

Application Note AN127:

Raspberry Pi UART Interface to COZIR CO2 Sensor

Introduction

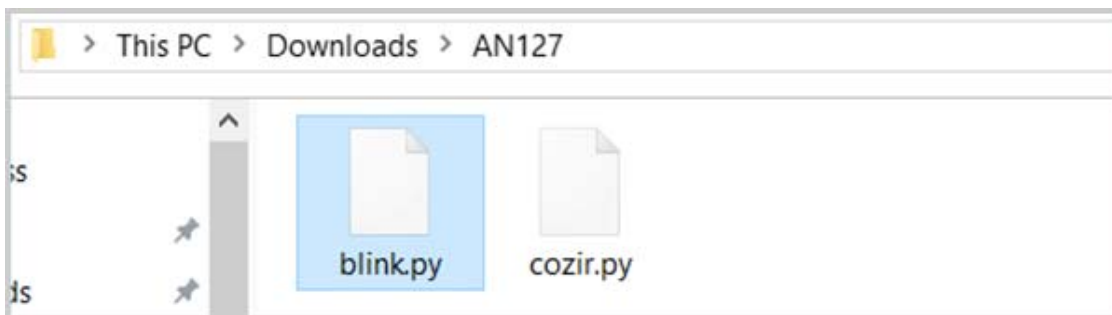
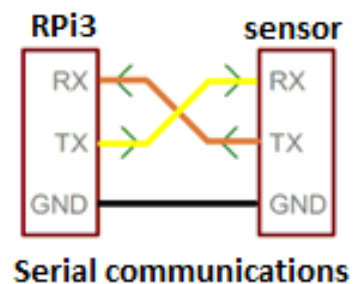
The Raspberry Pi (RPi) is a credit-card size computer. The RPi3 used here includes four USB ports plus Wi-Fi and Bluetooth connectivity. The HDMI video output requires a HDMI cable and appropriate HDMI-capable monitor. A USB keyboard and USB mouse will also be necessary.

In this example, the RPi3 connects to the COZIR CO2 sensor via the on-board UART. Example code for both the K-30 sensor is included below. The RPi3 offers a UART TXD-RXD connection for serial communication. The Raspian Linux operating system used is installed using the NOOBS installer. For more details on NOOBS, follow this link: <https://www.raspberrypi.org/help/faqs/#generalNoobs>

In order to proceed you will need to set up your RPi3 with Raspian 8.

Set up your RPi3

1. Connect your Raspberry Pi to your keyboard, mouse, monitor and 5VDC power supply.
2. Power the RPi3 and LCD monitor. In order to access to the attached python files, you will need an internet connection. RPi3 offers a built-in Wi-Fi device. On your monitor, find the icon of a terminal. Select the icon, select your Wi-Fi network, and enter your network password.
3. Verify that you are connected via Wi-Fi. On your monitor, select the globe icon to open an internet browser. Verify your internet connection by going to Google.com.
4. Search for AN-127 in the App Notes tab at <http://www.CO2meter.com>. Download the zip file to your Raspian downloads directory. Unzip this directory to access the python files.

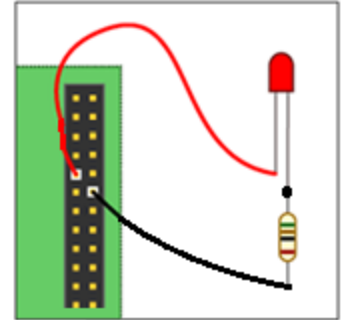


Run the Blink Example

We recommend you run the Blink example to verify your hardware and software is operating properly before you connect the CO2 sensor. Included in the downloaded folder is the blink.py python code.

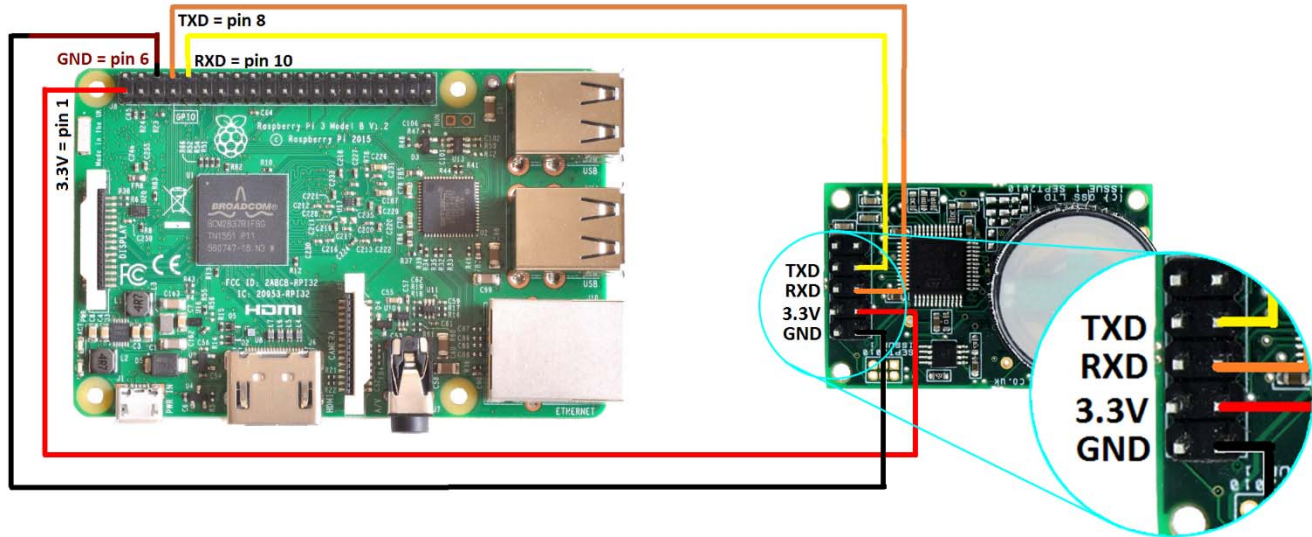
You can open it with the Geany, the Raspbian code editor.

1. Connect a LED and resistor to the RPi pin header to GPIO pins 9 and 12. The longer lead on the LED is positive, and connects through a 270 ohm, ¼ watt resistor to pin 12. The other LED lead connects to pin 9 which is GND (ground).
2. Select Menu > Programming > Geany.
3. Select File > Open > pi > AN127 > Blink. You should see the screen below.
4. Select Build > Compile.
5. Select Build > Execute. The LED will blink 12 times if everything is set up correctly. Otherwise, go back and trace your steps before continuing.



Connecting the COZIR Sensor

Connect the RPi3 to the COZIR CO2 sensor as shown. Power the sensor with the 3.3VDC (recommended) pin on the RPi 3 board. If you decide to use an external power, supply, it must provide between 3.3 and 5 VDC. If power drops below 3.3VDC, the sensor will not work as described.

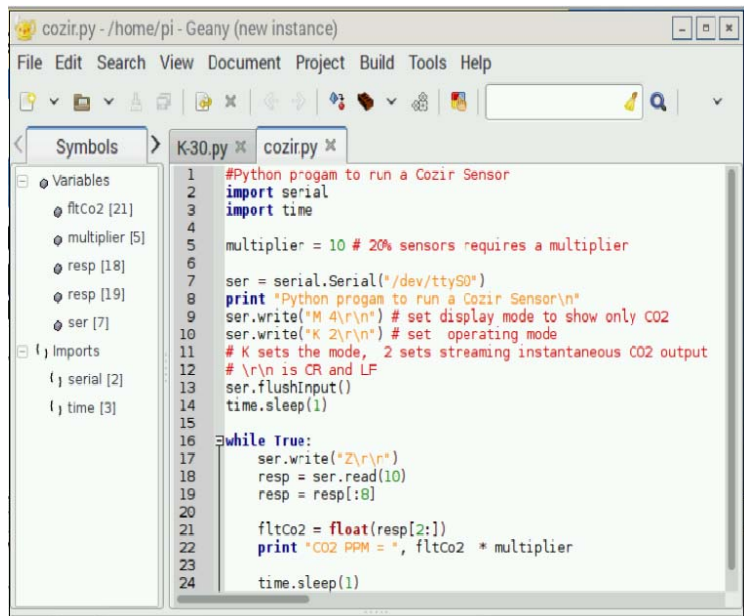


Creating your RPi3-COZIR Project

1. Select Menu > Programming > Geany
2. Select File > Open > pi > AN127 > cozir
3. Observe the following screen:

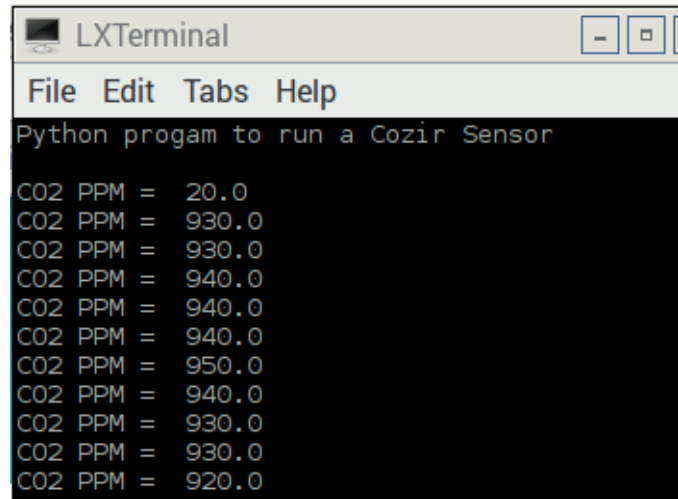
Note: On line 6, the UART connection is ttyS0. Your tty connection may differ.

4. Select Build > compile.
5. Select Build > execute.
6. The output is shown below:



```

1  #Python program to run a Cozir Sensor
2  import serial
3  import time
4
5  multiplier = 10 # 20% sensors requires a multiplier
6
7  ser = serial.Serial("/dev/ttyS0")
8  print "Python program to run a Cozir Sensor\n"
9  ser.write("M 4\r\n") # set display mode to show only CO2
10 ser.write("K 2\r\n") # set operating mode
11 # K sets the mode, 2 sets streaming instantaneous CO2 output
12 # \r\n is CR and LF
13 ser.flushinput()
14 time.sleep(1)
15
16 while True:
17     ser.write("Z\r\n")
18     resp = ser.read(10)
19     resp = resp[:8]
20
21     fltCo2 = float(resp[2:1])
22     print "CO2 PPM = ", fltCo2 * multiplier
23
24     time.sleep(1)
    
```



The screenshot shows an LXTerminal window with a menu bar (File, Edit, Tabs, Help) and a title bar (LXTerminal). The terminal output displays a series of CO2 PPM readings: 20.0, 930.0, 930.0, 940.0, 940.0, 940.0, 950.0, 940.0, 930.0, 930.0, and 920.0. The text 'Python program to run a CozIR Sensor' is visible at the top of the terminal output.

COZIR Python Code

```
#Python program to run a COZIR CO2 Sensor
import serial
import time

multiplier = 10 # 20% sensors requires a multiplier
ser = serial.Serial("/dev/ttyS0")
print "Python program to run a CozIR Sensor\n"
ser.write("M 4\r\n") # set display mode to show only CO2
ser.write("K 2\r\n") # set operating mode
# K sets the mode, 2 sets streaming instantaneous CO2 output
# \r\n is CR and LF
ser.flushInput()
time.sleep(1)

while True:
    ser.write("Z\r\n")
    resp = ser.read(10)
    resp = resp[:8]
    fltCo2 = float(resp[2:])
    print "CO2 PPM = ", fltCo2 * multiplier

    time.sleep(1)
```

COZIR Serial Commands

The following python instructions set the display mode and operating mode.

```
ser.write("M 4\r\n") # set display mode to show only CO2
ser.write("K 2\r\n") # set operating mode
```

Refer to the following instruction manual:

<http://www.co2meters.com/Documentation/Manuals/Manual-GSS-Sensors.pdf>

Page 11 of the manual describes output fields. These should be sent via UART to the sensor first, before the Operating Mode commands beginning on page 13 are used.

Output Fields are set and sent via the M command. The letter M is followed by a space, followed by a base-10 number, and followed by crlf. This command sends serial ASCII characters via the UART. The above python statement `ser.write("M 4\r\n")` sets the output field to display only CO2.

The base-10 number is the numerical sum of the “Mask Value” shown in the table.

Examples:

- To display only Humidity send M4096crlf. Or as defined in the Manual M4096\r\n/ .
- To display only Temperature send M64crlf. Or as defined in the Manual M64\r\n/ .
- To display only CO2, digitally filtered, send M4crlf. Or as defined in the Manual M4\r\n/ .
- To display Humidity and Temperature, add 4096+64=4160: M4160crlf or M4160\r\n .
- To display Humidity, Temperature and CO2, add 4096+64+4 =4164: M4164crlf or M4164\r\n .

The output as described in the Manual will be returned from the COZIR as

H 12345 T 12345 Z 00010\r\n

Operating Mode is set and sent via the **M command**. Set the Operating mode via python instruction after setting the display mode. For additional details, refer to the Manual page 13. The above program sets Polling Mode via this instruction: `ser.write("K 2\r\n")`.

Examples:

- To set Command Mode, send K 0crlf or K 0\r\n. There is a mandatory space following the K. This is used when extracting larger chunks of data from the COZIR.
 - Example: send `ser.write("K 0\r\n")`.
- To set Streaming Mode, send K 1crlf or K 1\r\n. There is a mandatory space following the K. This is the factory default. Measurements are reported up to twice per second.
 - Example: send `ser.write("K 1\r\n")`.
- To set Polling Mode, send K 2crlf or K 2\r\n. There is a mandatory space following the K. This sensor reports readings only when requested.
 - Example: send `ser.write("K 2\r\n")`.