Building Machine Vision Applications using OpenMV

Class 5: Designing a Machine Vision Application

June 12, 2020
Jacob Beningo

Presented by:

DesignNews

CEC CONTINUING EDUCATION CENTER

Digi-Key ELECTRONICS

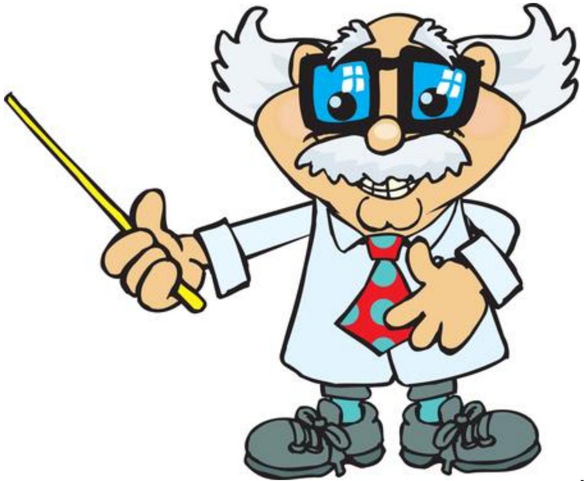
Course Overview

Topics:

- Introduction to Machine Vision and OpenMV
- Writing our First OpenMV Application
- Working with the OpenMV I/O
- Utilizing Machine Learning to Detect Objects
- **Designing a Machine Vision Application**

Session Overview

- Introduction
- Energy Management
- Using the SD Card
- Best Practices

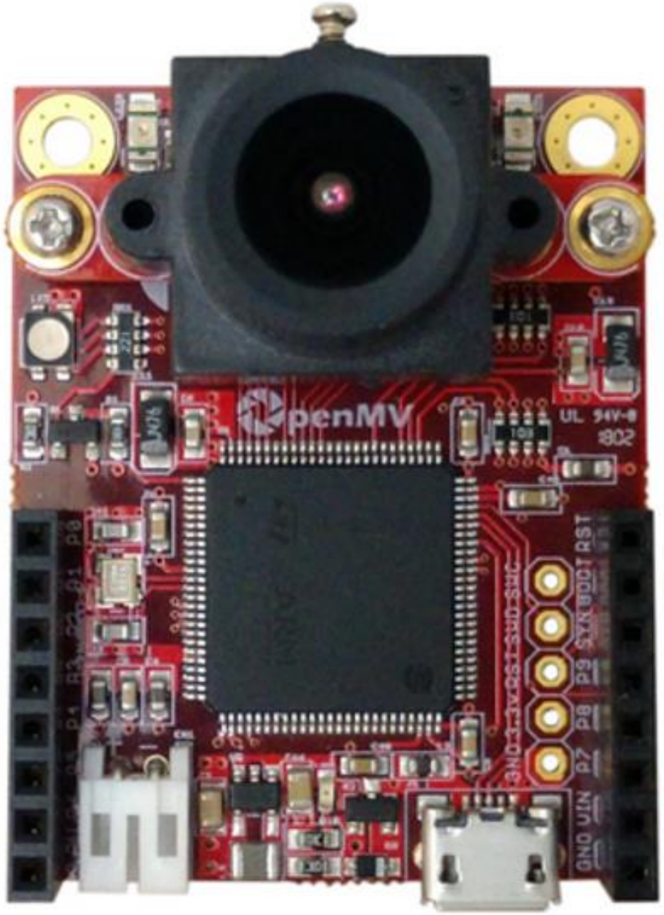


Presented by:

Introduction

Presented by:

Battery Management



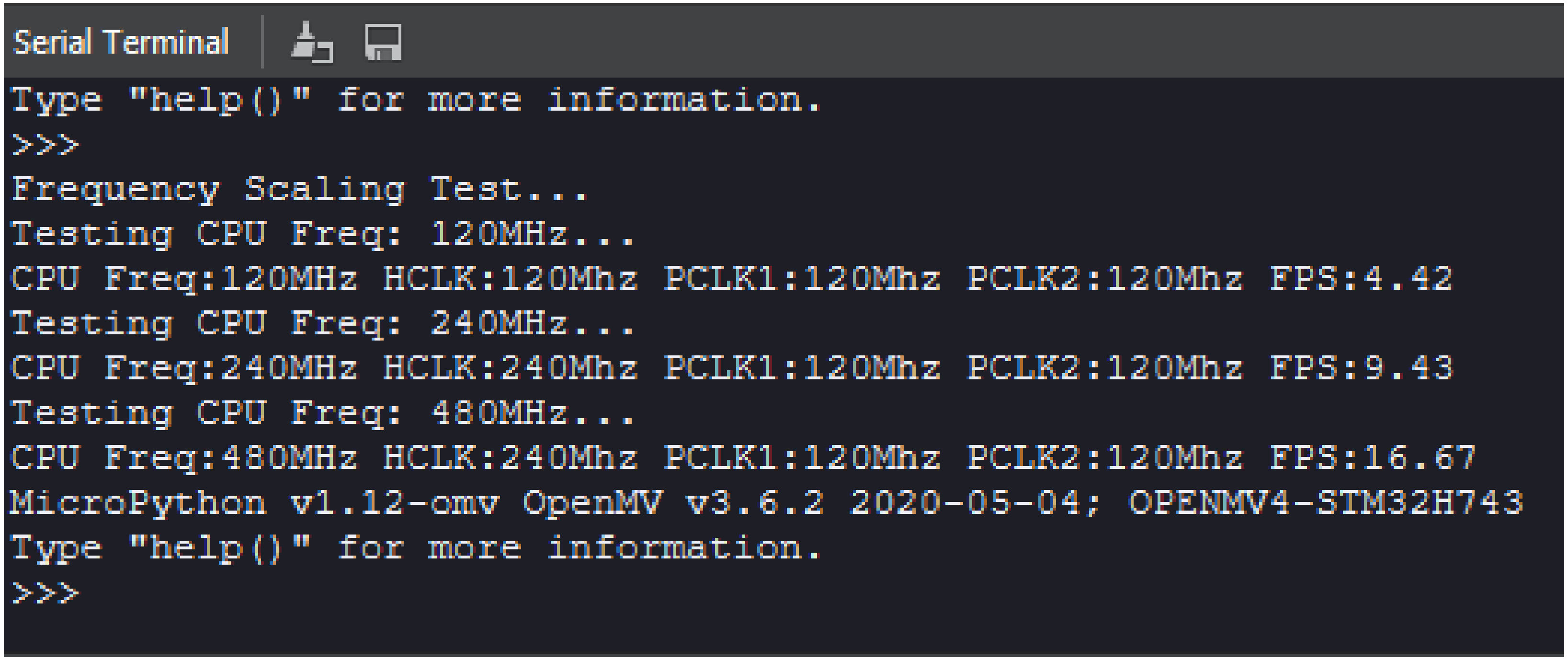
Battery Management

Energy Optimization

- Scale the CPU
- Deep sleep modes
- Sensor sleep
- Stop sleep mode

Battery Management

Frequencies Script for Scaling:

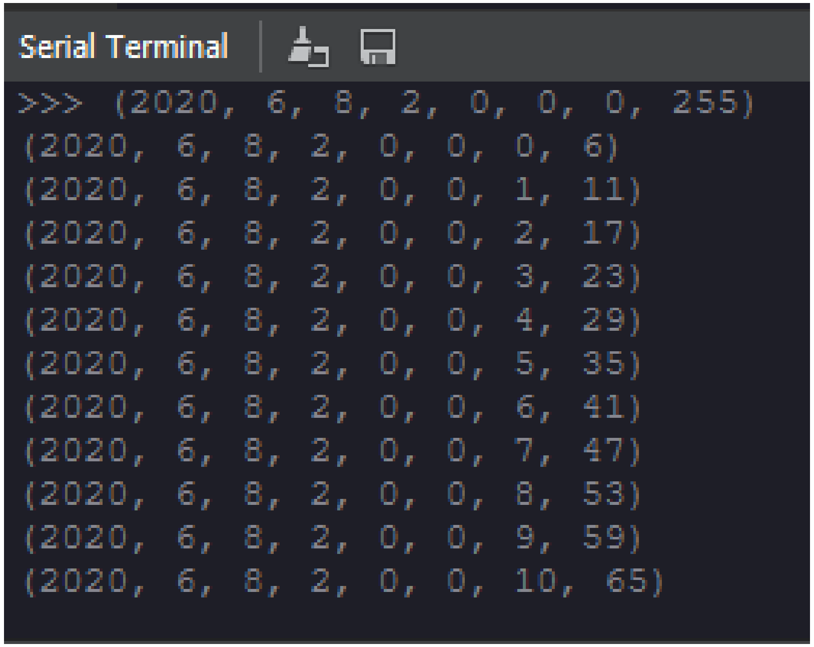


```
Type "help()" for more information.
>>>
Frequency Scaling Test...
Testing CPU Freq: 120MHz...
CPU Freq:120MHz HCLK:120Mhz PCLK1:120Mhz PCLK2:120Mhz FPS:4.42
Testing CPU Freq: 240MHz...
CPU Freq:240MHz HCLK:240Mhz PCLK1:120Mhz PCLK2:120Mhz FPS:9.43
Testing CPU Freq: 480MHz...
CPU Freq:480MHz HCLK:240Mhz PCLK1:120Mhz PCLK2:120Mhz FPS:16.67
MicroPython v1.12-omv OpenMV v3.6.2 2020-05-04; OPENMV4-STM32H743
Type "help()" for more information.
>>>
```

Battery Management

The RTC

```
import time
from pyb import RTC

rtc = RTC()
rtc.datetime((2020, 6, 8, 2, 0, 0, 0, 0))

while (True):
    print(rtc.datetime())
    time.sleep(1000)
```



```
>>> (2020, 6, 8, 2, 0, 0, 0, 255)
(2020, 6, 8, 2, 0, 0, 0, 6)
(2020, 6, 8, 2, 0, 0, 1, 11)
(2020, 6, 8, 2, 0, 0, 2, 17)
(2020, 6, 8, 2, 0, 0, 3, 23)
(2020, 6, 8, 2, 0, 0, 4, 29)
(2020, 6, 8, 2, 0, 0, 5, 35)
(2020, 6, 8, 2, 0, 0, 6, 41)
(2020, 6, 8, 2, 0, 0, 7, 47)
(2020, 6, 8, 2, 0, 0, 8, 53)
(2020, 6, 8, 2, 0, 0, 9, 59)
(2020, 6, 8, 2, 0, 0, 10, 65)
```

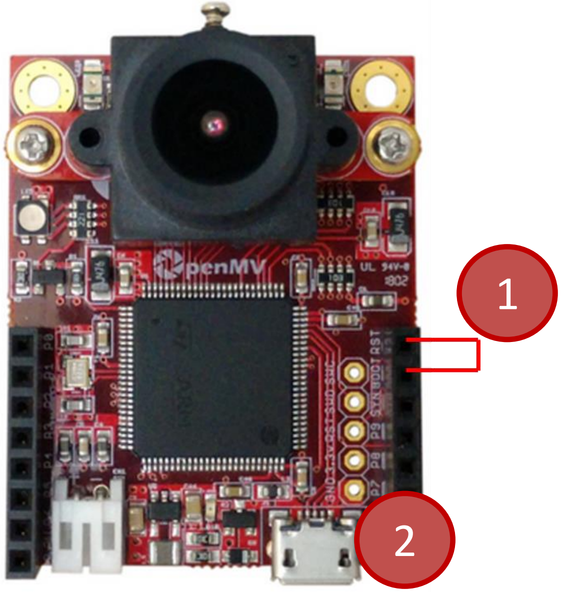
Battery Management

```
# Sensor Sleep Mode Example.
# This example demonstrates the sensor sleep mode. The sleep mode saves around
# 40mA when enabled and it's automatically cleared when calling sensor reset().

import sensor, image, time

sensor.reset()                             # Reset and initialize the sensor.
sensor.set_pixformat(sensor.RGB565)   # Set pixel format to RGB565 (or GRAYSCALE)
sensor.set_framesize(sensor.QVGA)     # Set frame size to QVGA (320x240)
sensor.skip_frames(time = 3000)        # Capture frames for 3000ms.
sensor.sleep(True)                         # Enable sensor sleep mode (saves about
                                           40mA).
```

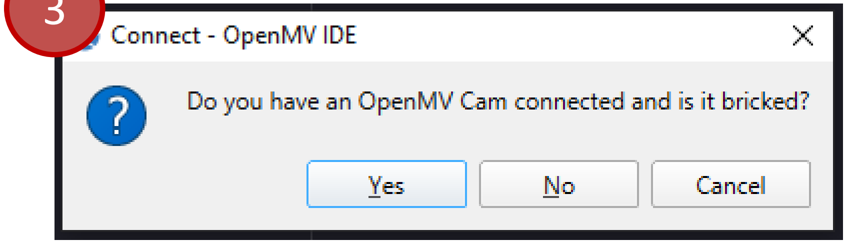
Source: OpenMV IDE sensor_sleep.py

Presented by:

Restoring your Firmware

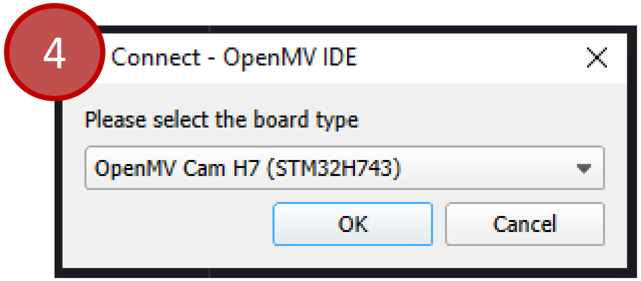


3


Connect - OpenMV IDE

Do you have an OpenMV Cam connected and is it bricked?

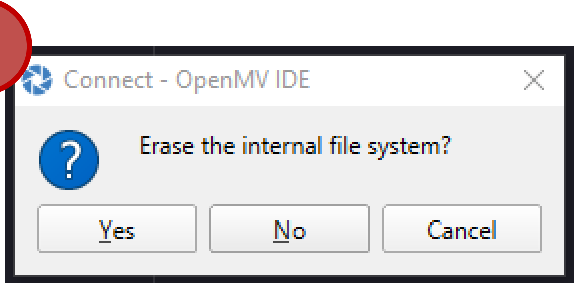
4


Connect - OpenMV IDE

Please select the board type

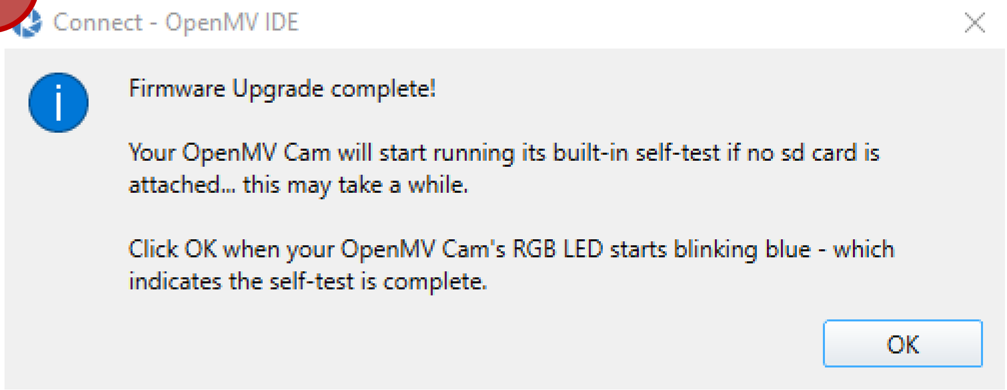
OpenMV Cam H7 (STM32H743)

5


Connect - OpenMV IDE

Erase the internal file system?

6


Connect - OpenMV IDE

Firmware Upgrade complete!

Your OpenMV Cam will start running its built-in self-test if no sd card is attached... this may take a while.

Click OK when your OpenMV Cam's RGB LED starts blinking blue - which indicates the self-test is complete.

Using the SD Card

uSD Card slot:

- Up to 32 GB (Fat file system)
- Replaces internal flash

Several Test Scripts can be used:

- Video-Recording -> image_writer.py
- Video-Recording -> image_reader.py
- Image-Filters

Using the SD Card

Video-Recording -> image_writer.py

Using the SD Card

```
img_reader = None if snapshot_source else image.ImageReader("/stream.bin")

while(True):
    clock.tick()
    img = sensor.snapshot() if snapshot_source else
                img_reader.next_frame(copy_to_fb=True, loop=True)
    img.laplacian(1)
    print(clock.fps())
```

Using the SD Card

Best Practices for Getting Started with OpenMV

1) Insert your SD card prior to powering up the cam.
2) Explore each example application program.
3) Experiment with multiple lenses.
4) Read through the MicroPython documentation
5) Read through the OpenMV tutorial documentation.
6) Test each image filter and understand how it affects our image.
7) Walk through a machine learning example.
8) Build simple test applications and then once proven add them to a larger application.
9) Create a high-level architecture that describes what you end application will do.
10) Share on social media what you've accomplished.

Additional Resources

- Beningo.com
  - Blog, White Papers, Courses
  - Embedded Bytes Newsletter
    - http://bit.ly/1BAHYXm
- OpenMV.io

From www.beningo.com under

- Blog > CEC – Building Machine Vision Applications using OpenMV