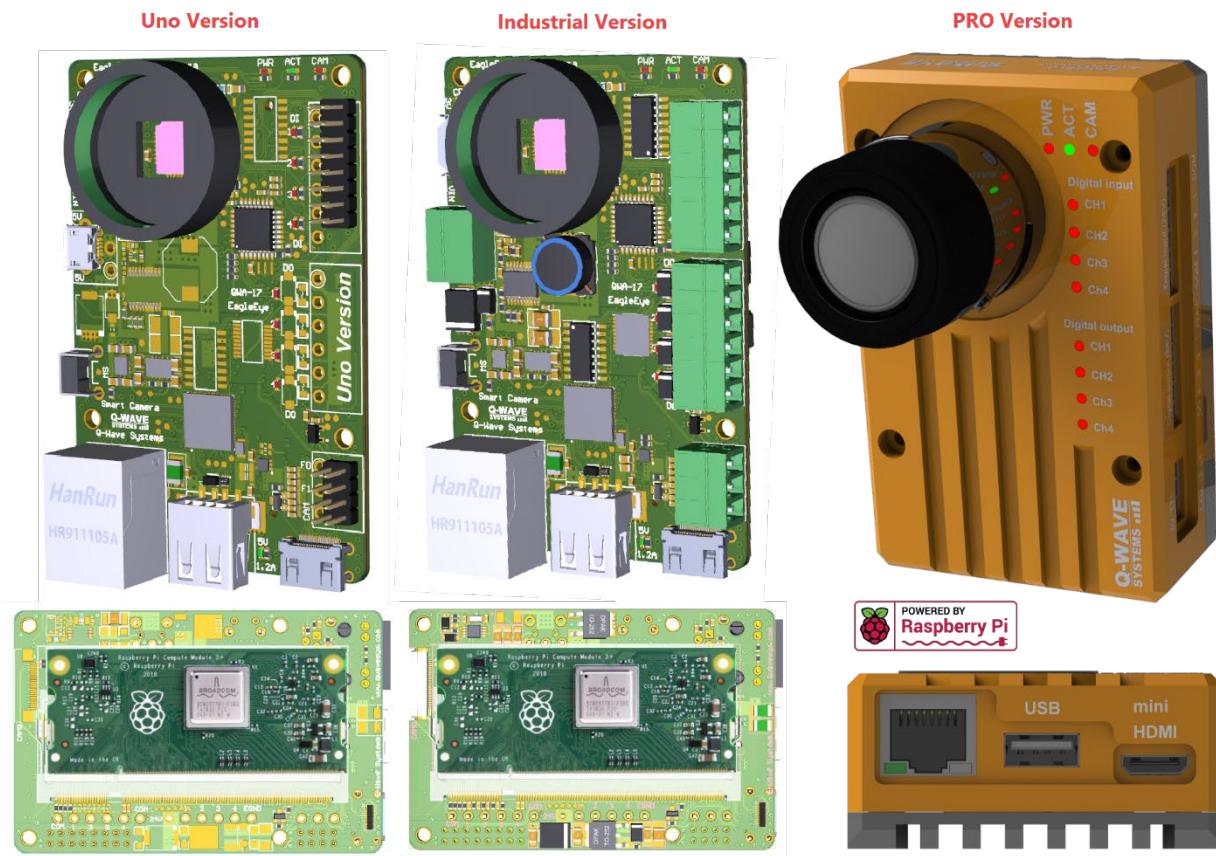


## EagleEYE Smart Camera



[https://github.com/QWaveSystems/QwaveSmartCamera\\_EagleEYE](https://github.com/QWaveSystems/QwaveSmartCamera_EagleEYE)

Product Datasheet  
Developer Manual  
Getting Start Guide

## Contents

Order Part Number .....	4
Target Application.....	5
Product Specification.....	5
EagleEYE Industrial verson.....	6
Pro Version (Industrial Version + Enclosure+ Heat-sink) .....	7
EagleEYE Uno version .....	8
Raspberry Pi Compute Module 3+ (CM3+) Specification .....	9
EagleEYE and Raspberry Pi CM3+ Interface Board view.....	9
Dimension .....	10
Update OS via EagleEYE Flasher CM3+ board .....	10
Power Supply Connection Diagram .....	11
Digital Input Connection Diagram .....	12
Digital Output Connection Diagram .....	13
CAM status and Light Source Output: RGB (WS2812) Connection Diagram.....	14
Getting Start Guide: Connect to HDMI monitor .....	15
Getting Start Guide: Connect PC via LAN cable and using VNC viewer.....	16
Getting Start Guide: File transfer using Network shared drive.....	18
Getting Start Guide: File transfer using FTP software.....	18
Getting Start Guide: Remote SSH.....	19
Getting Start Guide: Turn OFF start-up demo program.....	19
Getting Start Guide: Examples files.....	20
Getting Start Guide: Test camera using V4L2 .....	21
Getting Start Guide: Test camera using raspistill.....	21
Getting Start Guide: Python 3.5.6 face detect app using OpenCV 4.1.0 .....	22
Getting Start Guide: Face detect app using C++ and OpenCV 4.1.0 .....	24
Getting Start Guide: Using NodeRed program .....	25
Getting Start Guide: GPIO access.....	26
Getting Start Guide: using LabVIEW 2017 with Smart Camera EagleEYE.....	28
Getting Start Guide: using LabVIEW 2017 and OpenCV function development.....	32
Getting Start Guide: Install or upgrade the Raspbian OS.....	35

HW Design Schematics: GPIO .....	38
HW Design Schematics: Power Supply.....	38
HW Design Schematics: HDMI Display.....	39
HW Design Schematics: Real-Time Clock.....	39
HW Design Schematics: USB host.....	40
HW Design Schematics: eMMC Boot.....	40
HW Design Schematics: MCU Interface .....	41
HW Design Schematics: Image Sensor Interface .....	41
HW Design Schematics: Bill of Material (BOM) .....	42
HW Design Schematics: PCB Layout .....	43
Developer Manual: Recovery File systems Mode.....	44

Revision History

Revision	Date	Comment	Editor
A 1.0.1	23/11/2019	Update Recovery mode instruction	Amornthe P
A 1.0.0	22/11/2019	Initial version	Amornthe P.

Order Part Number

Product	P/N	Price USD/THB
EagleEYE Uno Board	EY-UNO	
EagleEYE Industrial Board	EY-IND	
Full Size Heat-sink	EY-HSK	
EagleEYE Uno+ Heat sink+ "CM3+ 32GB"	EY-UNO-32	
EagleEYE Industrial+ Heat sink+ "CM3+ 32GB"	EY-IND-32	
EagleEYE Pro "CM3+ 32GB"	EY-PRO-32	
EagleEYE Pro Developer Kit - Raspberry Pi CM3+ 32GB - EagleEYE Flasher CM3+ board - x1 Power supply 24V 2A - x5 Lens CS mount 4/6/8/12/5-50mm - x1 RGB 40 LEDS (WS2812) - x1 mini HDMI to HDMI cable 1m - x1 LAN 100Mbps cable 1m - x1 USB 2.0 hub 3 port Slim version - x1 USB 2.0 WiFi 2.4GHz 802.11b/g/n (150Mbit/s) - x1 Din rail clipper	EY-PRO-KIT	
EagleEYE Uno Developer Kit - Raspberry Pi CM3+ 32GB - EagleEYE Flasher CM3+ board - x1 Power supply 5V 2.5A (micro USB) - x5 Lens CS mount 4/6/8/12/5-50mm - x1 RGB 40 LEDS (WS2812) - x1 mini HDMI to HDMI cable 1m - x1 LAN 100Mbps cable 1m - x1 USB 2.0 hub 3 port Slim version - x1 USB 2.0 WiFi 2.4GHz 802.11b/g/n (150Mbit/s)	EY-UNO-KIT	

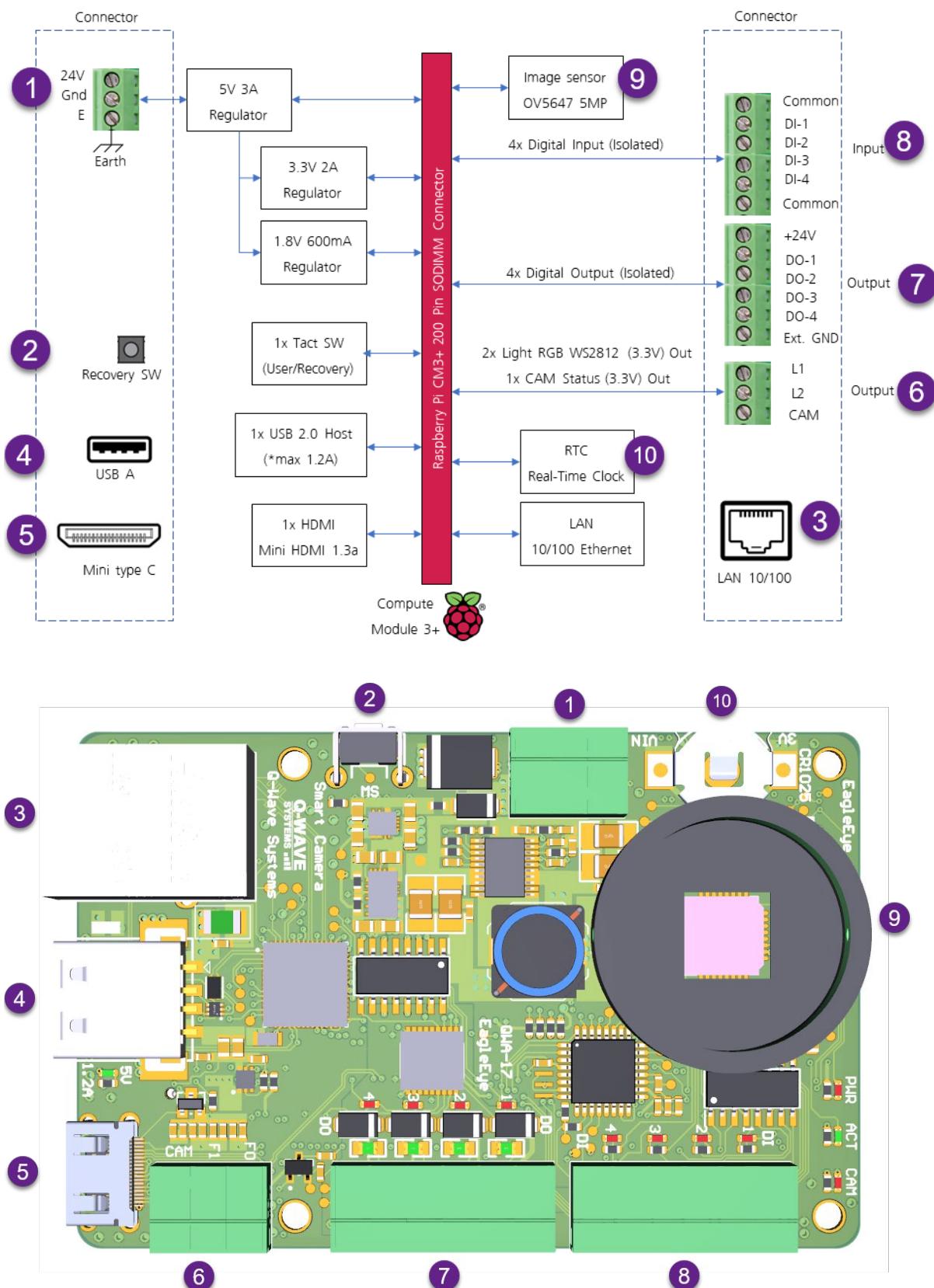
## Target Application

- Embedded vision
- industrial machine vision application
- Prototype computer vision algorithm
- Image processing analysis
- AI vision machine learning
- Drones navigation systems
- Robots vision, AGV (Automated Guided Vehicles)
- ADAS (Advanced Driver Assistance System)

## Product Specification

Specification	Industrial/Pro Version	Uno Version
CPU	<a href="#">Raspberry Pi Compute Module 3+ (CM3+)</a> via SODIMM connector, CPU Broadcom BCM2837B0 1.2GHz quad core cortex-A53 processor, 1GB LPDDR2 RAM, eMMC 8GB/16GB/32GB flash	
Camera Sensor	5MP OV5647 1/4inchs CMOS RAW image sensor (CSI-2)	
Camera Resolution	QSXGA 2592x1944 (max), Video QVGA 320x240 @120fps (max)	
Lens Configuration	CS mount lens	
Video Output	1x Mini HDMI Port (HDMI V1.3a)	
Networking	1x 10/100M Ethernet	
USB Host	1x USB 2.0 host port up to 1.2A	
Camera Status	1x 3.3V Output (CAM)	
Light RGB Output	2x 3.3V Output (L1,L2) *Required external +5V 3A (RGB WS2812)	
Thermal Solution	Full size heat-sink (98x61mm)	
RTC	Real-Time clock onboard	-
HW Watchdog	Yes (onboard MCU)	-
Voltage Input	12V-24V Input (*min 25W)	5V 3A Input via uUSB
Circuit Protection	Polarity protection, Short circuit, Over voltage/current, Thermal shutdown	-
Temperature Range	-20 C – +85 C	0 C – +45 C
Digital Input	4x Isolated 4 Channel 24V input	4x 3.3V via GPIO header
Digital Output	4x Isolated 4 Channel 24V Output (Required external +24V supply)	4x 3.3V via GPIO header (*50mA total)
Dimension (W/L/H)	Board size 85x56x19.5mm	Board size 85x56x19.5mm
Weight	-	-
Dimension(heatsink)	with heat-sink 98x61x31mm	with heat-sink 98x61x28mm
Power Consumption	25W (max)	15W (max)
Software	Standard Raspbian OS, OpenCV, C++ and Python	

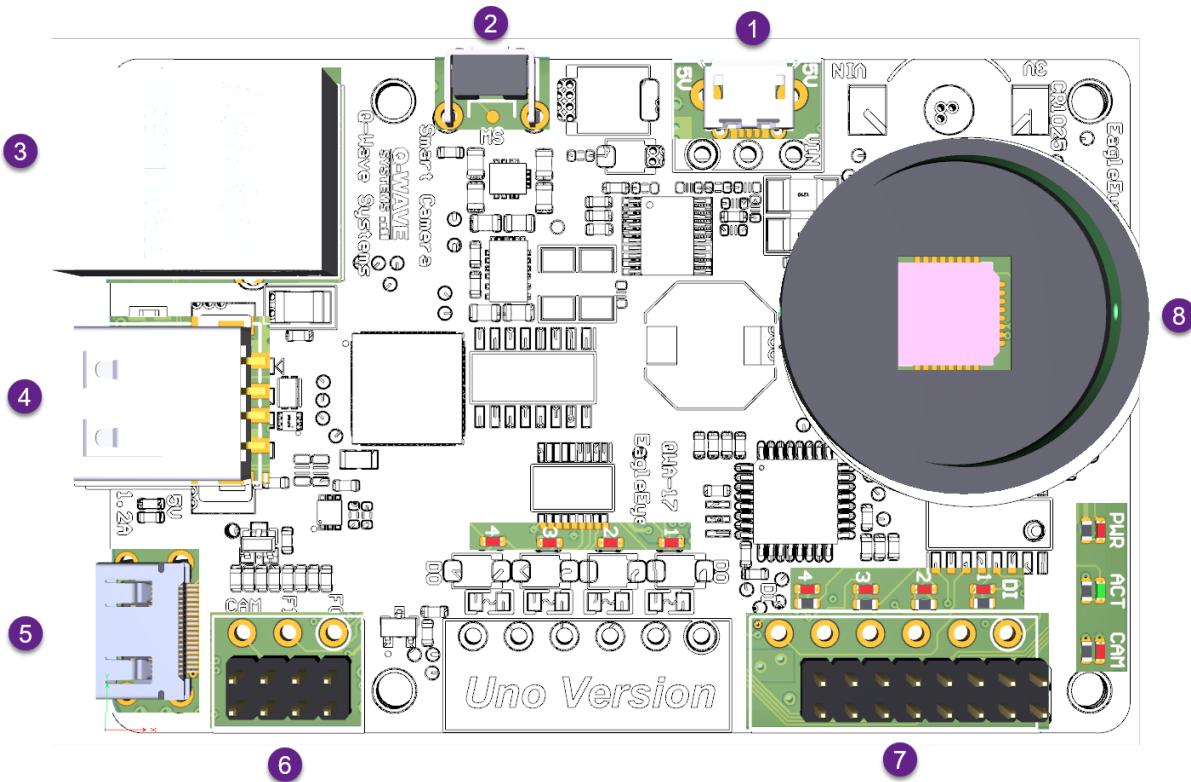
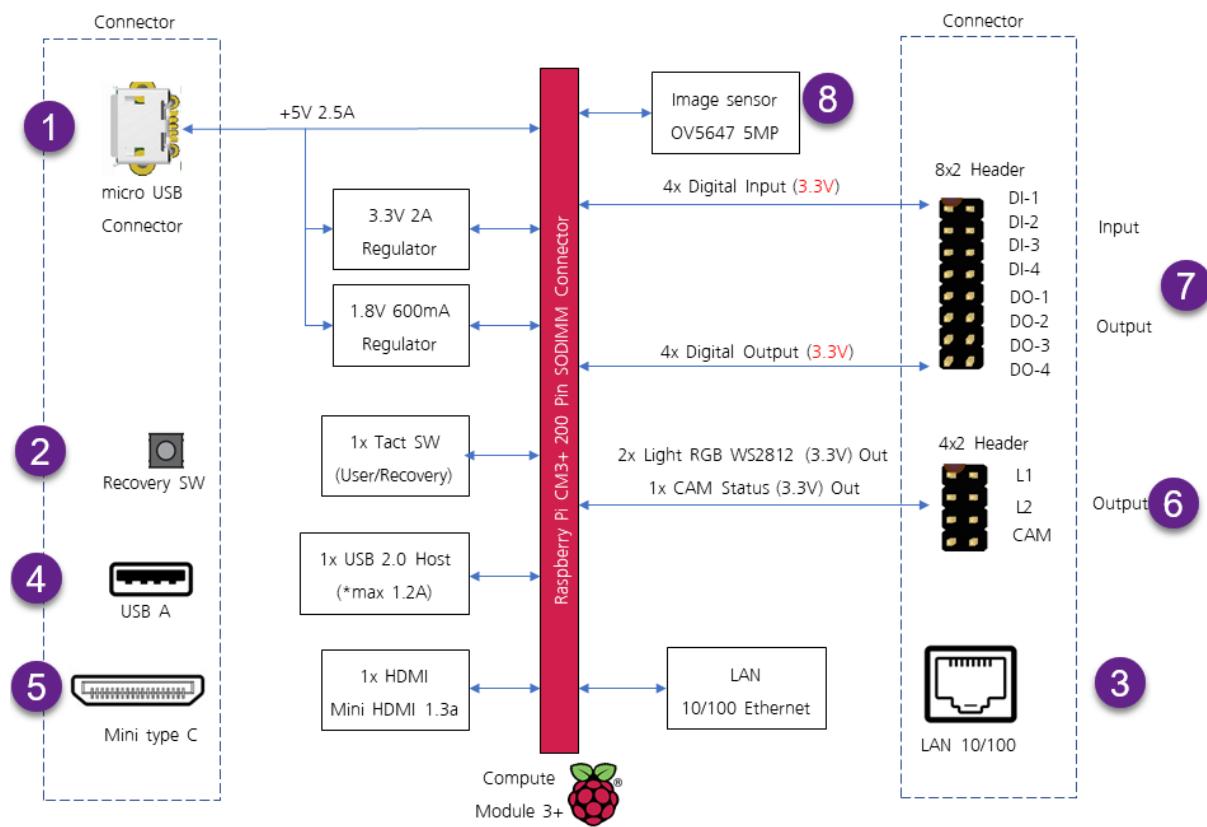
## EagleEYE Industrial version



Pro Version (Industrial Version + Enclosure+ Heat-sink)



## EagleEYE Uno version



## Raspberry Pi Compute Module 3+ (CM3+) Specification

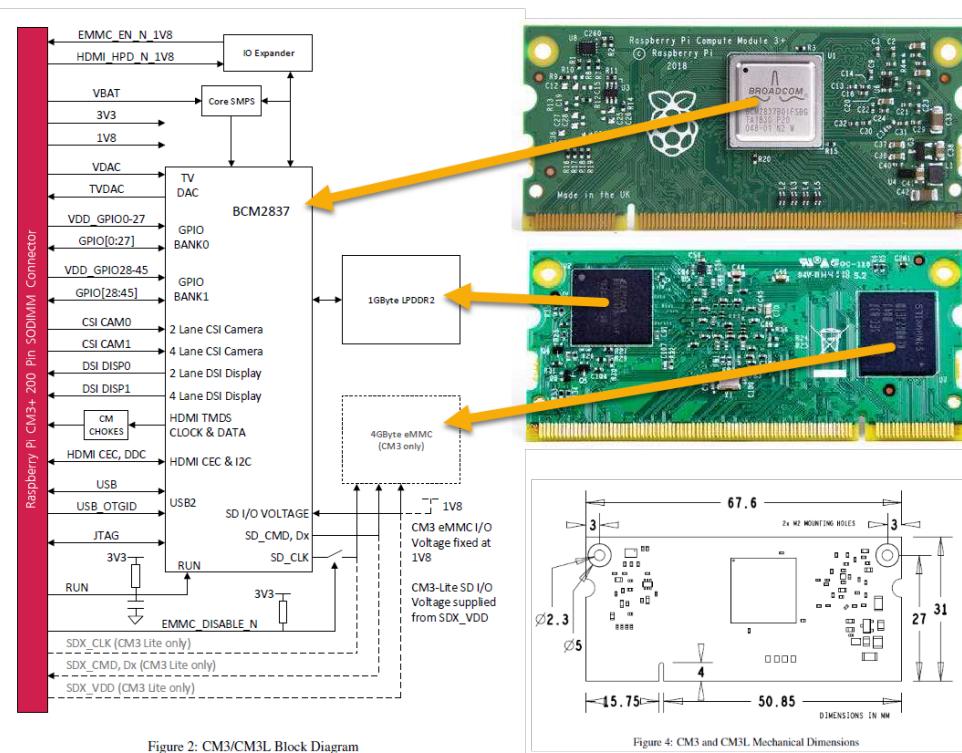
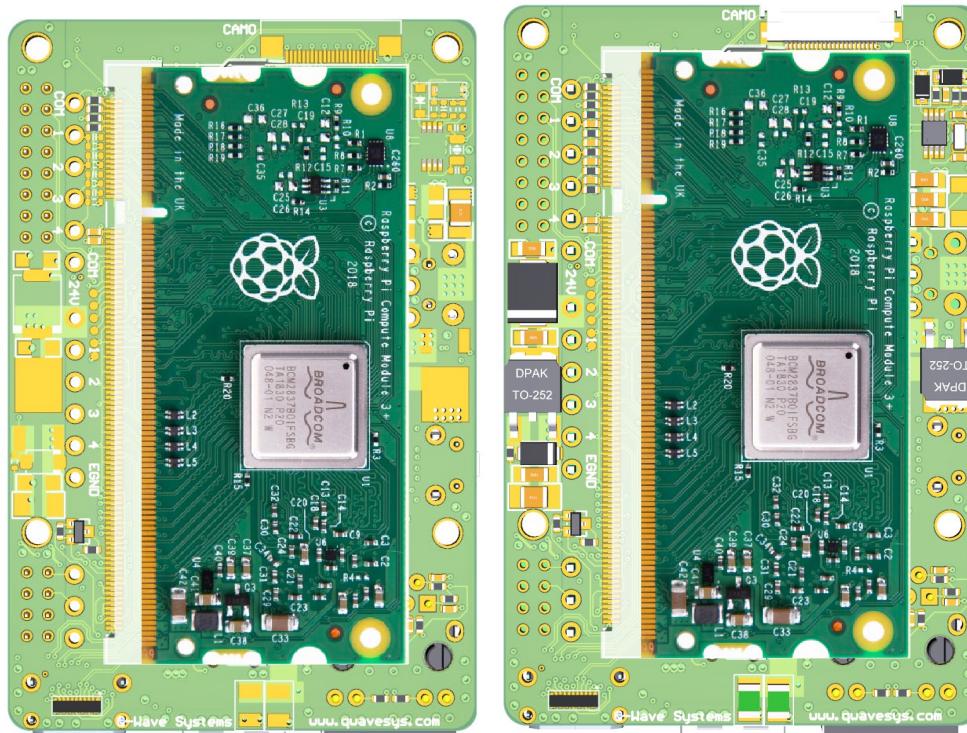


Figure 2: CM3/CM3L Block Diagram

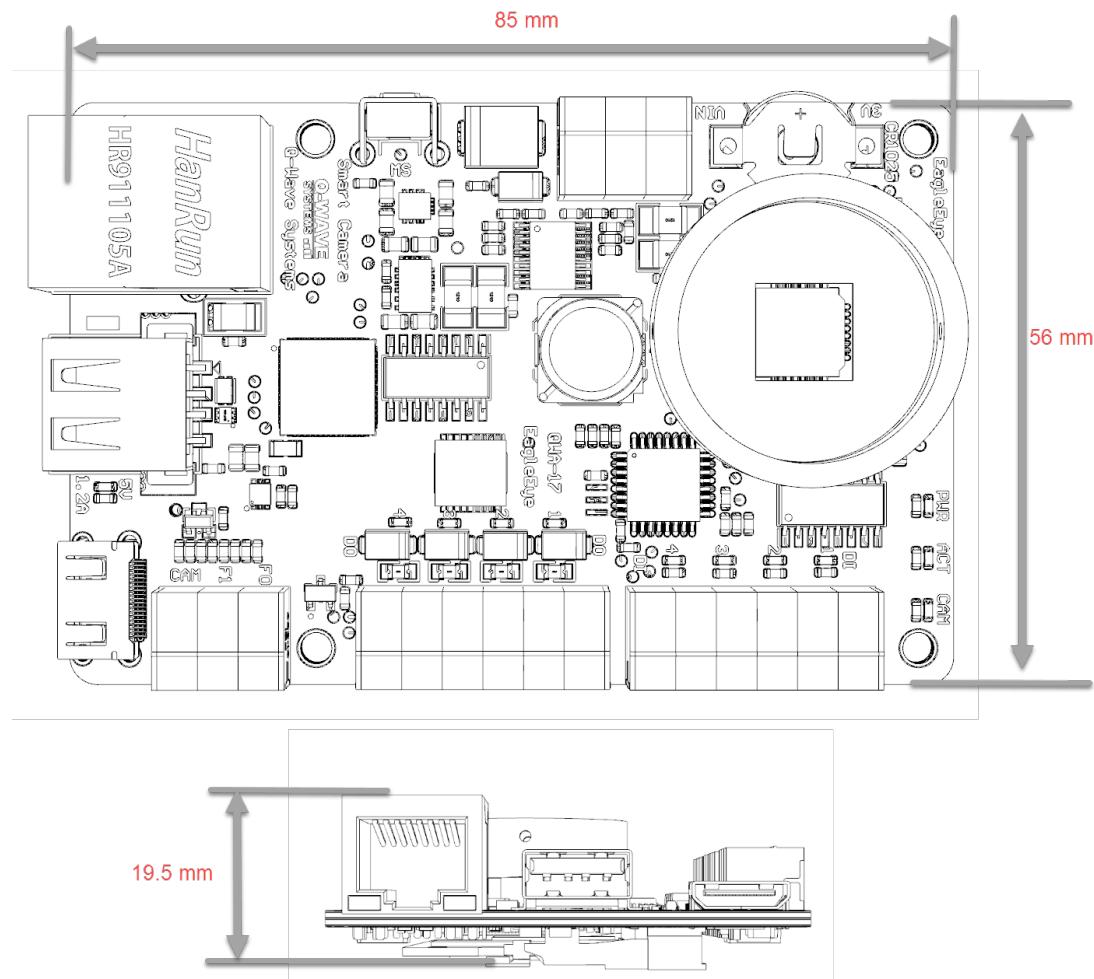
Figure 4: CM3 and CM3L Mechanical Dimensions

## EagleEYE and Raspberry Pi CM3+ Interface Board view

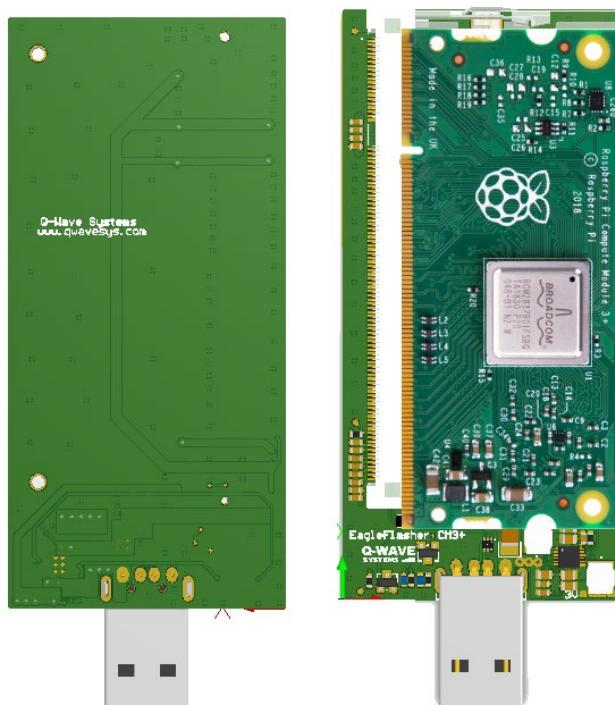


Left (Uno), Right (Industrial)

## Dimension



## Update OS via EagleEYE Flasher CM3+ board



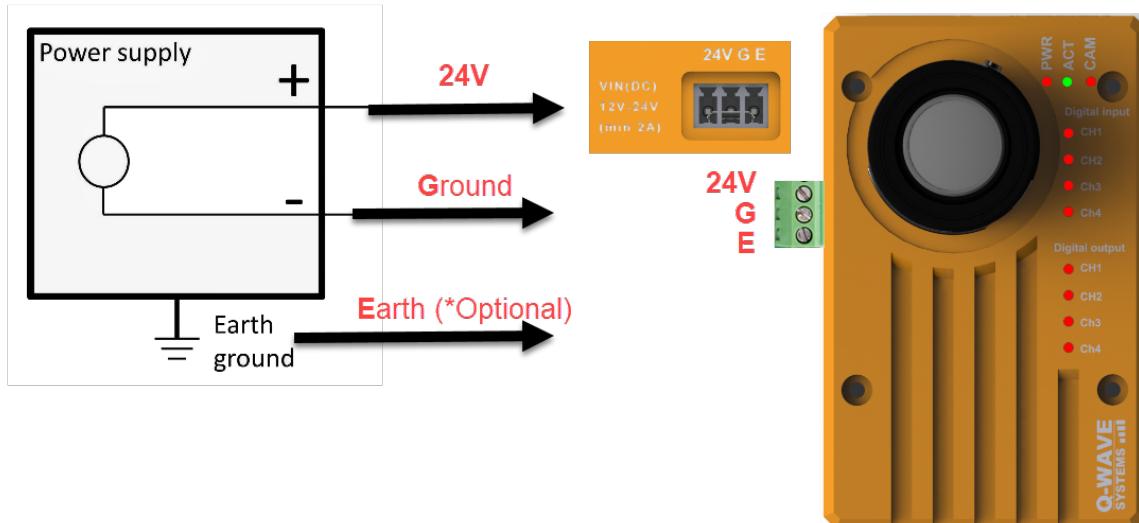
## Power Supply Connection Diagram

Industrial/Pro version:

Power Requirement: 25W Power supply

Vin support range: 5.5V-35V

Typical: 12V or 24V (2A)

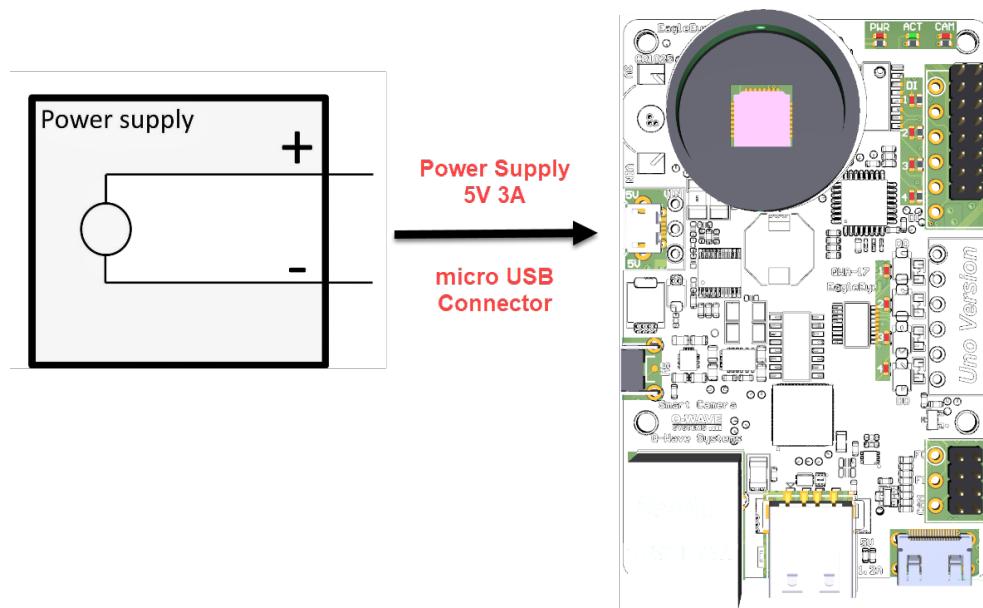


Uno version:

Power Requirement: 15W Power supply

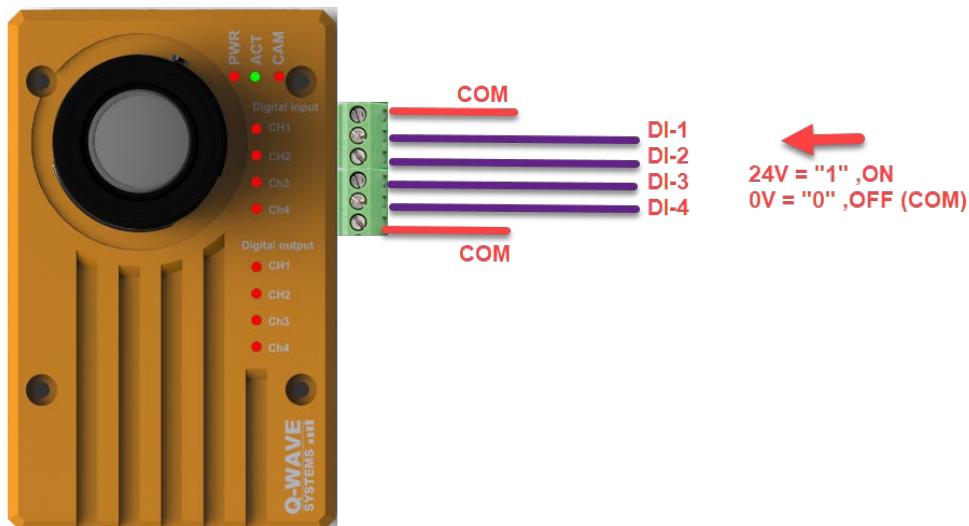
Vin support range: 4.75V-5.25V

Typical: 5V 3A

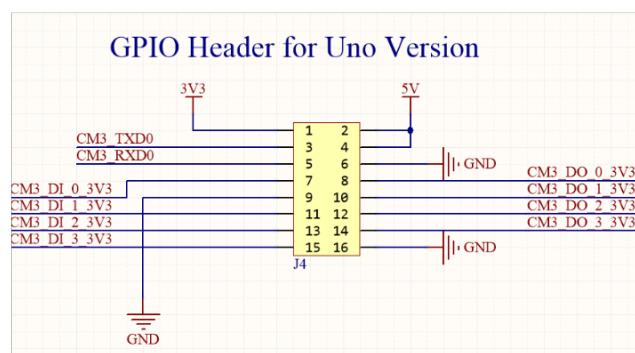
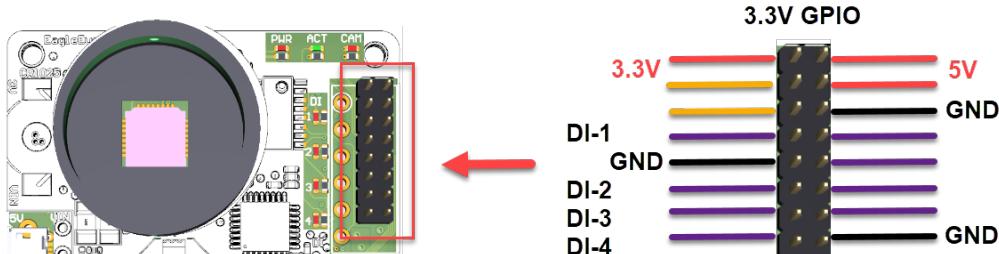


## Digital Input Connection Diagram

Industrial/Pro version: Digital Input 4 CH Isolate

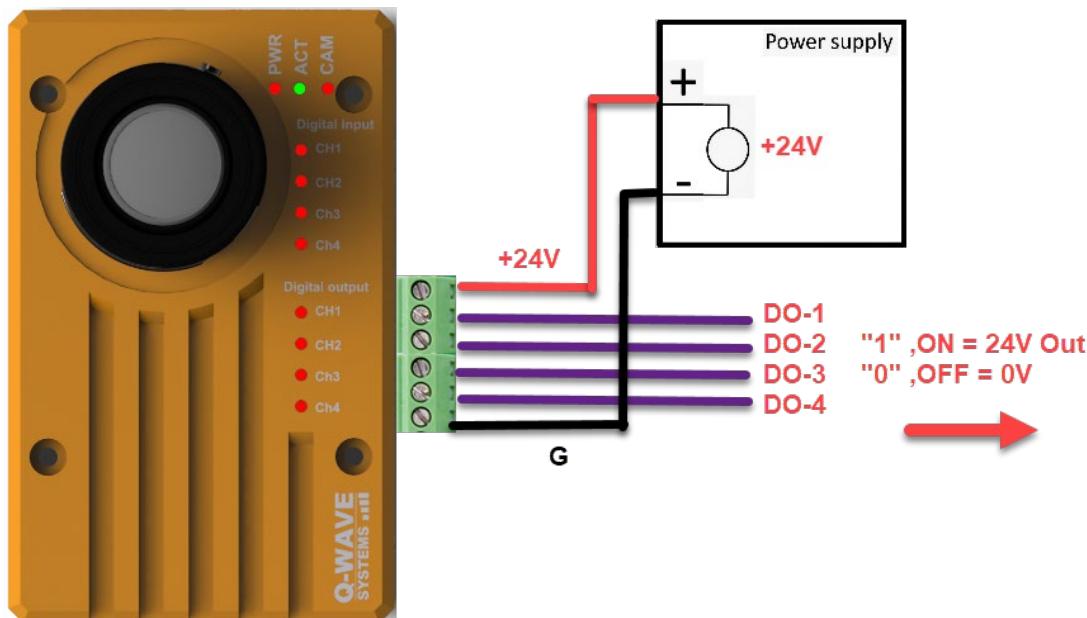


Uno version: 3.3V Input Only

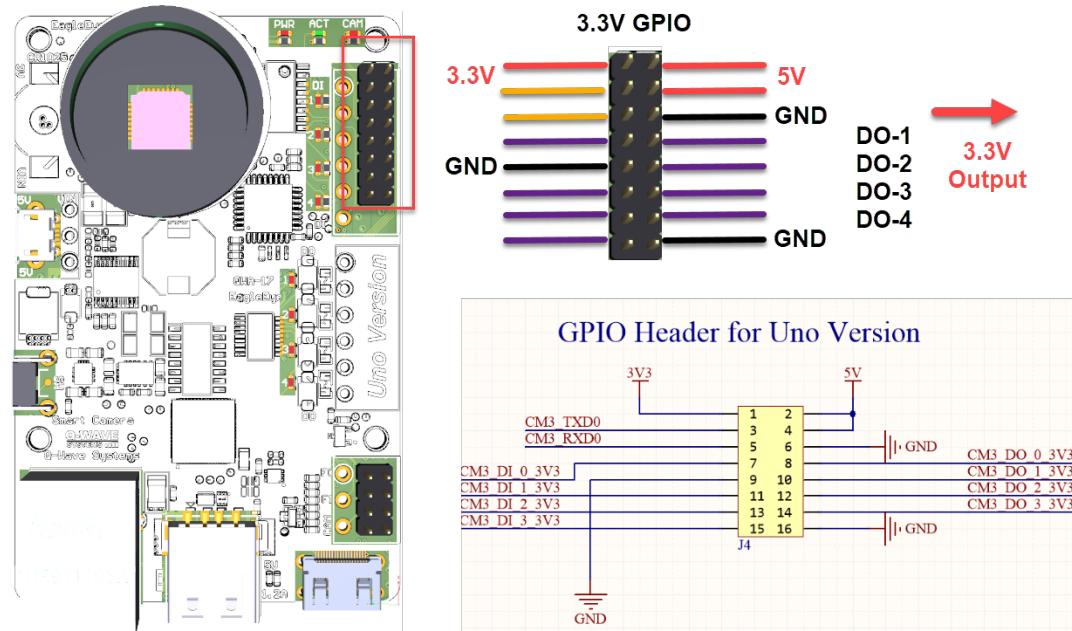


## Digital Output Connection Diagram

Industrial/Pro version: Digital Output 4 CH Isolate



Uno version: 3.3V Output Only (\*total 50mA max)



## CAM status and Light Source Output: RGB (WS2812) Connection Diagram

CAM: 3.3V Output. If camera is ON this pin = 3.3V. If camera is OFF = 0V

L1: 3.3V Output (RGB WS2812)

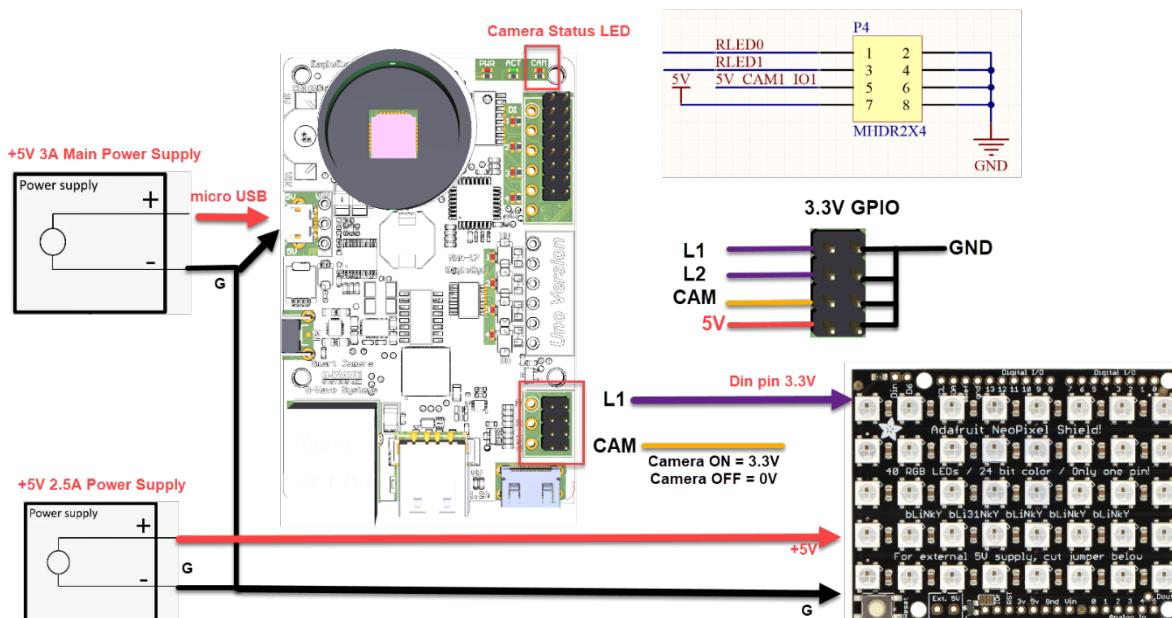
L2: 3.3V Output (RGB WS2812)

\*\*\*L1 and L2 can NOT use both at the same time.

Industrial/Pro version: 3.3V Output



Uno version: 3.3V Output



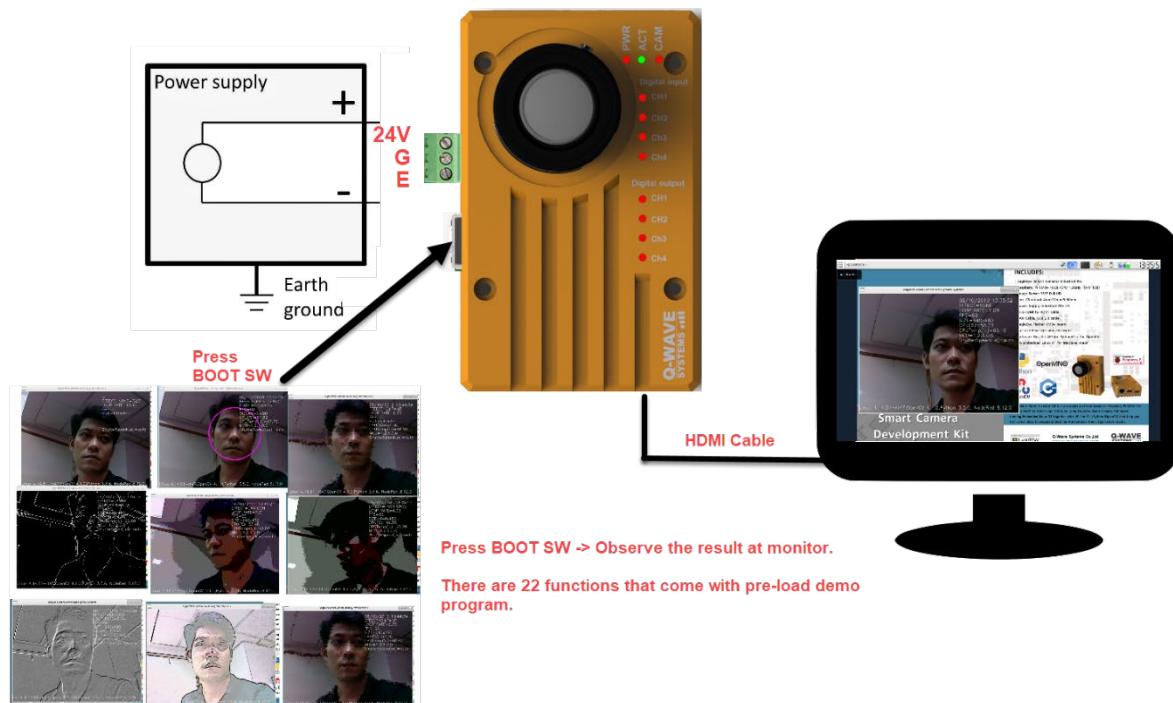
## Getting Start Guide: Connect to HDMI monitor

Once received Smart Camera you can experience with provided demo program by connect power supply (+24V for Industrial/Pro version or +5V USB for Uno version) then connect HDMI cable to monitor. The boot period will be around 15s-20sec for the first time.

The "ACT" (Green) LED blinking indicated that the board is running normally.

The "PWR" (Red) LED indicated that power supply status.

The "CAM" (Red) LED indicated that camera ON/OFF status.

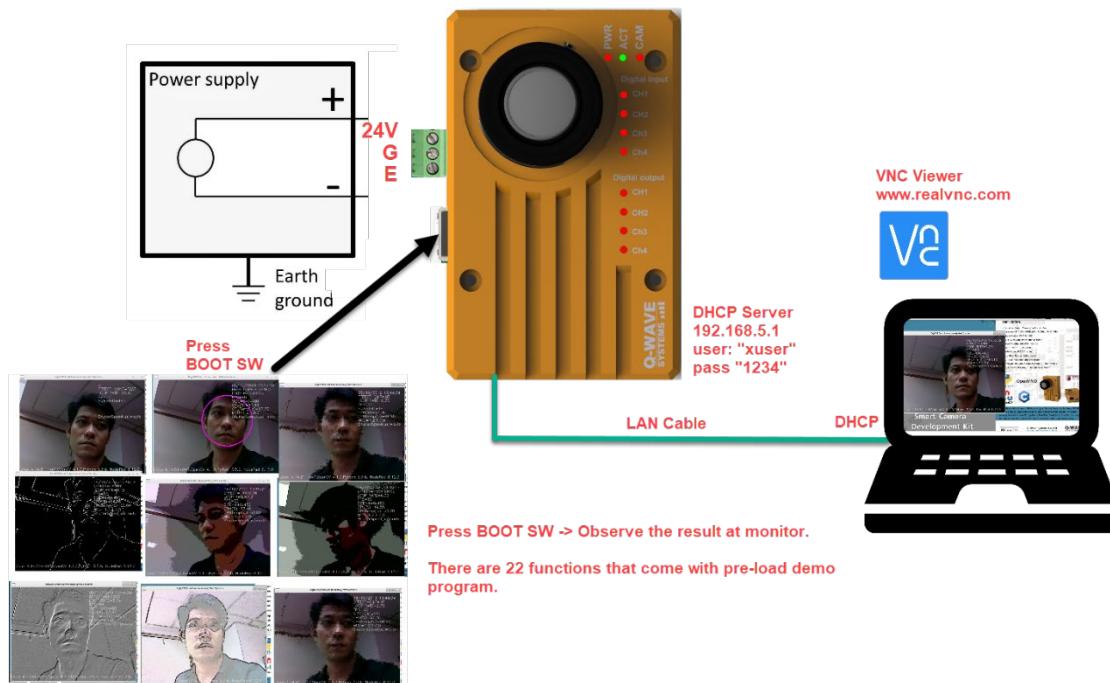


The demo program will be continually read "BOOT SW" status, Once button has pressed the effect will change accordingly, There will be 22 effect display at monitor screen.

At the screen will also display version of pre-load software, library and etc.. for examples.

- Linux Kernel version: 4.14.81
- OpenCV: 4.1.0
- Python: 3.5.6
- NodeRed: 8.12.0

## Getting Start Guide: Connect PC via LAN cable and using VNC viewer.



Smart Camera has setup to be “DHCP server and fix IP address to 192.168.5.1”, Once connect board to PC via LAN cable, The PC will received IP address from Smart Camera for example 192.168.5.xx.

(\*Optional) Open command prompt windows, then type “ipconfig” command to check current IP address on ethernet port.

```
C:\ Command Prompt
Microsoft Windows [Version 10.0.18363.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\MSI>ifconfig
'ifconfig' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\MSI>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

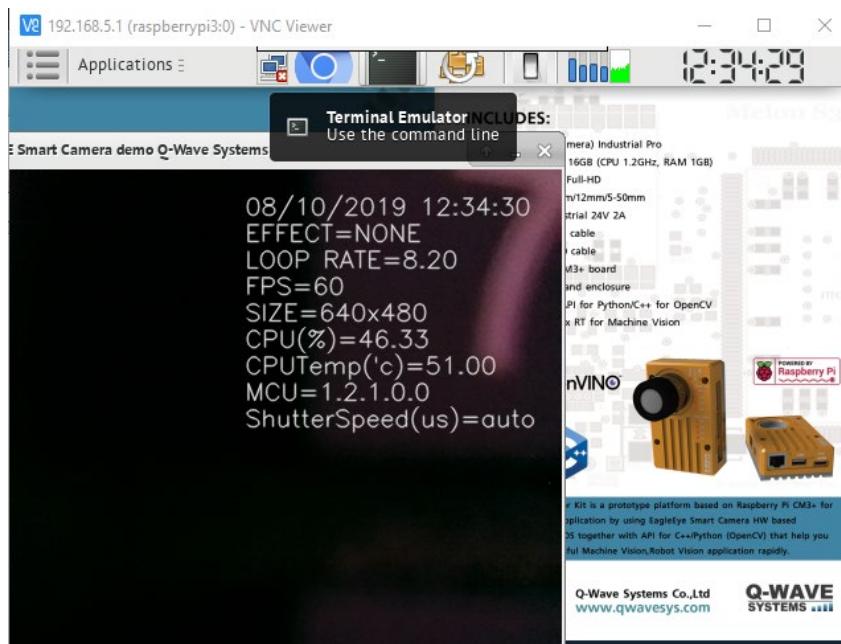
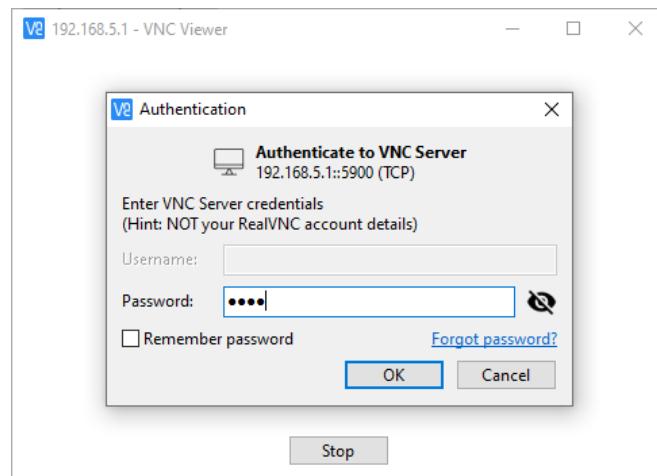
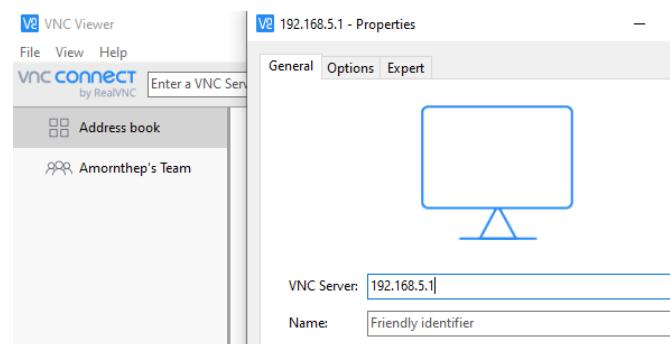
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::88ce:4f34:3eb3:6425%19
IPv4 Address. . . . . : 192.168.5.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
```

Open "VNC Viewer" Connect to Smart Camera using following info.

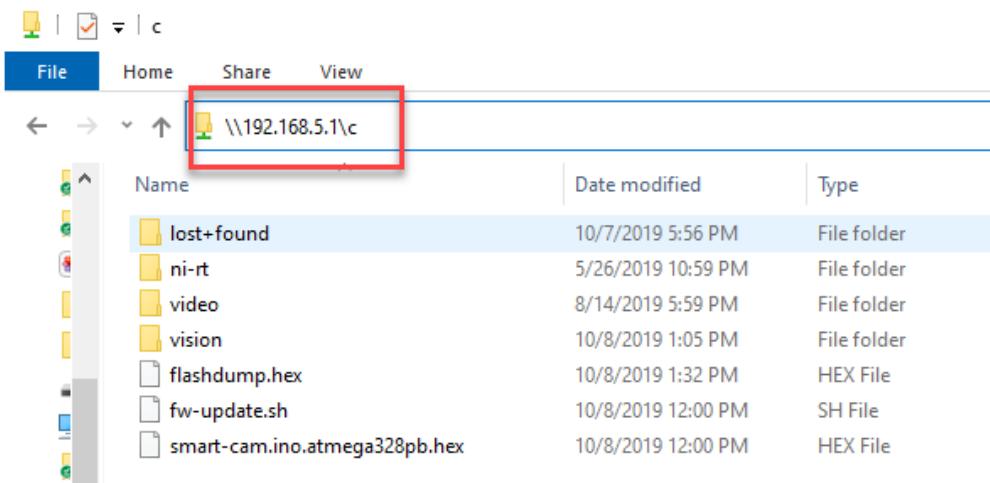
IP Address: "192.168.5.1"

User: "xuser"

Pass "1234"



## Getting Start Guide: File transfer using Network shared drive.



## Getting Start Guide: File transfer using FTP software.

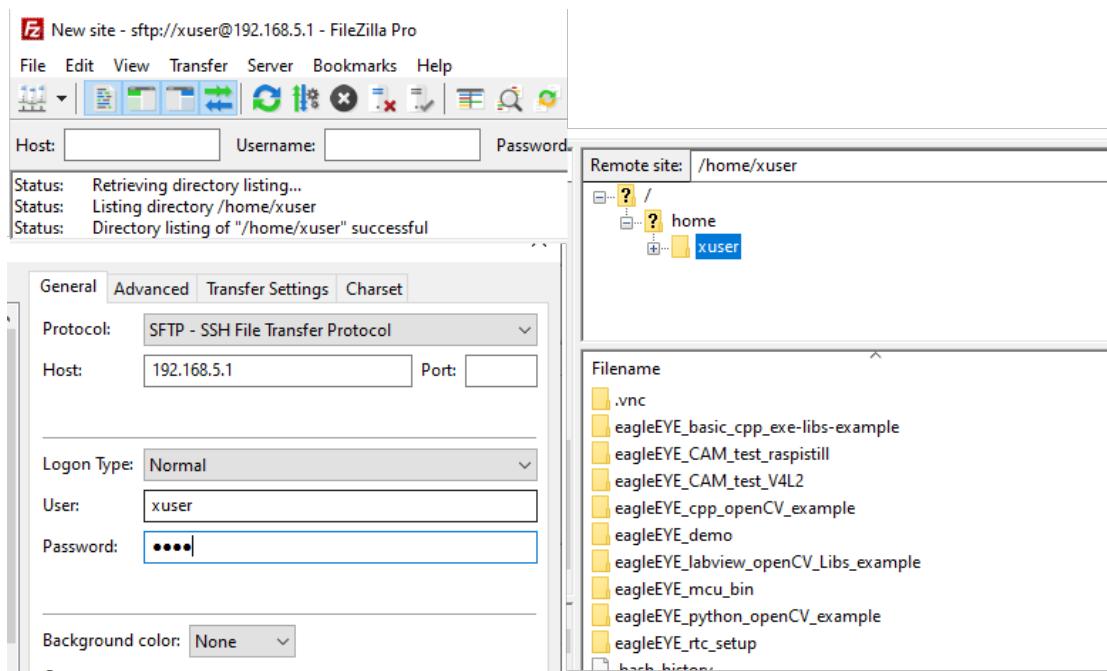
The default home directory for xuser is “/home/xuser”

Protocol: “SFTP”

Host: “192.168.5.1”

User: “xuser” or “root”

Pass: “1234” , There is NO password for root user.

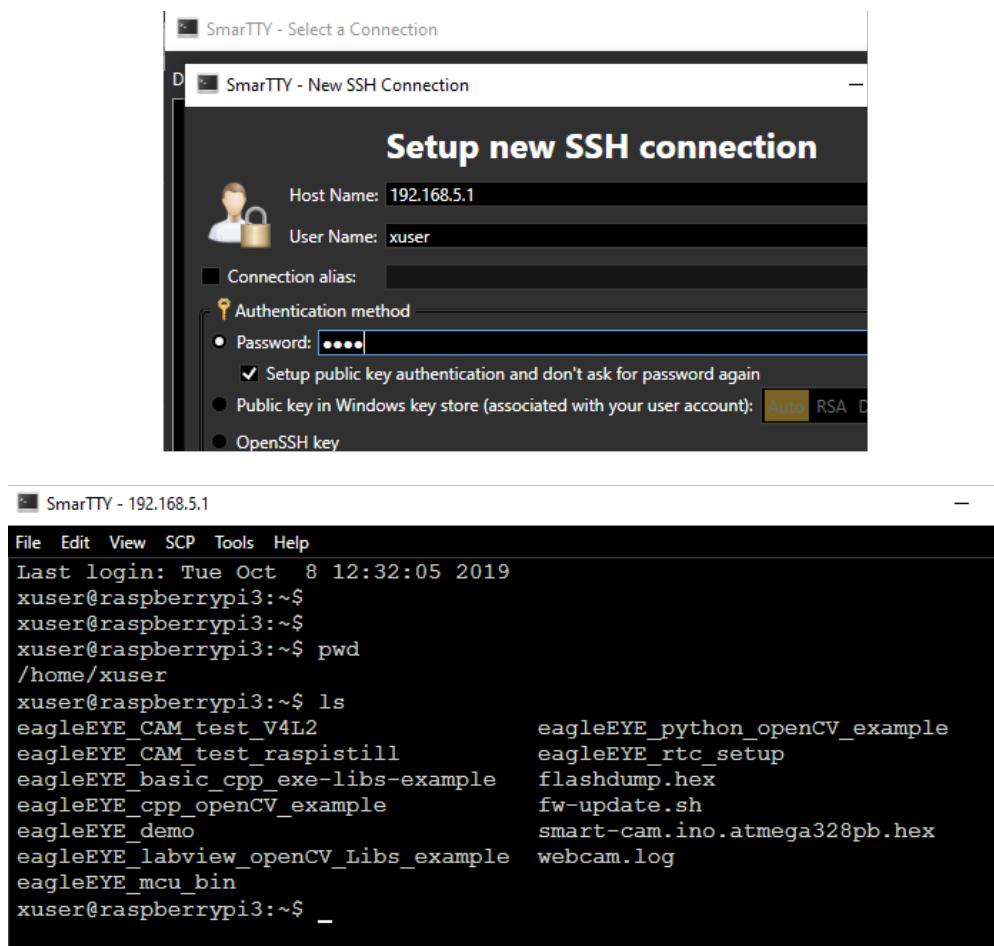


## Getting Start Guide: Remote SSH.

Host Name: "192.168.5.1"

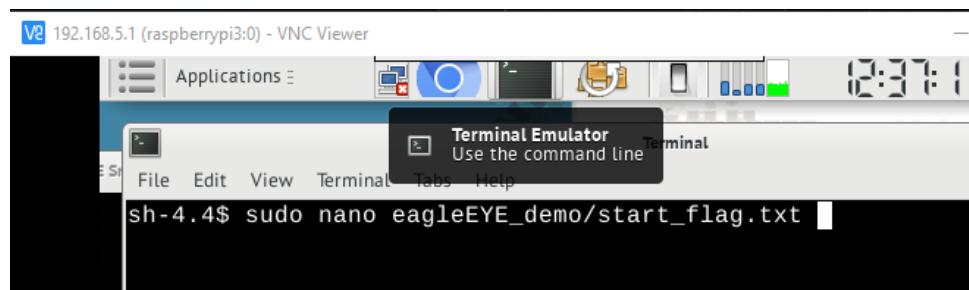
User Name: "xuser" , or "root"

Password: "1234" , There is NO password for root user.



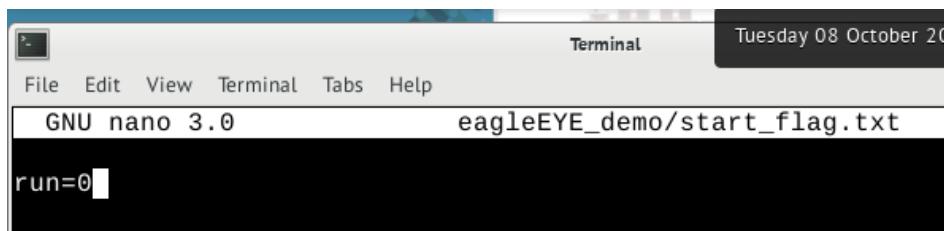
## Getting Start Guide: Turn OFF start-up demo program.

Open "Terminal Emulator", > "sudo nano eagleEYE\_demo/start\_flag.txt"

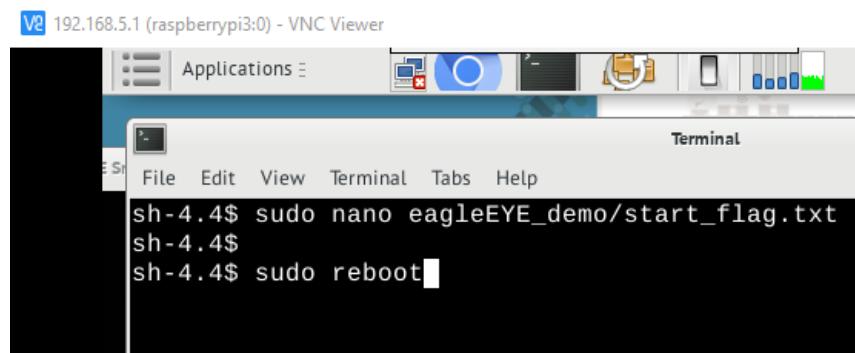


Edit text file to > "run=0"

Press "Ctrl+X" to exit and then press "Shift+Y" enter to save the file.

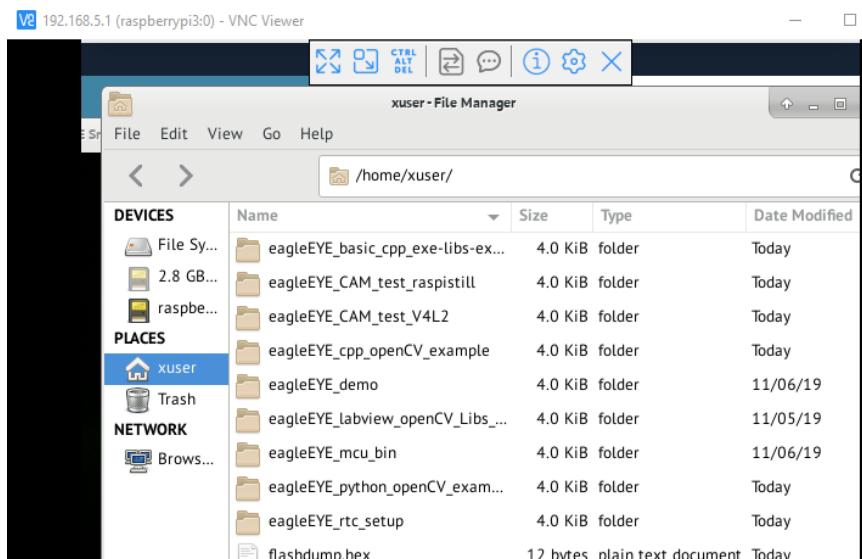


Using "sudo reboot" for reboot the board. After next boot the demo program will be not running anymore. From now you can write your own program to access the camera.



## Getting Start Guide: Examples files.

All example file located at "/home/xuser"



## Getting Start Guide: Test camera using V4L2

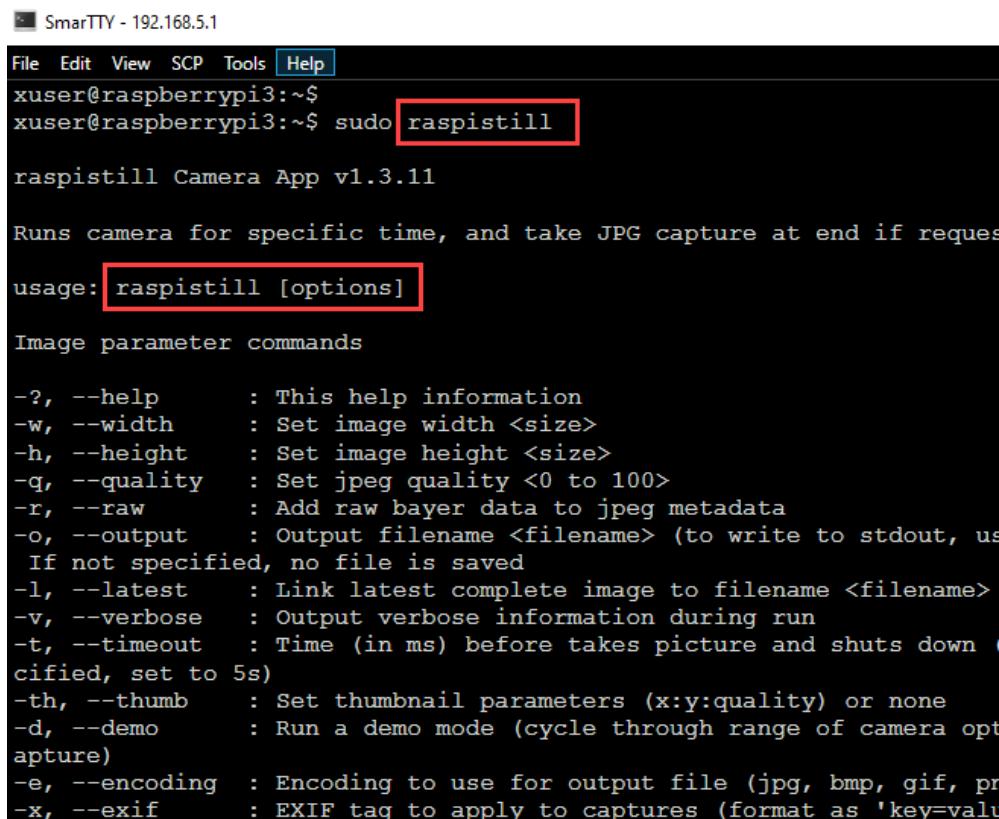
```
> cd eagleEYE_CAM_test_V4L2  
  
> sudo ./V4L2_cam_ON.sh  *for Turn ON camera  
  
> sudo ./V4L2_cam_OFF.sh  *for Turn OFF camera
```

## Getting Start Guide: Test camera using raspistill

```
> cd eagleEYE_CAM_test_raspistill  
  
> sudo ./raspistill_15sec.sh
```

Raspistill camera application 1.3.11  
command line tool for capturing still photographs with the camera module.

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspistill.md>  
[https://elinux.org/Rpi\\_Camera\\_Module#RaspiStill](https://elinux.org/Rpi_Camera_Module#RaspiStill)



The screenshot shows a terminal window titled "SmarTTY - 192.168.5.1". The window has a menu bar with File, Edit, View, SCP, Tools, and Help. The Help menu is currently selected. The terminal prompt is "xuser@raspberrypi3:~\$". The user types "sudo raspistill" and presses Enter. The terminal then displays the raspistill Camera App v1.3.11 help text. The help text includes the usage command "usage: raspistill [options]" and a detailed list of image parameter commands with their descriptions. Several lines of the help text are highlighted with red boxes, specifically the "usage:" line and several command descriptions.

```
File Edit View SCP Tools Help  
xuser@raspberrypi3:~$  
xuser@raspberrypi3:~$ sudo raspistill  
  
raspistill Camera App v1.3.11  
  
Runs camera for specific time, and take JPG capture at end if requested.  
usage: raspistill [options]  
  
Image parameter commands  
  
-?, --help      : This help information  
-w, --width     : Set image width <size>  
-h, --height    : Set image height <size>  
-q, --quality   : Set jpeg quality <0 to 100>  
-r, --raw        : Add raw bayer data to jpeg metadata  
-o, --output    : Output filename <filename> (to write to stdout, use - to  
If not specified, no file is saved  
-l, --latest     : Link latest complete image to filename <filename>  
-v, --verbose    : Output verbose information during run  
-t, --timeout   : Time (in ms) before takes picture and shuts down (specify, set to 5s)  
-th, --thumb     : Set thumbnail parameters (x:y:quality) or none  
-d, --demo       : Run a demo mode (cycle through range of camera options)  
-e, --encoding   : Encoding to use for output file (jpg, bmp, gif, png)  
-x, --exif       : EXIF tag to apply to captures (format as 'key=value')
```

Raspivid camera application 1.3.12  
command line tool for capturing video with the camera module

<https://www.raspberrypi.org/documentation/usage/camera/raspicam/raspivid.md>  
[https://elinux.org/Rpi\\_Camera\\_Module#RaspiVid](https://elinux.org/Rpi_Camera_Module#RaspiVid)

```
xuser@raspberrypi3:~$ sudo raspivid
raspivid Camera App v1.3.12

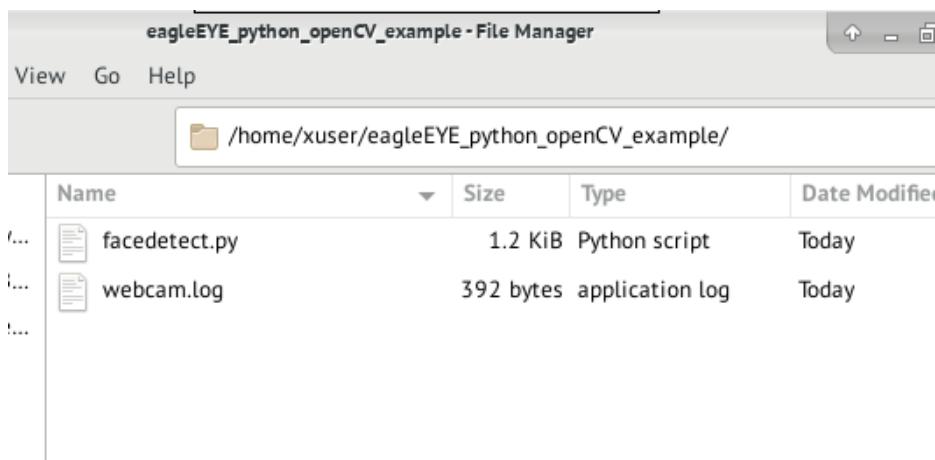
Display camera output to display, and optionally saves an H264 capture at
estimated bitrate

usage: raspivid [options]

Image parameter commands

-?, --help      : This help information
-w, --width     : Set image width <size>. Default 1920
-h, --height    : Set image height <size>. Default 1080
-b, --bitrate   : Set bitrate. Use bits per second (e.g. 10MBits/s would be
10000000)
-o, --output    : Output filename <filename> (to write to stdout, use '-o /dev/stdout')
                  Connect to a remote IPv4 host (e.g. tcp://192.168.1.2:1234)
                  To listen on a TCP port (IPv4) and wait for an incoming
connection use -l
                  (e.g. raspivid -l -o tcp://0.0.0.0:3333 -> bind to all network
interfaces, raspivid -l -o tcp://192.168.1.1:3333 -> bind to a certain
IPv4)
```

## Getting Start Guide: Python 3.5.6 face detect app using OpenCV 4.1.0

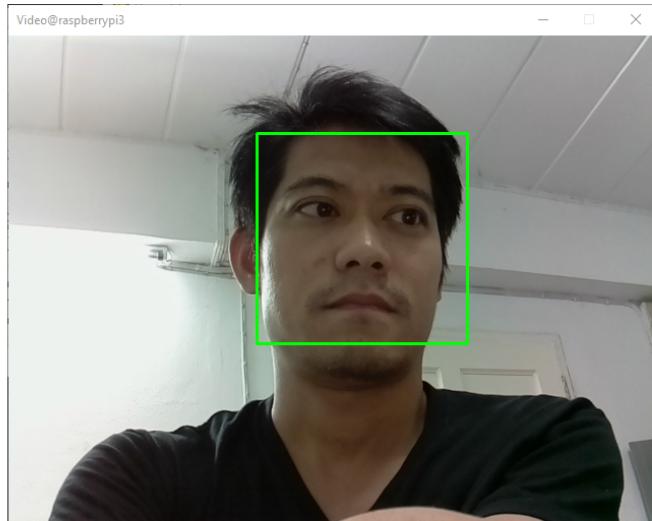


> cd eagleEYE\_python\_OpenCV\_example

> sudo python3 facedetect.py

\*\*press Ctrl+c to terminate the program

```
xuser@raspberrypi3:~$  
xuser@raspberrypi3:~$ cd eagleEYE_python_openCV_example/  
xuser@raspberrypi3:~/eagleEYE_python_openCV_example$  
xuser@raspberrypi3:~/eagleEYE_python_openCV_example$  
xuser@raspberrypi3:~/eagleEYE_python_openCV_example$  
xuser@raspberrypi3:~/eagleEYE_python_openCV_example$ ls  
facedetect.py  
xuser@raspberrypi3:~/eagleEYE_python_openCV_example$ sudo python3 facedetect.py  
libGL error: No matching fbConfigs or visuals found  
libGL error: failed to load driver: swrast  
(python3:1303): dbind-WARNING **: 12:40:56.645: Error retrieving accessibility  
bus address: org.ally.Bus.Error: Failed to execute child process ?/usr/bin? (Permission denied)
```



Using nano or others text editor to view/edit the source code.

```
> sudo nano facedetect.py  
**press Ctrl+x to exit from nano editor.
```

```
GNU nano 3.0          facedetect.py

import cv2
import sys
import logging as log
import datetime as dt
from time import sleep

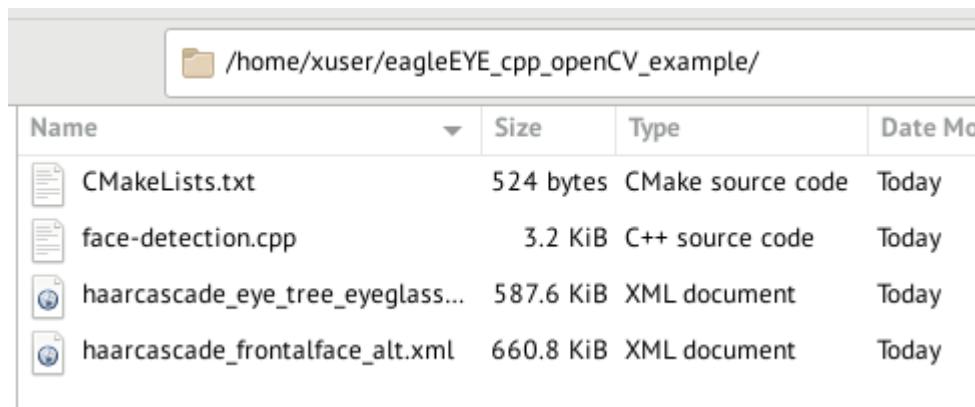
cascPath = "/c/vision/haarcascade_frontalface_alt.xml"
faceCascade = cv2.CascadeClassifier(cascPath)
log.basicConfig(filename='webcam.log', level=log.INFO)

video_capture = cv2.VideoCapture(0)
anterior = 0

while True:
    if not video_capture.isOpened():
        print('Unable to load camera.')
        sleep(5)
        pass

    # Capture frame-by-frame
    ret, frame = video_capture.read()
```

## Getting Start Guide: Face detect app using C++ and OpenCV 4.1.0



```
> cd eagleEYE_cpp_openCV_example
```

```
> mkdir build
```

```
> cd build
```

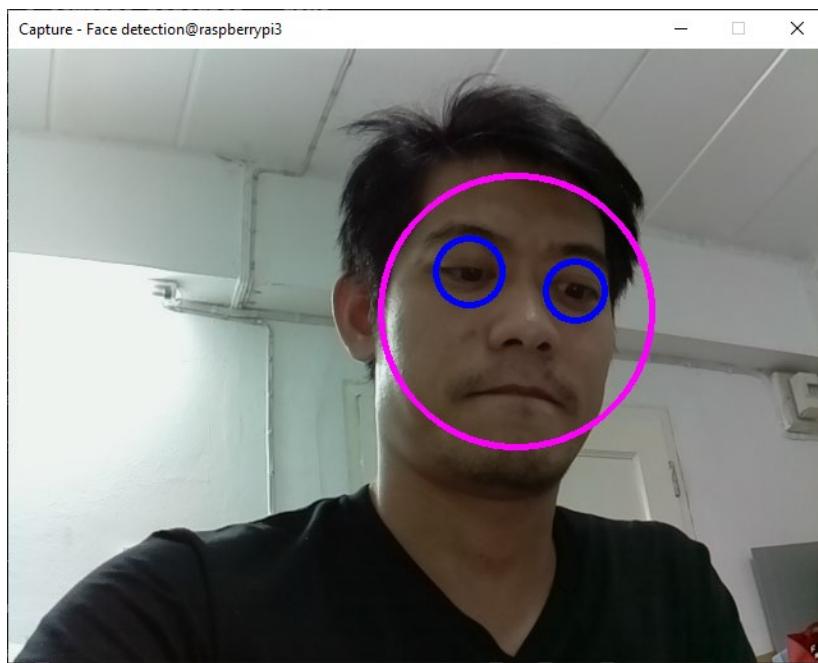
```
> cmake ..
```

```
> make
```

```
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example$ cd eagleEYE_cpp_openCV_example/
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example$ ls
CMakeLists.txt      haarcascade_eye_tree_eyeglasses.xml
build               haarcascade_frontalface_alt.xml
face-detection.cpp
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example$ ls
CMakeLists.txt      haarcascade_eye_tree_eyeglasses.xml
face-detection.cpp  haarcascade_frontalface_alt.xml
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example$ mkdir build
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example$ cd build
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example/build$ cmake ..
-- The C compiler identification is GNU 8.2.0
-- The CXX compiler identification is GNU 8.2.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found OpenCV: /usr (found version "4.1.0")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/xuser/eagleEYE_cpp_openCV_example/build
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example/build$ make
Scanning dependencies of target face-detection
[ 50%] Building CXX object CMakeFiles/face-detection.dir/face-detection.cpp.o
[100%] Linking CXX executable face-detection
[100%] Built target face-detection
xuser@raspberrypi3:~/eagleEYE_cpp_openCV_example/build$ _
```

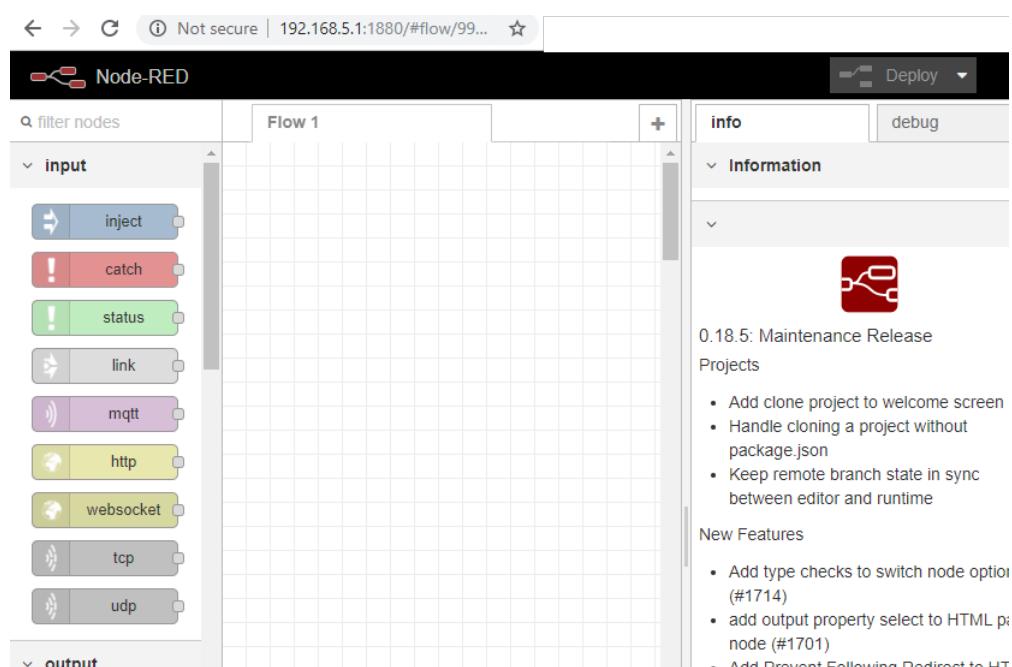
Run the program "face-detection"

```
> sudo ./face-detection  
**press Ctrl+c to terminate the program
```

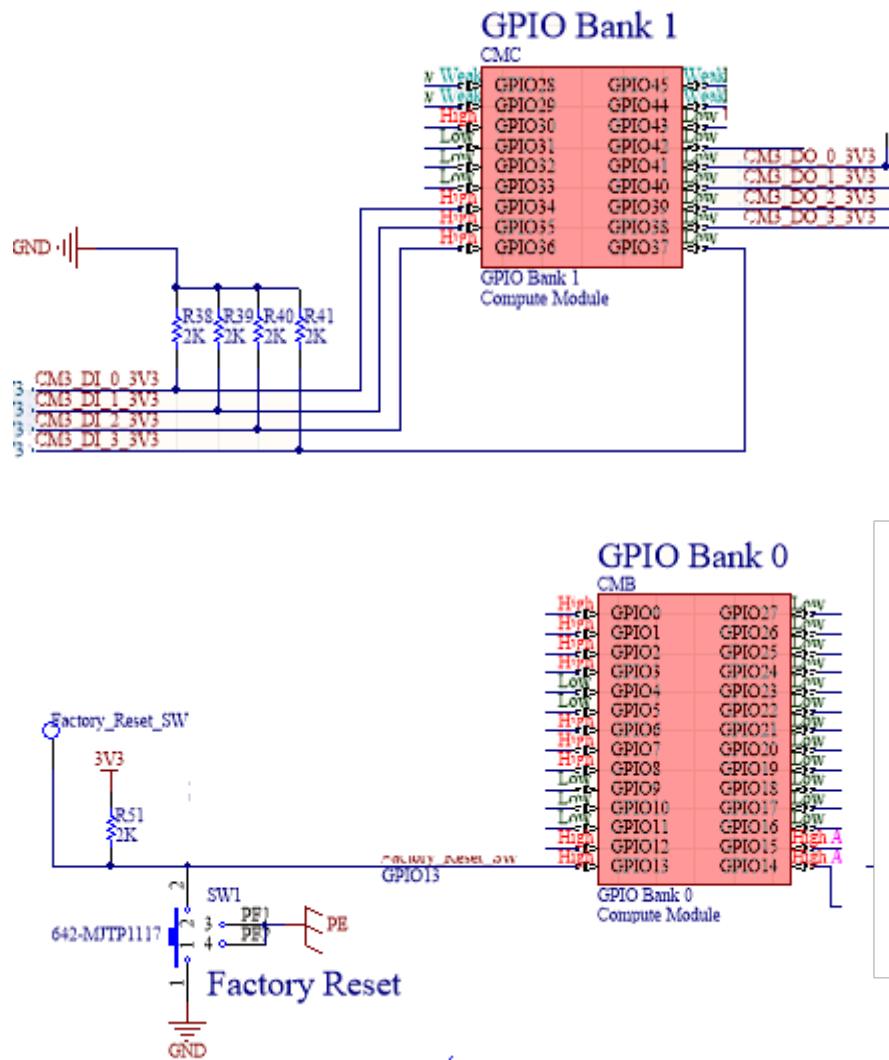


## Getting Start Guide: Using NodeRed program

At remoted PC, Open web browser for example "Google Chrome" goto this link  
["http://192.168.5.1:1880/"](http://192.168.5.1:1880/)



## Getting Start Guide: GPIO access.



DI-1 = GPIO34 (Input)

DI-2 = GPIO35 (Input)

DI-3 = GPIO36 (Input)

DI-4 = GPIO37 (Input)

DO-1 = GPIO41 (Output)

DO-2 = GPIO39 (Output)

DOI-3 = GPIO38 (Output)

DO-4 = GPIO37 (Output)

BOOT SW = GPIO13 (Input)

Using WiringPi: The GPIO utility for testing

```
sh-4.4$ gpio -v
gpio version: 2.54
Copyright (c) 2012-2018 Gordon Henderson
This is free software with ABSOLUTELY NO WARRANTY.
For details type: gpio -warranty

Raspberry Pi Details:
  Type: CM3+, Revision: 00, Memory: 1024MB, Maker: Sony
  * Device tree is enabled.
  *--> Raspberry Pi Compute Module 3 Plus Rev 1.0
  * This Raspberry Pi supports user-level GPIO access.
sh-4.4$ █
```

Example: Turn ON/OFF DO-1 (GPIO41) using GPIO utility

```
> gpio mode 41 out
```

```
> gpio write 41 1
```

```
> gpio write 41 0
```

More detail please refer this link.

<http://wiringpi.com/the-gpio-utility/>

## Getting Start Guide: using LabVIEW 2017 with Smart Camera EagleEYE

Evaluation Version can be download from this link

[https://ntsolutionscorp-my.sharepoint.com/:f/q/personal/amornthep\\_ntsolutionscorp\\_onmicrosoft\\_com/Ev\\_nLw7BrMIDk6TOlYmCZABtenswqpb-SiGfaAtMhFJkA?e=2tENE2](https://ntsolutionscorp-my.sharepoint.com/:f/q/personal/amornthep_ntsolutionscorp_onmicrosoft_com/Ev_nLw7BrMIDk6TOlYmCZABtenswqpb-SiGfaAtMhFJkA?e=2tENE2)

s > LabVIEW\_2017\_EagleEYE ↗

Name	Modified	Modified By	File Size
1.2017LV-WinEng.exe	23 พฤษภาคม, 2560	Amornthep Phunsi	1.40 GB
2.2017RealTime-Eng.exe	20 พฤษภาคม	Amornthep Phunsi	1.03 GB
3.VISION_2017_SP1.zip	20 พฤษภาคม	Amornthep Phunsi	2.45 GB
4.NIVISA1700full.exe	22 มิถุนายน	Amornthep Phunsi	732 MB
q_wave_systems_lib_qwaveeagleeye-1.0....	เมื่อวาน at 20:36	Amornthep Phunsi	6.15 MB

### Requirement software

LabVIEW 2017 (32bit)

LabVIEW Real-Time module 2017 (32bit)

LabVIEW Vision Development SP1 2017 (32bit)

q\_wave\_systems\_lib\_qwaveeagleeye-1.0.0.x.vip

After installed "q\_wave\_systems\_lib\_qwaveeagleeye-1.0.0.x.vip",

The examples located at "C:\Program Files (x86)\National Instruments\LabVIEW 2017\vi.lib\Q-Wave Systems\QwaveEagleEYE\Examples"

Home Share View

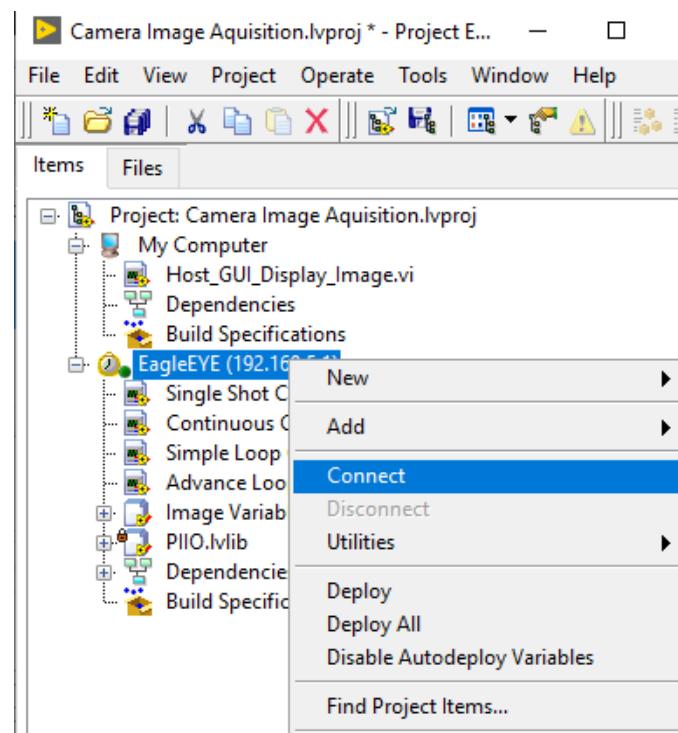
▼ C:\Program Files (x86)\National Instruments\LabVIEW 2017\vi.lib\Q-Wave Systems\QwaveEagleEYE\Examples

Name	Date modified	Type	Size
_eagleEYE_basic_cpp_exe-libs-example	11/18/2019 11:19 PM	File folder	
_eagleEYE_CAM_test_raspistill	11/18/2019 11:19 PM	File folder	
_eagleEYE_CAM_test_V4L2	11/18/2019 11:19 PM	File folder	
_eagleEYE_cpp_openCV_example	11/18/2019 11:19 PM	File folder	
_eagleEYE_demo	11/18/2019 11:19 PM	File folder	
_eagleEYE_labview_openCV_Libs_example	11/18/2019 11:19 PM	File folder	
_eagleEYE_python_openCV_example	11/18/2019 11:19 PM	File folder	
Camera Image Aquisition	11/18/2019 11:19 PM	File folder	
Demo Real Time processing with display	11/18/2019 11:19 PM	File folder	
Digital Input Output	11/18/2019 11:19 PM	File folder	
Display Image with Processing at RPi monitor	11/18/2019 11:19 PM	File folder	
Light Source Output WS2812 RGB	11/18/2019 11:19 PM	File folder	
MachineVision_AOI PCB Inspector	11/18/2019 11:19 PM	File folder	
MachineVision_Barcod Detector	11/18/2019 11:19 PM	File folder	
MachineVision_Barcod Reader	11/18/2019 11:19 PM	File folder	
MachineVision_Pills Color Inspector	11/18/2019 11:19 PM	File folder	
MachineVision_Water Level Measure	11/18/2019 11:19 PM	File folder	
MachineVision_Wood Surface Inspection	11/18/2019 11:19 PM	File folder	
Network Communication_PID_HMI	11/18/2019 11:19 PM	File folder	
OpenCV Face Detect	11/18/2019 11:19 PM	File folder	
OpenCV Function Development	11/18/2019 11:19 PM	File folder	
OpenCV-IMAQ Conversion	11/18/2019 11:19 PM	File folder	
RT FIFO Variables_NetworkShare	11/18/2019 11:19 PM	File folder	

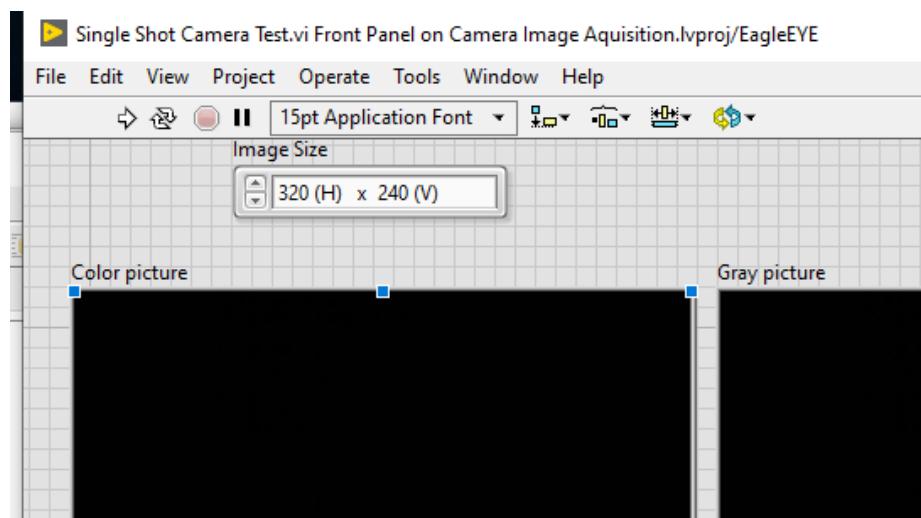
Test acquired image from camera, Open "Camera Image Acquisition\Camera Image Aquisition.lvproj"

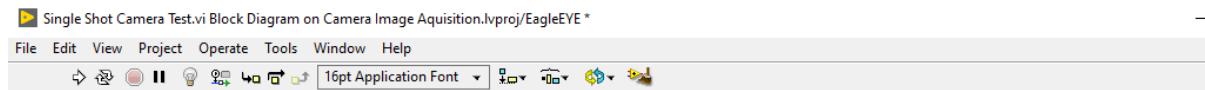
.. \ National Instruments \ LabVIEW 2017 \ vi.lib \ Q-Wave Sy

Name	Date modified
Advance Loop Get Images.vi	11/18/2019 11:19 PM
Camera Image Aquisition.aliases	11/22/2019 1:26 AM
Camera Image Aquisition.lvproj	11/18/2019 8:59 PM
Continuous Camera Test.vi	11/18/2019 11:19 PM
Host_GUI_Display_Image.vi	11/18/2019 11:19 PM
Image Variables.lvlib	11/18/2019 10:57 PM
Simple Loop Get Images to Host GUI.vi	11/18/2019 11:19 PM
Single Shot Camera Test.vi	11/18/2019 11:19 PM

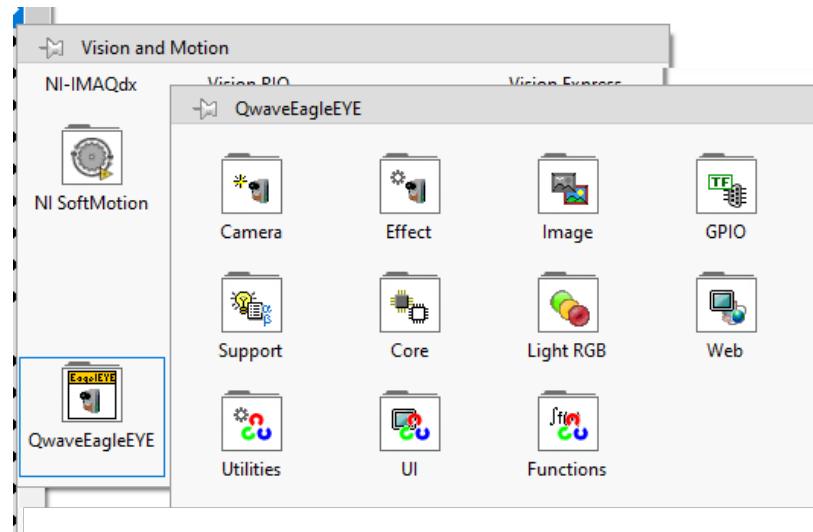


Run "Single Shot Camera Test.vi"





The EagleEYE library are located at "Vision and Motion\QwaveEagleEYE" pallet.

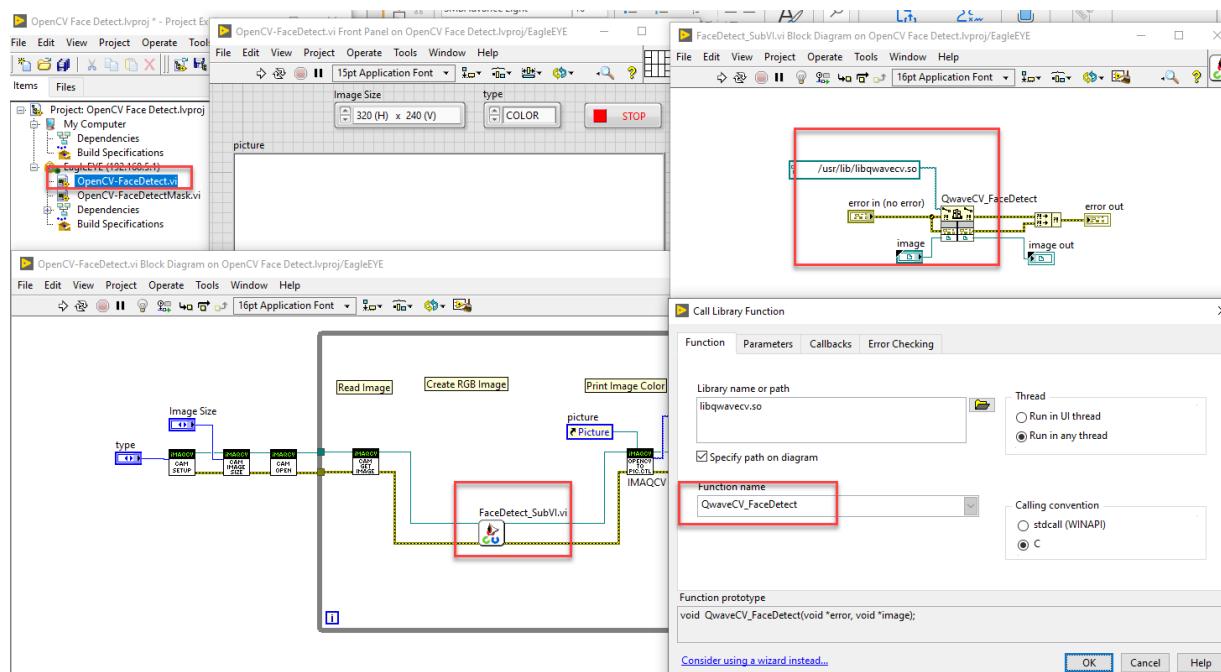


## Getting Start Guide: using LabVIEW 2017 and OpenCV function development.

Open "OpenCV Face Detect\ OpenCV Face Detect.lvproj"

Name	Date modified	Type
masks	11/18/2019 11:19 PM	File folder
FaceDetect.cpp	10/25/2018 2:14 PM	C++ Source File
FaceDetect_SubVI.vi	11/18/2019 11:19 PM	LabVIEW Instrument
FaceDetectMask.cpp	11/20/2018 3:57 PM	C++ Source File
FaceDetectMask_SubVI.vi	11/18/2019 11:19 PM	LabVIEW Instrument
haarcascade_frontalface_alt2.xml	10/25/2018 2:07 PM	XML Document
Learn OpenCV by Examples- Face Detection us...	11/21/2018 12:38 AM	Internet Shortcut
Learn OpenCV by Examples- Putting a mask on ...	11/21/2018 12:37 AM	Internet Shortcut
OpenCV Face Detect.alises	11/22/2019 1:33 AM	ALIASES File
OpenCV Face Detect.lvproj	11/18/2019 11:09 AM	LabVIEW Project
OpenCV-FaceDetect.vi	11/18/2019 11:19 PM	LabVIEW Instrument
OpenCV-FaceDetectMask.vi	11/18/2019 11:19 PM	LabVIEW Instrument

Navigated to "OpenCV-FaceDetect.vi"



The source code for C++ function is "FaceDetect.cpp" or "FaceDetectMask.cpp"

Name	Size	Type	Date Modified
build	4.0 KiB	folder	11/05/19
masks	4.0 KiB	folder	11/05/19
CMakeLists.txt	316 bytes	CMake source code	11/05/19
FaceDetect.cpp	1.3 KiB	C++ source code	11/05/19
FaceDetectMask.cpp	2.7 KiB	C++ source code	11/05/19
haarcascade_frontalface_alt2.xml	527.9 KiB	XML document	11/05/19

Steps to build share library “libqwavecv.so”

```
> cd eagleEYE_labview_OpenCV_Libs_example
```

```
> mkdir build
```

```
> cd build
```

```
> cmake ..
```

```
> make
```

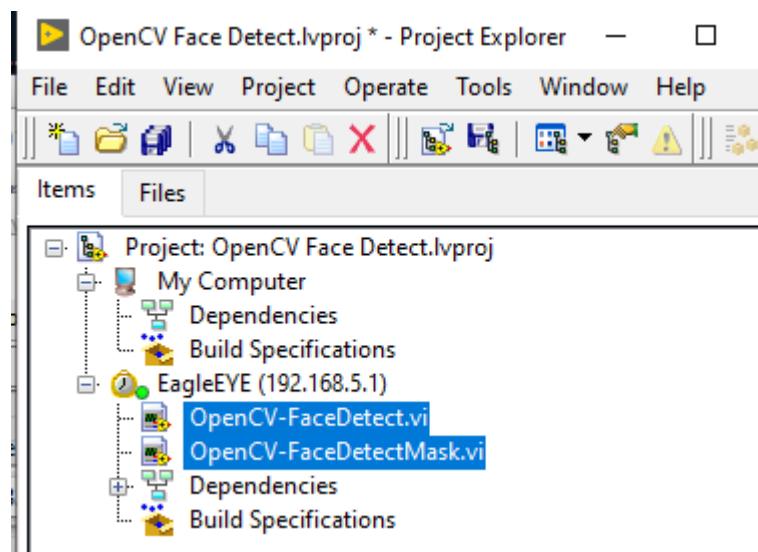
```
> sudo make install
```

\*The library located at “usr/lib/libqwavecv.so”

```

/home/xuser
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example/
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_examples$ ls
CMakeLists.txt FaceDetectMask.cpp haarcascade_frontalface_alt2.xml
FaceDetect.cpp build masks
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example$ mkdir build
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example$ cd build
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example/build$ cmake ..
-- The C compiler identification is GNU 8.2.0
-- The CXX compiler identification is GNU 8.2.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found OpenCV: /usr (found version "4.1.0")
-- Configuring done
-- Generating done
-- Build files have been written to: /home/xuser/eagleEYE_labview_OpenCV_Libs_
example/build
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example/build$ make
-- Configuring done
-- Generating done
-- Build files have been written to: /home/xuser/eagleEYE_labview_OpenCV_Libs_
example/build
Scanning dependencies of target qwavecv
make[2]: Warning: File '../FaceDetect.cpp' has modification time 2453571 s in
the future
[ 33%] Building CXX object CMakeFiles/qwavecv.dir/FaceDetect.cpp.o
[ 66%] Building CXX object CMakeFiles/qwavecv.dir/FaceDetectMask.cpp.o
[100%] Linking CXX shared library libqwavecv.so
make[2]: warning: Clock skew detected. Your build may be incomplete.
[100%] Built target qwavecv
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example/build$ sudo make ins
tall
-- Configuring done
-- Generating done
-- Build files have been written to: /home/xuser/eagleEYE_labview_OpenCV_Libs_
example/build
Scanning dependencies of target qwavecv
make[2]: Warning: File '../FaceDetect.cpp' has modification time 2453538 s in
the future
[ 33%] Building CXX object CMakeFiles/qwavecv.dir/FaceDetect.cpp.o
[ 66%] Building CXX object CMakeFiles/qwavecv.dir/FaceDetectMask.cpp.o
[100%] Linking CXX shared library libqwavecv.so
make[2]: warning: Clock skew detected. Your build may be incomplete.
[100%] Built target qwavecv
Install the project...
-- Install configuration: ""
-- Installing: /usr/lib/libqwavecv.so
xuser@raspberrypi3:~/eagleEYE_labview_OpenCV_Libs_example/build$
```

Run the "OpenCV-FaceDetect.vi" or "OpenCV-FaceDetectMask.vi"



## Getting Start Guide: Install or upgrade the Raspbian OS.

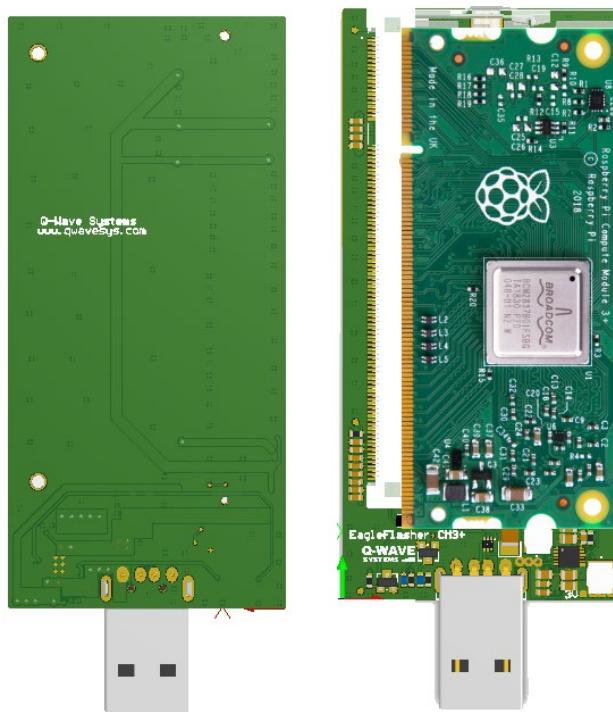
The latest Raspbian located at this link.

<https://www.raspberrypi.org/downloads/raspbian/>

A screenshot of the Raspbian download page. It shows three download options:

- Raspbian Buster with desktop and recommended software**: Image with desktop and recommended software based on Debian Buster. Version: September 2019, Release date: 2019-09-26, Kernel version: 4.19, Size: 2541 MB. Includes [Release notes](#), [Download Torrent](#), and [Download ZIP](#). SHA-256: 549da0fa9ed52a8d7c2d66cb06afac9fe856638b06d8f23df4e6b72e67ed4cea.
- Raspbian Buster with desktop**: Image with desktop based on Debian Buster. Version: September 2019, Release date: 2019-09-26, Kernel version: 4.19, Size: 1123 MB. Includes [Release notes](#), [Download Torrent](#), and [Download ZIP](#). SHA-256: 2c4067d59acf891b7aa1683cb1918da78d76d2552c02749148d175fa7f766842.
- Raspbian Buster Lite**: Minimal image based on Debian Buster. Version: September 2019, Release date: 2019-09-26, Kernel version: 4.19, Size: 435 MB. Includes [Release notes](#), [Download Torrent](#), and [Download ZIP](#). SHA-256: a50237c2f718bd8d806b96df5b9d2174ce8b789eda1f03434ed2213.

In order to write the image to CM3+, the "EagleEYE Flasher CM3+ board" is required.



Please download the “rpiboot” software. Follow instruction as this link

<https://www.raspberrypi.org/documentation/hardware/computemodule/cm-emmc-flashing.md>

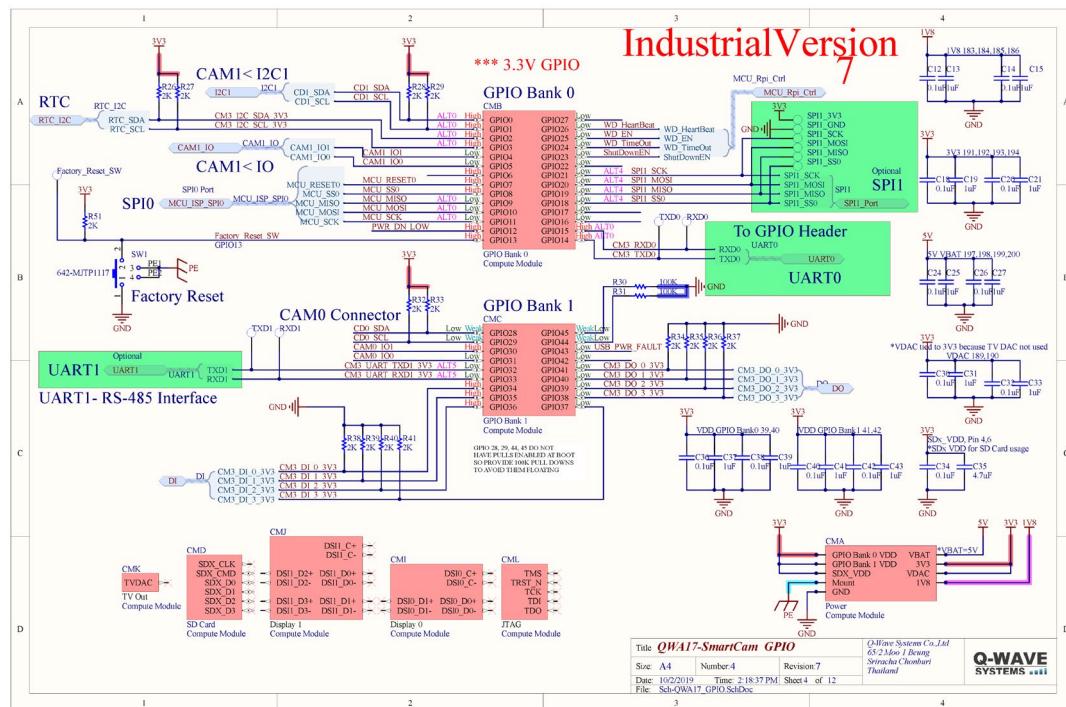
[https://github.com/raspberrypi/usbboot/blob/master/win32/rpiboot\\_setup.exe](https://github.com/raspberrypi/usbboot/blob/master/win32/rpiboot_setup.exe)

The Smart Camera EagleEYE: camera driver “dt-blob.bin” is required. After write the latest image to CM3+ the “dt-blob.bin” file must be copy to boot location below.

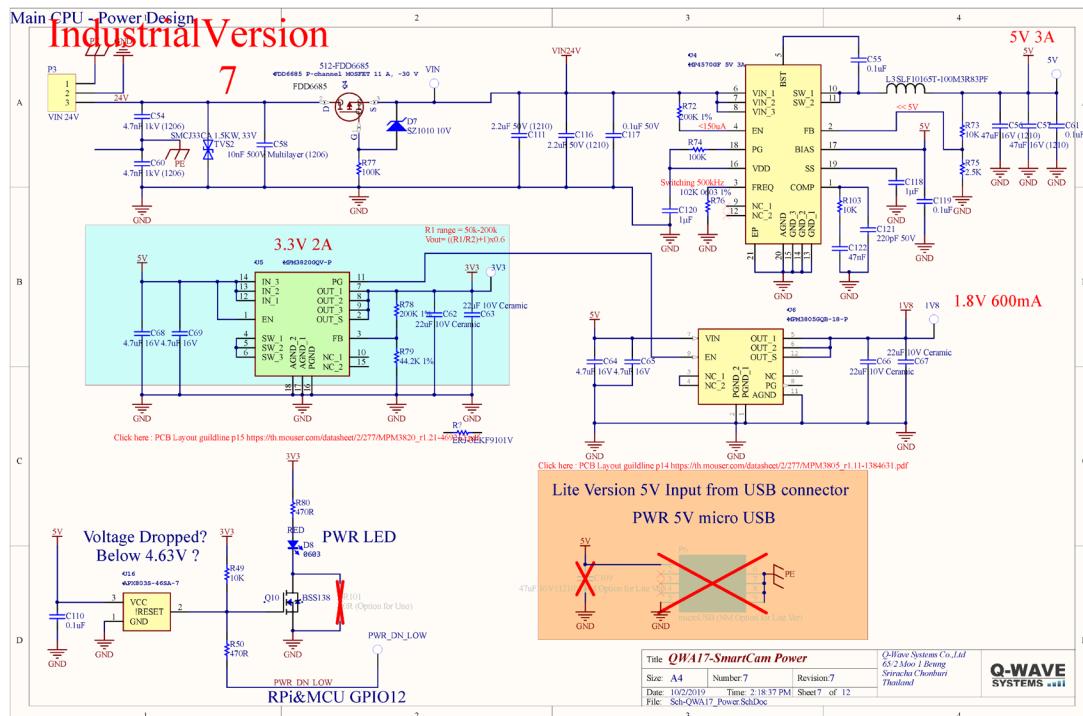
[https://github.com/QWaveSystems/QwaveSmartCamera\\_EagleEYE/blob/master/Linux%20Camera%20Driver/dt-blob.bin](https://github.com/QWaveSystems/QwaveSmartCamera_EagleEYE/blob/master/Linux%20Camera%20Driver/dt-blob.bin)

Name	Size	Packed Size	Modified	Created
overlays	36 216	41 472	2019-10-07 10:24	2019-10-07 10:24
setting	32	512	1980-01-01 00:00	1980-01-01 00:00
System Volume Informa...	88	1 024	2019-10-08 11:47	2019-10-08 11:47
bcm2708-rpi-0-w.dtb	23 315	23 552	2019-10-07 10:24	2019-10-07 10:24
bcm2708-rpi-b-plus.dtb	23 071	23 552	2019-10-07 10:24	2019-10-07 10:24
bcm2708-rpi-b.dtb	22 812	23 040	2019-10-07 10:24	2019-10-07 10:24
bcm2708-rpi-cm.dtb	22 589	23 040	2019-10-07 10:24	2019-10-07 10:24
bcm2709-rpi-2-b.dtb	24 115	24 576	2019-10-07 10:24	2019-10-07 10:24
bcm2710-rpi-3-b-plus.d...	25 574	25 600	2019-10-07 10:24	2019-10-07 10:24
bcm2710-rpi-3-b.dtb	25 311	25 600	2019-10-07 10:24	2019-10-07 10:24
bcm2710-rpi-cm3.dtb	24 087	24 576	2019-10-07 10:24	2019-10-07 10:24
bcm2835-bootfiles-1.20...	0	0	2019-10-07 10:24	2019-10-07 10:24
bootcode.bin	52 296	52 736	2019-10-07 10:24	2019-10-07 10:24
cmdline.txt	169	512	2019-10-07 10:24	2019-10-07 10:24
config.txt	307	36 864	2019-09-10 15:40	2019-10-07 17:56
dt-blob.bin	8 222	8 704	2019-05-26 15:59	2019-05-26 15:59
fixup.dat	6 701	7 168	2019-10-07 10:24	2019-10-07 10:24
fixup_cd.dat	2 622	3 072	2019-10-07 10:24	2019-10-07 10:24
fixup_db.dat	9 750	10 240	2019-10-07 10:24	2019-10-07 10:24
fixup_x.dat	9 752	10 240	2019-10-07 10:24	2019-10-07 10:24
image-version-info	37	512	2019-10-07 10:24	2019-10-07 10:24
kernel7.img	4 991 024	4 991 488	2019-10-07 10:24	2019-10-07 10:24
start.elf	2 873 444	2 873 856	2019-10-07 10:24	2019-10-07 10:24
start_cd.elf	683 460	683 520	2019-10-07 10:24	2019-10-07 10:24
start_db.elf	4 830 756	4 831 232	2019-10-07 10:24	2019-10-07 10:24
start_x.elf	3 774 980	3 775 488	2019-10-07 10:24	2019-10-07 10:24

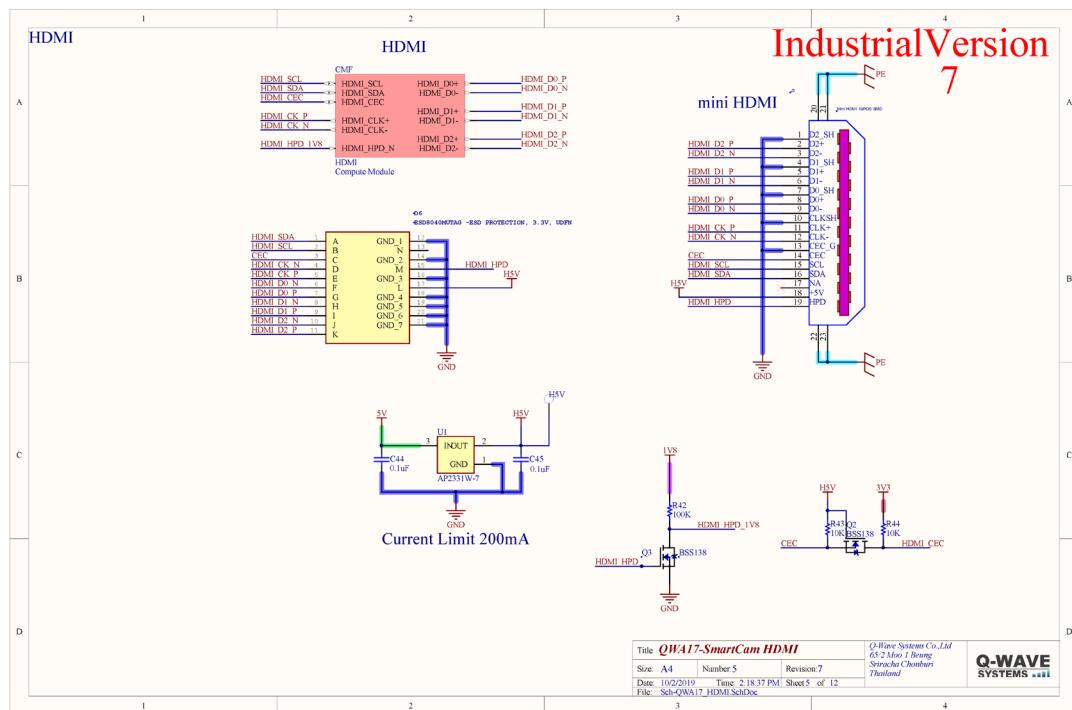
## HW Design Schematics: GPIO



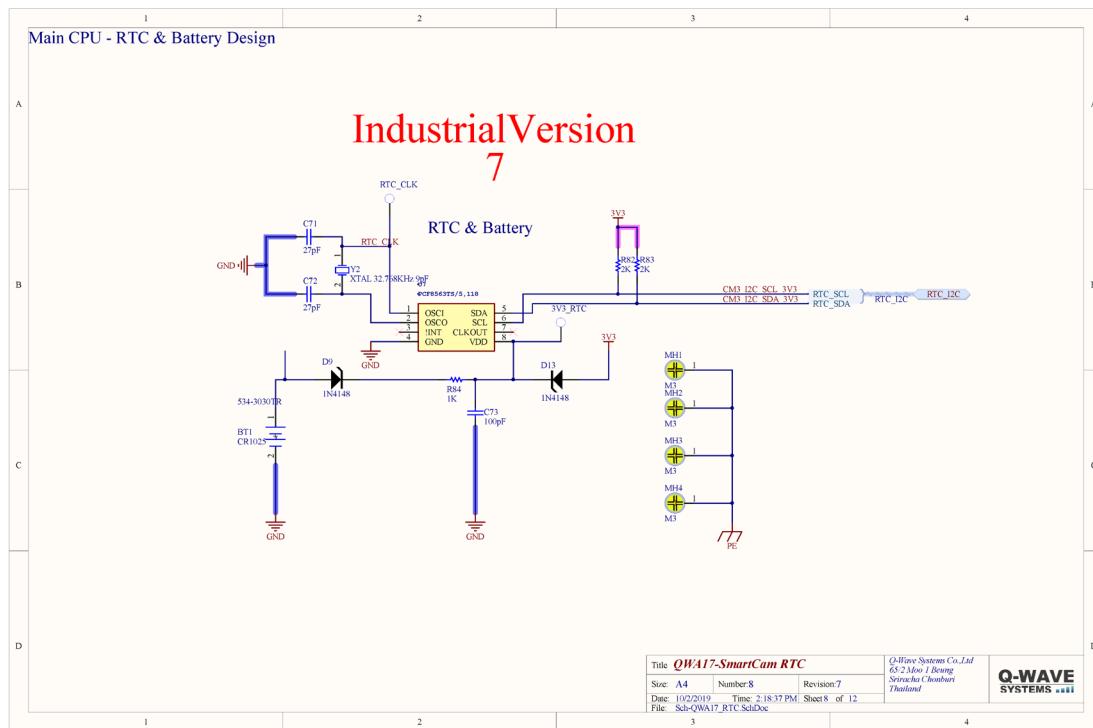
## HW Design Schematics: Power Supply



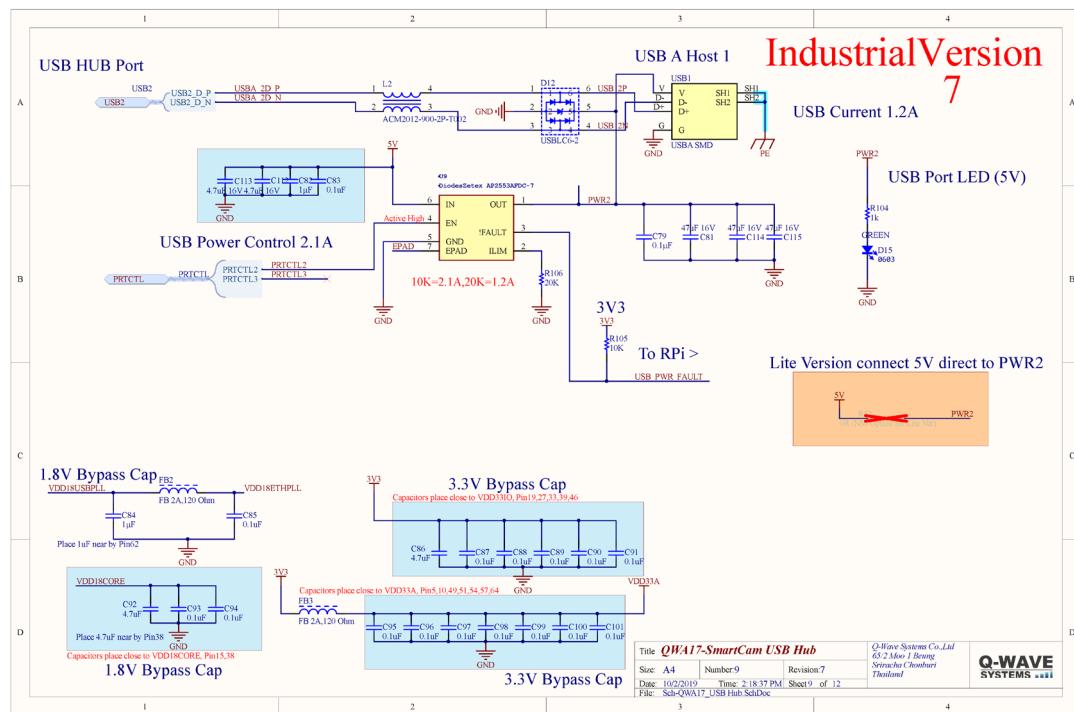
## HW Design Schematics: HDMI Display



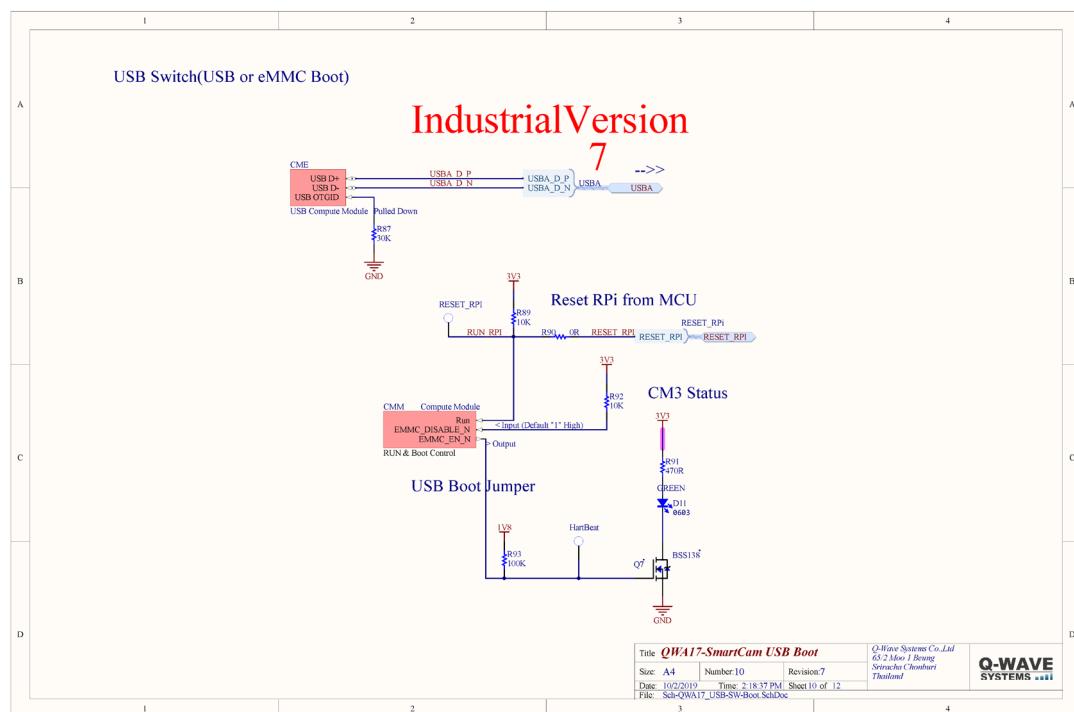
## HW Design Schematics: Real-Time Clock



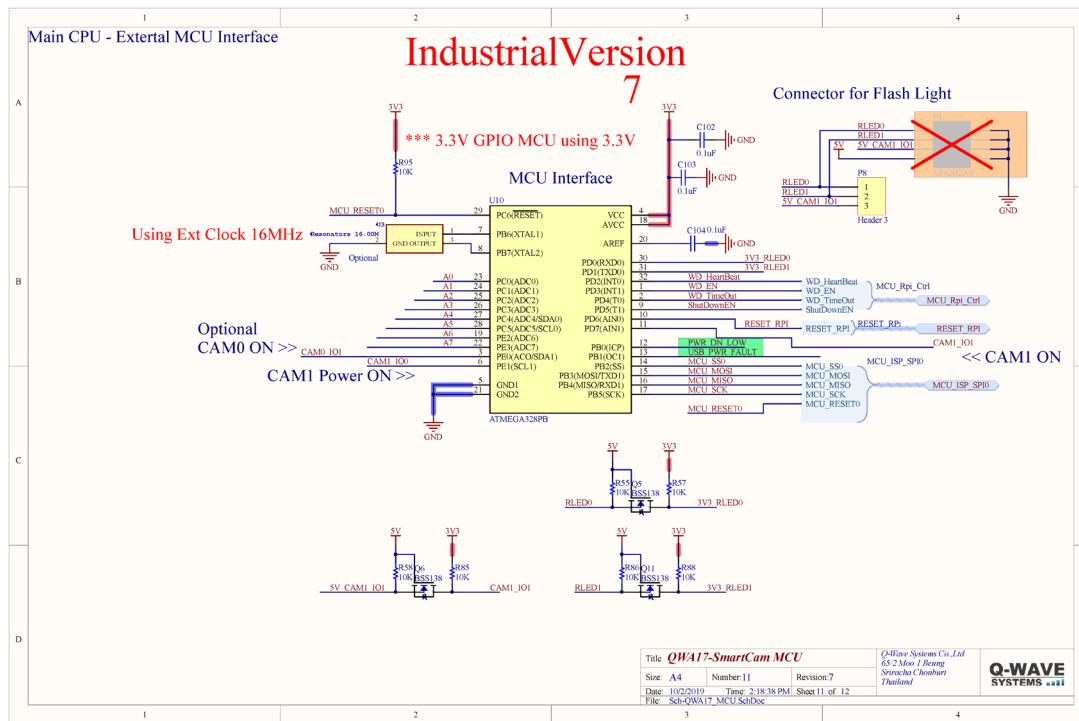
## HW Design Schematics: USB host



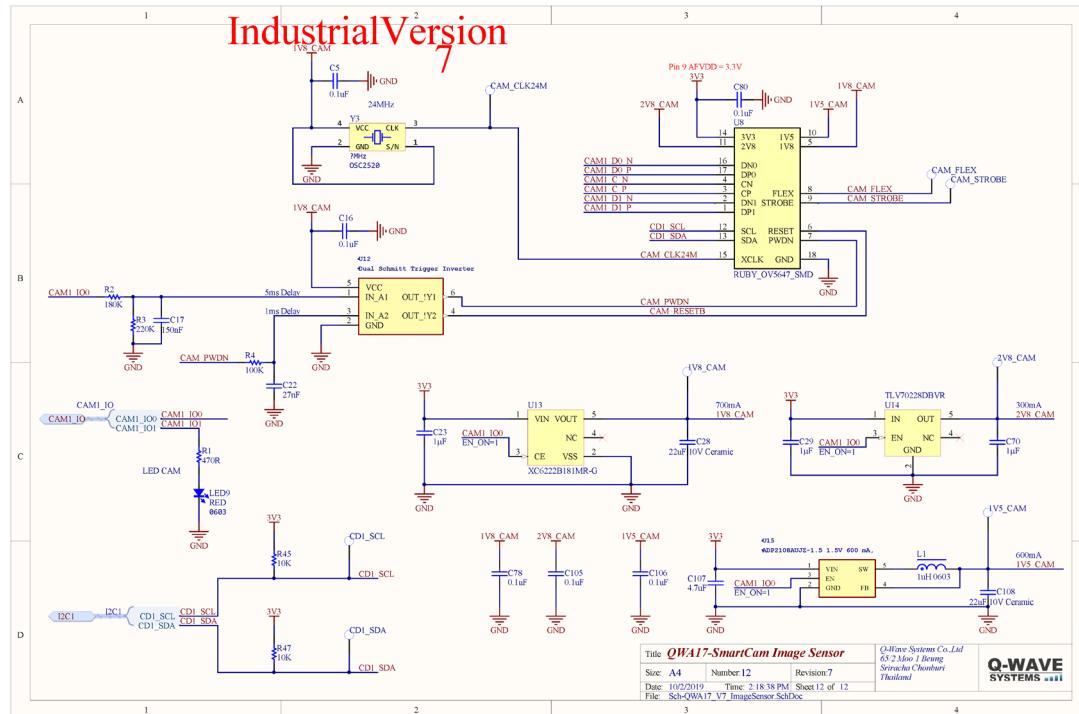
## HW Design Schematics: eMMC Boot



# HW Design Schematics: MCU Interface



# HW Design Schematics: Image Sensor Interface



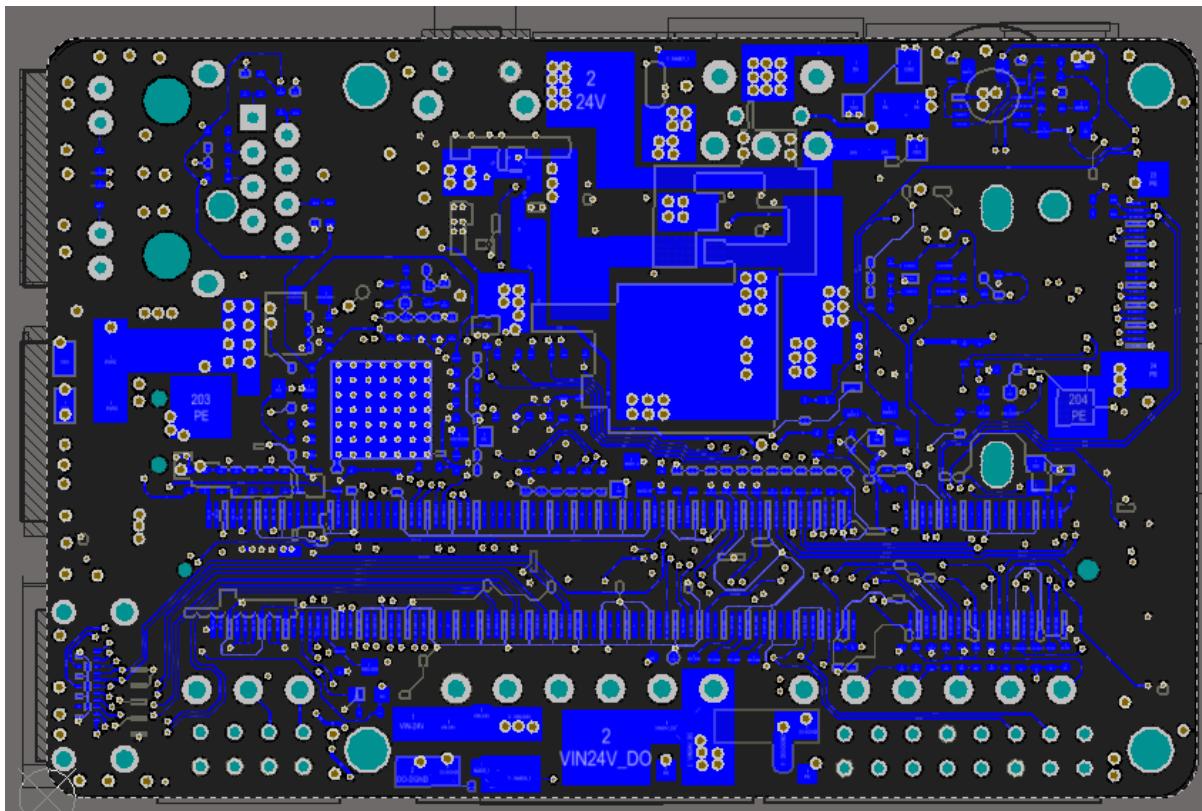
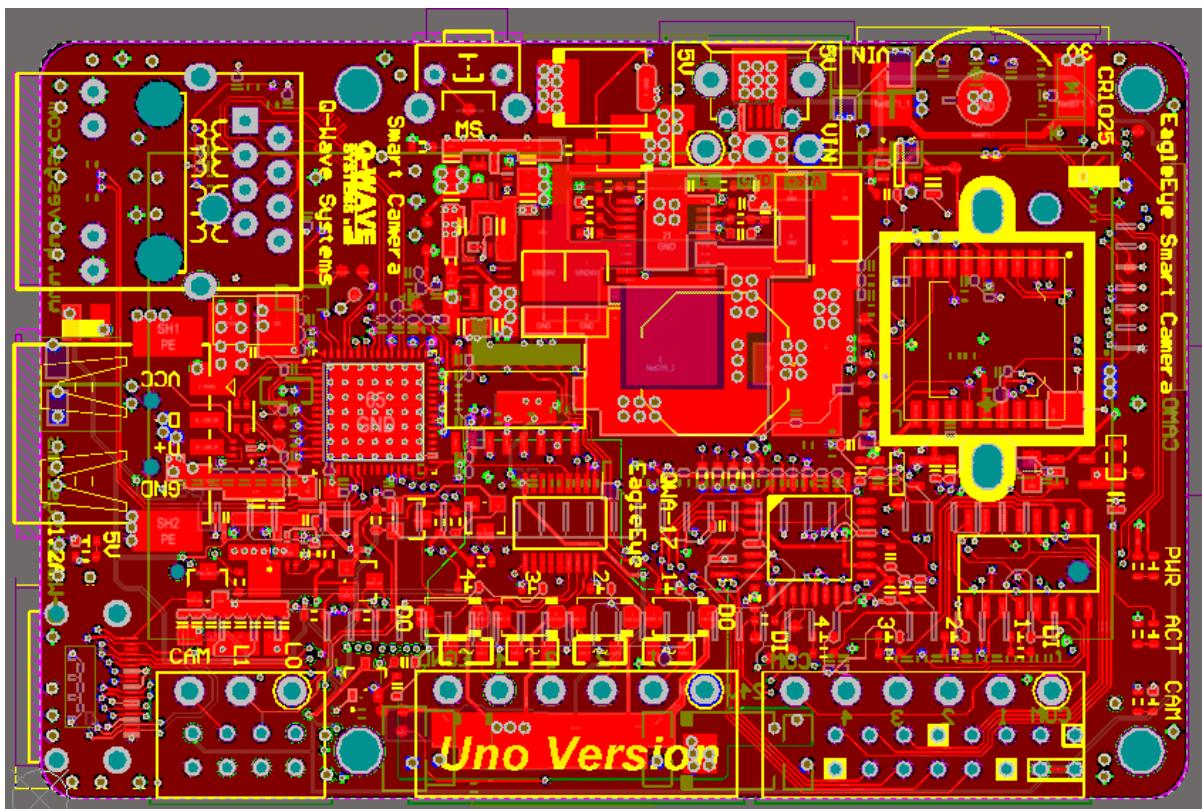
## HW Design Schematics: Bill of Material (BOM)

Bill of Materials						QWA17-SmartCam
Footprint	Comment	LibRef	Designator	Description	Quantity	
TEST_POINT	1V5_CAM	TEST_POINT	1V5_CAM	Test Point	1	
TEST_POINT	1V8	TEST_POINT	1V8	Test Point	1	
TEST_POINT	1V8_CAM	TEST_POINT	1V8_CAM	Test Point	1	
TEST_POINT	2V8_CAM	TEST_POINT	2V8_CAM	Test Point	1	
TEST_POINT	3V3	TEST_POINT	3V3	Test Point	1	
TEST_POINT	3V3_RTC	TEST_POINT	3V3_RTC	Test Point	1	
TEST_POINT	5V	TEST_POINT	5V	Test Point	1	
TEST_POINT	5V_PWR3	TEST_POINT	5V_PWR3	Test Point	1	
SOIC127P700	TLP292-4(V4-LA)	TLP292-4(V4-LA)	B1, B2	Transistor Output Optocouplers 4ch Transis	2	
SOP65P640X	TBD62783AFNG	TBD62783AFNG	B3	Gate Drivers DMOS Transistor Array 7-CH,	1	
3030TR	CR1025	Battery	BT1	Coin Cell Battery Holders SM COIN CELL E	1	
CAP_0603	0.1?F	Cap Semi	C1, C2, C3, C4, C79	0.1?F ?% 50V Ceramic Capacitor X7R 060	5	
CAP_0603	0.1uF	Cap Semi	C5, C12, C14, C16, C18, C20, C24, C26,	0.1?F ?% 50V Ceramic Capacitor X7R 060	44	
CAP_0603	104	Cap Semi	C6	0.1?F ?% 50V Ceramic Capacitor X7R 060	1	
CAP_0603	105	Cap Semi	C7	1?F ?% 50V Ceramic Capacitor X7R 0603	1	
C 1206	4.7nF 1kV (1206)	Cap	C8, C11, C54, C60	4.7nF 1kV (1206) GRM31BR73A472KW01L	4	
C 1210	10uF 50V (1210)	Cap	C9	10uF 50V Ceramic Capacitor X5R 1210	1	
C 1206	0.33uF 50V (120)	Cap	C10	0.33uF 50V Ceramic Capacitor X5R 1206	1	
CAP_0603	1uF	Cap Semi	C13, C15, C19, C21, C25, C27, C31,	1?F ?% 50V Ceramic Capacitor X7R 0603	12	
CAP_0603	150nF	Cap Semi	C17	150nF 16VCeramic Capacitor X7R 0603	1	
CAP_0603	27nF	Cap Semi	C22	27nF 16VCeramic Capacitor X7R 0603	1	
CAP_0603	1?F	Cap Semi	C23, C29, C70, C82, C84, C118, C120	1?F ?% 50V Ceramic Capacitor X7R 0603	7	
CAP_0805	22uF 10V Ceram	Cap	C28, C62, C63, C66, C67, C108	22uF 10V Ceramic	6	
TEST_POINT	TXDU	TEST_POINT	TXDU	Test Point	1	
TEST_POINT	TXD1	TEST_POINT	TXD1	Test Point	1	
Diodes_AP23	IC_Diodes_AP23	IC_Diodes_AP23	U1	IC LOAD SWITCH SC59	1	
_LAN9514	LAN9512I-JZX	CMP-1537-00002	U2	USB 2.0 Hub and 10/100 Ethernet Contolle	1	
CSTNE16M0V	Resonators 16.0	CSTNE16M0VH3	U3	CSTNE16M0VH3C000R0	1	
SOP65P640X	MP4570GF 5V 3	MP4570GF-Z	U4	MP4570GF-Z	1	
MPM3820GQ	MPM3820GQV-F	MPM3820GQV-F	U5	MPM3820GQV-P VIN 6V, 1.2 MHz, 2A	1	
MPM3805GQ	MPM3805GQB-1	MPM3805GQB-1U6		MPM3805GQB-18-P 6V 4MHz 600mA	1	
SOT65P490X	PCF8563TS/5, 11	PCF8563TS_5, 1	U7	REAL TIME CLOCK, 5.5V, I2C, TSSOP-8	1	
RUBY_OV564	RUBY_OV5647	RUBY_OV5647	U8	RUBY_OV5647_SMD	1	
SONG65P201X	DiodesZetex AP2	AP2553AFDC-7	U9	Power Distribution 2.1A Precision ADJ 5.5V	1	
LQFP_P0 8_3	ATMEGA328PB	ATMEGA328PB	U10		1	
SOT95P275X	Dual Schmitt Trig	NL27WZ14DTT10	U12	NL27WZ14DTT1G	1	
TORX-SOT-25	XC6222B181MR	CMP-0659-0000	U13	High Speed LDO Regulator, 700 mA, 1.8V	1	
DBV0005A_M	TLV70228DBVR	CMP-0394-0050	U14	Single Output LDO, 300 mA, Fixed 2.8 V O	1	
SOT95P280X	ADP2108AUJZ-1	ADP2108AUJZ-1	U15		1	
SOT95P230X	APX803S-46SA-1	APX803-46SAG	U16	APX803S-46SA-7	1	
USB-A/S_B	USBA SMD	USBA SMD	USB1	UMRA04FTWNY-03	1	
TEST_POINT	USB3_DN	TEST_POINT	USB3_DN	Test Point	1	
TEST_POINT	USB3_DP	TEST_POINT	USB3_DP	Test Point	1	
TEST_POINT	USB3_GND	TEST_POINT	USB3_GND	Test Point	1	
TEST_POINT	VDD18CORE	TEST_POINT	VDD18CORE	Test Point	1	
TEST_POINT	VDD18ETHPLL	TEST_POINT	VDD18ETHPLL	Test Point	1	
TEST_POINT	VDD18USBPLL	TEST_POINT	VDD18USBPLL	Test Point	1	
TEST_POINT	VDD33A	TEST_POINT	VDD33A	Test Point	1	
TEST_POINT	VGQ1	TEST_POINT	VGQ1	Test Point	1	
TEST_POINT	VIN	TEST_POINT	VIN	Test Point	1	
TEST_POINT	VIN-24V	TEST_POINT	VIN-24V	Test Point	1	
XT_2016_4	XTAL 25MHz XRO	XTAL_2016_4P	Y1	Crystal Oscillator	1	
XT_3215_2	XTAL 32.768kHz	XTAL	Y2	Crystal Unit 32.768KHz ?20ppm, 9pF	1	
XT_2520_4	FJ2400002	OSC_2520_4P	Y3	Crystal Oscillator 24MHz	1	

331

Approved Notes

## HW Design Schematics: PCB Layout

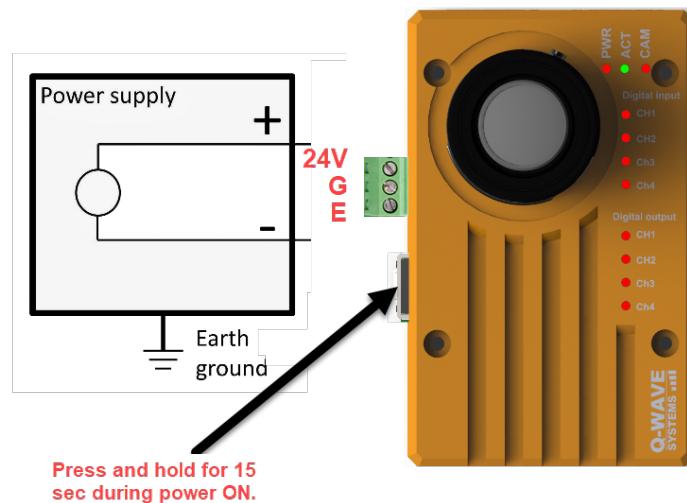


## Developer Manual: Recovery File systems Mode

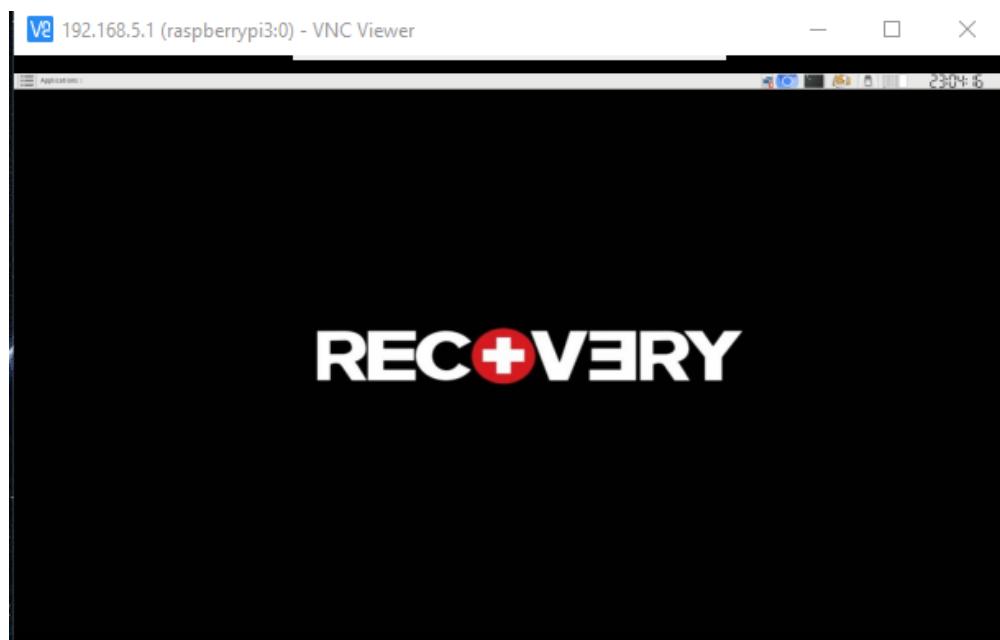
- A. In case of files system has been crash and the Smart Camera can not boot.

Please following these steps to recover system from restore point.

Press and hold “BOOT” button while turn ON power supply for 10-15 sec.



Then the Smart Camera will boot to recovery mode. Observed the background image will show recovery logo which mean recovery mode.



Open Terminal program then using command below.

> restore –list (\*to check the restore point)

> sudo restore –last

```
sn-4.4$  
sh-4.4$  
sh-4.4$ sudo restore --last  
Are you sure you want to restore from 20191106165431? (y/n) > █
```

Press "y" to confirm.

> sudo reboot

```
sh-4.4$  
sh-4.4$ sudo restore --last  
Are you sure you want to restore from 20191106165431? (y/n) > y  
System restore from point 20191106165431 ...done  
sh-4.4$ sudo reboot█
```

(\*\*\* DO NOT TURN OFF POWER by yourself, please wait the system reboot automatically, normally will take a while o 5 min (max).

(Due to file system for restoring are very large and it cannot show the progress for restoring files, Then you have to wait until board has reboot itself.)

B. Steps for make your own restore point.

> backup

> restore –list (\*to check the restore point)

In order make a system backup, you can use "backup" command and if would to see the restore point which is in date-time format filename please use command "restore --list",

```
xuser@raspberrypi3:~$ restore --list  
20190526225814 20190526230040 20191106165431  
xuser@raspberrypi3:~$
```