Алгоритм работы программы:

1. Включение камеры (ничего не происходит просто выводит изображение)
2. Если кнопка была удержана в течении 5 сек
   1. Включается режим Настройки
   2. Появляется надпись (уберите деталь из кадра и удерживайте кнопку 3 сек.)
   3. Если кнопка была удержана в течении 3 секунд
      1. Начинается калибровка уровня усиления камеры (пока все белы объекты не исчезнут)
      2. Запоминаем это значение (gain\_max – значение, выше которого нельзя поднимать усиление, т.к. будут распознаны другие объекты)
   4. Появляется надпись (поставе деталь и удерживайте кнопку 3 сек.)
   5. Если кнопка была удержана в течении 3 секунд
      1. Начинается калибровка уровня усиления камеры (пока не появиться деталь, при этом gain не может быть выше gain\_max)
      2. Запоминаем значение gain
      3. Если деталь удалось обнаружить, то переходим в обычный режим
      4. Обнаружение наибольшего белого объекта в кадре с фиксированным уровнем усиления = gain.

https://cctvlens.ru/publications/polyarizatsionnye-svetofiltry/

**sensor.flush()**

**sensor.get\_fb()**

**sensor.disable\_delays(*disable:***[***bool***](https://docs.openmv.io/library/builtins.html#bool)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= None*)→**[**bool**](https://docs.openmv.io/library/builtins.html#bool)

**find\_edges(*edge\_type***, ***threshold=(100, 200)*) image.EDGE\_CANNY**

**image.VFLIP*:***[***int***](https://docs.openmv.io/library/builtins.html#int) **image.HMIRROR*:***[***int***](https://docs.openmv.io/library/builtins.html#int) **image.ROTATE\_90*:***[***int***](https://docs.openmv.io/library/builtins.html#int)

**image.BLACK\_BACKGROUND*:***[***int***](https://docs.openmv.io/library/builtins.html#int)

**rotation\_corr**

**Filtering Methods**

**bilateral(*size:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***color\_sigma:***[***float***](https://docs.openmv.io/library/builtins.html#float)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= 0.1***, ***space\_sigma:***[***float***](https://docs.openmv.io/library/builtins.html#float)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= 1.0***, ***threshold:***[***bool***](https://docs.openmv.io/library/builtins.html#bool)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= False***, ***offset:***[***int***](https://docs.openmv.io/library/builtins.html#int)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= 0***, ***invert:***[***bool***](https://docs.openmv.io/library/builtins.html#bool)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= False***, ***mask:***[***Image***](https://docs.openmv.io/library/omv.image.html#image.Image)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= None*)→**[**Image**](https://docs.openmv.io/library/omv.image.html#image.Image)

**mode(*size:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***threshold:***[***bool***](https://docs.openmv.io/library/builtins.html#bool)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= False***, ***offset:***[***int***](https://docs.openmv.io/library/builtins.html#int)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= 0***, ***invert:***[***bool***](https://docs.openmv.io/library/builtins.html#bool)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= False***, ***mask:***[***Image***](https://docs.openmv.io/library/omv.image.html#image.Image)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= Nonee*)→**[**Image**](https://docs.openmv.io/library/omv.image.html#image.Image)

**dilate(*size:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***threshold:***[***int***](https://docs.openmv.io/library/builtins.html#int)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= None***, ***mask:***[***Image***](https://docs.openmv.io/library/omv.image.html#image.Image)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= None*)→**[**Image**](https://docs.openmv.io/library/omv.image.html#image.Image)

**gamma(*gamma:***[***float***](https://docs.openmv.io/library/builtins.html#float)***= 1.0*, *contrast:***[***float***](https://docs.openmv.io/library/builtins.html#float)***= 1.0*, *brightness:***[***float***](https://docs.openmv.io/library/builtins.html#float)***= 0.0*)→**[**Image**](https://docs.openmv.io/library/omv.image.html#image.Image)

**mask\_rectange(*x:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***y:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***w:***[***int***](https://docs.openmv.io/library/builtins.html#int), ***h:***[***int***](https://docs.openmv.io/library/builtins.html#int)**)**

**clear(*mask:***[***Image***](https://docs.openmv.io/library/omv.image.html#image.Image)***|***[***None***](https://docs.python.org/3.5/library/constants.html#None)***= None*)→**[**Image**](https://docs.openmv.io/library/omv.image.html#image.Image)

**запись чтение**

## класс ImageIO – объект ImageIO

**bytearray()→ [bytearray](https://docs.openmv.io/library/builtins.html" \l "bytearray" \o "bytearray)**

Функция Final\_Detecting\_Object отвечает за обнаружение объекта и реагирование на его присутствие или отсутствие.

1. Функция check\_object\_in\_frame проверяет наличие объекта в кадре в течение заданного времени (200 миллисекунд). Если объект не найден в течение заданного времени, возвращается 0, иначе — 1.
2. В функции Final\_Detecting\_Object происходит мониторинг наличия объекта в течение 1 секунды, на основе результата работы check\_object\_in\_frame. Если объект присутствует всё время, выводится сообщение «Объект обнаружен».
3. Если по истечении времени объект не был обнаружен, подаётся сигнал на реле, после чего сохраняется изображение в буфер изображений (75). Затем функция снова начинает мониторинг.
4. При отсутствии объекта в течение двух секунд выводится сообщение об ошибке обнаружения, после чего сохраняется аварийное изображение.