

```

1  from machine import Pin
2  from time import sleep
3  from machine import ADC, Pin
4  import machine
5
6
7  sdaPIN= Pin(0)#IC2 0
8  sclPIN= Pin(1)#IC2 0
9  sda2PIN = Pin(10) #IC2 1
10 scl2PIN = Pin(11) #IC2 1
11
12 accelerometer1 = adafruit_adxl34x.ADXL345(i2c0)
13 accelerometer2 = adafruit_adxl34x.ADXL345(i2c1)
14 i2c = machine.I2C(1,sda=sda2PIN, scl=scl2PIN, freq=400000)
15
16 devices = i2c.scan()
17 if len(devices) == 0:
18     print("No i2c device !")
19 else:
20     print('i2c devices found:',len(devices))
21 for device in devices:
22     print("At address: ",hex(device))
23
24 i2c2 = machine.I2C(0,sda=sdaPIN, scl=sclPIN, freq=400000)
25 devices = i2c2.scan()
26 if len(devices) == 0:
27     print("No i2c device !")
28 else:
29     print('i2c devices found:',len(devices))
30 for device in devices:
31     print("At address: ",hex(device))
32
33 ## Initialize ADXL345 supposedly this will work
34 def init_adxl345():
35     i2c.writeto_mem(ADXL345_ADDRESS, ADXL345_POWER_CTL, bytearray([0x08])) # Set bit 3 to 1 to
enable measurement mode
36     i2c.writeto_mem(ADXL345_ADDRESS, ADXL345_DATA_FORMAT, bytearray([0x0B])) # Set data format to
full resolution, +/- 16g
37
38 # Read acceleration data
39 def read_accel_data():
40     data = i2c.readfrom_mem(ADXL345_ADDRESS, ADXL345_DATA0, 6)
41     x, y, z = ustruct.unpack('<3h', data)
42     return x, y, z
43
44 # Main loop
45 init_adxl345()
46 while True:
47     x, y, z = read_accel_data()
48     print("X: {}, Y: {}, Z: {}".format(x, y, z))
49     time.sleep(0.1)

```