Application Note AN137:
Raspberry Pi UART Interface to K-30 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 SenseAir K-30 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.

Serial communications

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-237 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.
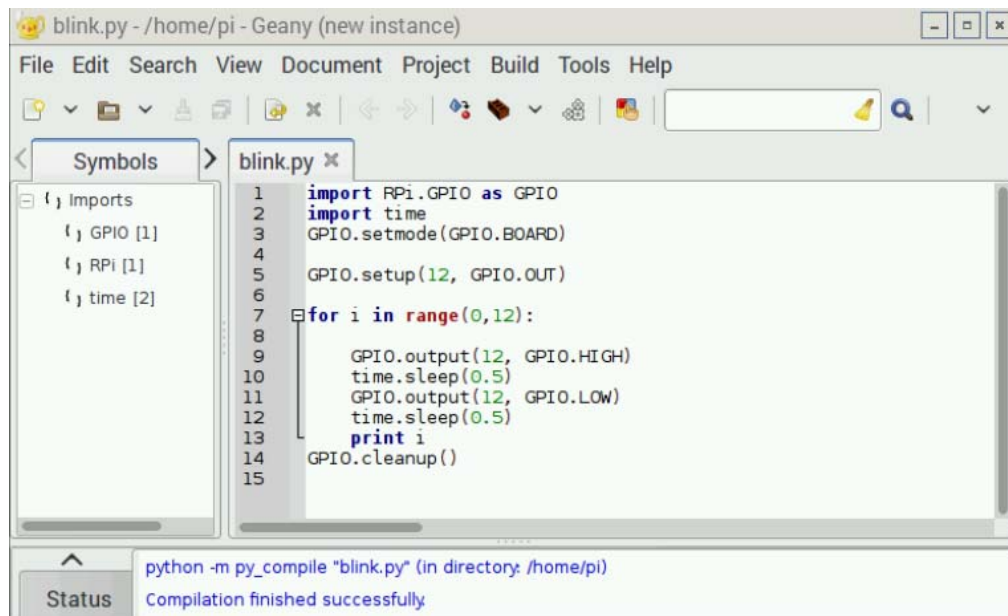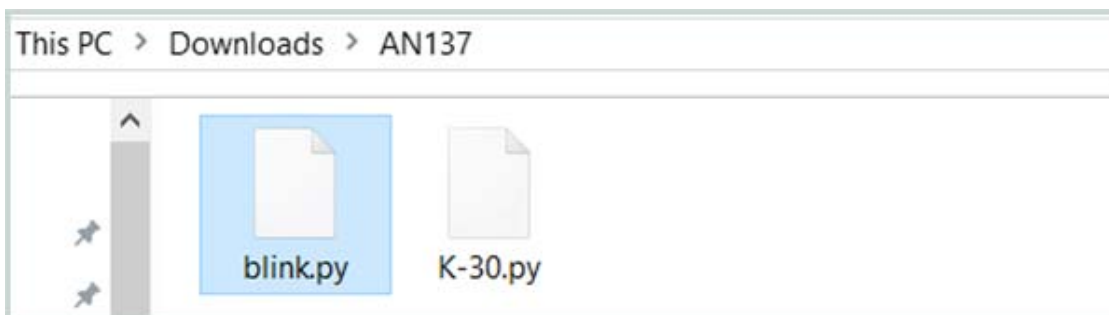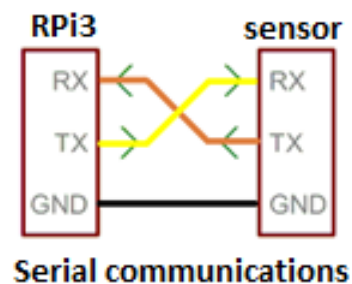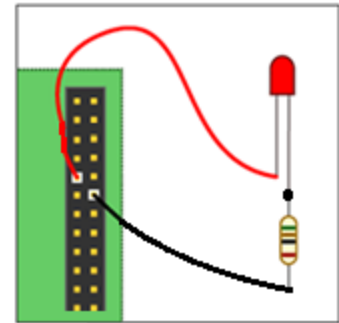
Run the Blink Example

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 Raspian 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 > AN137 > 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.

blink.py - /home/pi - Geany (new instance)

File   Edit   Search   View   Document   Project   Build   Tools   Help

Symbols   blink.py

Imports
  GPIO [1]
  RPi [1]
 time [2]

```
 1    import RPi.GPIO as GPIO
 2    import time
 3    GPIO.setmode(GPIO.BOARD)
 4
 5    GPIO.setup(12, GPIO.OUT)
 6
 7    for i in range(0,12):
 8
 9        GPIO.output(12, GPIO.HIGH)
10        time.sleep(0.5)
11        GPIO.output(12, GPIO.LOW)
12        time.sleep(0.5)
13        print i
14    GPIO.cleanup()
15
```

Status
python -m py_compile "blink.py" (in directory: /home/pi)
Compilation finished successfully.

Connecting the K-30 Sensor

Connect the RPi3 to the K-30 sensor as shown. Power the sensor with an external 6-9VDC (recommended) power supply capable of 300mA output. When the K-30 sensor takes a measurement, the internal circuitry draws a comparatively larger peak current for approximately 100 milliseconds. Note that if power drops below 5.5VDC, the sensor will not work as described.
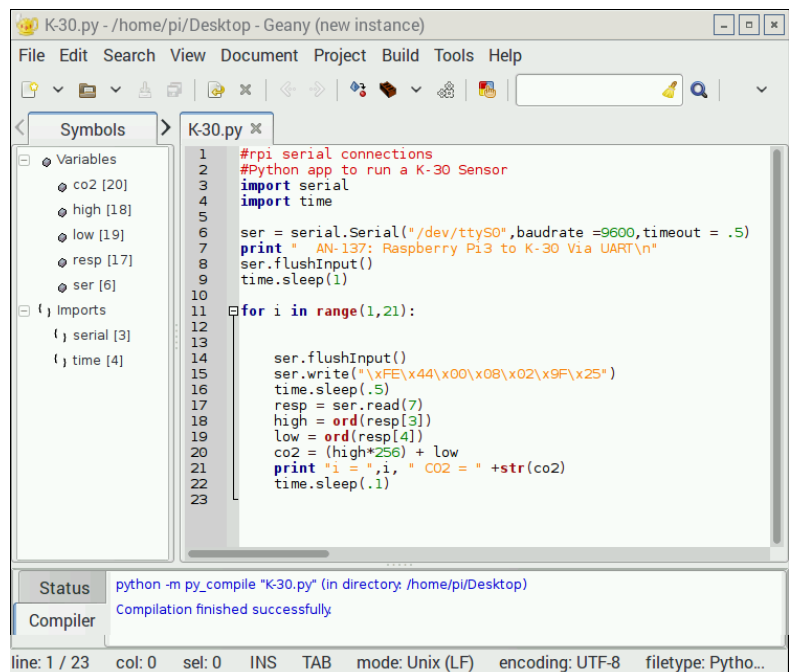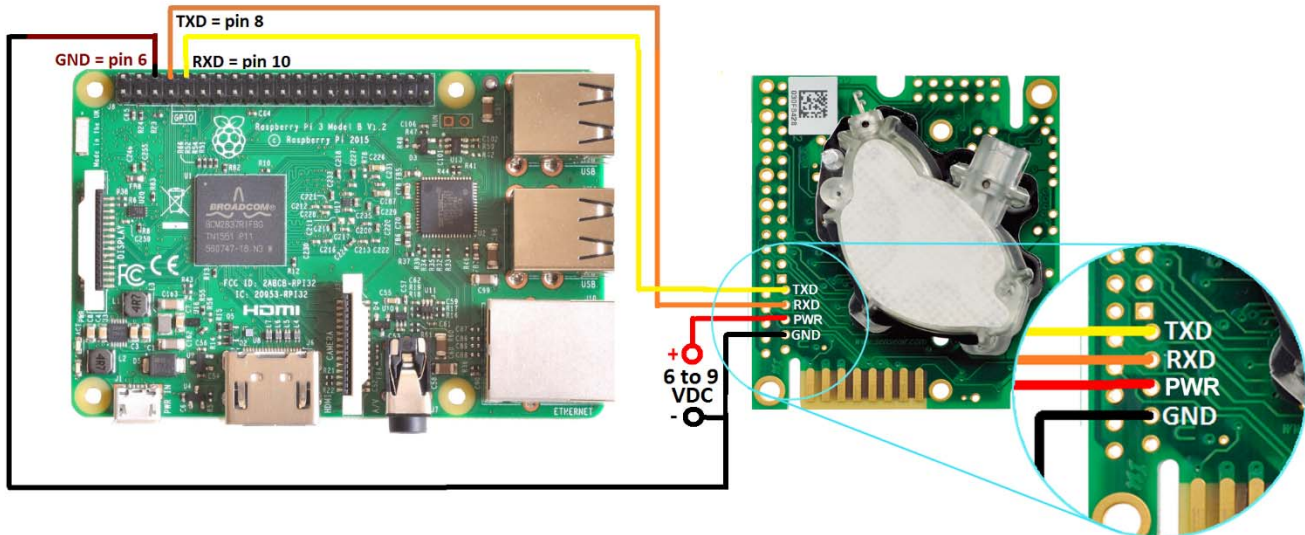

Creating your RPi3-K30 Project

1. Select Menu > Programming > Geany
2. Select File > Open > pi > AN137 > K30
3. Observe the following screen:

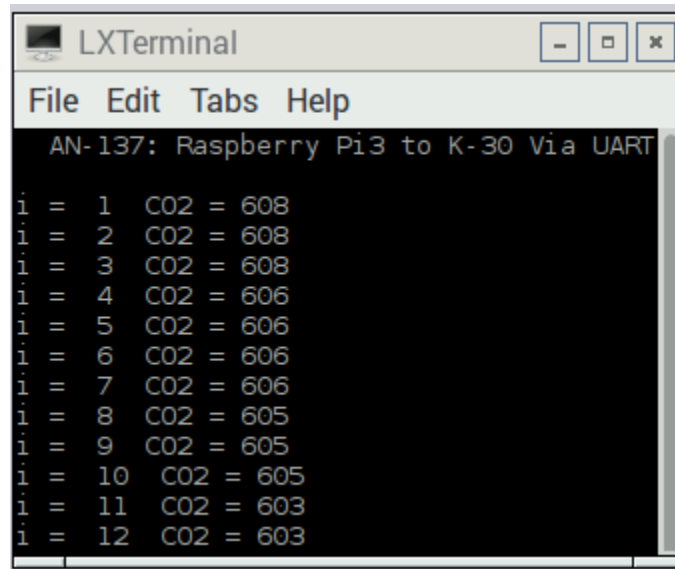**Note: On line 6, the UART connection is ttyS0. Your tty connection may differ.**

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

```
LXTerminal                              [ _ ][ □ ][ × ]
File  Edit  Tabs  Help
     AN-137: Raspberry Pi3 to K-30 Via UART

i =   1   CO2 = 608
i =   2   CO2 = 608
i =   3   CO2 = 608
i =   4   CO2 = 606
i =   5   CO2 = 606
i =   6   CO2 = 606
i =   7   CO2 = 606
i =   8   CO2 = 605
i =   9   CO2 = 605
i =   10  CO2 = 605
i =   11  CO2 = 603
i =   12  CO2 = 603
```

K-30 Python Code

```python
#rpi serial connections
#Python app to run a K-30 Sensor
import serial
import time

ser = serial.Serial("/dev/ttyS0",baudrate =9600,timeout = .5)
print "  AN-137: Raspberry Pi3 to K-30 Via UART\n"
ser.flushInput()
time.sleep(1)

for i in range(1,21):

    ser.flushInput()
    ser.write("\xFE\x44\x00\x08\x02\x9F\x25")
    time.sleep(.5)
    resp = ser.read(7)
    high = ord(resp[3])
    low = ord(resp[4])
    co2 = (high*256) + low
    print "i = ",i, " CO2 = " +str(co2)
    time.sleep(.1)
```