

AI For Kids with M5Stick-V board

Top Robot Kids



A Guide by • Mr. Apirak Sang-ngenchai • March 11, 2020

AI For Kids with M5Stick-V



เวิร์กช็อปเรียนรู้การพัฒนาอุปกรณ์ด้วยระบบปัญญาประดิษฐ์ด้วยบอร์ด M5Stick-V โดยเน้นการใช้งานเบื้องต้น ตั้งแต่การพัฒนาโปรแกรมพื้นฐาน การใช้งานบอร์ด รวมไปถึงกำความเข้าใจกระบวนการทำงานปัญญาประดิษฐ์อย่างง่าย ผ่านการประมวลผลภาพ

ชิ้นบอร์ด M5Stick-V มีชิป K210 สำหรับประมวลผลด้าน AI โดยเฉพาะ ทำให้สามารถสร้างอุปกรณ์ทำงานด้วยระบบปัญญาประดิษฐ์อย่างง่ายได้ ไม่ว่าจะเป็นการตรวจจับใบหน้า การจำแนกใบหน้า รวมไปถึงการจำแนกสิ่งของต่าง ๆ ทำให้ผู้พัฒนามือใหม่ สามารถนำไปต่อยอด และศึกษาการพัฒนาอุปกรณ์ปัญญาประดิษฐ์ได้ในอนาคต



AI For Kids

With M5Stick-V

→ Introduce

What's Artificial Intelligence
M5StickV Specification

→ Software preparing

MaixPy IDE
K-Flash GUI

→ Hand-on

Board Example
Image and Face Detection
Object Classification

Demo~



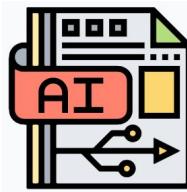
Introduce



- What's Artificial Intelligence
- M5StickV Specification

What is Artificial Intelligence

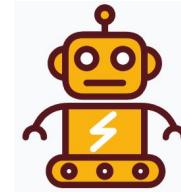
“กระบวนการคิด และใช้เหตุผล โดยมีมนุษย์เป็นต้นแบบ”



การประมวลผลภาษาธรรมชาติ
(Natural Language Processing : NLP)



คอมพิวเตอร์วิทัศน์
(Computer Vision)



หุ่นยนต์
(Robotics)



ระบบผู้เชี่ยวชาญ
(Expert System)

อ้างอิงจากหนังสือ AI สร้างได้ด้วยแมชชีนเลิร์นนิ่ง

AlphaGo



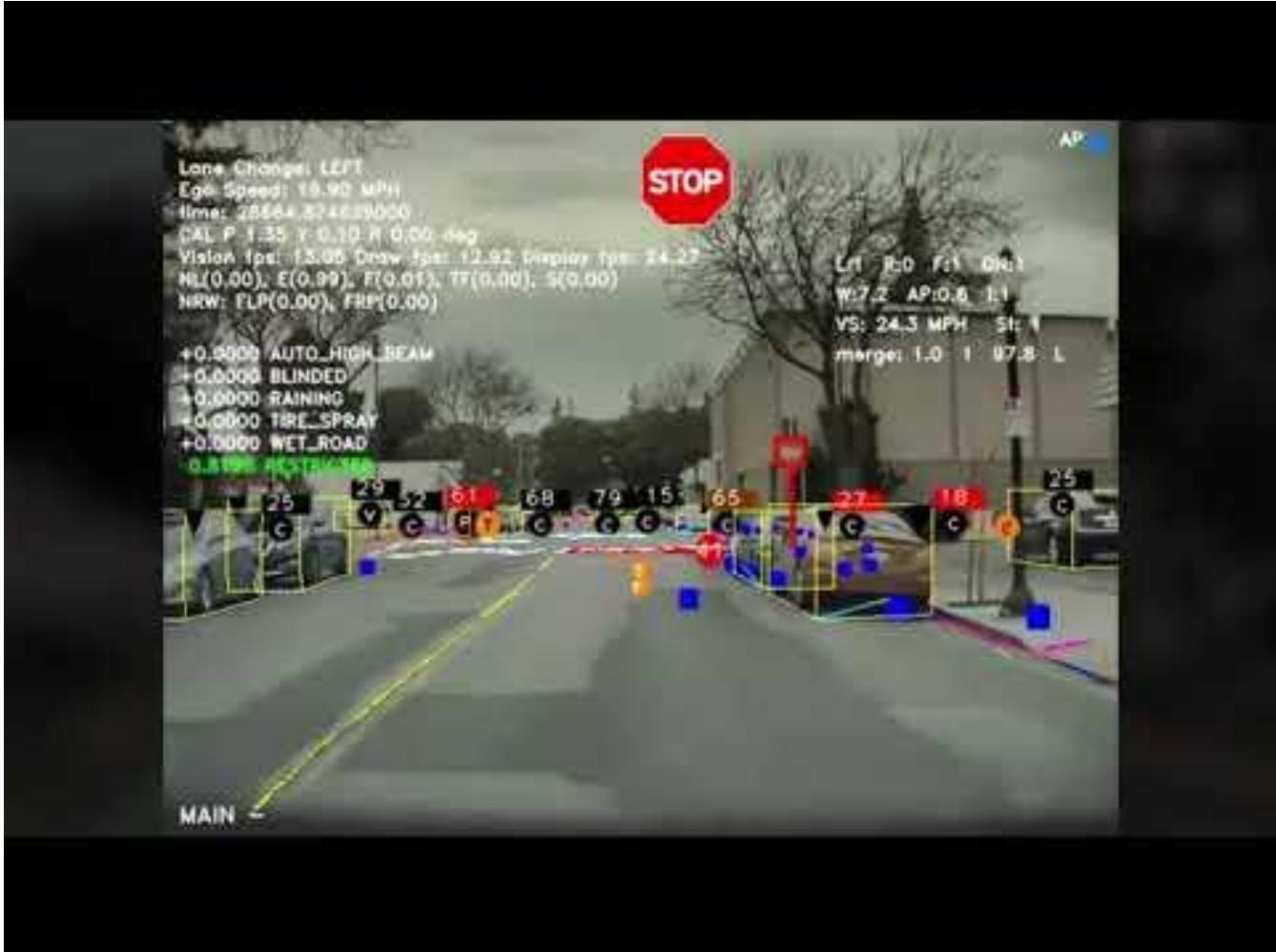


OpenAI



OpenAI



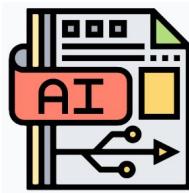


comma.ai (D.I.Y. Autopilot Car Kit)



Machine Learning

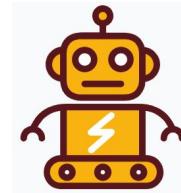
“การให้เครื่องจักรเรียนรู้งานใดงานหนึ่งจากตัวอย่าง หรือข้อมูล เพื่อให้ทำงานนั้นได้อย่างมีประสิทธิภาพ”



การประมวลผลภาษาธรรมชาติ
(Natural Language Processing : NLP)



คอมพิวเตอร์วิทัศน์
(Computer Vision)



หุ่นยนต์
(Robotics)



ระบบผู้เชี่ยวชาญ
(Expert System)

ARTIFICIAL INTELLIGENCE

Early artificial intelligence stirs excitement.



MACHINE LEARNING

Machine learning begins to flourish.



DEEP LEARNING

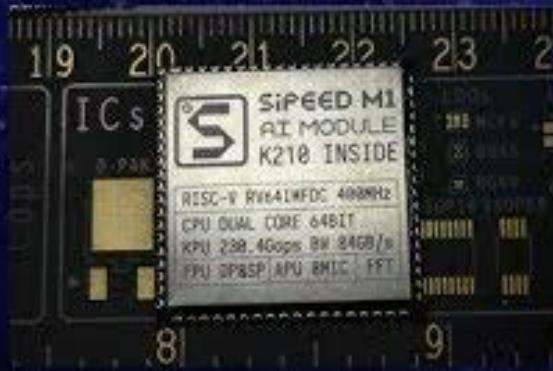
Deep learning breakthroughs drive AI boom.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

World's First Square Inch RV64 AI Module

Sipeed MAIX: AIoT





M5StickV

0.8 Tops



Raspberry Pi

6.4 GFLOPs



Jetson Nano

472 GFLOPs



Jetson TX2

874 GFLOPs



Movidius

100 GFLOPs

#GFLOPs : GiGaFlops (1,000,000,000)

#Tops : Tera Operations per Second (1,000,000,000,000)

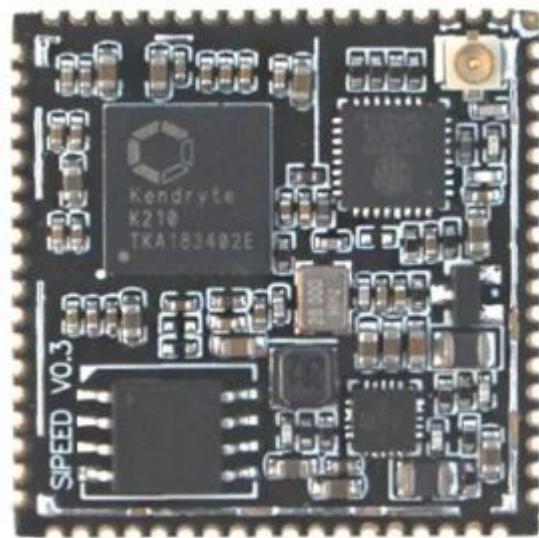
M5StickV



K210
AI
CAMERA

Specification

- Dual-Core 64-bit RISC-V RV64IMAFDC (RV64GC) CPU / 400Mhz (Normal)
- Dual Independent Double Precision FPU
- 8MiB 64bit width On-Chip SRAM
- Neural Network Processor(KPU) / 0.8 Tops (Tera Operations per Seco
- Field-Programmable IO Array (FPIOA)
- Dual hardware 512-point 16bit Complex FFT
- SPI, I2C, UART, I2S, RTC, PWM, Timer Support
- AES, SHA256 Accelerator
- Direct Memory Access Controller (DMAC)
- **Micropython Support**
- Firmware encryption support
- Case Material: PC + ABS





- On-board Hardware resources:
 - Flash : 16M
 - TFT : ST7789 135*240 IPS 1.14 inch SPI
 - Camera : OV7740
 - PCM : MAX98357 I2S Class-D Mono Amp
 - PMIC : AXP192 Power System Management
 - Button : Front and side
 - Battery : 200mAh
 - Indicator light : RGB-W
 - External storage : TF card / Micro SD card
 - Gyro : MPU6886
 - Interface : GROVE wafer
 - Package size : 144mm x 44mm x 43mm

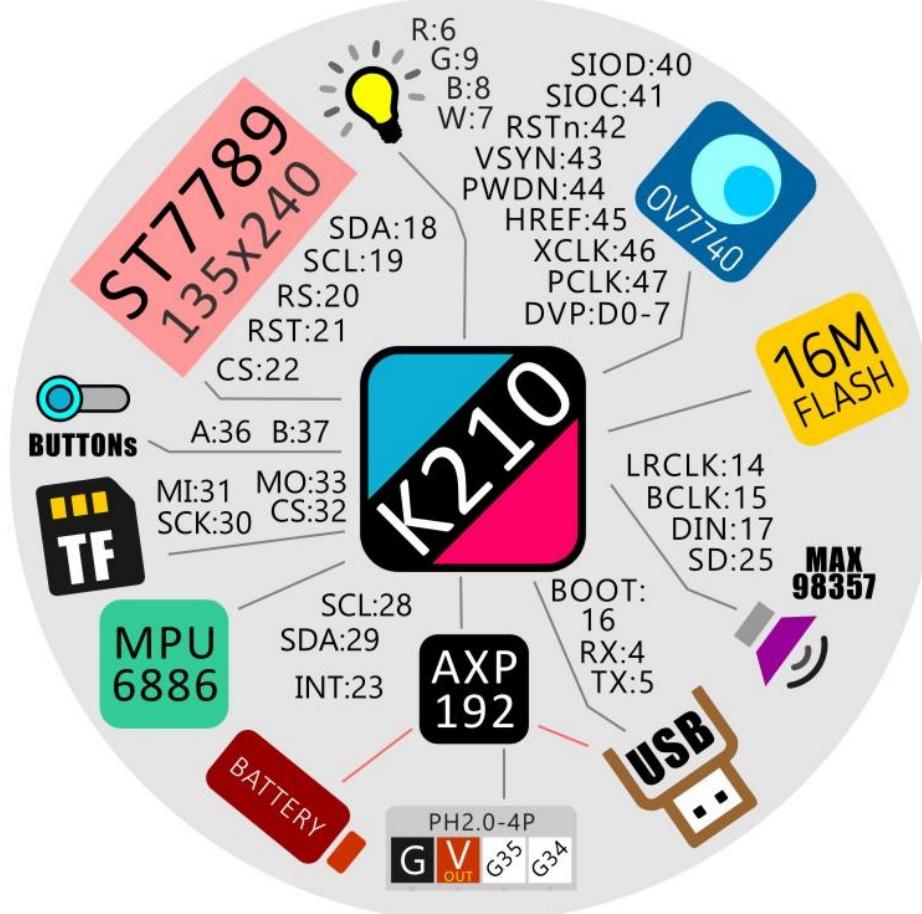


OK



NG

Pinout



Software preparing

- MaixPY IDE
 - K-Flash GUI



The screenshot shows a terminal window with the title "demo_find_face.py - ManyIDE". The code implements a face detection loop using OpenCV's Haar cascade classifier. It reads frames from a camera, applies grayscale conversion and Gaussian blur, and then uses the classifier to detect faces. For each detected face, it draws a bounding box and prints the face ID. The terminal also displays a histogram and RGB color space analysis of the current frame.

```
demo_find_face.py * x
File Edit Tools Window Help
demo_find_face.py * x
21 #utime.sleep_ms(1000)
22
23
24
25 #utime.sleep_ms(1000)
26 #led_g.value(1)
27 utime.sleep_ms(1000)
28
29
30 lcd.init()
31
32 sensor.reset()
33 sensor.set_pixformat(sensor.RGB565)
34 sensor.set_framesize(sensor.QVGA)
35 sensor.set_vflip(1)
36
37 task = kpu.load(0x300000) # you need put model(face.kfpkg) in flash at address 0x300000
38 #task = kpu.load(0x100000)
39 #task = kpu.load(1, 1189, 2, 52465, 3, 94056, 1, 94056, 3, 99987, 5, 3658, 5, 155437, 6, 92275, 6, 718375, 9, 01025)
40 a = kpu.run_yolo(task, 0.5, 0.3, 5, anchor)
41 while True:
42     img = sensor.snapshot()
43     code = kpu.run_yolo2(task, img)
44     if code != None:
45         for i in code:
46             a = img.draw_rectangle(i.rect())
47
48     print(a)
49
50     s = json.dumps(a)
51     buf = memoryview(bytearray.loads(s))
52
53     if buf[buffind(b'face')] <= 120 :
54         led_r.value(0)
55     else:
56         led_r.value(1)
57
58     if buf[buffind(b'face')] > 120 :
59         led_g.value(0)
60     else:
61         led_g.value(1)
62
63
64     img.draw_string(10, 10, str(buf[buffind(b'value')]), color=(0,0,0), scale=2)
65     img.draw_string(10, 10, str(buf[buffind(b'value')]), color=BLACK, led.WHILE)
66     a = lcd.display(img)
67
68     lcd.deinit()
69
70 #["x":1140, "y":146, "w":82, "h":111, "value":3.168161, "classid":0, "index":0, "objnum":3]
71
```

MaixPy IDE

Download on Website : <http://dl.sipeed.com/MAIX/MaixPy/ide/v0.2.4>

The screenshot shows a web interface for downloading the MaixPy IDE. At the top, there's a navigation bar with a home icon and the word 'MAIX'. Below it, a section titled '2. Downloads' is shown. Under this, a heading says 'Download from dl.sipeed.com:'. A table lists several files with their download URLs. To the right of the table, a 'SHA256SUM' column is visible, with a note 'from 2019-09-07' and an eye icon.

#	Name	SHA256SUM
1	maixpy-ide-windows-0.2.4-installer-archive.7z:	from 2019-09-07
	maixpy-ide-windows-0.2.4.exe:	
	maixpy-ide-mac-0.2.4.dmg:	
	maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z:	
	maixpy-ide-linux-x86_64-0.2.4.run:	

Download from dl.sipeed.com:

```
-----  
# Name  
1 maixpy-ide-windows-0.2.4-installer-archive.7z:  
maixpy-ide-windows-0.2.4.exe:  
maixpy-ide-mac-0.2.4.dmg:  
maixpy-ide-linux-x86_64-0.2.4-installer-archive.7z:  
maixpy-ide-linux-x86_64-0.2.4.run:  
-----
```

SHA256SUM

from 2019-09-07

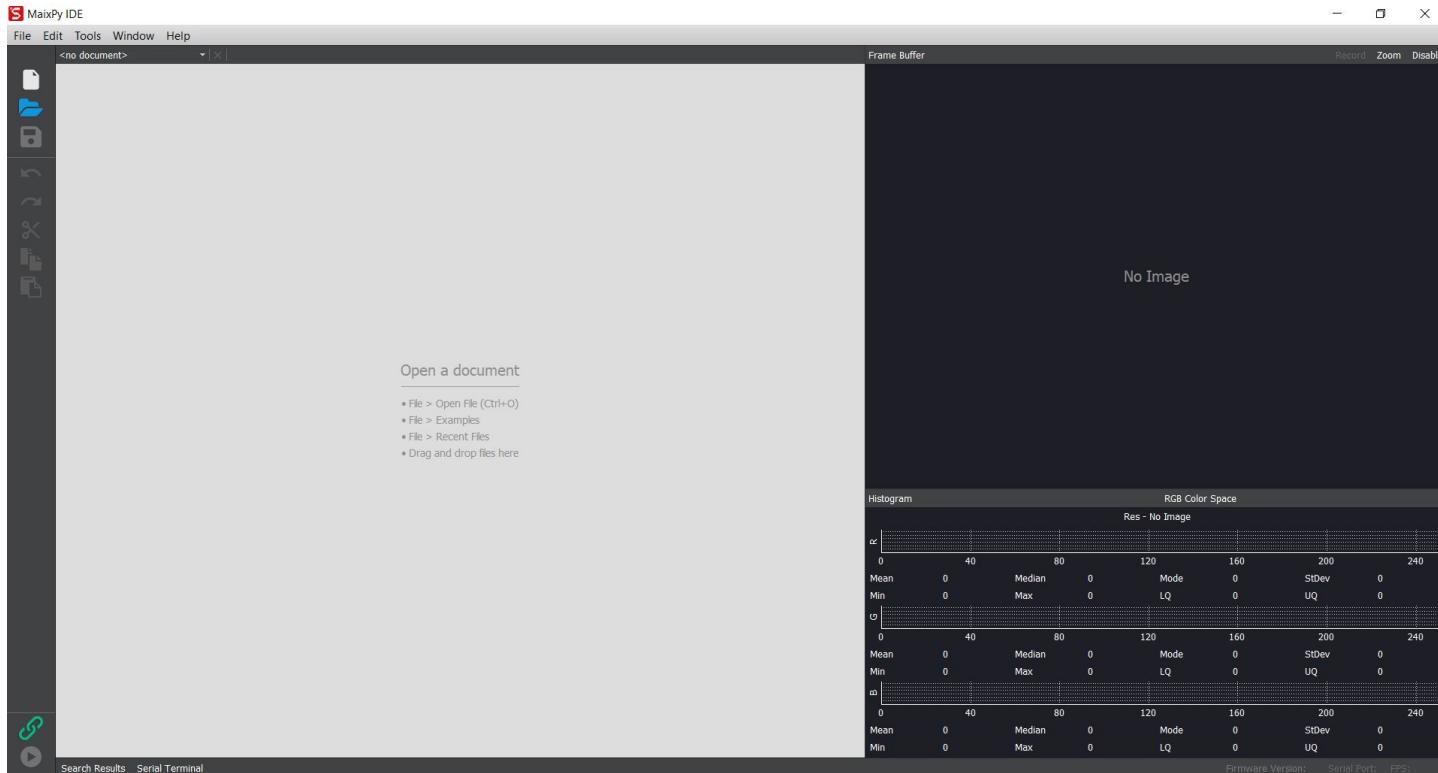
You will get .7z file. Then, Extract it



maixpyide.exe may inside the folder



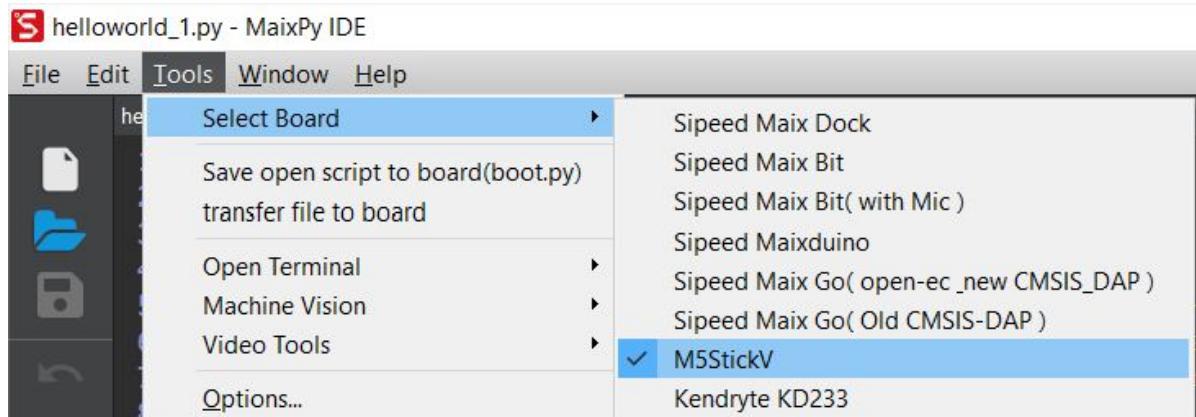
Open Maixpy IDE v2.4.0 on your computer



Connect M5StickV to your computer via USB Type C



Click Tools --> Select Board --> M5StickV



Click Connect and Select the Comport

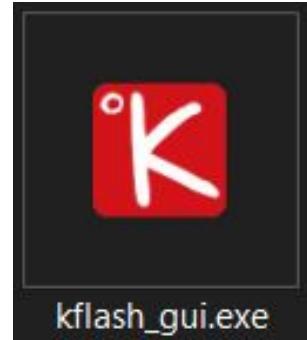


K-Flash GUI

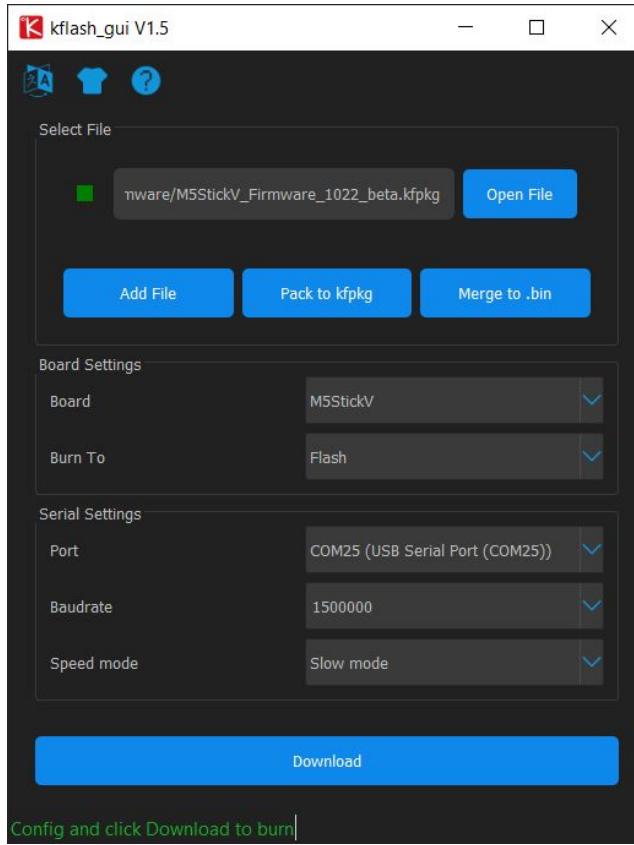
Download on Website : <https://github.com/kendryte/kflash.py/releases>

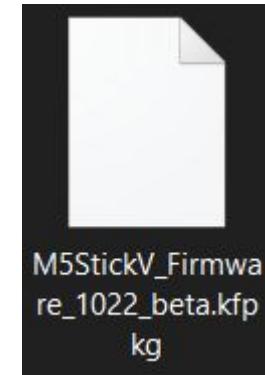
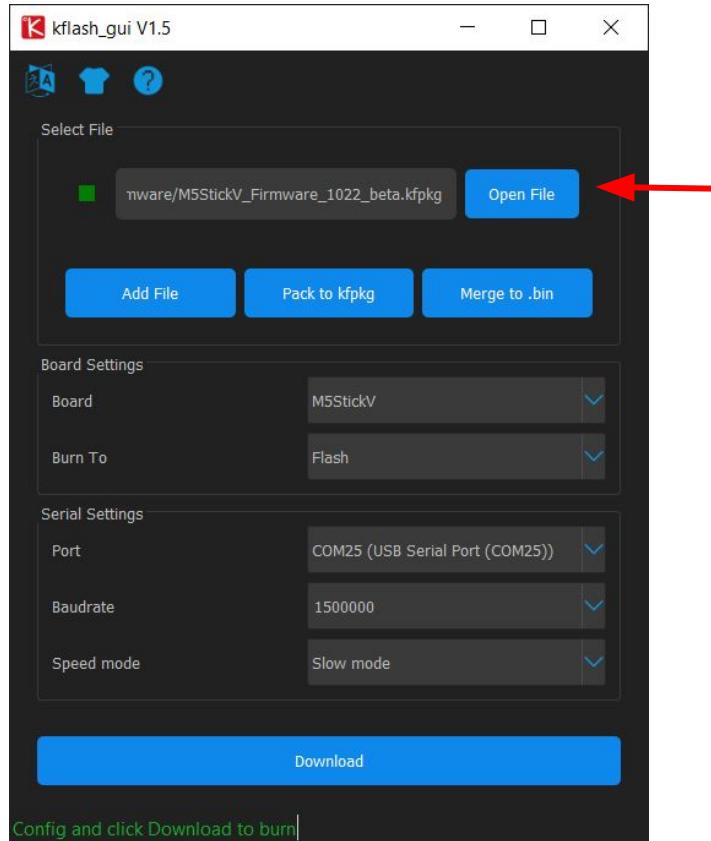
The screenshot shows the GitHub repository page for `kendryte / kflash.py`. The top navigation bar includes 'Code' (selected), 'Issues 6', 'Pull requests 3', 'Actions', 'Projects 0', 'Security', and 'Insights'. Below this, there are tabs for 'Releases' and 'Tags' (selected). A red box highlights the release entry for `v0.9.1`, which was tagged by `vowstar` on Jul 4, 2019, with 1 commit to master since this tag. The release notes mention 'Close serial port when crash'. Under 'Assets', there are two download links: 'Source code (zip)' and 'Source code (tar.gz)'. The URL for the page is <https://github.com/kendryte/kflash.py/releases/tag/v0.9.1>.

Extract file. Then you will get kflash_gui.exe



open kflash_gui v1.5





Advance (Optional)

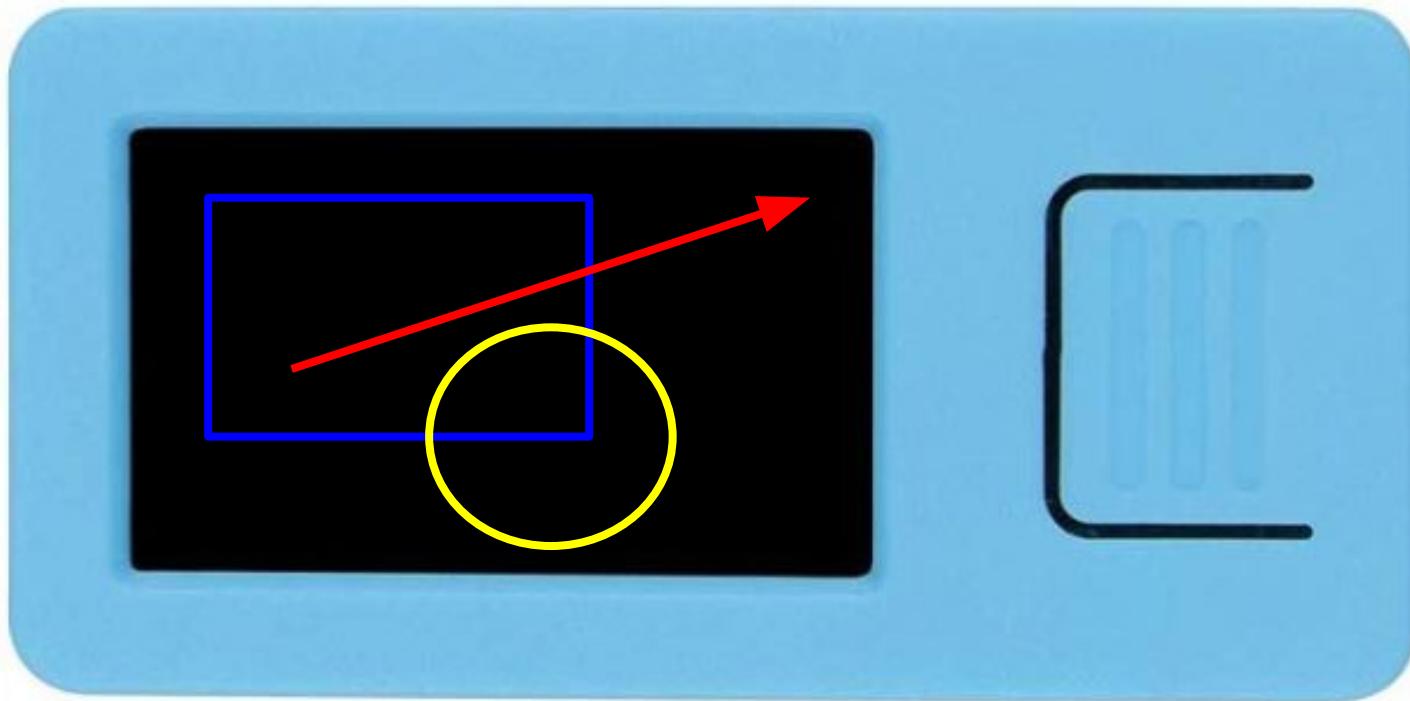
- Anaconda
- LabelImg
- Maix_Toolbox



Hand-on 1

- Board Example

Lab 1 : Display



0100_Display_TEST.py

```
lcd.clear(lcd.WHITE)
time.sleep_ms(1000)

lcd.clear(lcd.BLACK)
time.sleep_ms(1000)

while True:
    lcd.rotation(2)
    lcd.draw_string(0, 0, "Hello World!")
    lcd.draw_string(0, 15, "M5 Stick V")
    lcd.draw_string(0, 30, "LCD Display 135x240")
```

```
lcd.clear(lcd.WHITE)  
time.sleep_ms(1000)
```



เครื่องหน้าจอ โดยพื้นหลังเป็นสีขาว

```
lcd.clear(lcd.BLACK)  
time.sleep_ms(1000)
```

```
while True:
```

```
    lcd.rotation(2)
```



หมุนจอในตำแหน่งที่ 2

```
    lcd.draw_string(0, 0, "Hello World!")
```



เขียนข้อความ ที่ตำแหน่ง X=0 และ Y=0

0101_Display_DrawString.py

```
import lcd
import image

lcd.init(type=1, freq=15000000, color=lcd.WHITE)
lcd.rotation(2)

while True:
    lcd.draw_string(0, 0, "M5Stick V 240x135 pixels", lcd.BLACK, lcd.WHITE)
    lcd.draw_string(0, 15, "Line 2")
    lcd.draw_string(0, 30, "Line 3")
    lcd.draw_string(0, 45, "Line 4")
    lcd.draw_string(0, 60, "Line 5")
    lcd.draw_string(0, 75, "Line 6")
    lcd.draw_string(0, 90, "Line 7")
    lcd.draw_string(0, 105, "Line 8")
```

0102_Display_DrawShape.py

```
import lcd
import image
import sensor

lcd.init(type=1, freq=15000000, color=lcd.WHITE)
lcd.rotation(2)
lcd.clear(lcd.BLACK)

sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.run(1)

lcd.draw_string(0, 0, "M5StickV Capture & Draw Image")
time.sleep_ms(3000)

img = sensor.snapshot()

while True:
    img = img.draw_string(60, 60, "Draw Shape", scale=2)
    img = img.draw_line(50, 150, 240, 100, color = (255,255,0), thickness = 5)
    img = img.draw_arrow(70, 90, 170, 170, color = (0,0,255), thickness = 5)
    img = img.draw_circle(100, 150, 30, color = (255,0,0), thickness=3)
    img = img.draw_ellipse(220, 150, 30, 20, color = (0,255,0), thickness=3)
    lcd.display(img)
    time.sleep_ms(1)
```

Lab 2 : Camera



0200_Display_LiveStream.py

```
import lcd
import image
import sensor

lcd.init()
lcd.rotation(2)

sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.run(1)

lcd.draw_string(40, 60, "M5StickV Live Stream", lcd.WHITE, lcd.BLACK)
time.sleep_ms(3000)

while True:
    img = sensor.snapshot()
    img.draw_string(60, 60, "Live Stream", color = (255,0,0), scale=2)
    lcd.display(img)
```

```
import lcd
import image
import sensor
lcd.init()
lcd.rotation(2)
while True:
    img = sensor.snapshot()
    img.draw_string(60, 60, "Live Stream", color = (255,0,0), scale=2)
    lcd.display(img)
```

เรียกใช้งานไลบรารี LCD, Image, Sensor

เริ่มต้นใช้งาน LCD และตั้งค่าการหมุนจอที่ตำแหน่ง 2

อ่านค่าจากกล้อง เก็บไว้ในตัวแปร img

แสดงภาพออกจากแสดงผล

Lab 3 : LED

LED RGB W



0300_LED_TEST.py

```
# Setup LED Mode
led_w = GPIO(GPIO.GPIO3, GPIO.OUT)
led_r = GPIO(GPIO.GPIO4, GPIO.OUT)
led_g = GPIO(GPIO.GPIO5, GPIO.OUT)
led_b = GPIO(GPIO.GPIO6, GPIO.OUT)

# LED is Active Low (0 is ON, 1 is OFF)
led_w.value(1)
led_r.value(1)
led_g.value(1)
led_b.value(1)

while(True):
    led_w.value(0)
    led_r.value(1)
    led_g.value(1)
    led_b.value(1)
    time.sleep(1)
```

```
from Maix import GPIO  
from fpioa_manager import *  
from board import board_info
```

← เรียกใช้งานไลบรารี GPIO ของบอร์ด

```
fm.register(board_info.LED_W, fm.fpioa.GPIO3)
```

← ระบุตำแหน่งรีเซเตอร์ขาใช้งาน

```
led_w = GPIO(GPIO.GPIO3, GPIO.OUT)
```

← กำหนดขาใช้งานเป็น OUTPUT Mode

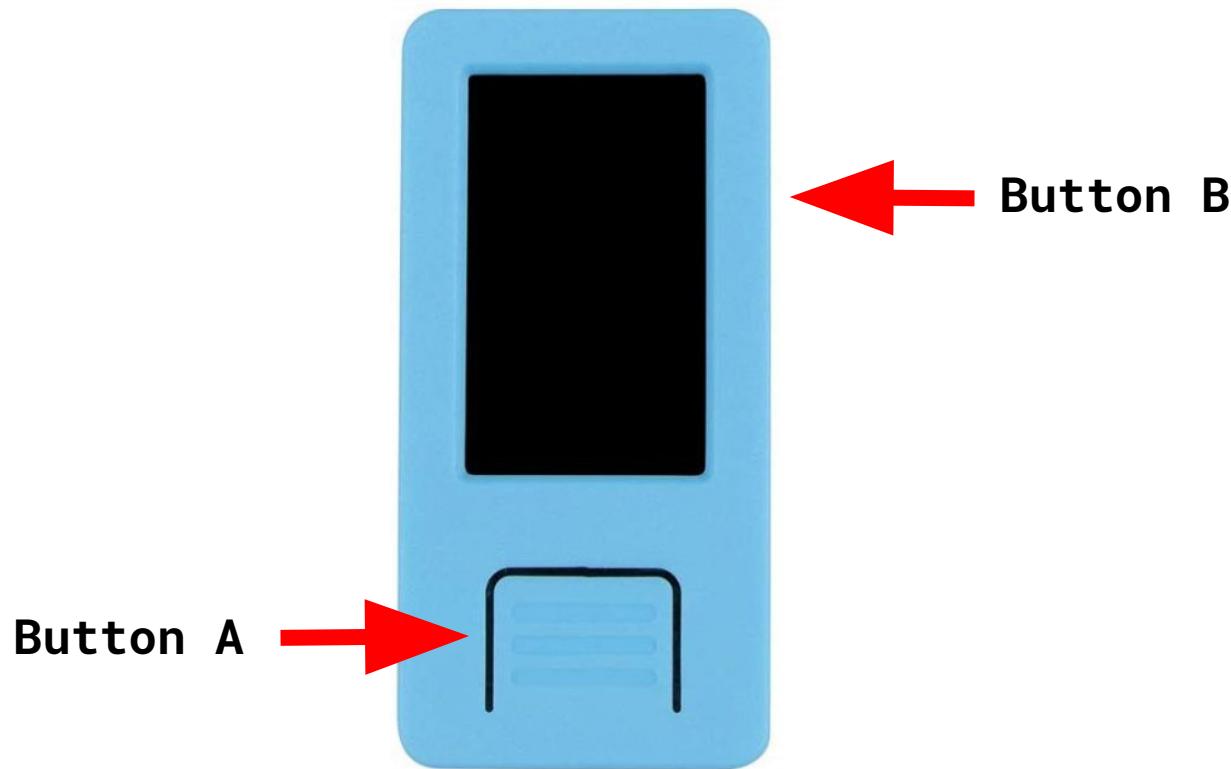
```
led_w.value(1)
```

← ควบคุมการติด ดับของหลอดไฟ

โดย value = 0 คือ ไฟติด

value = 1 คือ ไฟดับ

Lab 4 : Button



0400_LED_Button.py

```
# Setup LED Mode
led_w = GPIO(GPIO.GPIO3, GPIO.OUT)
led_r = GPIO(GPIO.GPIO4, GPIO.OUT)
led_g = GPIO(GPIO.GPIO5, GPIO.OUT)
led_b = GPIO(GPIO.GPIO6, GPIO.OUT)

# LED is Active Low (0 is ON, 1 is OFF)
led_w.value(1)
led_r.value(1)
led_g.value(1)
led_b.value(1)

while(True):
    if button_a.value() == 0 and button_b.value() == 0:
        time.sleep_ms(200)
        print("Button A and B Press")
        led_r.value(0)
        led_g.value(1)
        led_b.value(1)
```

```
from Maix import GPIO  
from fpioa_manager import *  
from board import board_info
```

← เรียกใช้งานไลบรารี GPIO ของบอร์ด

```
fm.register(board_info.BUTTON_A, fm.fpioa.GPIO1) ← ระบุตำแหน่งรีจิสเตอร์ขาใช้งาน
```

```
button_a = GPIO(GPIO.GPIO1, GPIO.IN, GPIO.PULL_UP)
```

↑
กำหนดขาใช้งานปุ่มกดเป็น INPUT
แบบ PULL UP

```
button_a.value()
```

← ค่าสถานะของปุ่มกด จะถูกเก็บไว้ในตัวแปร button_a
โดย value = 0 คือ กดปุ่ม
value = 1 คือ ไม่กดปุ่ม

Lab 5 : PWM

LED RGB W



0500_LED_PWM.py

```
# PWM Pin
pwm_led = board_info.LED_W

pwm = Timer(Timer.TIMER0, Timer.CHANNEL0, mode=Timer.MODE_PWM)
PWM_ch = PWM(pwm, freq=500000, duty=0, pin=pwm_led)

duty_val = 100

try:
    while(True):
        # PWM Value 0 is ON and 100 is OFF
        for duty_val in range(100, 0, -1):
            PWM_ch.duty(duty_val)
            time.sleep_ms(10)
        for duty_val in range(0, 100, 1):
            PWM_ch.duty(duty_val)
            time.sleep_ms(10)
```

```
pwm_led = board_info.LED_W
```



กำหนดตัวแปร `pwm_led` ให้เป็นหลอดไฟสีขาว

```
pwm = Timer(Timer.TIMER0, Timer.CHANNEL0, mode=Timer.MODE_PWM)
```



เรียกใช้งาน Timer0

```
PWM_ch = PWM(pwm, freq=500000, duty=0, pin=pwm_led)
```



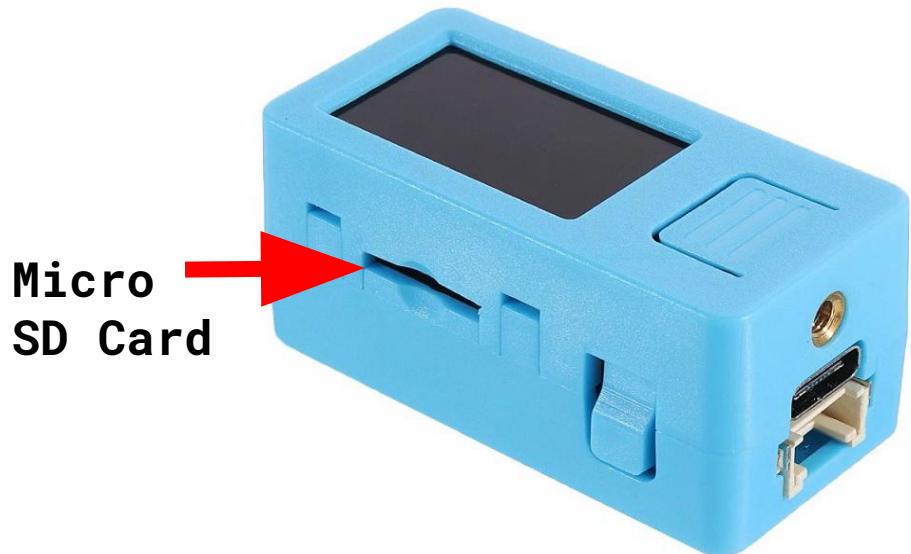
ตั้งค่าการใช้งานสัญญาณพัลส์
และกำหนดความถี่ที่ใช้ 500khz

```
PWM_ch.duty(duty_val)
```



ส่งสัญญาณ PWM โดยมีค่าระหว่าง 0 - 100

Lab 6 : SD Card & WAV play



Micro
SD Card



Speaker

06oo_File_Take_Photos.py

img = sensor.snapshot() ← อ่านค่าจากกล้อง เก็บไว้ในตัวแปร img

img.save("/sd/photos/1.jpg" , quality=95) ← บันทึกรูปไปยังโฟลเดอร์ที่ระบุ

```
# Take Photos
if button_a.value() == 0:
    cnt_save+=1
    fname = "/sd/photos/" +str(cnt_save)+ ".jpg"
    print (fname)

    img.save(fname, quality=95)
```

06oo_File_Take_Photos.py

```
def play_sound(filename):
    try:
        player = audio.Audio(path = filename)
        player.volume(100)
        wav_info = player.play_process(wav_dev)
        wav_dev.channel_config(wav_dev.CHANNEL_1, I2S.TRANSMITTER,resolution =
        wav_dev.set_sample_rate(wav_info[1]))
        while True:
            ret = player.play()
            if ret == None:
                break
            elif ret==0:
                break
            player.finish()
    except:
        pass

print(os.listdir())

try:
    while True:
        lcd.draw_string(5, 5, "Play WAV from SD Card", lcd.WHITE, lcd.BLACK)
        lcd.draw_string(80, 60, "Press Button A -->", lcd.WHITE, lcd.BLACK)
        if button_a.value() == 0:
            play_sound("/sd/wav/logo.wav")
            print("Play WAV song")
```

```
def play_sound(filename):
    try:
        player = audio.Audio(path = filename)
        player.volume(100)
        wav_info = player.play_process(wav_dev)
        wav_dev.channel_config(wav_dev.CHANNEL_1, I2S.TRANSMITTER,resolution = I2S.RESOLUTION_16_BIT,
align_mode = I2S.STANDARD_MODE)
        wav_dev.set_sample_rate(wav_info[1])
        while True:
            ret = player.play()
            if ret == None:
                break
            elif ret==0:
                break
        player.finish()
    except:
        pass
```

ฟังก์ชันเล่นไฟล์นามสกุล .wav
ที่รับ filename ด้วยรับตัวแปรชื่อ filename

play_sound("/sd/wav/logo.wav")

เรียกใช้งานฟังก์ชัน โดยป้อน Path ของไฟล์เดอร์ที่มีไฟล์นามสกุล .wav

Lab 7 : Gyro & Acceleration



0700_MP6886.py

```
MPU6866_init()

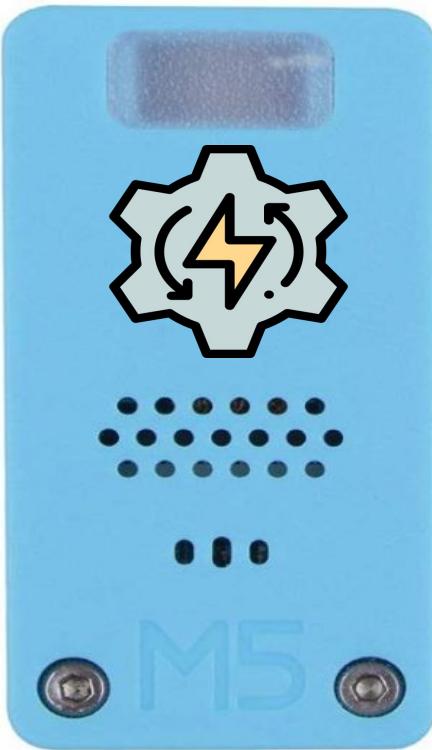
aRes = 8.0/32768.0;

while True:
    x,y,z=MPU6866_read()
    accel_array = [x*aRes, y*aRes, z*aRes]
    print(accel_array);

    lcd.draw_string(10,10,"Gyro & Acc MPU6886")
    lcd.draw_string(20,50,"x:"+str(accel_array[0]))
    lcd.draw_string(20,70,"y:"+str(accel_array[1]))
    lcd.draw_string(20,90,"z:"+str(accel_array[2]))
    time.sleep_ms(10)
```

```
def write_i2c(address, value) ← พังก์ชันติดต่อสื่อสาร I2C  
  
def MPU6886_init() ← พังก์ชันเริ่มต้นการทำงานของ MPU6886  
  
def MPU6886_read() ← พังก์ชันอ่านค่าจากเซ็นเซอร์  
  
MPU6886_init() ← เรียกใช้งาน MPU6886  
  
x,y,z = MPU6886_read() ← อ่านค่าจากเซ็นเซอร์เก็บไว้ในตัวแปร x, y, z  
  
accel_array = [x*aRes, y*aRes, z*aRes] ← คำนวณมุกการเอียง  
  
lcd.draw_string(20,50,"x:"+str(accel_array[0])) ← แสดงผลลัพท์ที่หน้าจอ
```

Lab 8 : Power management



0800_Power_Management.py

```
axp = pmu.axp192()
axp.enableADCs(True)

while True:
    vbat = axp.getVbatVoltage()
    usb_vol = axp.getUSBVoltage()
    usb_cur = axp.getUSBInputCurrent()
    connext_vol = axp.getConnextVoltage()
    connext_input_current = axp.getConnextInputCurrent()
    bat_current= axp.getBatteryChargeCurrent()
    bat_dis_current = axp.getBatteryDischargeCurrent()
    bat_instant_watts = axp.getBatteryInstantWatts()
    temp = axp.getTemperature()

    lcd.drawString(20,0,"usb_vol:"+str(usb_vol)+"mV : "+str(usb_vol/1000)+"V")
    lcd.drawString(20,15,"usb_cur:"+str(usb_cur)+"mA")
    lcd.drawString(20,30,"connext_vol:"+str(connext_vol)+"mV")
    lcd.drawString(20,45,"connext_input_current:"+str(connext_input_current))
    lcd.drawString(20,60,"bat_current:"+str(bat_current))
    lcd.drawString(20,75,"bat_dis_current:"+str(bat_dis_current))
    lcd.drawString(20,90,"bat_instant_watts:"+str(bat_instant_watts))
    lcd.drawString(20,105,"temp:"+str(temp))
```

```
import pmu
```

เรียกใช้งานไลบรารี pmu

```
axp = pmu.axp192()
```

กำหนดคุณลักษณะของ axp

```
axp.enableADCs(True)
```

เปิดการใช้งาน การอ่านค่าแบบ ADC (Analog to Digital)

```
vbat = axp.getVbatVoltage()
```

อ่านค่าแรงดันจาก Battery มีหน่วยเป็น mV

```
usb_vol = axp.getUSBVoltage()
```

อ่านค่าแรงดันจาก USB มีหน่วยเป็น mV

```
usb_cur = axp.getUSBInputCurrent()
```

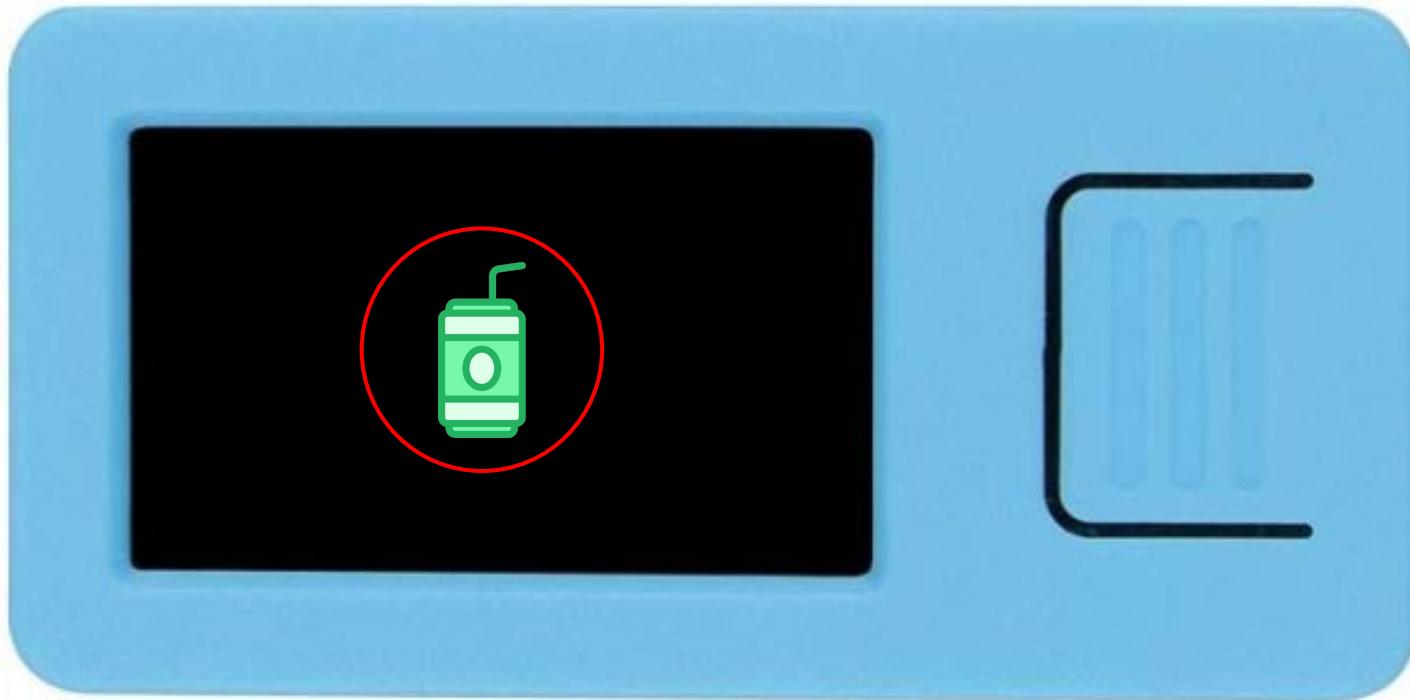
อ่านค่ากระแสจาก USB มีหน่วยเป็น mA

Hand-on 2

- Images
and Face Detection



Lab 9 : Find Green



0900_Find_Green.py

```
#thresholds = (30, 100, 15, 127, 15, 127)      # RED
thresholds = (30, 100, -64, -8, -32, 32)       # GREEN
#thresholds = (0, 30, 0, 64, -128, 0)           # BLUE

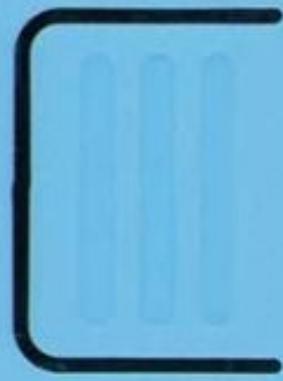
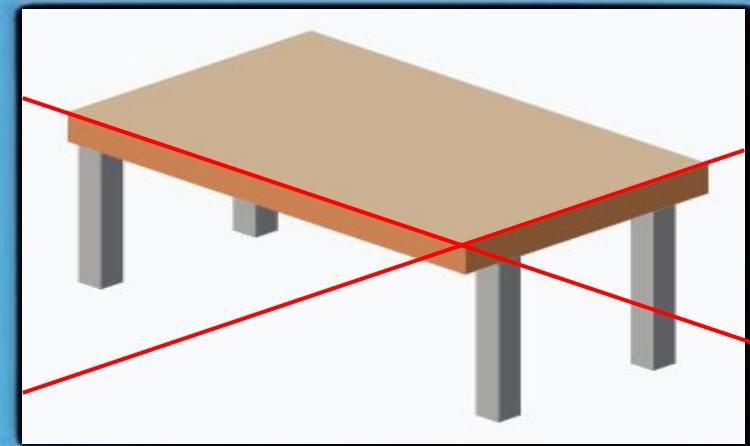
while True:
    img=sensor.snapshot()
    blobs = img.find_blobs([thresholds])
    if blobs:
        for b in blobs:
            tmp=img.draw_rectangle(b[0:4])
            tmp=img.draw_cross(b[5], b[6])
            c=img.get_pixel(b[5], b[6])
    lcd.display(img)
```

```
#thresholds = (30, 100, 15, 127, 15, 127)      # Find RED color
thresholds = (30, 100, -64, -8, -32, 32)        # Find GREEN color
#thresholds = (0, 30, 0, 64, -128, 0)           # Find BLUE color
```

img.find_blobs([thresholds]) ← color block object

img.draw_rectangle(b[0:4]) ← วาดรูปกล่องสีเหลือง

Lab 10 : Find Line



1000_Find_Line.py

```
while True :  
    img  = sensor.snapshot ()  
    lines = img.find_lines ()  
  
    if lines:  
        print (str(lines))  
        print (str(lines[0][0]) + " to " + str(lines[0][1]))  
        print (str(lines[0][2]) + " to " + str(lines[0][3]))  
        for l in lines:  
            img.draw_line(l.line(), color = (255, 0, 0))  
        lcd.display(img)
```

```
img = sensor.snapshot () ← ดึงภาพจากกล้อง  
lines = img.find_lines () ← ค้นหารูปร่างเส้นตรง และเก็บไว้ในตัวแปร lines
```

```
if lines:  
    print (str(lines))  
    print (str(lines[0][0]) + " to " + str(lines[0][1])) ← แสดงผลลัพธ์  
    print (str(lines[0][2]) + " to " + str(lines[0][3]))  
for l in lines:  
    img.draw_line(l.line(), color = (255, 0, 0)) ← วาดเส้นจากจุดที่ตรวจจับเจอ  
    lcd.display(img)
```

Lab 11 : Find Circles



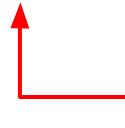
1100_Find_Circle.py

```
while True:
    img = sensor.snapshot()
    for c in img.find_circles(threshold = 3500, x_margin = 10, y_margin = 10,
        img.draw_circle(c.x(), c.y(), c.r(), color = (255, 0, 0)))
    img.draw_rectangle(80,60,160,120)
    lcd.display(img)

print("finish")
lcd.clear()
```

```
img = sensor.snapshot() ← แสดงผลลัพธ์
```

```
for c in img.find_circles(threshold = 3500, x_margin = 10, y_margin = 10,  
    r_margin = 10, r_min = 2, r_max = 100, r_step = 2, roi=(80,60,160,120)):
```

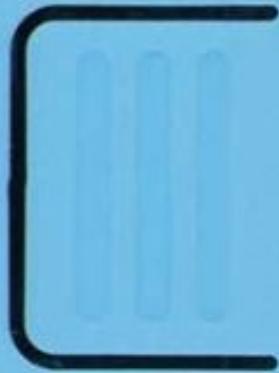


ตรวจจับรูปทรงวงกลม

```
img.draw_circle(c.x(), c.y(), c.r(), color = (255, 0, 0)) ← วาดเส้นวงกลม
```

```
Lcd.display(img) ← แสดงผลลัพธ์ที่หน้าจอ
```

Lab 12 : Find Rectangle.



1200_Find_Rectangle.py

```
while True:
    img = sensor.snapshot()
    for c in img.find_circles(threshold = 3500, x_margin = 10, y_margin = 10,
        img.draw_circle(c.x(), c.y(), c.r(), color = (255, 0, 0)))
    img.draw_rectangle(80,60,160,120)
    lcd.display(img)

print("finish")
lcd.clear()
```

```
img = sensor.snapshot () ← ดึงภาพจากกล้อง  
lines = img.find_lines () ← ค้นหารูปร่างเส้นตรง และเก็บไว้ในตัวแปร lines
```

```
if lines:  
    print (str(lines))  
    print (str(lines[0][0]) + " to " + str(lines[0][1])) ← แสดงผลลัพธ์  
    print (str(lines[0][2]) + " to " + str(lines[0][3]))  
for l in lines:  
    img.draw_line(l.line(), color = (255, 0, 0)) ← วาดเส้นจากจุดที่ตรวจจับเจอ  
    lcd.display(img)
```

Lab 13 : Find Face with Cascade



1300_Find_Face_Eyes.py

```
while True:  
    # Capture snapshot  
    img = sensor.snapshot()  
  
    objects = img.find_features(face_cascade, threshold=0.5, scale_factor=1.5)  
    for face in objects:  
        img.draw_rectangle(face, color = (255, 0, 0))  
        eyes = img.find_features(eyes_cascade, threshold=0.5, scale_factor=1.2,  
        for e in eyes:  
            img.draw_rectangle(e, color = (0, 0, 255))  
lcd.display(img)
```

```
face_cascade = image.HaarCascade("frontalface", stages=25)  
eyes_cascade = image.HaarCascade("eye", stages=24)
```

โหลดโมเดล Cascade

```
img = sensor.snapshot()
```

ดึงภาพจากกล้อง

```
objects = img.find_features(face_cascade, threshold=0.5, scale_factor=1.5)  
for face in objects:  
    img.draw_rectangle(face, color = (255, 0, 0))  
    eyes = img.find_features(eyes_cascade, threshold=0.5, scale_factor=1.2, roi=face)  
    for e in eyes:  
        img.draw_rectangle(e, color = (0, 0, 255))  
lcd.display(img)
```

ตรวจจับใบหน้า

ตรวจจับดวงตา

Lab 14 : Find Face with kmodel



1400_FaceDetection.py

```
print("load model...")
task = kpu.load("/sd/model/face.kmodel")
print("done...!")

anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275,
          7.5245, 8.5245, 9.5245, 10.5245, 11.5245, 12.5245, 13.5245, 14.5245, 15.5245)

a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)

while(True):
    img = sensor.snapshot()
    code = kpu.run_yolo2(task, img)
    print(code)
    if code:
        for i in code:
            print(i)
            a = img.draw_rectangle(i.rect())
    a = lcd.display(img)
```

```
task = kpu.load("/sd/model/face.kmodel") ← โหลดโมเดลจาก SD Card
```

```
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275, 6.718375, 9.01025)
```

```
a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor) ← เริ่มต้นใช้งาน yolo2
```

```
img = sensor.snapshot() ← ดึงภาพจากกล้อง
```

```
code = kpu.run_yolo2(task, img) ← สั่งให้ yolo2 ทำงาน  
ส่งเอาต์พุตเก็บไว้ที่ตัวแปร code
```

```
if code:
```

```
    for i in code:
```

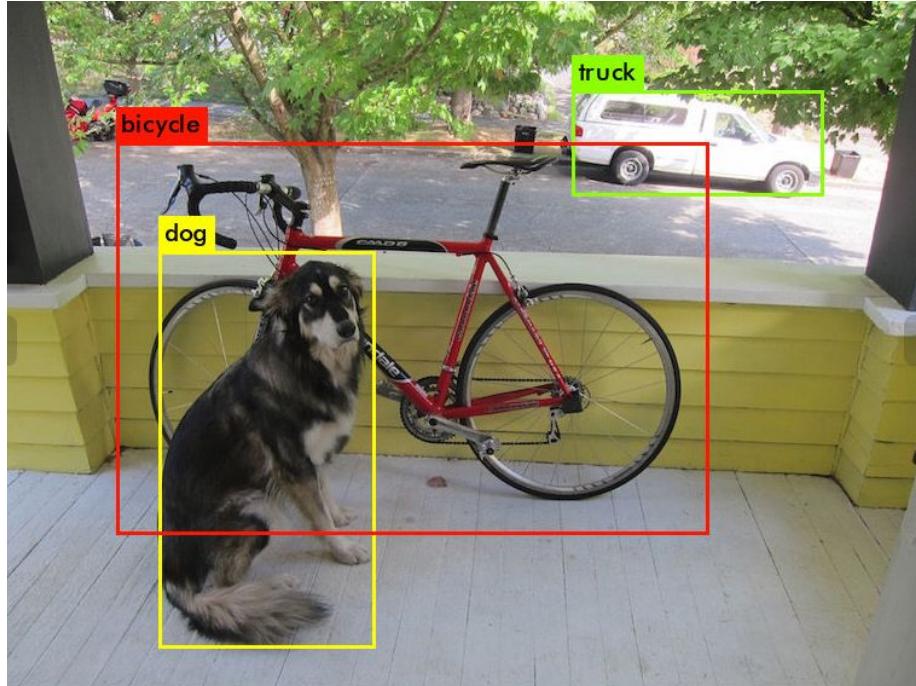
```
        print(i)
```

```
        a = img.draw_rectangle(i.rect()) ← วาดรอบสี่เหลี่ยม มีเจือใบหน้า
```

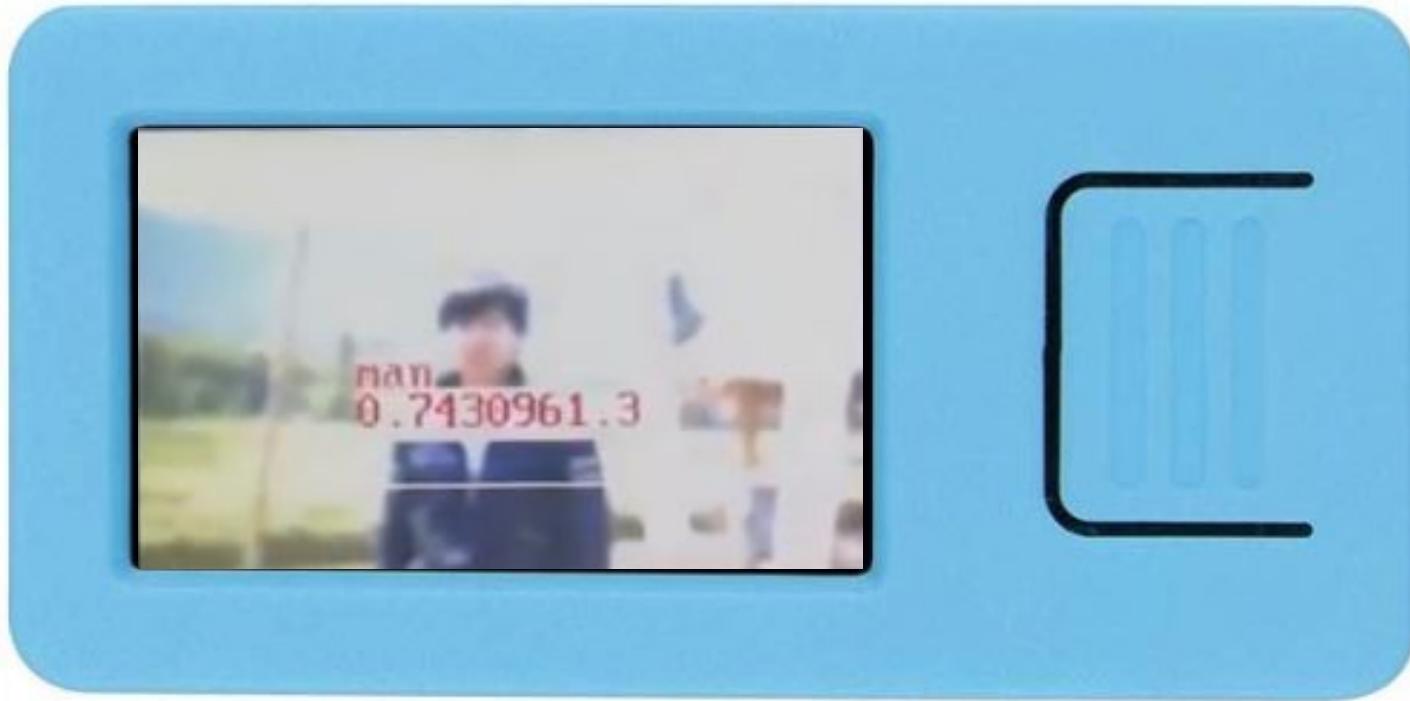
```
a = lcd.display(img)
```

Hand-on

- Object Classification



Lab 15 : Object Classification



1500_ObjectClassification.py

```
classes = ['aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus', 'car', 'task = kpu.load("/sd/model/20class.kmodel")
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, 5.3658, 5.155437, 6.92275
a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor)

while(True):
    clock.tick()
    img = sensor.snapshot()
    code = kpu.run_yolo2(task, img)
    print(clock.fps())
    if code:
        for i in code:
            a=img.draw_rectangle(i.rect())
            a = lcd.display(img)
            for i in code:
                lcd.draw_string(i.x(), i.y(), classes[i.classid()], lcd.RED,
                                lcd.draw_string(i.x(), i.y()+12, '%f1.3' %i.value(), lcd.RED,
else:
    a = lcd.display(img)
```

```
classes = ['aeroplane', 'bicycle', 'bird', 'boat', 'bottle', '~'] ← ดึงภาพจากกล้อง
```

```
task = kpu.load("/sd/model/20class.kmodel") ← ดึงภาพจากกล้อง
```

```
anchor = (1.889, 2.5245, 2.9465, 3.94056, 3.99987, ~)
```

```
a = kpu.init_yolo2(task, 0.5, 0.3, 5, anchor) ← เริ่มต้นใช้งาน yolo2
```

```
img = sensor.snapshot()
```

```
code = kpu.run_yolo2(task, img) ← สั่งให้ yolo2 ทำงาน
```

```
if code: ← ส่งเอาต์พุตเก็บไว้ที่ตัวแปร code
```

```
for i in code:
```

```
    a=img.draw_rectangle(i.rect()) ← วาดกรอบสี่เหลี่ยมรอบวัตถุที่ตรวจพบ
```

```
    a = lcd.display(img)
```

```
    for i in code:
```

```
        lcd.draw_string(i.x(), i.y(), classes[i.classid()], lcd.RED, lcd.WHITE) ← เขียนชื่อคลาส
```

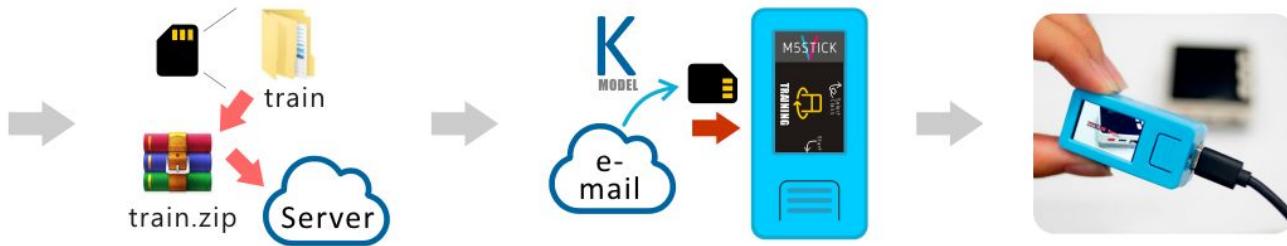
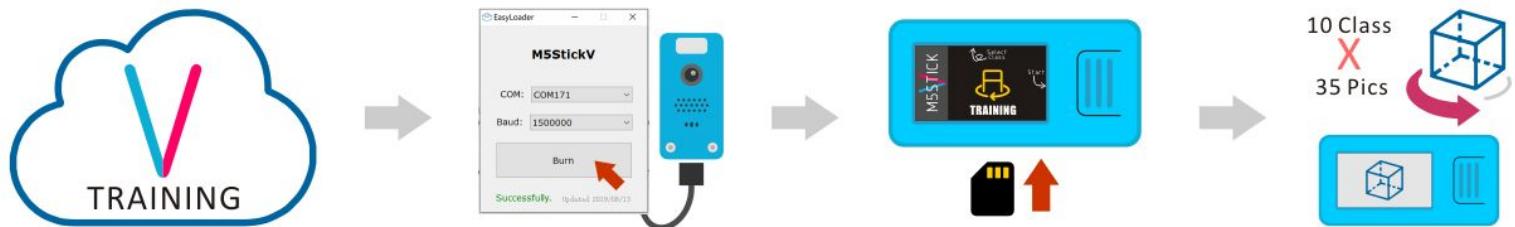
```
        lcd.draw_string(i.x(), i.y()+12, '%f1.3%i.value()', lcd.RED, lcd.WHITE)
```

บันหน้าจอ



Advance :

<http://v-training.m5stack.com>



ZIP &
Upload

Download
Model

Reset &
Run