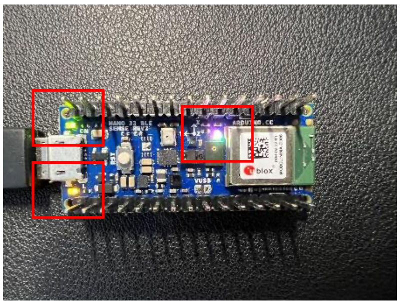
Tutorial 2 – Nano BLE Sense Rev2



In MicroPython, these four LEDs are defined as

LED-Orange Pin13	LED(4)
LED-Red	LED(1)
LED-Green	LED(2)
LED-Blue	LED(3)

Note: the RGB LED is controlled by 3 pins (Pin 24 16 06). You can control these LEDs simply by on/off methods.

Example Code:

```
from board import LED
import time
led_red = LED(1)
led_green = LED(2)
led_blue = LED(3)
led_builtin = LED(4)
while (True):

    # Turn on LEDs
    led_red.on()
    led_green.on()
    led_blue.on()
    led_builtin.on()
    # Wait 0.25 seconds
    time.sleep_ms(250)

    # Turn off LED
    led_builtin.off()
    led_red.off()
    led_green.off()
    led_blue.off()
    # Wait 0.25 seconds
    time.sleep_ms(250)
```

Sensors

Compared with complicated C/C++ codes, MicroPython provides much easier APIs to control and access the sensors’ data.

IMU

The gyroscope and accelerometer (BMI270) and magnetometer (BMM150) are connected to the microcontroller via I2C bus on pin 14 (SDA) and 15(SCL).

Example Code

```
import time
import imu
from machine import Pin, I2C

bus = I2C(1, scl=Pin(15), sda=Pin(14))
imu = imu.IMU(bus)

while (True):
    print('Accelerometer: x:{:>8.3f} y:{:>8.3f}
z:{:>8.3f}'.format(*imu.accel()))
    print('Gyroscope:     x:{:>8.3f} y:{:>8.3f}
z:{:>8.3f}'.format(*imu.gyro()))
    print('Magnetometer:  x:{:>8.3f} y:{:>8.3f}
z:{:>8.3f}'.format(*imu.magnet()))
    print("")
    time.sleep_ms(100)
```

Temperature and Humidity

The temperature and humidity sensors (HS3003) are connected to the microcontroller via I2C bus on pin 14 (SDA) and 15 (SCL).

Example Code:

```
import time
import hs3003
from machine import Pin, I2C
bus = I2C(1, scl=Pin(15), sda=Pin(14))
hs = hs3003.HS3003(bus)

while (True):
    rH   = hs.humidity()
    temp = hs.temperature()
    print ("rH: %.2f%% T: %.2fC" %(rH, temp))
    time.sleep_ms(100)
```

Pressure

The barometric pressure and temperature sensors (LPS22HB) are connected to the microcontroller via I2C bus on pin 14 (SDA) and 15 (SCL).

```
import time
import lps22h
from machine import Pin, I2C

bus = I2C(1, scl=Pin(15), sda=Pin(14))
lps = lps22h.LPS22H(bus)

while (True):
    pressure = lps.pressure()
    temperature = lps.temperature()
    print("Pressure: %.2f hPa Temperature: %.2f C"%(pressure,
temperature))
    time.sleep_ms(200)
```

Ambient Light and Proximity

The ambient light and proximity sensors (APDS9960) are connected to the microcontroller via I2C bus on pin 14 (SDA) and 15(SCL).

```
from time import sleep_ms
from machine import Pin, I2C
from apds9960.const import *
from apds9960 import uAPDS9960 as APDS9960

bus = I2C(1, sda=Pin(14), scl=Pin(15))
apds = APDS9960(bus)

apds.enableLightSensor()

while True:
    sleep_ms(250)
    val = apds.readAmbientLight()
    print("AmbientLight={}".format(val))
```

```
from time import sleep_ms
from machine import Pin, I2C

from apds9960.const import *
from apds9960 import uAPDS9960 as APDS9960

bus = I2C(1, sda=Pin(14), scl=Pin(15))
apds = APDS9960(bus)

apds.setProximityIntLowThreshold(50)
apds.enableProximitySensor()
while True:
    sleep_ms(250)
    val = apds.readProximity()
    print("proximity={}".format(val))
```

Note: the magic value of 50 in setProximityIntLowThreshold() is empirically chosen. Please try other values e.g., 0 and 100, to see their impacts.

Conclusion

This tutorial shows how to control/access onboard sensors.