

About SerialMonitor.rb

Lazurite Pi Gateway の ruby\SerialMonitor.rb のコードについて説明します。

```
#! /usr/bin/ruby
                                                            Ruby のコードであることの宣言
# -*- coding: utf-8; mode: ruby -*-
# Function:
    Lazurite Sub-GHz/Lazurite Pi Gateway Sample program
require 'net/http'
require 'date'
# BP3596 API
##
class BP3596PipeApi
  # func : Read the data from the receiving pipe
  # input : Receive pipe fp
  # return : Receive data
                                                                 Time out 待ち時間を設定
  def read_device(fp)
    # Data reception wait (timeout = 100ms)
    ret = select([fp], nil, nil, 0.1)
                                                                 ドライバの仕様
                                                                 2 バイトの read を行うと、受信
    # Reads the size of the received data
    len = fp.read(2)
                                                                 したデータ長を返します。
    if ((len == "") | | (len == nil)) then # read result is empty
     return -1
                                                                 受信していなかったら終了
    size = len.unpack("S*")[0]
    # The received data is read
                                                                受信したデータを取得
    recv_buf = fp.read(size)
    if ((recv_buf == "") | | (recv_buf == nil)) then # read result is empty
     return -1
    end
   return recv_buf
  end
  def print_raw(raw)
          fc = raw.unpack("H*")
          len = raw.length
          print(len,"bytes: ",fc,"\fr\fr\)
  end
```



About SerialMonitor.rb

```
def print_raw2(raw)
   len = raw.length
   header = raw.unpack("H4")[0]
        # unsupported format
                                     "21a8"は IEEE802.15.4e の MAC ヘッダです。
        if header != "21a8" then
                                     Lazurite Sub-GHz から送信したデータは、この値
          unsupported_format(raw)
          return
                                     になります。
        end
        # supported format
                                     3 バイト目のデータが省略されていますが、ここには
   seq = raw[2].unpack("H2")[0]
                                     シーケンス番号が格納されています。ドライバ内で使
                                     用していますが、Ruby 側では不要なデータです。
        # PANID
   myPanid = raw[3..4].unpack("S*")[0]
        # RX Address
                                             送信先(Raspberry Pi 側)の PANID,アドレス,
        rxAddr = raw[5..6].unpack("S*")[0]
                                             送信元のアドレスが格納されています。
        # TX Address
        txAddr = raw[7..8].unpack("S*")[0]
                                             Payload 値を文字列として認識しています。
        str_buf = raw[9..len-2].unpack("Z*")[0]
        print(sprintf("PANID=0x%04X, rxAddr=0x%04X, txAddr=0x%04X,
Strings:: %s",myPanid,rxAddr,txAddr,str_buf))
 \# printing unsupported format
 def unsupported_format(raw)
                                             本プログラムが未対応なフォーマットを受
   data = raw.unpack("H*")
                                             信した時の処理。
        print("unsupported format::",data,"\format")
                                             MAC ヘッダを含む全データを表示します。
 end
end
```



About SerialMonitor.rb

```
# Main function
##
class MainFunction
 ### Variable definition
                                                                 デバイスファイルの指定
 bp3596_dev = "/dev/bp3596" # Receiving pipe
 finish_flag = 0
                         # Finish flag
 # Process at the time of SIGINT Receiving
 Signal.trap(:INT)\{
                                                                  Ctrl+C を押したときの処理
   finish_flag=1
 # Open the Receiving pipe
                                                                  デバイスドライバを開く
 bp3596\_fp = open(bp3596\_dev, "rb")
 bp_api = BP3596PipeApi.new
 \# Loop until it receives a SIGINT
 while finish_flag==0 do
   # Read device
                                                                  ドライバから raw データを取得
   recv\_buf = bp\_api.read\_device(bp3596\_fp)
   if recv_buf == -1
     next
   # Create a transmit buffer from the receive buffer
                                                                 Raw データをモニターに出力
         bp_api.print_raw2(recv_buf)
 end
 # Close the Receiving pipe
 bp3596\_fp.close
end
```