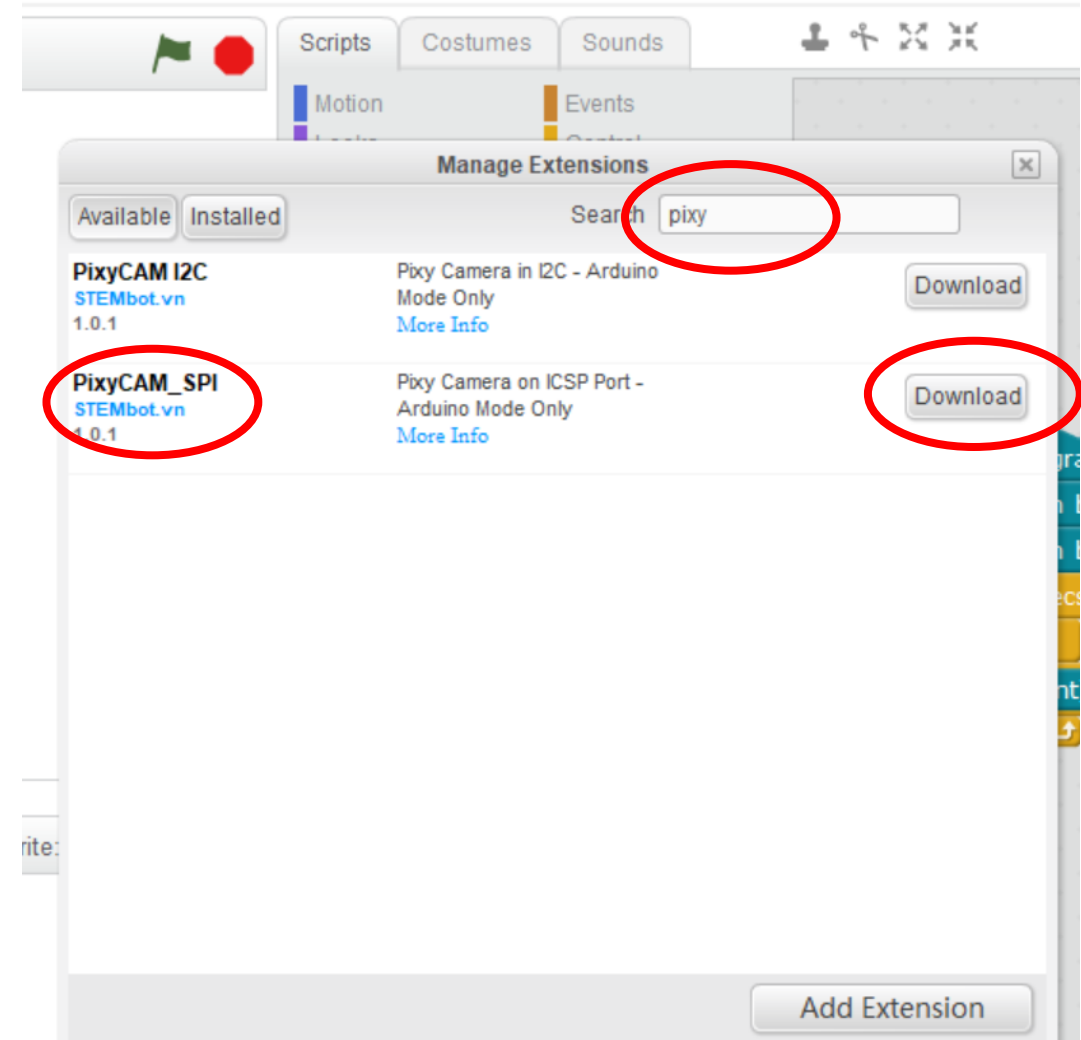
Installing Pixy

mBlock - Based On Scratch From the MIT Media Lab(v3.4.11) - Disconnected - Not saved

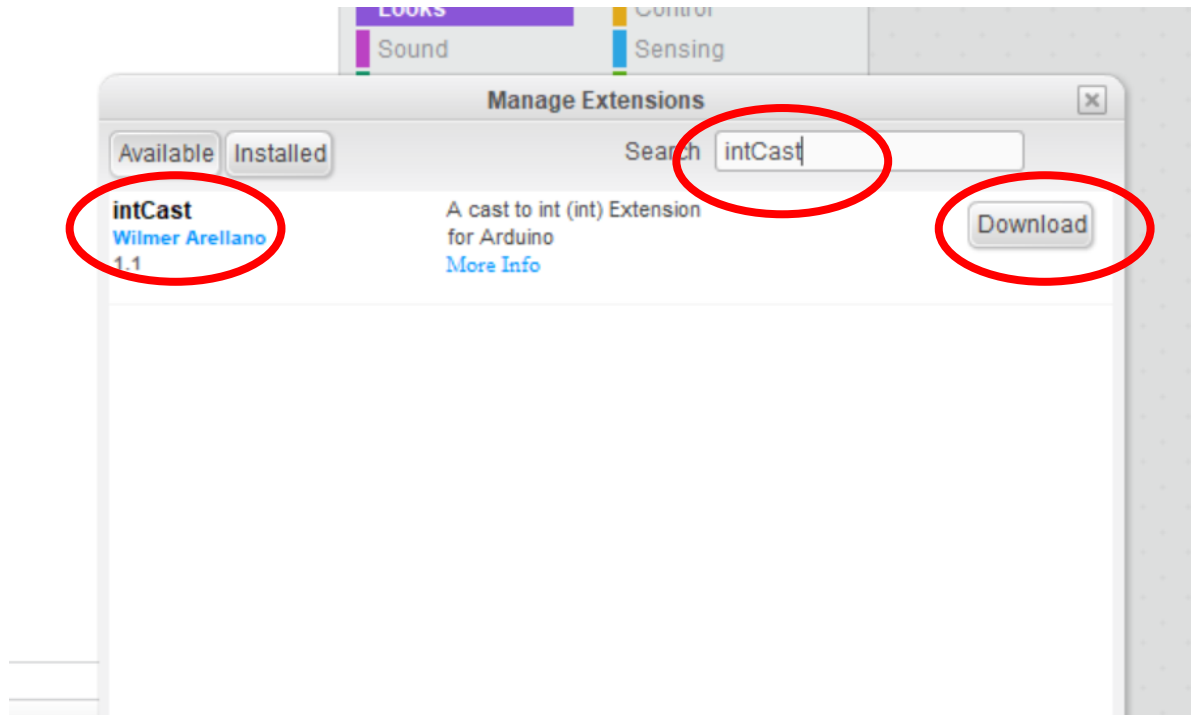
File Edit Connect Boards Extensions Language Help



Disconnected - Not saved

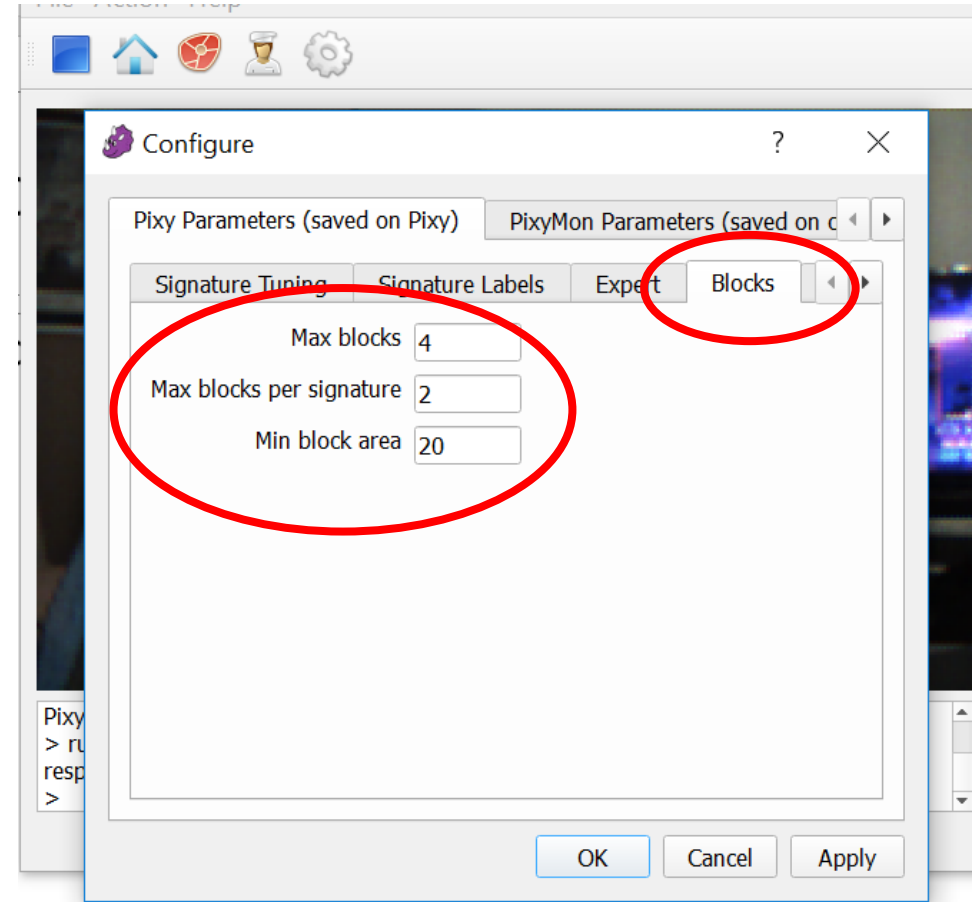


Installing Int Casting

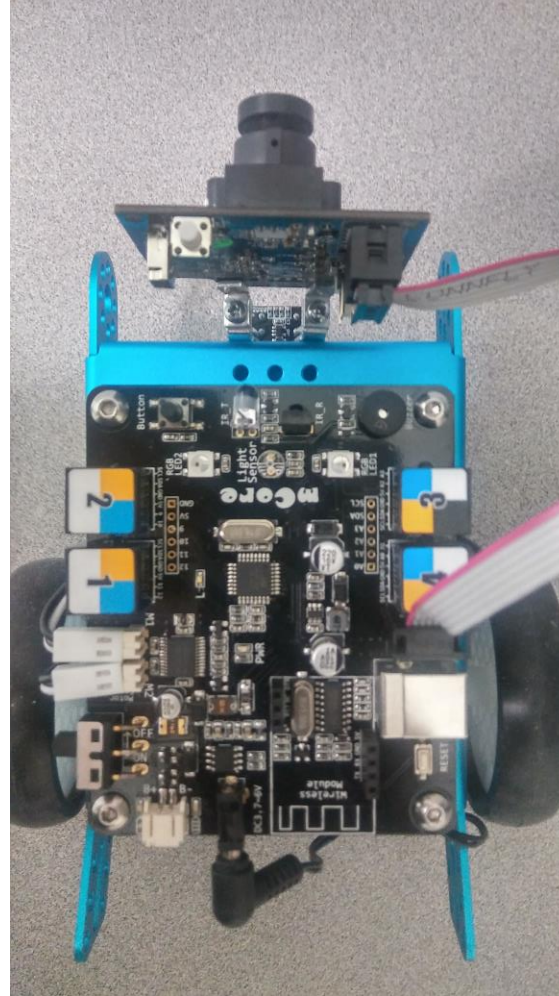


Tweaking the Camera Output

- Due to lighting conditions, a single object may appear as several objects.
- By reducing the number of blocks the pixy detects, object detection can be improved.
- The Pixy gives priority at reporting larger objects.
- Set this values according to your needs.

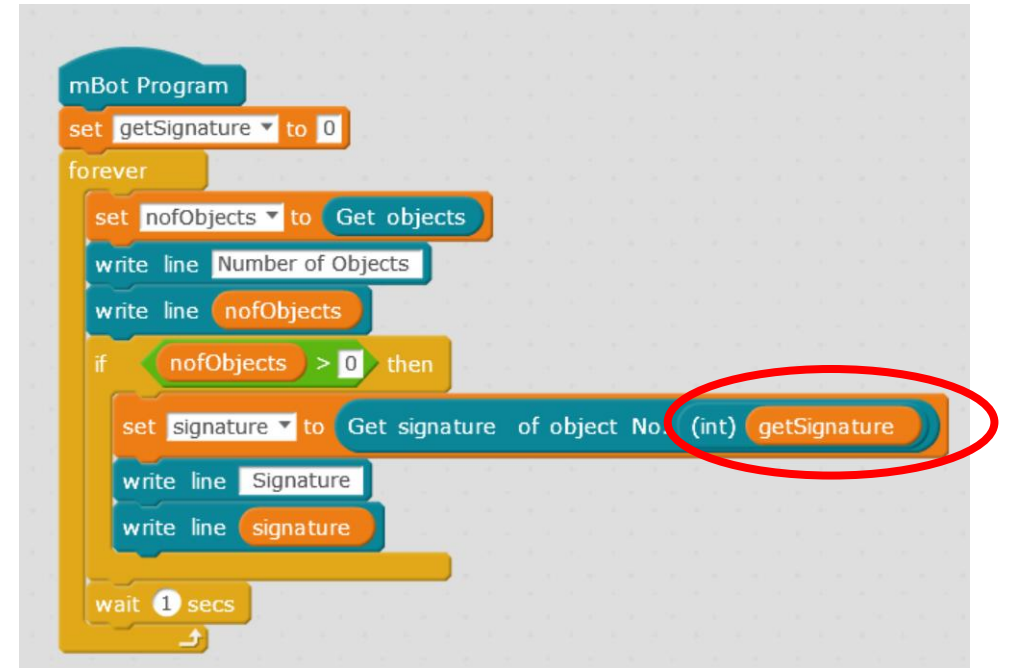
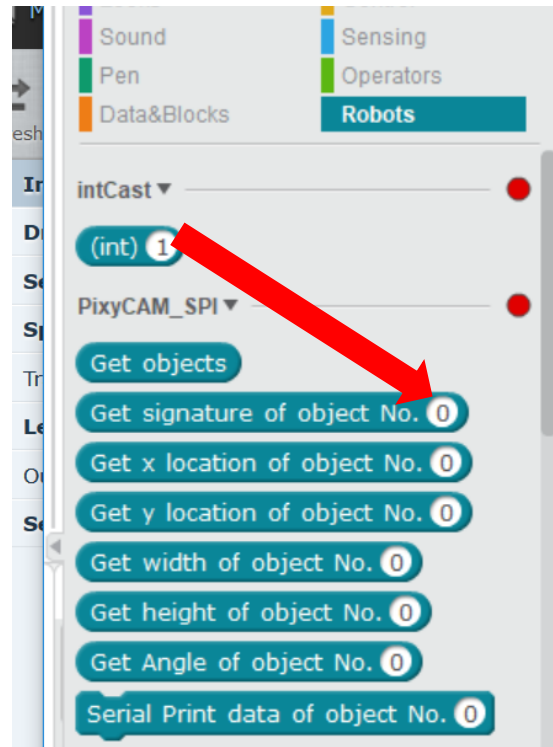


Connecting Pixy to mBot



Example 1

- If one or more objects are detected, this program prints the signature of object 1.
- The arguments for the Pixy functions must be integers. Scratch works with doubles. The function (int) is needed to do the type casting.



Range of Functions Values

- **Get signature of object:**
The signature number of the detected object (1-7 for normal signatures)
- **Get x location of object:**
The x location of the center of the detected object (0 to 319)
- **Get Y location of object:**
The y location of the center of the detected object (0 to 199)
- **Get width of object:**
The width of the detected object (1 to 320)
- **Get height of object:**
The height of the detected object (1 to 200)
- **Get angle of object:**
The angle of the object detected object if the detected object is a color code.
- **Serial print data of object:**
A member function that prints the detected object information to the serial port

Hello World Arduino Style

- The program on the right is a popular Pixy “Hello World” demo with a minor modification.
- Open the Arduino IDE from within mBlock. Copy the program on the right and paste it into the IDE.
- Upload to mBot and run.
- Program will detect blocks and print the associated data.
- The instruction `if (i % 10 == 0)` limits the printing to 1 every 10 cycles.

```
#include <Arduino.h>
#include <Wire.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include "Pixy.h"

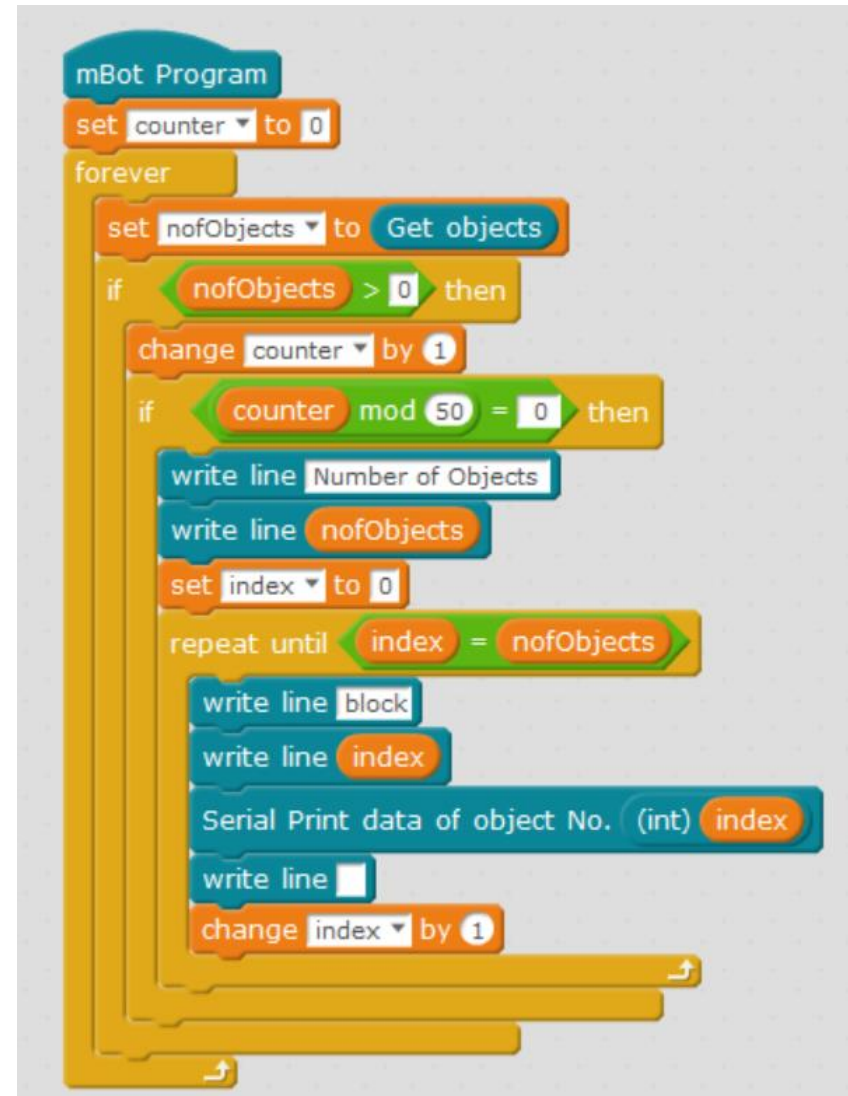
Pixy ;

void setup(){
  Serial.begin(115200);
  Serial.print("Starting...\n");
  pixy.init();
}

void loop(){
  static int i = 0;
  int j;
  uint16_t blocks;
  char buf[32];
  blocks = pixy.getBlocks();
  if (blocks) {
    i++;
    if (i % 10 == 0) {
      sprintf(buf, "Detected %d: \n", blocks);
      Serial.println(buf);
      for (j = 0; j < blocks; j++) {
        sprintf(buf, "block %d: ", j);
        Serial.print(buf);
        pixy.blocks[j].print();
        Serial.println();
      }
    }
  }
}
```


Hello World mBot Style v1

- The program on the right is an mBlock version of the “Hello World” program in the previous slide.
- The program has the original 50 cycles update period.

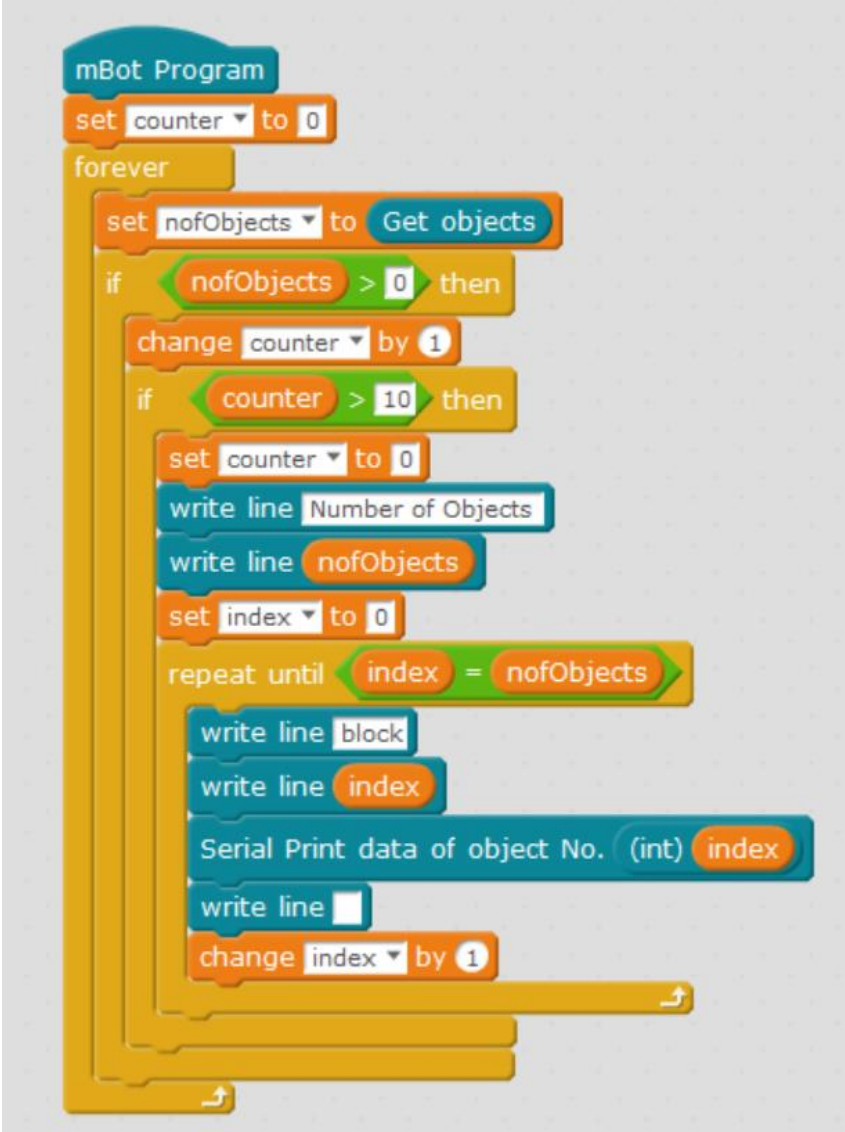


```
mBot Program
set counter to 0
forever
  set nofObjects to Get objects
  if nofObjects > 0 then
    change counter by 1
    if counter mod 50 = 0 then
      write line Number of Objects
      write line nofObjects
      set index to 0
      repeat until index = nofObjects
        write line block
        write line index
        Serial Print data of object No. (int) index
        write line 
        change index by 1
```

The image shows a screenshot of the mBlock programming environment. The code is written in a block-based style. It starts with a blue 'mBot Program' block, followed by an orange 'set counter to 0' block. A yellow 'forever' loop block contains several nested blocks: an orange 'set nofObjects to Get objects' block, a green 'if nofObjects > 0 then' block, an orange 'change counter by 1' block, a green 'if counter mod 50 = 0 then' block, a blue 'write line Number of Objects' block, a blue 'write line nofObjects' block, an orange 'set index to 0' block, a green 'repeat until index = nofObjects' block, a blue 'write line block' block, a blue 'write line index' block, a blue 'Serial Print data of object No. (int) index' block, a blue 'write line ' block, and an orange 'change index by 1' block. The code is visually organized with indentation to show the nested structure of the loops and conditionals.

Hello World mBot Style v2

- The program on the right is a modification from the one in the previous slide.
- Prints information once every 11 frames.
- The modulo operation was substituted by “greater than” check.

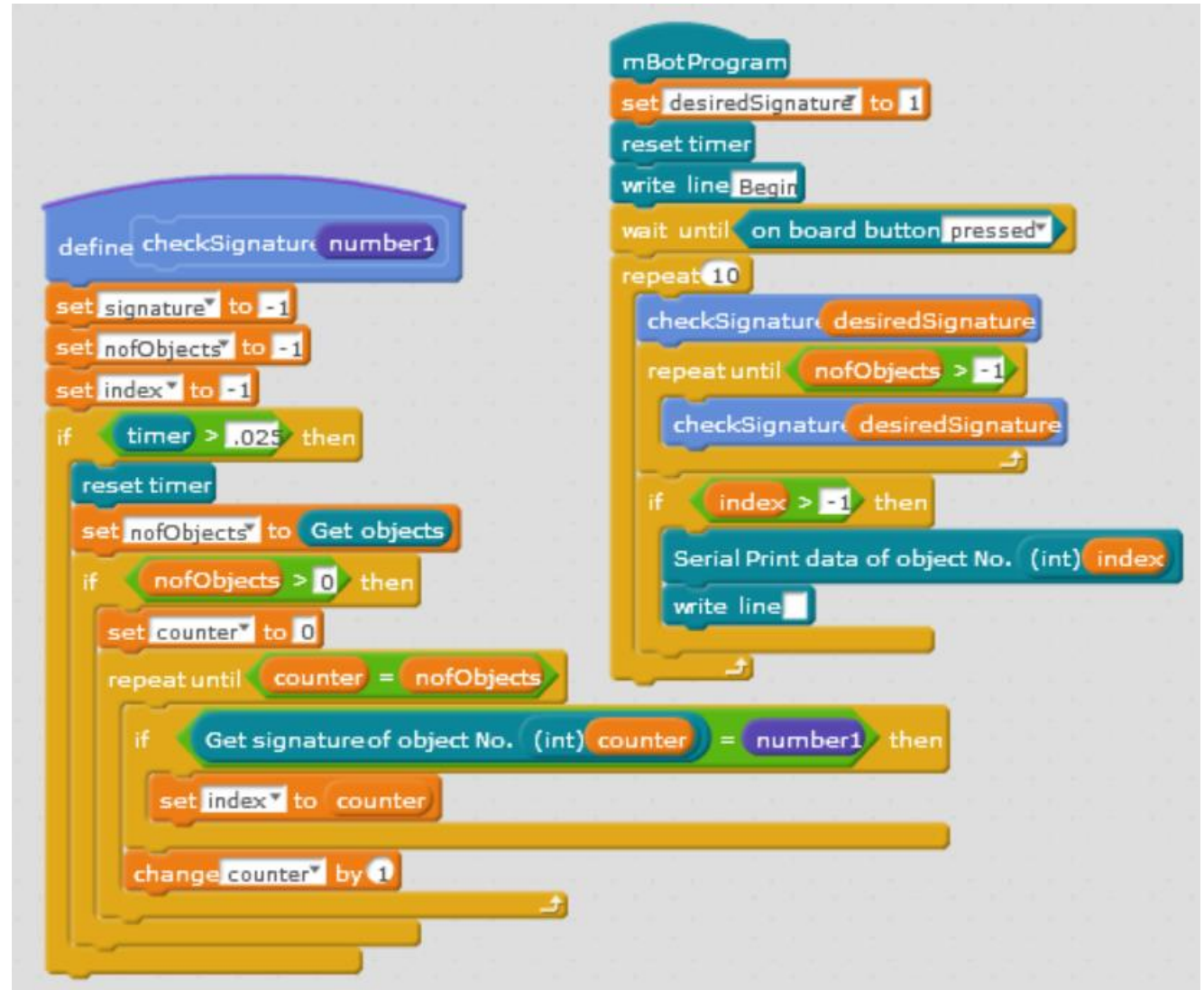


```
mBot Program
set counter to 0
forever
  set nofObjects to Get objects
  if nofObjects > 0 then
    change counter by 1
    if counter > 10 then
      set counter to 0
      write line Number of Objects
      write line nofObjects
      set index to 0
      repeat until index = nofObjects
        write line block
        write line index
        Serial Print data of object No. (int) index
        write line 
        change index by 1
```

The image shows a Scratch-style code editor for an mBot program. The code is written in a block-based language. It starts with a blue 'mBot Program' block, followed by an orange 'set counter to 0' block. A yellow 'forever' loop block contains several nested blocks: an orange 'set nofObjects to Get objects' block, a yellow 'if nofObjects > 0 then' block containing an orange 'change counter by 1' block and another yellow 'if counter > 10 then' block. The inner 'if' block contains an orange 'set counter to 0' block, a blue 'write line Number of Objects' block, a blue 'write line nofObjects' block, an orange 'set index to 0' block, a yellow 'repeat until index = nofObjects' block, and a blue 'change index by 1' block. The 'repeat until' block contains a blue 'write line block' block, a blue 'write line index' block, a blue 'Serial Print data of object No. (int) index' block, and a blue 'write line ' block. The code ends with a yellow arrow pointing to the right, indicating the end of the program.

Getting Blocks too often Version 1

- Calling the function Get Objects too often may falsely return 0 objects when objects are present.
- The frame rate of Pixy is 20 ms. You should not call twice in the same period.
- The program on the right returns -1 when you call the function again in less than 20 ms.



```
mBotProgram
set desiredSignature to 1
reset timer
write line Begin
wait until on board button pressed
repeat 10
  checkSignature desiredSignature
  repeat until nofObjects > -1
  checkSignature desiredSignature
  if index > -1 then
    Serial Print data of object No. (int) index
    write line

define checkSignature number1
set signature to -1
set nofObjects to -1
set index to -1
if timer > .025 then
  reset timer
  set nofObjects to Get objects
  if nofObjects > 0 then
    set counter to 0
    repeat until counter = nofObjects
    if Get signature of object No. (int) counter = number1 then
      set index to counter
      change counter by 1
```

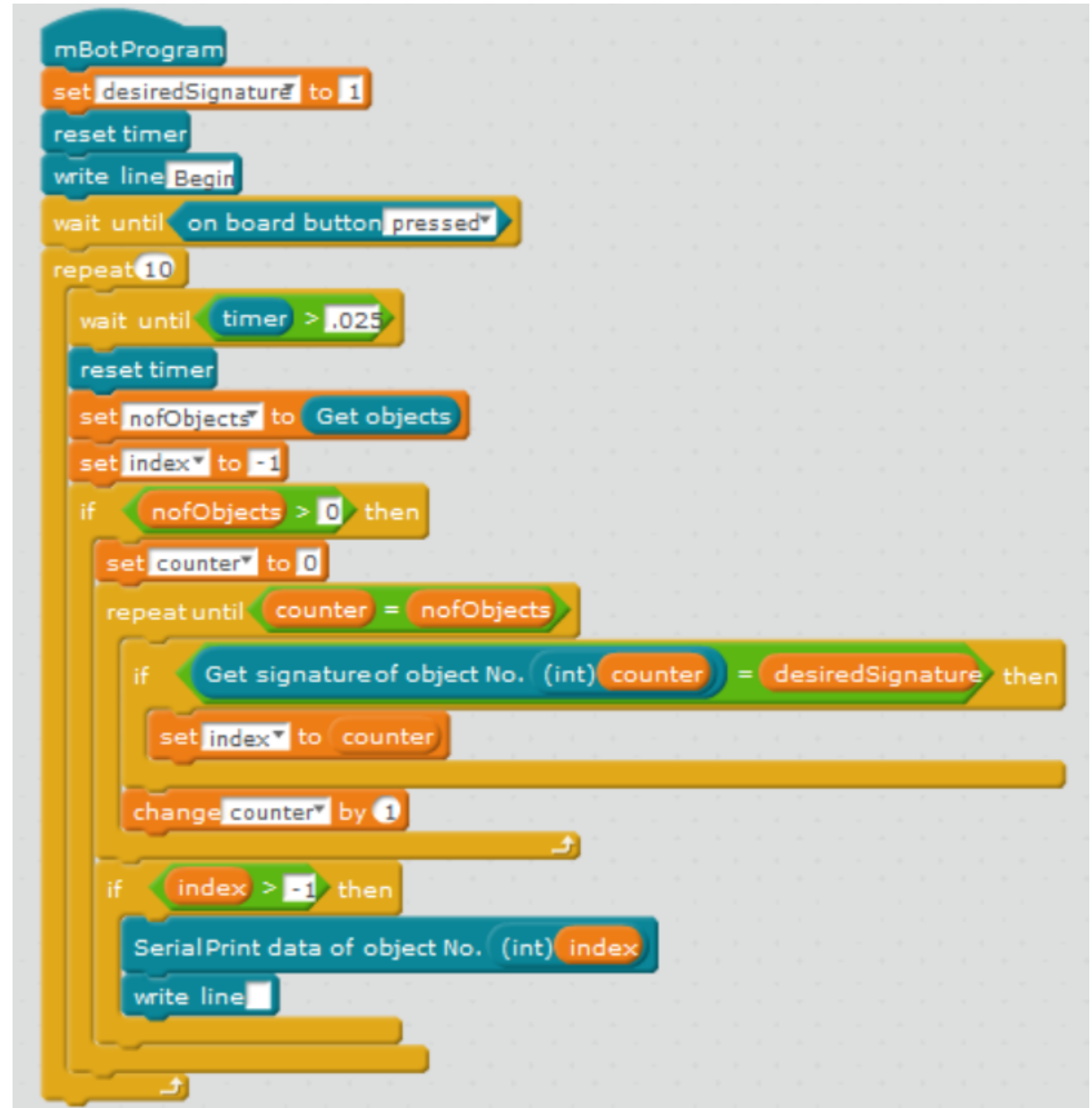
Getting Blocks too often Version 1

- With an object with signature 1 in front of the camera, Run the program and watch the output in the serial monitor.
- To re-run, please press reset button first.
- Change .02 to .001 (1 ms) and repeat.
- What is the difference?

```
mBotProgram
set desiredSignature to 1
reset timer
write line Begin
wait until on board button pressed
repeat 10
  checkSignature desiredSignature
  repeat until nofObjects > -1
  checkSignature desiredSignature
  if index > -1 then
    Serial Print data of object No. (int) index
    write line
define checkSignature number1
set signature to -1
set nofObjects to -1
set index to -1
if timer > .025 then
  reset timer
  set nofObjects to Get objects
  if nofObjects > 0 then
    set counter to 0
    repeat until counter = nofObjects
    if Get signature of object No. (int) counter = number1 then
      set index to counter
      change counter by 1
```

Getting Blocks too often Version 2

- Another version of this program without using Block functions.



```
mBotProgram
set desiredSignature to 1
reset timer
write line Begin
wait until on board button pressed
repeat 10
  wait until timer > .025
  reset timer
  set nofObjects to Get objects
  set index to -1
  if nofObjects > 0 then
    set counter to 0
    repeat until counter = nofObjects
      if Get signature of object No. (int) counter = desiredSignature then
        set index to counter
        change counter by 1
    if index > -1 then
      SerialPrint data of object No. (int) index
      write line
```

Example 2

- Everything together
- In this case we set, MaxBlocks 2, MaxBlocks per signature 1.
- We use mBot to track an object with signature 1.
- Please note that the Arduino is detecting one object, but the index for that object is 0
- See video at: <https://youtu.be/pmbLD0JBqyw>

```
mBot Program
reset timer
run forward at speed 75
wait 0.02 secs
run forward at speed 0
wait until on board button pressed
forever
  wait until timer > .02
  reset timer
  set nofObjects to Get objects
  if nofObjects < 1 then
    run forward at speed 0
  else
    set xPos to Get x location of object No. 0
    if 120 < xPos and xPos < 200 then
      run forward at speed 0
    else
      if xPos < 120 then
        turn left at speed 80
      else
        turn right at speed 80
```

Example 3

- The program in next slide will sequentially track signature 1 and then get away from signature 2 by turning left.
- In this case we set, MaxBlocks 2, MaxBlocks per signature 1, Min block area 40.
- After pressing the button, the robot will track signature 1 until it has been centered or away for more than 3 seconds, after that a tone is played.
- Then the robot will get away from signature 2, until it is not visible for 3 seconds, a different tone is played and the program stops.
- A video demonstrating this program can be found at:
<https://youtu.be/9Ury4ilYpew>

```

mBot Program
run forward at speed 75
wait 0.02 secs
run forward at speed 0
wait until on board button pressed
reset timer
trackSignature1
play tone on note C4 beat Double
reset timer
avoidSignature2
play tone on note A4 beat Double

```

```

define checkIndex
  This function finds th...
  set index to -1
  set counter to 0
  repeat until counter = noOfObjects
  if Get signature of object No. (int) counter = signature then
    set index to counter
  change counter by 1

```

```

define trackSignature1
  set signature to 1
  set stop to 0
  repeat until stop = 1
  wait until timer > .025
  set noOfObjects to Get objects
  if noOfObjects < 1 then
    run forward at speed 0
  else
    checkIndex
    if index > -1 then
      set xPos to Get x location of object No. (int) index
      if 120 < xPos and xPos < 200 then
        run forward at speed 0
      else
        if xPos < 120 then
          turn left at speed 80
          reset timer
        else
          turn right at speed 80
          reset timer
    if timer > 3 then
      set stop to 1

```

```

define avoidSignature2
  set signature to 2
  set stop to 0
  repeat until stop = 1
  wait until timer > .025
  set noOfObjects to Get objects
  if noOfObjects < 1 then
    run forward at speed 0
  else
    checkIndex
    if index > -1 then
      turn left at speed 80
      reset timer
    if timer > 3 then
      set stop to 1

```



Example 4

- The program in next slide will rotate to find signature 1 and then move towards it.
- In this case we set, MaxBlocks 2, MaxBlocks per signature 1, Min block area 40.
- After pressing the button, the robot will rotate to find signature 1 until it has been centered then moves towards signature 1.
- The “y” value of the object is use to stop the robot, it must be adjusted for the particular camera height.
- The scoreGoal function must be developed, it may be very similar to the getBall function.
- A video demonstrating this program can be found at:
<https://www.youtube.com/watch?v=GdPtPGIU5Fk>

```

mBot Program
run forward at speed 75
wait 0.02 secs
run forward at speed 0
wait until on board button pressed
reset timer
getBall
play tone on note C4 beat Double
reset timer
scoreGoal
play tone on note A4 beat Double

```

```

define checkIndex
  This function finds th...
  set index to -1
  set counter to 0
  set nofObjects to Get objects
  if nofObjects > 0 then
    repeat until counter = nofObjects
      if Get signature of object No. (int) counter = signature then
        set index to counter
      change counter by 1

```

```

define getBall
  set signature to 1
  set stop to 0
  repeat until stop = 1
    wait until timer > .02
    checkIndex
    if index = -1 then
      turn right at speed 60
      reset timer
    else
      set xPos to Get x location of object No. (int) index
      if 100 < xPos and xPos < 220 then
        run forward at speed 80
        reset timer
        wait 0.01 secs
      else
        if xPos < 100 then
          turn left at speed 75
          reset timer
        else
          turn right at speed 75
          reset timer
      if Get y location of object No. (int) index > 185 then
        run forward at speed 0
        set stop to 1

```

```

define scoreGoal
  set signature to 2
  set stop to 0
  repeat until stop = 1
    if timer > 3 then
      set stop to 1

```



x: -3
y: 20