

Using I2C

RDK X5 enables I2C5 (physical pins 3 and 5) and I2C0 (physical pins 27 and 28) on 40pin by default, with IO voltage 3.3V.

Please refer to `/app/40pin_samples/test_i2c.py` for details on how to use I2C.

Testing Method

- Run the test program `python3 /app/40pin_samples/test_i2c.py`
- List the currently enabled I2C buses
- Scan the bus to see which devices are connected to the bus by inputting the bus number
- Input the device address (in hexadecimal), and the test program will read one byte of data from that device

Test Result

```
Starting demo now! Press CTRL+C to exit
List of enabled I2C controllers:
/dev/i2c-0 /dev/i2c-1
Please input I2C BUS num:1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:      -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: 40 -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- --
Please input I2C device num(Hex):40
Read data from device 40 on I2C bus 1
read value= b''
```

Test Code

```
#!/usr/bin/env python3

import sys
import os
import time

# import i2cdev
from i2cdev import I2C

def i2cdevTest():
    # device, bus = 0x51, 0
    bus = input("Please input I2C BUS num:")
    os.system('i2cdetect -y -r ' + bus)
```

```
device = input("Please input I2C device num(Hex):")
print("Read data from device %s on I2C bus %s" % (device, bus))
i2c = I2C(eval("0x" + device), int(bus))
value = i2c.read(1)
i2c.write(value)
print("read value=", value)
i2c.close()

if __name__ == '__main__':
    print("Starting demo now! Press CTRL+C to exit")
    print("List of enabled I2C controllers:")
    os.system('ls /dev/i2c*')
    while True:
        i2cdevTest()
```