Eng:

Pixy 2 is a camera with a built-in processor that can process images and provide ready-made information about the image.

Pixy official website: <https://pixycam.com/pixy2/>

Pixycam 2 documentation: <https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start>

For the camera to work correctly with ev3, you need to install pixyMon v2 (<https://pixycam.com/downloads-pixy2/> ), install “general frameware”, and in the settings in the “interface” menu select I2C mode (not lego I2C!!!)

This repository has a module for running ev3 with pixie 2 camera on clover. As well as test programs to check the functionality of the camera.

**Repository contents:**

“Pixy2.bpm” - main module

“getBlocks\_test.bp” - a program that tracks specified colored objects

“specifics\_test.bp” - a program for testing basic camera functions

“getMainFeature\_test.bp” - a program that tracks straight lines (for example, a line)

“getRGB\_test.bp” - a program that gets the rgb of “each” pixel (takes a photo)

“\_internal/ and rgbToPNG.exe” - a program for saving an image in png format based on the results of “getRGB\_test.bp”

You can see how test programs work here: \*link to video\*

If you move the “\_internal” folder, the “rgbToPNG.exe” application, the text file “rgb.txt” into one folder (this file must be downloaded from the ev3 unit after running the “getRGB\_test.bp” program, an image in RGB format is written to this file) and run “rgbToPNG.exe”, the application will create a PNG image in this folder

**I divided the commands for working with pixies (documentation) into 4 categories:**

- Specifics

- Color Connected Components

- Line Tracking

-Video

green commands - already implemented and in the module

red commands are commands that are described in the documentation (<https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:porting_guide> ), but have not yet been added to the module



**Specific** - these are general commands, such as turning on the backlight, getting fps, etc.

**getVersion()** - this command returns information about the current camera firmware

The command has 2 input parameters: camera port, I2C camera address.

Returns 5 output parameters about the current firmware version.

**printVersion()** - this command prints information about the camera firmware in a convenient form on the block screen.

The command has 2 input parameters: camera port, I2C camera address.

**getResolution()** - This command returns the camera resolution.

The command has 2 input parameters: camera port, I2C camera address.

Returns 2 parameters: the number of pixels in width and the number of pixels in height.

**setCameraBrightness()** - this command is not implemented(

**setServos()** - this command is not implemented(

**setLED()** - this command lights the RGB LED based on the specified RGB components.

The command has 5 input parameters: camera port, camera I2C address, R component, G component, B component.

**setLamp()** - This command turns the top and bottom LEDs on and off.

The command has 4 input parameters: camera port, I2C camera address, status.

Upper LEDs (0-1), status of the lower LED (0-1).

0 - disabled

1 - enabled

**getFPS()** - This command returns the current frames per second refresh rate of the camera. ***Attention!!! ev3 can accept significantly fewer frames per second due to the low information transfer rate of the I2C bus***

The command has 2 input parameters: camera port, I2C camera address.

Returns 1 parameter: the current number of frames per second.

**Color Connected Components** are commands that allow you to track specified objects. Objects are set through the PixyMon2 application (<https://pixycam.com/downloads-pixy2/> )

**getMainBlock()** - This command returns information about the largest object being tracked.

The command has 2 input parameters: camera port, I2C camera address.

Returns 8 parameters: object number, object center along the X axis, object center along the Y axis, object width, object height, angle, identification number, number of frames during which the object is in the field of view (0-255).

For each new object in the frame, the camera gives its own unique identification number. If you remove an object from the frame and move it back into the frame, the ID number will be different.

For normal objects, the angle will always be 0. For color codes, the angle is the degree of inclination of the code (-180/180).

If the tracked object is a color code, then instead of the object number there will be a color code (does not work correctly yet).

**getBlocks()** - This command returns information about all tracked objects. Please note that this command returns information about objects in the form of arrays, where counting starts from index 0. Objects in the array are sorted in descending order, that is, the 0th element of the array will contain information about the largest object, etc.

The command has 2 input parameters: camera port, I2C camera address.

Returns 9 parameters (1 numeric value and 8 numeric arrays): the number of objects in the frame (numeric value, subsequent arrays), object number, object center along the X axis, object center along the Y axis, object width, object height, angle, identification number, the number of frames during which the object is in the field of view (0-255).

**Line Tracking** are commands that allow you to track straight lines (for example, to follow a line).

**getMainFeature()** - This command returns information about a vector, a vector is a line between two points. A vector is the main (largest) straight line in the frame.

The command has 2 input parameters: camera port, I2C camera address.

Returns 6 parameters: the x-coordinate of the starting point,

y-coordinate for start point, x-coordinate for end point, y-coordinate for end point, ID number, flag

For each new object in the frame, the camera gives its own unique identification number. If you remove an object from the frame and move it back into the frame, the ID number will be different.

A flag is unique information about a vector, for example, if the vector is adjacent to an intersection, then the flag will be equal to 4, otherwise it will be equal to 0.

**getAllFeatures()** - this command is not implemented(

**setMode()** - this command is not implemented(

**setNextTurn()** - this command is not implemented(

**setDefaultTurn()** - this command is not implemented(

**setVector()** - this command is not implemented(

**reverseVector()** - this command “reverses” the direction of the vector (switches the starting and ending points).

The command has 2 input parameters: camera port, I2C camera address.

**Video** are commands that allow you to obtain information about a picture in the RGB format of each pixel

**getRGB()** - This command returns the RGB values of a specified area in a frame.

The command has 5 input parameters: camera port, camera I2C address, x coordinate of the area center, y coordinate of the area center, saturation (0-1).

Returns 3 parameters: RGB of the specified area

An “area” is a collection of 25 pixels.

If the saturation input parameter is 0, then the RGB values will be “raw”. If the input parameter “saturation” is equal to 1, then the largest value among RGB will be equal to 255, the rest will be stretched in proportion to the maximum