Interfacing the Cypress CY7C65213 to the Raspberry Pi

Background:

The CY7C65213 from Cypress Semiconductor is a full USB to UART Bridge. It makes all eight serial port lines available (TXD, RXD, RTS, CTS, DSR, DTR, DCD and RI) plus eight general purpose input/output (GPIO) pins that can be used to control other functions. I've been using it to interface the Iridium 9603 satellite communication short burst data modem to a Windows PC:

https://github.com/PaulZC/Iridium_9603_Lite_USB_Prototype

https://github.com/PaulZC/Iridium_9603_Lite_USB

But what I would really like to be able to do is use the CY7C65213 to interface the 9603 to Raspberry Pi and use python code to create a satellite-communication-enabled system for high altitude balloons and other remote monitoring applications.

I'm writing the following in real time as I connect the CY7C65213 to Raspberry Pi for the first time, using my fork of Taisuke Yamada's library python-ucdev

I am using the Cypress CYUSB232 development board:

- http://www.cypress.com/documentation/development-kitsboards/cyusbs232-usb-uart-lp-reference-design-kit
- Available from Mouser (Part# 727-CYUSBS232)

But you might want to use the new SparkFun USB UART Serial Breakout which comes with a comprehensive hookup guide:

- https://learn.sparkfun.com/tutorials/sparkfun-usb-uart-breakout-cy7c65213-hookup-guide

Step 1: Configure the CY7C65213

If you want to change the function of the CY7C65213 GPIO pins or configure the USB current draw, you'll first need to do that using a Windows PC as sadly Cypress don't seem to offer a version of their USB Serial Configuration Utility for Linux. Follow the instructions in Sparkfun's hookup guide, or my guide included at the end of this document.

The test code we'll be using later (pyserial_and_gpio_test.py) will assume that you have:

- Set RX Notification to None
- Set TX Notification to None
- Set RX or TX Notification to GPIO_0
- Made GPIO_1 an output (Drive 0)
- Made GPIO_2 an input

Step 2: Update your Pi

Make sure your Pi is running the latest version of Raspbian and that it is up to date:

- https://www.raspberrypi.org/downloads/raspbian/

Open a Terminal window and run:

- sudo apt-get update
- sudo apt-get upgrade

Step 3: Download Cypress' USB-Serial Software Development Kit

Download Cypress' USB-Serial Software Development Kit:

Open a web browser and navigate to:

- http://www.cypress.com/documentation/software-and-drivers/usb-serial-software-development-kit

Scroll down to the Linux section and download the zip file:

- http://www.cypress.com/?docID=45729

(You will need to register with Cypress first. Registration is free - in return for your email address.)

Open a File Manager window and navigate to the Downloads folder. Right-click on the CyUSBSerial_SDK_Linux.zip file and select "Extract Here".

Now navigate into the CyUSBSerial_SDK_Linux/CyUSBSerial_SDK_Linux/libusb subdirectory. Right-click on libusb-1.0.9.tar.bz2 and select "Extract Here".

Open a Terminal window and build libusb-1.0.9 by:

- cd Downloads/CyUSBSerial_SDK_Linux/CyUSBSerial_SDK_Linux/libusb/libusb-1.0.9
- sudo ./configure
- sudo make
- sudo make install

Change to the linux/library directory:

- cd ../..
- cd linux/library

Compile the Cypress USB library:

- sudo make

Compile the Cypress CyUSBSerialTestUtility. We won't actually use this code but the makefile does useful things like copy 90-cyusb.rules into /etc/udev/rules.d

- cd ..
- cd testUtility
- sudo make

Step 4: Install CFFI

If you haven't installed it before, you will also need CFFI.

- https://pypi.python.org/pypi/cffi?
- http://cffi.readthedocs.io/en/latest/

You can install it by running the following command in a Terminal window:

- sudo apt-get install python-cffi

Step 5: Download python-ucdev

Download my fork of Taisuke Yamada's library python-ucdev. This is easiest in a Terminal window using git:

- cd ~
- git clone https://github.com/PaulZC/python-ucdev.git
- cd python-ucdev
- sudo python setup.py install

Step 6: Connect the CY7C65213

First have a look at the serial ports already present on the Raspberry Pi. In a Terminal window, type

- ls /dev/tty*

In that list, you should see /dev/ttyAMA0 which is the Pi's serial port.

Connect your chosen CY7C65213 board to a spare USB port.

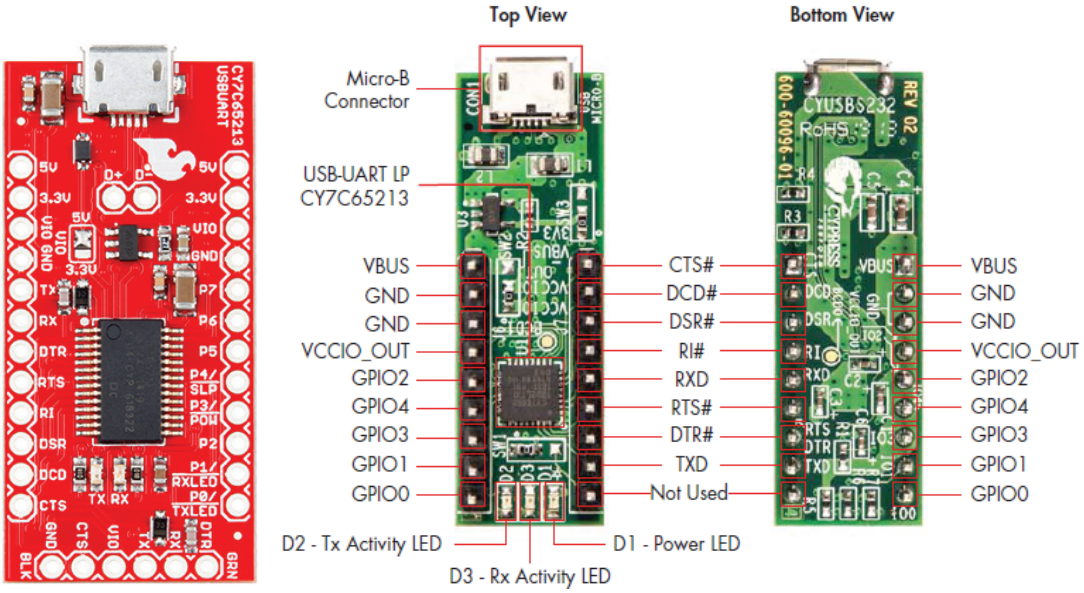
Type ls /dev/tty* again and you should now see an extra port called /dev/ttyACM0 which is the CY7C65213's serial port. We will be using that name in our python code.

Type lsusb to see a list of all the USB devices connected to the Pi. One of them should show: ID 04b4:0003 Cypress Semiconductor Corp.

Step 7: Install loop-back connections on the CY7C65213

Using jumper wires, link the following pins together so we can perform loop-back tests with the python code:

- TXD (TX) to RXD (RX)
- RTS to CTS
- DTR to DSR
- GPIO_1 (P1) to GPIO_2 (P2)



Step 8: Modify the test code for Linux

The test code gpio_test.py and pyserial_and_gpio_test.py has been written for use under Windows. To let the code run under Linux you need to make two small changes:

- Open IDLE by selecting "Python 2 (IDLE)" from the Programming menu
- Click on "File" then "Open", select the directory "python-ucdev" and then open "pyserial_and_gpio_test.py"
- Comment out the following two lines:
    - #com_port = 'COM1'

    - #dll = "C:\\Program Files (x86)\\Cypress\\USB-Serial SDK\\library\\cyusbserial\\x64\\cyusbserial.dll"
- Uncomment the following two lines:
    - com_port = '/dev/ttyAMC0"
    - dll = "/usr/local/lib/libcyusbserial.so"
- Save the code and then run it by selecting the Run menu and then Run_Module; or press F5
- Repeat for gpio_test.py

To run the modified code in a Terminal window, type:

- cd ~
- cd python-ucdev
- python pyserial_and_gpio_test.py

Thanks!

This project wouldn't have been possible without the open source designs and code kindly provided by:

- **Sparkfun**:
    - The USB UART Serial Breakout CY7C65213 (Product BOB-13830)
    - https://www.sparkfun.com/products/13830
- Taisuke Yamada (Tai)
    - Python library for Cypress CY7C65211/CY7C65215 USB-Serial bridge
    - https://github.com/tai/python-ucdev

The small print

How do I configure the CY7C65213 (under Windows)?

Start by reading Sparkfun's hookup guide:

- https://learn.sparkfun.com/tutorials/sparkfun-usb-uart-breakout-cy7c65213-hookup-guide

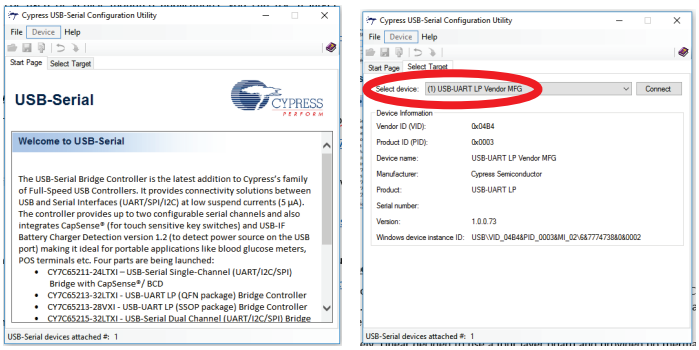
You will also want to download and install Cypress' USB-Serial Software Development Kit:

http://www.cypress.com/documentation/software-and-drivers/usb-serial-software-development-kit

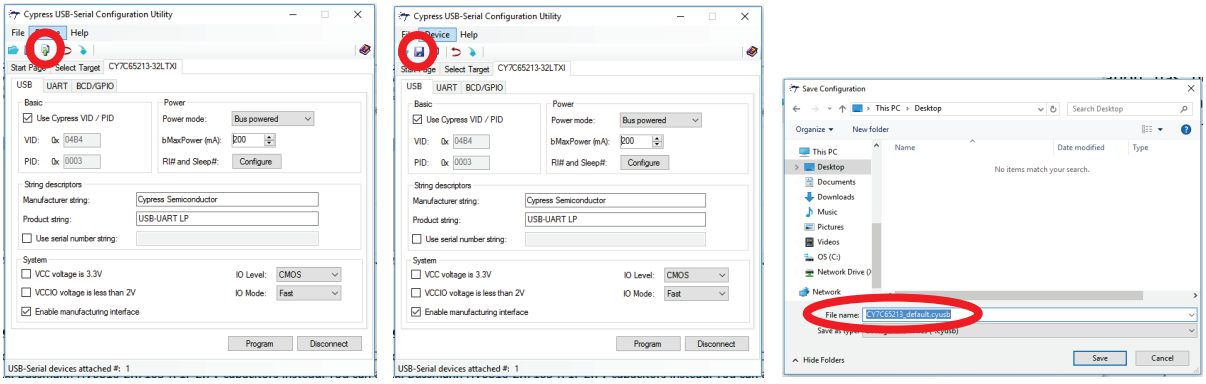
You might find Cypress' CY7C65213 Reference Design documentation useful too:

- http://www.cypress.com/documentation/development-kitsboards/cyusbs232-usb-uart-lp-reference-design-kit

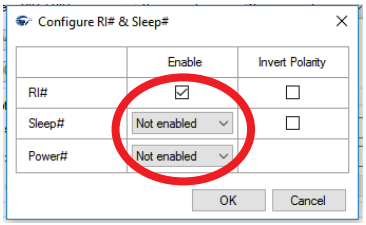
Plug in your board, run the USB Serial Configuration Utility and check that it detects the CY7C65213:



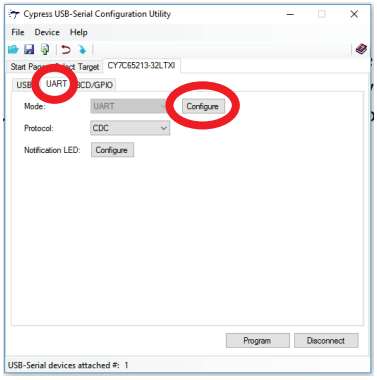
Select the "USB-UART LP Vendor MFG" and connect to it. Click the "Open Configuration from Device" icon to ensure the default device configuration has been read successfully. Click the Save icon and save the default configuration using a filename like "CY7C65213_Default.cyusb". That way, you can restore the default configuration if you need to.



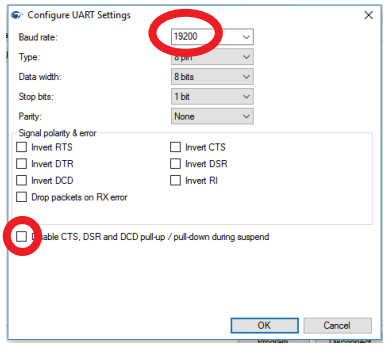
Click the Configure button next to “RI# and Sleep#”. Set “Sleep#” and “Power#” to “Not enabled”, then click OK so you can use all the GPIO pins for GPIO.



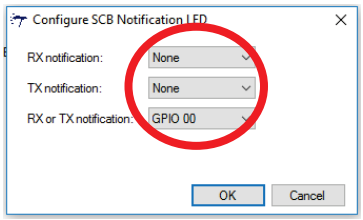
Select the UART tab and then click the Configure button next to “Mode”:



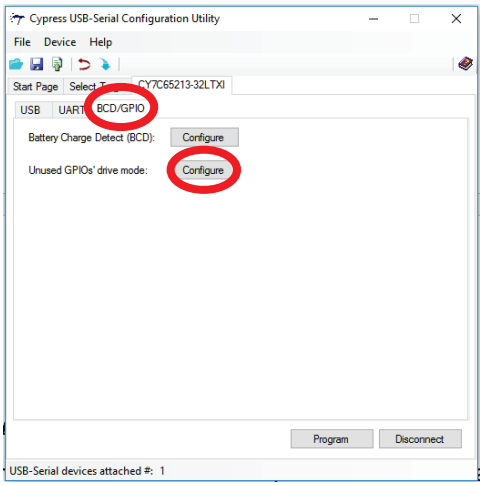
Set the Baud rate to 19200, untick the “Disable CTS…” box and then click OK.



Click the Configure button next to “Notification LED”. Set “RX notification” and “TX notification” to “None”. Set “RX or TX notification” to “GPIO 00”. Then click OK.

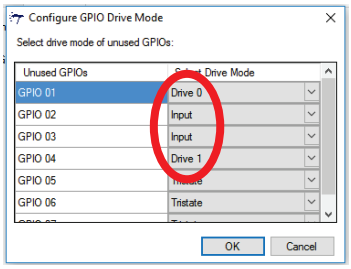


Select the BCD/GPIO tab and then click the Configure button next to “Unused GPIOs’ drive mode”:

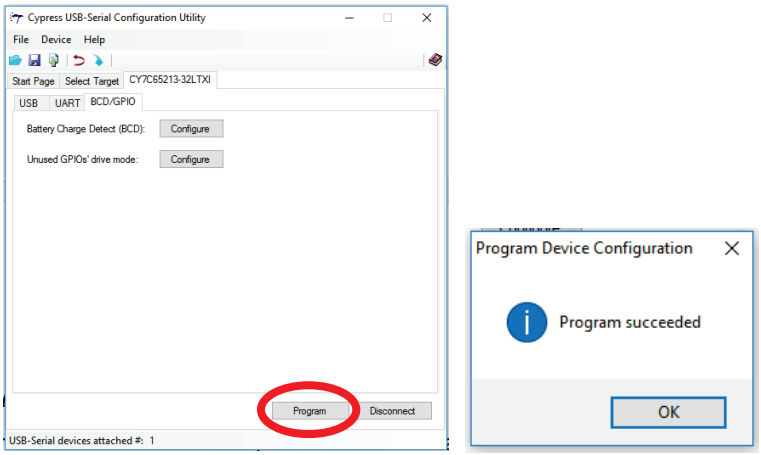


Select the following drive modes then click OK:

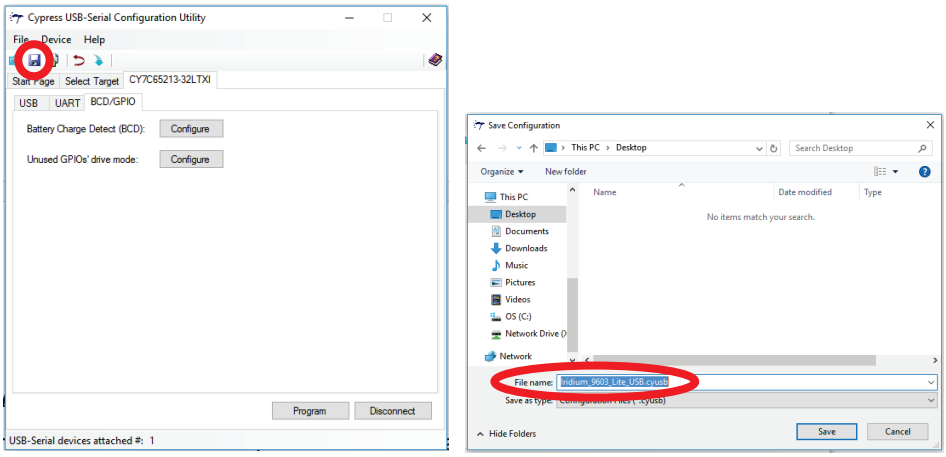
- GPIO 01: “Drive 0”
- GPIO 02: “Input”



Click “Program” to program these settings into the CY7C65213. Check that the programming is successful.



Save the configuration into a new .cyusb file called something like GPIO_Test.cyusb.



Finally, click “Disconnect” and unplug the CY7C65213. It will use the new configuration the next time it is powered up.