

Raspberry Pi (Raspbian)
を使用したリーダー・ライタの制御例
(USB 接続と UART 接続)

2022 年 12 月 8 日 第 1.2.3 版

株式会社アートファイネックス

はじめに

本書は、Raspberry Pi 3 Model B (OS: Raspbian) を使用し、アートファインックス社製 RFID リーダ・ライター (組込用モジュールと小型 USB 給電機器) を制御する例を記載したものです。

Raspberry Pi 3 Model B と RFID リーダ・ライターとの間は、USB の場合と UART の場合の2通りで行いました。

無断転載を禁じます。

本書の内容は、断りなく変更することがあります。

※ Microsoft Windows は、米国 Microsoft Corp. の登録商標です。

※ Raspberry Pi は、Raspberry Pi 財団の登録商標です。

※ FTDI は、Future Technology Devices International Limited の商標または登録商標です。

※ その他、商品名及び製品名などは一般に各社の商標または登録商標です。

1. 用意するもの

- ・開発用 PC
 - OS: Windows
- ・Raspberry Pi 3 Model B (以下 ラズパイ)
 - 本体
 - 電源 (micro USB から供給)
 - micro SD メモリーカードと変換アダプター (開発用 PC のため)
 - USB キーボード
 - USB マウス
 - モニタ (HDMI) と HDMI ケーブル
 - ネットワークケーブル
- ・RFID リーダ・ライター (f ファミリ) とアンテナ
- ・RF タグ
- ・ラズパイ- RFID リーダ・ライター接続ケーブル

2. 概要

手順の概要は以下のとおりです。

1. 開発用 PC に Raspbian のインストーラをダウンロードして解凍する。
2. microSD カードをフォーマットした後、1で解凍したファイル一式を書込む。
3. ラズパイに microSD カードを挿して起動し、インストールを行う。
4. 必要に応じてラズパイの設定を変更する。
(例: IP アドレス、パスワード、タイムゾーン、など)
5. ラズパイの UART で行うときは下の設定も必要です。

`/boot/config.txt` の末尾に下の2行を追記

```
dtoverlay=pi3-miniuart-bt
enable_uart=1
```

`/boot/cmdline.txt` に記載されている下の箇所を削除

```
console=serial0,115200
```

6. ラズパイと RFID リーダ・ライターを接続する。
7. ラズパイで使える python3 で RF タグの ID を読むアプリを開発する。

次章からは上の6と7について記載します。

※FTDI 社製の USB ドライバのインストール作業は不要でした。

3. 各種バージョン

1. OS(ラズパイ)

```
pi@raspberrypi:~ $ lsb_release -a
# 以下出力
# No LSB modules are available.
# Distributor ID: Raspbian
# Description: Raspbian GNU/Linux 9.4 (stretch)
# Release: 9.4
# Codename: stretch
```

2. Kernel

```
pi@raspberrypi:~ $ uname -a
# 以下出力
# Linux raspberrypi 4.14.34-v7+ #1110 SMP Mon Apr 16 15:18:51 BST 2018 amv71
GNU/Linux
```

3. python

```
pi@raspberrypi:~ $ python3 -V
# 以下出力
# Python 3.5.3
```

4. ラズパイと RFID リーダ・ライタとの接続例

4. 1. USB での接続例

UC-100f-J2

ラズパイ



キーボード



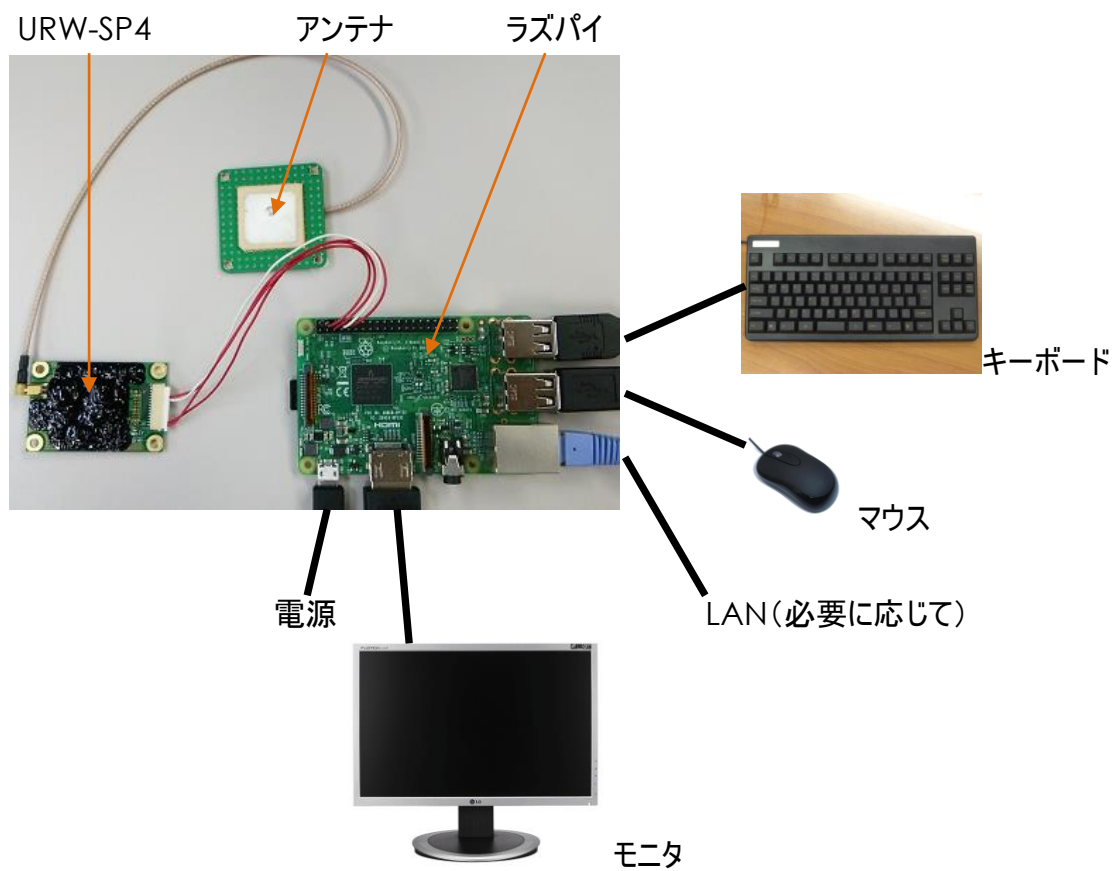
マウス

電源

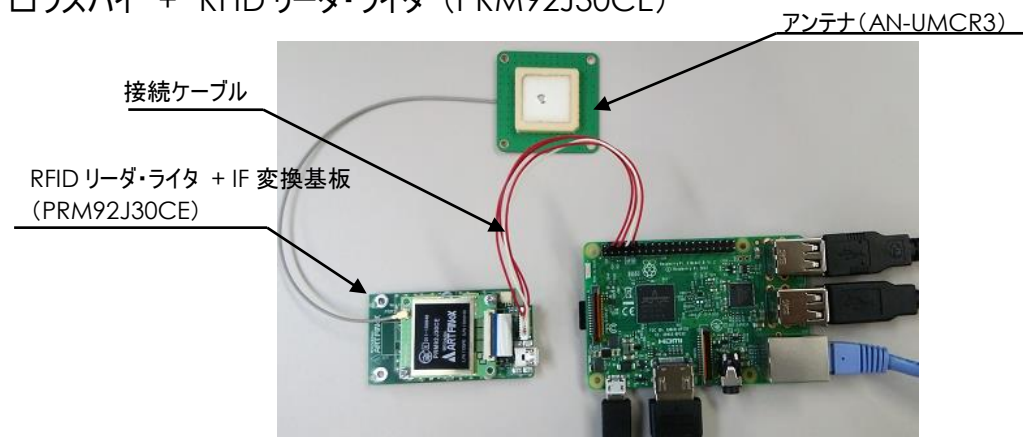


モニタ

4. 2. UART での接続例

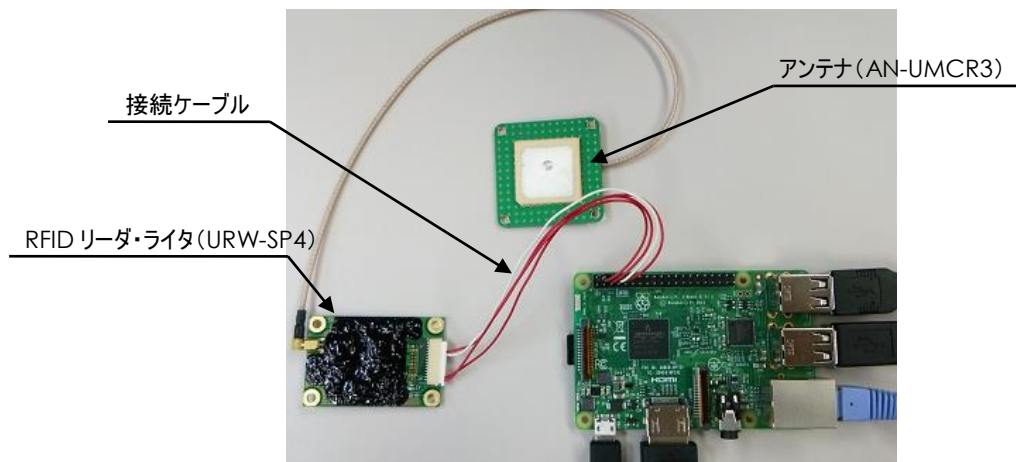


□ラズパイ + RFID リーダ・ライター (PRM92J30CE)



ラズパイ		RFID リーダ・ライター	
ピン番号	機能	ピン番号	機能
4	5V PWR	1	5V PWR
6	GND	2	GND
8	TXD	9	RXD
10	RXD	8	TXD
※	GPIO	5	RESET

□ラズパイ + RFID リーダ・ライター (URW-SP4)



ラズパイ		RFID リーダ・ライター	
ピン番号	機能	ピン番号	機能
4	5V PWR	1	5V PWR
6	GND	2	GND
8	TXD	11	RXD
10	RXD	12	TXD
※	GPIO	10	RESET

※リーダの RESET ピンは必要に応じてラズパイの GPIO ピンと接続してください

リセット信号は "L" (1ms 以上) でモジュールをリセットします

5. アプリケーションソフトウェア開発手順例

ラズパイ上で行います。

始めに、リーダー・ライターがつながっているポートを確認します。

下のアプリを作成します。(port.py)

```
import glob
ports = glob.glob('/dev/tty[A-Za-z]*')
for port in ports:
    print(port)
```

ラズパイにリーダー・ライターを接続して下のコマンドを実行すると

```
$python3 port.py
```

下のように表示されます。

```
/dev/ttyS0
/dev/ttyUSB0    USB で接続するとき
/dev/ttyprintk
/dev/ttyAMA0    UART で接続するとき
```

次に、RF タグの ID を取得するために下のアプリを作成します。(ReadEpc_usb.py)

```
import serial
ser = serial.Serial("/dev/ttyUSB0")
ser.baudrate = 115200
sendData = b'\xBB\x80\x22\x00\x02\x01\xA1\x7E'
ser.write(sendData)

while (True):

    recvData = ser.read(5)
    len1 = recvData[3]*256+recvData[4]+1
    temp1 = ser.read(len1)
    if (recvData[2] == 0xFF):
        print('No tag!!')
        break
    elif (recvData[2] == 0x27):
        break
    else:
        len2 = len(temp1)
        temp2 = temp1[2:len2-4]
        print(temp2.hex())

ser.close()
```

UART で接続するときは
ttyAMA0 にします。
ボーレートは使用する機器に
合わせてください。
\
は ¥ と表示されること
もあります。

ラズパイにリーダ・ライタを接続して下のコマンドを実行します。

```
$python3 ReadEpc_usb.py
```

RF タグがあれば、取得した RF タグの ID を表示します。(複数個取得したら複数個表示します。)

また、RF タグがかざされていない場合は「No tag!!」と表示されます。

「URW-SP プロトコル仕様書」内の「Read Type C Tag ID Single(メッセージ区分:0x22)」コマンドを使用したサンプルアプリです。この仕様書に記載されている「メッセージフォーマット」のご理解が必要です。

以上は f ファミリの RFID リーダ・ライタに関するものです。

最後に参考として、CB ファミリの RFID リーダ・ライタで RF タグの ID を取得するサンプルアプリを記します。

```
import serial
ser = serial.Serial("/dev/ttyUSB0")
ser.baudrate = 115200
sendData = b'\x53\x00\x00\x00\x20\x00\x00\x00\x20\x00\x00\x00\x00\x00\x00\x00\x93'
ser.write(sendData)
recvData = ser.read(16) # 最初の 16byte を読む
len1 = recvData[7]*256+recvData[6]+1 # 残りのデータ数を取得
tagData = ser.read(len1) # 残りのデータを読む
if (recvData[1] != 0x00): # RF タグが無いなどのエラー
    print('No tag!!')
else:
    numTag = tagData[0] # 読んだ RF タグ数を取得
    print('タグ数 : ' + str(numTag)) # して表示
    dataPointer = 1
    for num in range(0, numTag): # RF タグ数回繰り返す
        antNum = tagData[dataPointer] # アンテナ番号を取得
        dataPointer += 1
        idLen = tagData[dataPointer] # RF タグ情報の長さを取得
        dataPointer += 1
        # RF タグ情報は PC(2byte)+EPC(可変)+CRC(2byte)
        epc = tagData[dataPointer+2:dataPointer+idLen-2] # EPC を取得
        dataPointer += idLen
        print('Ant' + str(antNum) + ', ' + epc.hex())
ser.close()
```

「ART Finex リーダ・ライタプロトコル」 UHF 版仕様書 内の「タグ ID 取得(メッセージ区分:20h)」コマンドを使用したサンプルアプリです。この仕様書に記載されている「メッセージフォーマット」のご理解が必要です。

以上