

Pythonを用いたUSB-UARTデバイス制御

USBデバイスの活用

USBデバイスを利用する場合、ドライバと呼ばれるハードウェアとアプリケーションの間に入ってデータのやり取りを手助けする機能を持ったモジュールのインストールが必要になる。

一方で、USBデバイスのうち非同期送受信プロトコルの一つであるUARTを利用するものは、デバイスのドライバがOSに含まれていたり、ドライバの入手性が良い場合が多いため活用しやすい。

UART : Universal Asynchronous Receiver/Transmitter
RS232なども非同期送受信プロトコルのうちのひとつ

[UART - Wikipedia](#)

今回はRaspberry Pi OSにドライバが含まれているFTDI製「FT232」が搭載されているRFIDリーダーを用いて、シリアル通信デバイス制御のイメージを掴む。

FT232はPCにUSB接続したデバイスがシリアルポートとして認識させるUSB-UART機能を持ったICである。接続した機器はシリアルポートから利用することができる。

※Silicon Labs製「CP2102」もUSB-UARTデバイスとして頻出、Raspberry Pi OSでも比較的に簡単に利用できる上に情報も多い。

【参考】 シリアル通信とパラレル通信

シリアル通信

送信データのビット列を逐次的に送る方法. 時分割でパルス状のデータを送信するのが一般的である. 高速化しやすく現在多く利用されている.

- シリアル通信プロトコル例
 - モールス符号 電報
 - RS-232
 - USB
 - イーサネット
 - シリアルATA
 - PCI Express
 - SPI
 - I2C

パラレル通信

複数の信号線を用いて、送信データのビット列を一度に送る方法で、シリアル通信と比較した際、データリンクの速度が同じであれば複数のビットを展開して送ったほうが転送効率が良い。

一方で複数の信号線でより多くのデータをより早く送受信する技術革新の過程で、複数の信号線でパルスの立ち上がり、立ち下がりを厳密に同期しながら送信するのが難しくなってきた。
また、物理的に複数の信号線を利用することから相互の配線（バス）間でのクロストーク（他の信号線に別の信号線のデータが影響すること）が無視できなくなり、最近見かけなくなった。

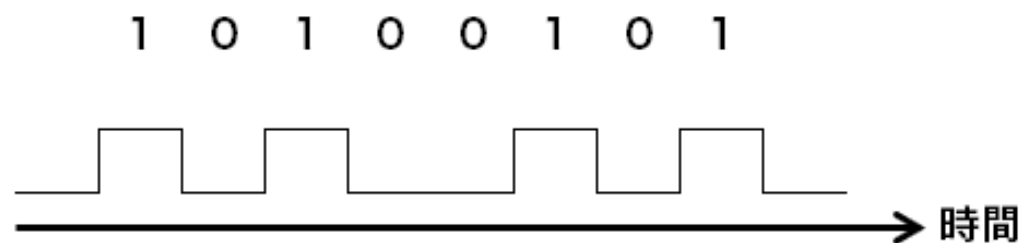
｜ GPIOなどを利用して実装することをイメージしてみると良い

”

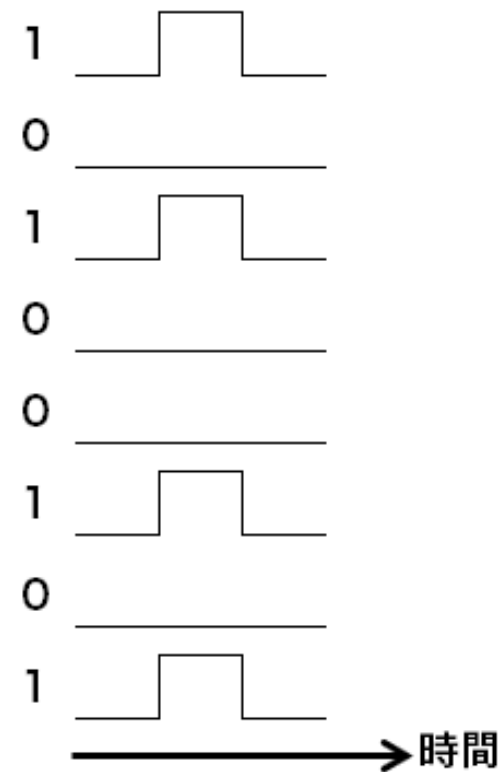
- パラレル通信プロトコル例
 - ISA
 - ATA
 - SCSI
 - PCI
 - IEEE 1284（パラレルポート）

シリアル通信とパラレル通信の伝送路上での波形

シリアル通信



パラレル通信



Raspberry piでのUSB-UARTデバイスの扱い方

本体にデバイスを接続した際、認識したかどうかは `lsusb` コマンドで確認することができる。

USB-UARTデバイスは本体に接続後、デバイスの認識が終わると `/dev/ttyUSBX`（`X` は0から始まる連番の数字）にデバイスファイルが現れる。

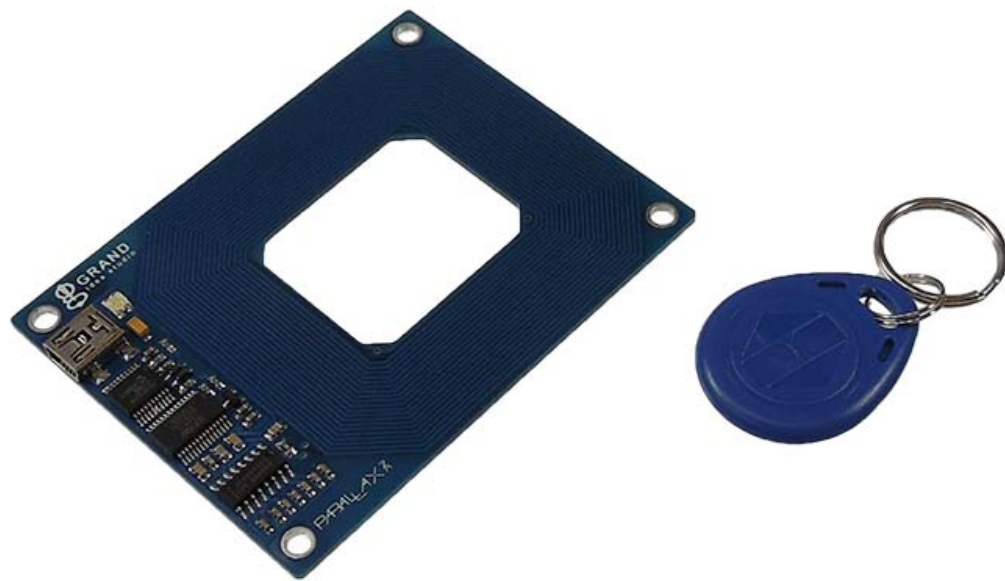
プログラムからデバイスを利用する場合はデバイスファイル名を指定して利用する。

USB-UARTを利用できるRFIDリーダーを利用する

下記のデバイスを利用して実習する.

秋月電子通商 | U S B 接続 R F I D リード (読み込み) モジュール (タグサンプル付き)

<https://akizukidenshi.com/catalog/g/gM-06826/>



- RFIDのリード(読み込み)に対応したモジュール基板です。付属のRFIDタグ(円盤型、カード型、キーホルダー型)の他に、125kHz EM4100のタグに対応します。通信はUSBで行いますので、PCでの使用に便利です。（Parallax Inc.製）
- 仕様
 - 電源：+5V(アイドル時：10mA以下、動作時：100～200mA)
 - 通信方式：USB(仮想COMポート接続：2,400bps,N,8,1 設定)
 - 基板寸法：6.2x8.26cm
- セット内容
 - RFIDのリードモジュール基板本体 x1個
 - 円盤型RFIDタグ(直径約50mm) x1枚
 - 円盤型RFIDタグ(直径約25mm) x1枚
 - カード型RFIDタグ x1枚
 - キーホルダー型RFIDタグ x1個
 - USBミニBケーブル 1本

利用手順

- 本体にRFIDリーダーを接続する
- USBデバイスが認識されていることを確認するとともに、デバイスファイル名を確認する
 - `lsusb` コマンドを実行してUSBデバイスが認識されていることを確認する
 - `/dev/ttyUSB0` が存在していることを確認する
- pyserialモジュールをインストールする
 - `$ python3 -m pip install pyserial`
- シリアル通信プログラムをpythonで実装する
- プログラムを実行後にタグを読んでタグ固有のIDを取得する
 - アプリケーションで利用する際にはタグ固有のIDをキーにしたテーブルをDBなどで備えておいて必要な情報を参照する.

USBデバイスの認識状況の確認

接続前：

```
$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

接続後：

```
$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 003: ID 0403:6001 Future Technology Devices International, Ltd FT232 Serial (UART) IC
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub

$ ls /dev/ttyUSB*
/dev/ttyUSB0
```

シリアル通信を行うコード

```
import serial

PORTNAME = "/dev/ttyUSB0"
BAUDRATE = 2400

if __name__ == '__main__':
    port = serial.Serial(PORTNAME, BAUDRATE, timeout=None)
    try:
        while(True):
            line = port.readline()
            code = str(line.decode().replace('\r\n',''))
            print(code)
    except KeyboardInterrupt:
        print('interrupted!') # Ctrl-Cが入力された場合の処理
        pass
    port.close()
    print("Done")
```

シリアル通信を行うコード

```
port = serial.Serial(PORTNAME, BAUDRATE, timeout=None)
```

シリアル通信を行う際には下記の設定が必要になる.

- ポート
 - デバイス名を指定する. Linuxでは `/dev/tty~`, Windowsでは `COMX` で割り当てられることが多い
- 通信速度 (baudrate)
 - 2400,4800,9600...115200bpsあたりがよく利用される (bps: bits per second)
- データ長さ
- ストップビット
- パリティ

※生活実感では「データ長さ8bit, ストップビット1bit, パリティなし」の設定が多い

シリアル通信を行うコード

今回のRFIDリーダーの仕様書にも記載がある.

利用したいデバイスを見つけた場合は（あまり気が進まないが）目を通してみる.

Connections (USB)

The Parallax RFID Card Reader USB version can be connected directly to any PC, Macintosh, or Linux machine that has a USB port and the appropriate drivers installed. The module is powered from the host computer's USB port and uses an industry-standard FTDI FT232R device to provide the USB connectivity. FTDI drivers are available from www.ftdichip.com/Drivers/VCP.htm.

Signal	Port Name	Function
RX	Serial Receive	Serial output to host. <u>2400 bps, 8 data bits, no parity, 1 stop bit.</u>
DTR	Data Terminal Ready	Module enable. Bring the DTR line HIGH to enable the RFID reader and activate the antenna. Bring the DTR line LOW to disable the RFID reader.