

Check Device I2C Address

Aug 2026



Introduction

When you attach a device on the I2C bus, even though you might know it's address, it's a great idea to run a test to see if you can read it's actual address. This achieves two things, it proves your wiring is good, because if you don't have the correct cabling, this script will fail, and it proves that the device does appear to function.

In this example, I have an LCD address on I2C bus zero, and I know it's address is 3C0 Hex, which in decimal is 60. So this test will confirm that my assumptions about my device address is good.

I2C Address Check

Below is the code we need to run to check for devices on the I2C.

[download](#)

```
from machine import Pin, I2C

i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=200000)
print("I2C addresses:", i2c.scan())
```

There are a couple of considerations around this code, and they are:

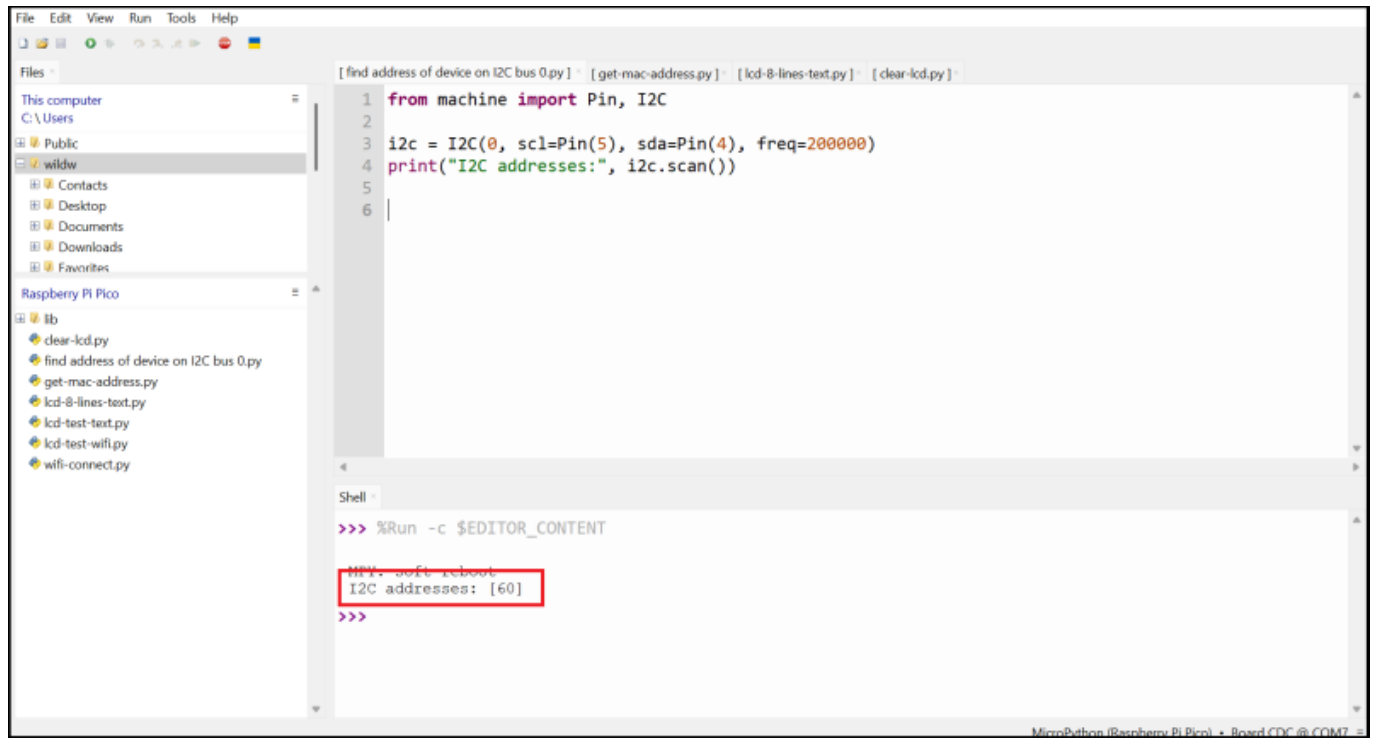
- `i2c = I2C(0)` - The Zero is for Bus 0

The Pico has Bus 0 and Bus 1, and you should know which you used.

- `scl=Pin(5), sda=Pin(4)`

[These are the Pico pins your device is connected to, so `scl\(5\)` is Pico Pin 7 and `sda\(4\)` is Pico pin 6.](#)

If you run the code above, you will see something similar to below:



```
[ find address of device on I2C bus 0.py ] [ get-mac-address.py ] [ lcd-8-lines-text.py ] [ clear-lcd.py ]  
1 from machine import Pin, I2C  
2  
3 i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=200000)  
4 print("I2C addresses:", i2c.scan())  
5  
6  
  
Shell  
>>> %Run -c $EDITOR_CONTENT  
  
NEW: soft reboot  
I2C addresses: [60]  
>>>
```

Here we can see the I2C address is listed as 60 and this is a Decimal address.

If you convert 60 Dec to Hex, then you get Hex 3C, which is exactly the address the LCD is currently set to.

From:

<https://wiki.alanwalker.uk/> - WalkerWiki - wiki.alanwalker.uk

Permanent link:

https://wiki.alanwalker.uk/doku.php?id=check_device_i2c_address&rev=1787141715

Last update: 2026/08/19 12:15

