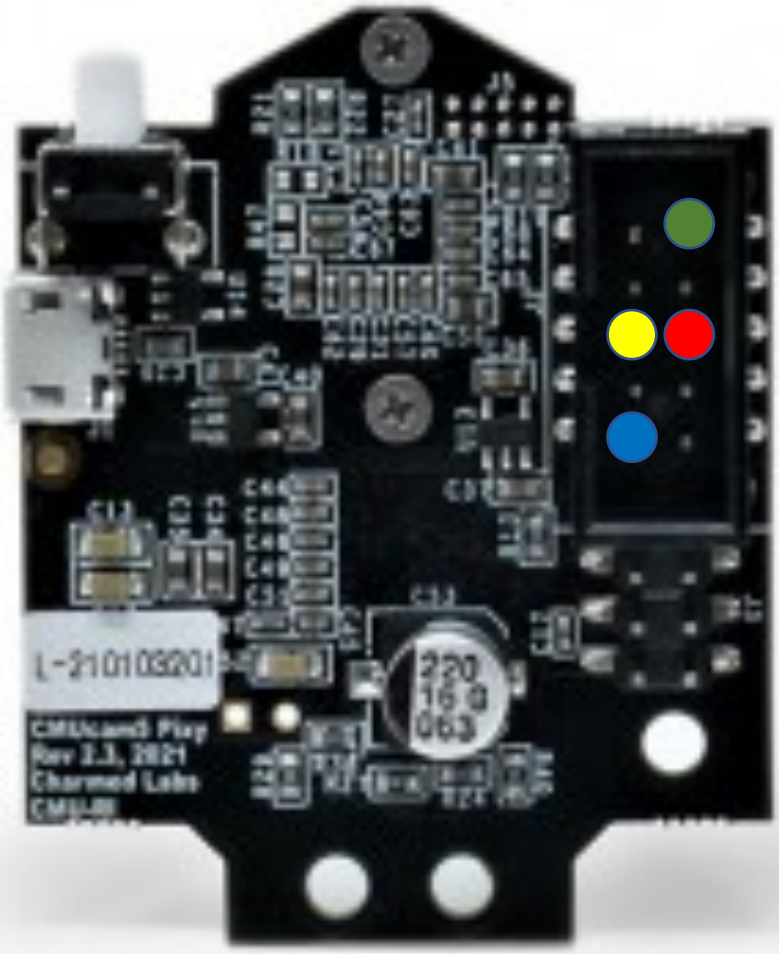
---Инструкция работы с камерой Pixy v2.1---

Оглавление

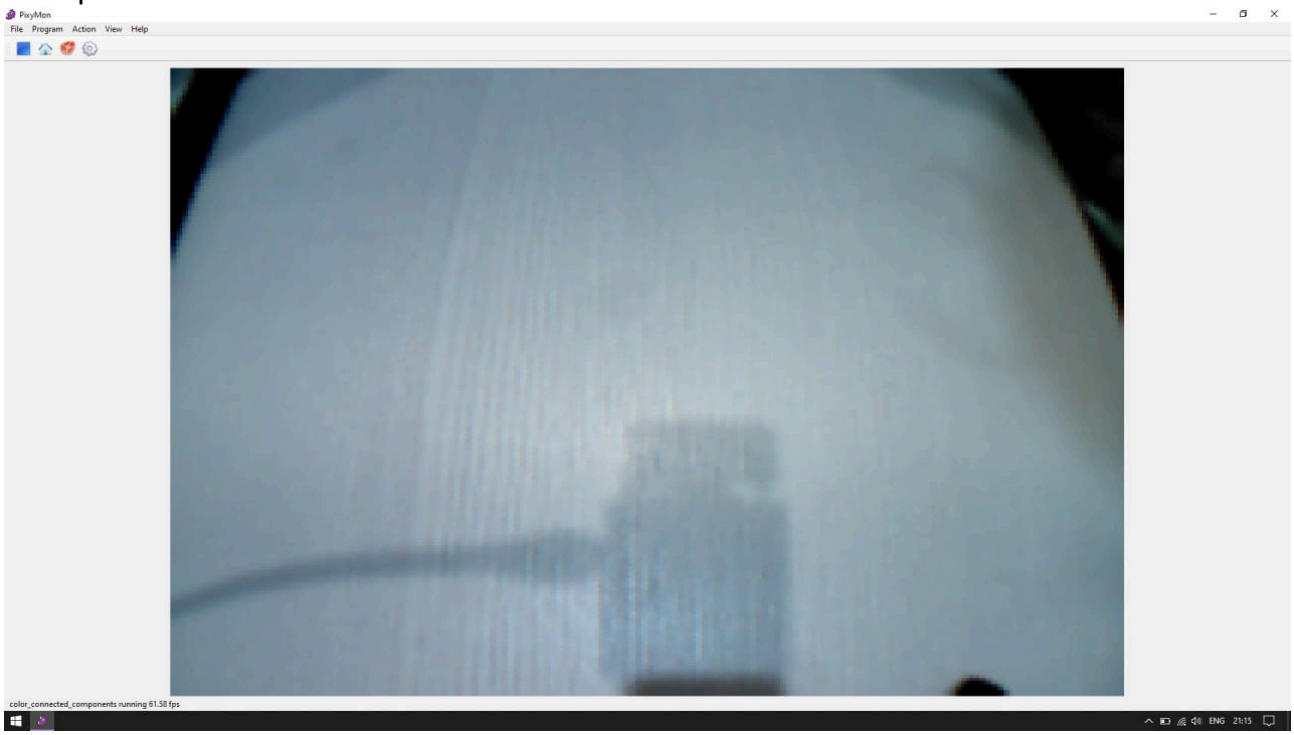
Подключение

Для подключения камеры нам необходимо разрезать изоляцию провода Ev3 и припаять коннекторы типа «мама». Далее их подключить к пинам указанным на изображении.
Цвет кружочка обозначает цвет провода, к которому нужно подключить данный контакт. (зеленый пин – к зеленому проводу).

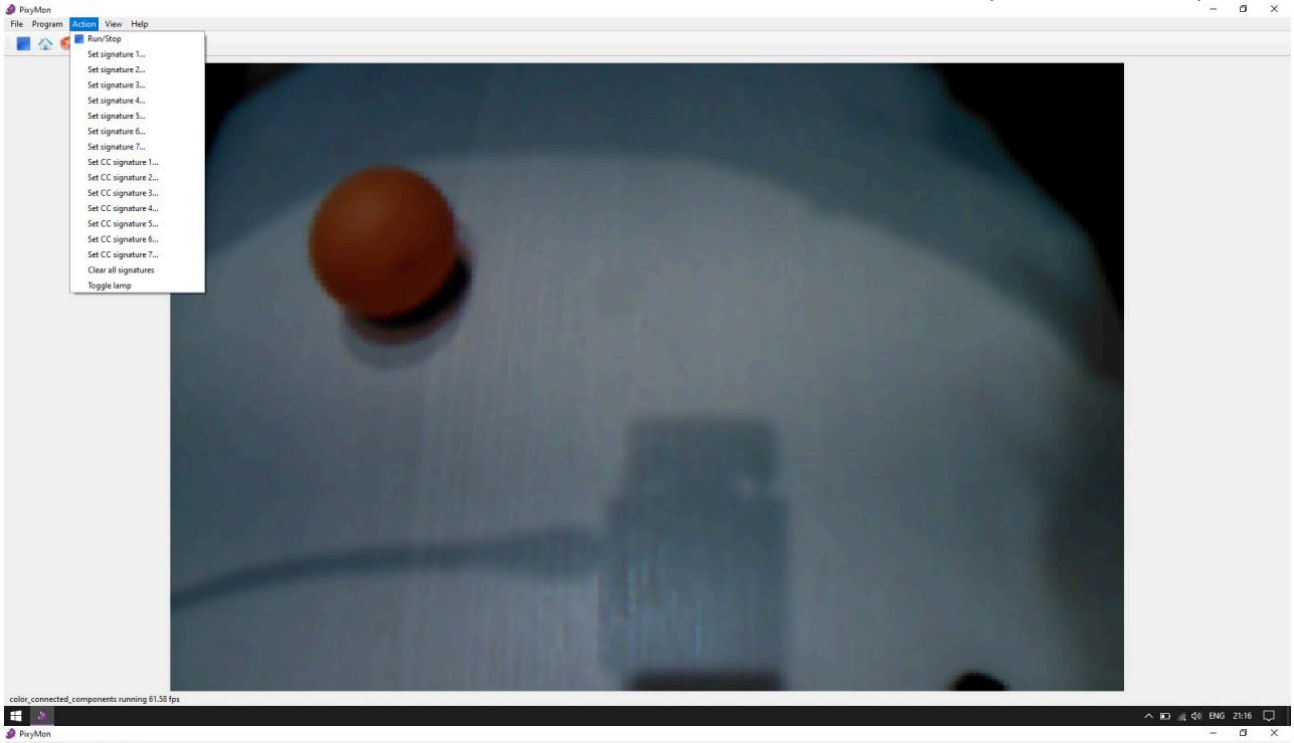


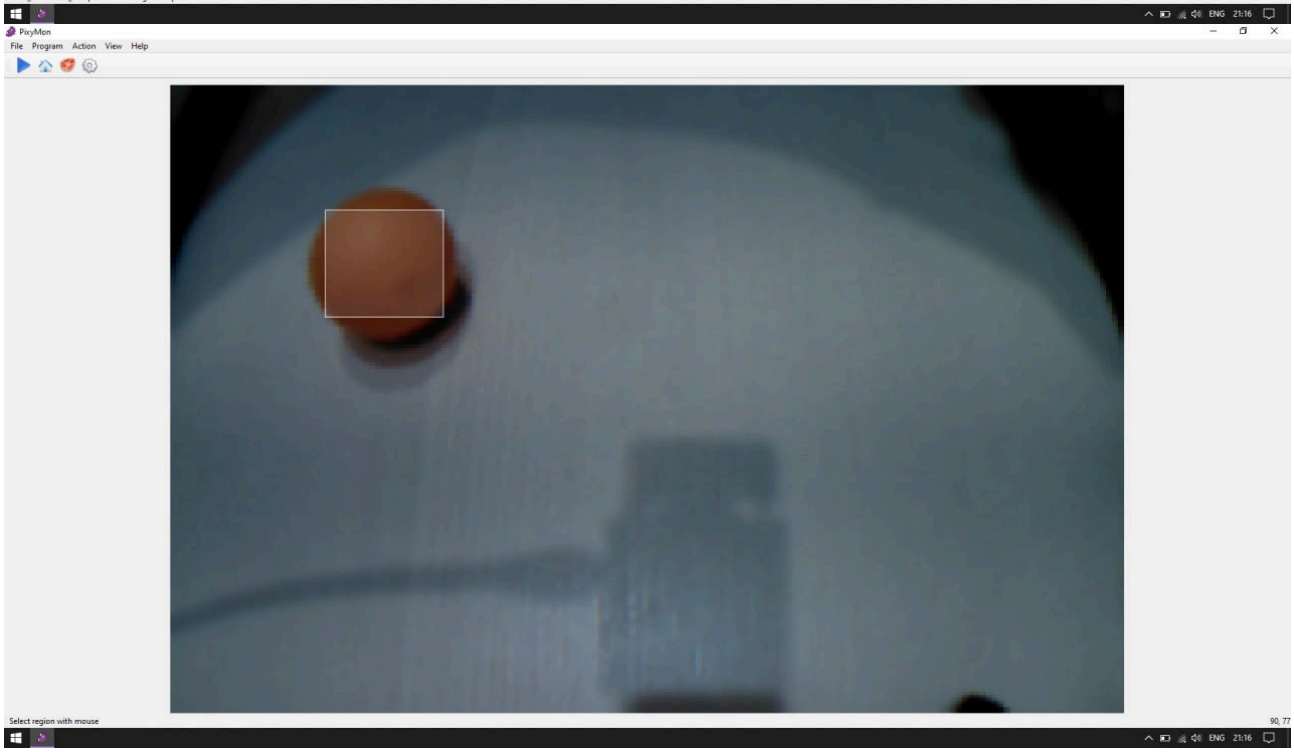
Настройка камеры

В данном пункте я расскажу, как настраивать камеру на определение объекта определенного цвета, в моем случает теннисного мячика оранжевого цвета.
Скачиваем программу PixyMon v2 по ссылке (https://pixycam.com/downloads-pixy2/) и открываем её.
Подключаем камеру по проводу microUSB к компьютеру и видим наше изображение с камеры.

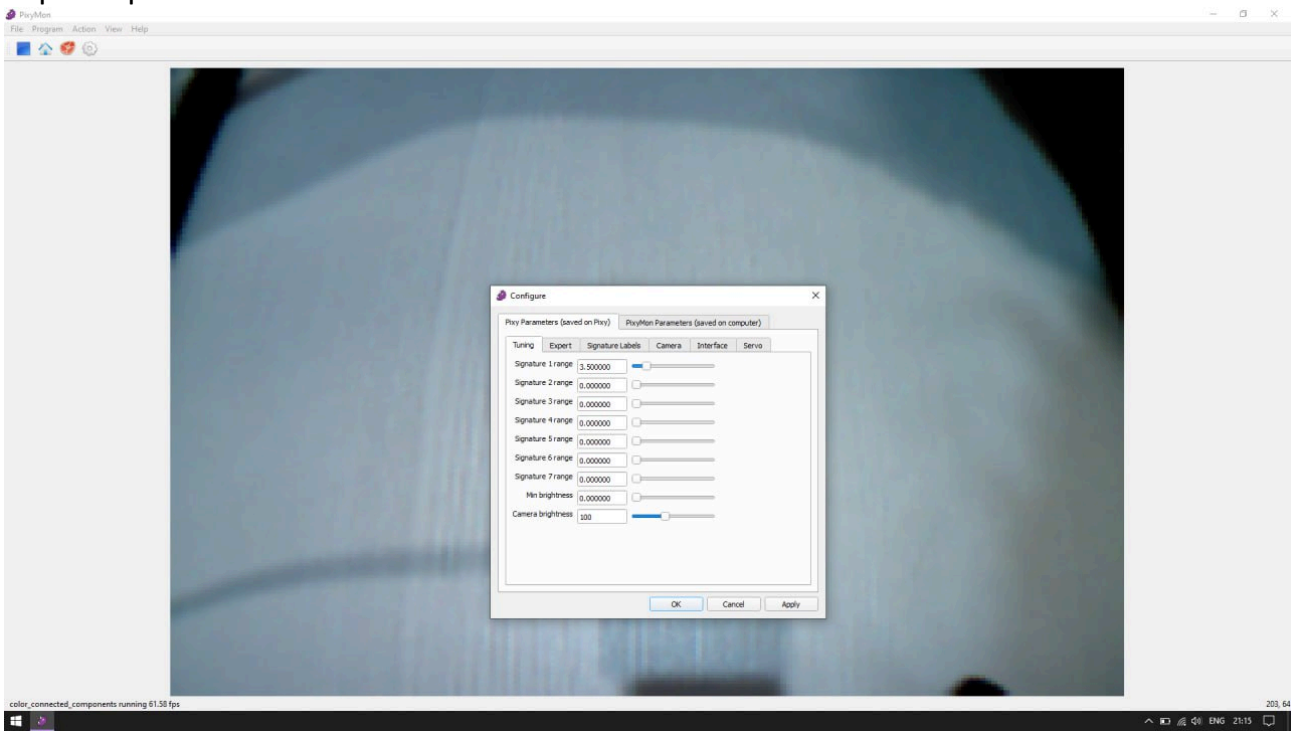


Далее мы показываем объект, который необходимо найти. Открываем вкладку Action. Там мы видим функции для задачи сигнатуры объекта. Нажимаем Set signature 1 и выделяем ЦВЕТ ОБЪЕКТА, обратите внимание не объект, а его цвет. Чем больше область цвета вы выделите, тем точнее будет определение.

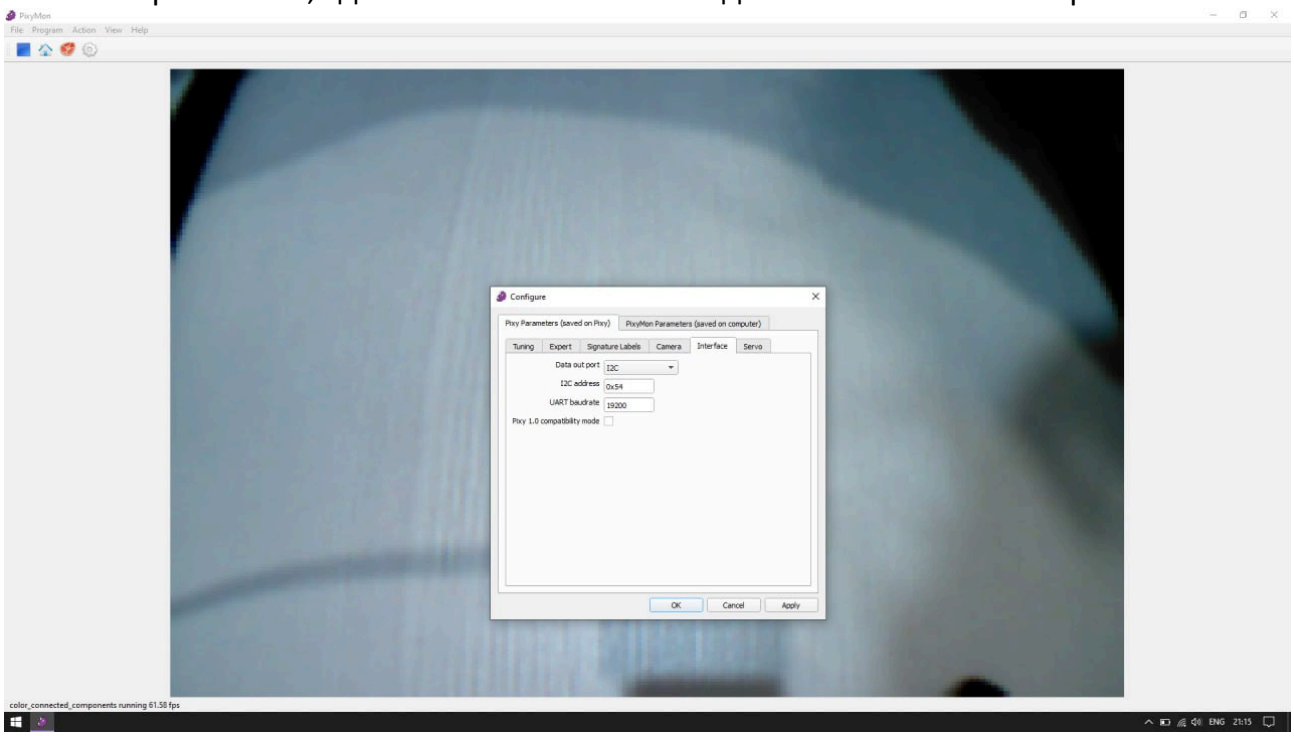




Всё, мы настроили определение объекта в приложении. Также можно изучить настройки во вкладке Settings и улучшить определение.
Например, если камера видит слишком много лишних объектов, то вы можете либо перенастроить камеру, как мы сделали раньше, либо уменьшить Range во вкладке Tuning. Начальное значение 3.5, советую самим экспериментировать с изменением данного параметра.



Для работы с камерой на ev3basic нам необходимо поменять способ общения камеры с компьютером на I2C, сделать это можно во вкладке Interface – data out port.



Считывание показаний с камеры

Я буду считывать показания на языке ev3basic в среде программирования clev3r. Скачать clev3r можно по ссылке (https://clev3r.ru/) Код для считывания выглядит так:

Вначале программы нам нужно задать константы:

```
I2C_adress = 84 'Камера
iCorrect = 0 'корректировка камеры
ArraySizeSend = 32
valuesSize = 19
ArraySend = Vector.Init(ArraySizeSend,0)
valuesCamera = Vector.Init(valuesSize, 0) 'массив данных с камеры
ArraySend[0] = 174
ArraySend[1] = 193
ArraySend[2] = 32
ArraySend[3] = 2
ArraySend[4] = 1
ArraySend[5] = 1
tvalues[0] = 0 'измененный массив
sti = 0 'коррекция массива
xcorrect = 0 'корректировка массива
```

Функция определения координат объекта:

```
Sub getCoordsBall
  While "true"
    iCorrect = 0

    valuesCamera = Sensor.CommunicateI2C(cameraPort,I2C_adress, ArraySizeSend, valuesSize, ArraySend)

    xcorrect=0
    iCorrect = 0
    While 16>= iCorrect + 3
      If valuesCamera[iCorrect] = 85 And valuesCamera[iCorrect+1] = 170 And valuesCamera[iCorrect+2] =85
And valuesCamera[iCorrect+3] = 170 Then
        xcorrect = iCorrect
        iCorrect = 99
      EndIf
      iCorrect = iCorrect + 1
    EndWhile

    If iCorrect <> 100 Then
      If valuesCamera[13] = 85 And valuesCamera[14] = 170 And valuesCamera[15] = 85 And valuesCamera[0]
= 170 Then
        xcorrect = 13
      EndIf
      If valuesCamera[14] = 85 And valuesCamera[15] = 170 And valuesCamera[0] = 85 And valuesCamera[1] =
170 Then
        xcorrect = 14
      EndIf
      If valuesCamera[15] = 85 And valuesCamera[0] = 170 And valuesCamera[1] = 85 And valuesCamera[2] =
170 Then
        xcorrect = 15
      EndIf
    EndIf

    If xcorrect <> 0 Then
      tvalues = Sensor.CommunicateI2C(cameraPort,I2C_adress, ArraySizeSend, valuesSize, ArraySend)
      iCorrect = xcorrect
      sti=  0
      While iCorrect <= 15
        valuesCamera[sti] = valuesCamera[iCorrect]
        iCorrect = iCorrect + 1
        sti = sti + 1
      EndWhile
      iCorrect = 0

      sti = 16 - xcorrect
      While iCorrect < xcorrect
        valuesCamera[sti] = tvalues[iCorrect]
        iCorrect = iCorrect + 1
        sti = sti + 1
      EndWhile
    EndIf

    centerX = valuesCamera[8] + valuesCamera[9] * 255
    centerY = valuesCamera[10] + valuesCamera[11] * 255
    width = valuesCamera[12] + valuesCamera[13] * 255
    hight = valuesCamera[14] + valuesCamera[15] * 255
  EndWhile
EndSub
```

Пояснение к коду

```
valuesCamera = Sensor.CommunicateI2C(cameraPort,I2C_adress, ArraySizeSend, valuesSize, ArraySend)
```

Здесь мы получаем значения с камеры

```
    xcorrect=0
    iCorrect = 0
    While 16>= iCorrect + 3
      If valuesCamera[iCorrect] = 85 And valuesCamera[iCorrect+1] = 170 And valuesCamera[iCorrect+2] =85
And valuesCamera[iCorrect+3] = 170 Then
        xcorrect = iCorrect
        iCorrect = 99
      EndIf
      iCorrect = iCorrect + 1
    EndWhile

    If iCorrect <> 100 Then
      If valuesCamera[13] = 85 And valuesCamera[14] = 170 And valuesCamera[15] = 85 And valuesCamera[0] =
170 Then
        xcorrect = 13
      EndIf
      If valuesCamera[14] = 85 And valuesCamera[15] = 170 And valuesCamera[0] = 85 And valuesCamera[1] =
170 Then
        xcorrect = 14
      EndIf
      If valuesCamera[15] = 85 And valuesCamera[0] = 170 And valuesCamera[1] = 85 And valuesCamera[2] =
170 Then
        xcorrect = 15
      EndIf
    EndIf

    If xcorrect <> 0 Then
      tvalues = Sensor.CommunicateI2C(cameraPort,I2C_adress, ArraySizeSend, valuesSize, ArraySend)
      iCorrect = xcorrect
      sti=  0
      While iCorrect <= 15
        valuesCamera[sti] = valuesCamera[iCorrect]
        iCorrect = iCorrect + 1
        sti = sti + 1
      EndWhile
      iCorrect = 0

      sti = 16 - xcorrect
      While iCorrect < xcorrect
        valuesCamera[sti] = tvalues[iCorrect]
        iCorrect = iCorrect + 1
        sti = sti + 1
      EndWhile
    EndIf
```

Это корректировка входного массива. Иногда камера смещает массив на несколько индексов, поэтому данный код корректирует входной массив.

```
    centerX = valuesCamera[8] + valuesCamera[9] * 255
    centerY = valuesCamera[10] + valuesCamera[11] * 255
    width = valuesCamera[12] + valuesCamera[13] * 255
    hight = valuesCamera[14] + valuesCamera[15] * 255
```

Получаем координаты центра объекта по оси X и У, также получаем длину и ширину.

Использование в основной программе

Мы объявляем блок и константы в начале программ. Далее запускаем его в параллельном цикле:

```
Thread.Run = getCoordsBall
```

Всё, вывести координаты на экран можно таким образом:

```
While "true"
    LCD.StopUpdate()
    LCD.Clear()
    LCD.Write(0,50,centerX)
    LCD.Write(50,50,centerY)
    LCD.Write(0,100,width)
    LCD.Write(50,100,hight)
    LCD.Update()
EndWhile
```

Алгоритм следования за мячом

Алгоритм состоит из обычного ПИД-регулятора движения, но «ошибка» рассчитывается по-другому. Если для езды по линии мы из показаний одного датчика вычитали показания другого, то в нашем случае мы из координаты объекта по X вычитаем координату по оси X к которой мы стремимся. Коэффициенты вы должны подобрать сами.

```
Thread.Run = getCoordsBall
While "True"
    err = centerX - centerNeed
    sum = sum + err - errOld
    dLine = (err - errOld)
    up = (err * kFollowingBall) + (dLine * kdFollowingBall) + (sum * kiFollowingBall)
    Motor.StartPower("B",vFollowingBall + up)
    Motor.StartPower("C",vFollowingBall - up)
    errOld = err
EndWhile
```