

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

```
#!/usr/bin/ruby
# -*- coding: utf-8; mode: ruby -*-
# Function:
#   Lazurite Sub-GHz/Lazurite Pi Gateway Sample program
#   SerialMonitor.rb

require 'net/http'
require 'date'
##
# BP3596 API
##
class BP3596PipeApi
  ##
  # func   : Read the data from the receiving pipe
  # input  : Receive pipe fp
  # return : Receive data
  ##
  def read_device(fp)
    # Data reception wait (timeout = 100ms)
    ret = select([fp], nil, nil, 0.1)

    # Reads the size of the received data
    len = fp.read(2)
    if ((len == "") || (len == nil)) then # read result is empty
      return -1
    end
    size = len.unpack("S*")[0]

    # The received data is read
    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, "¥r¥n")
  end
end
```

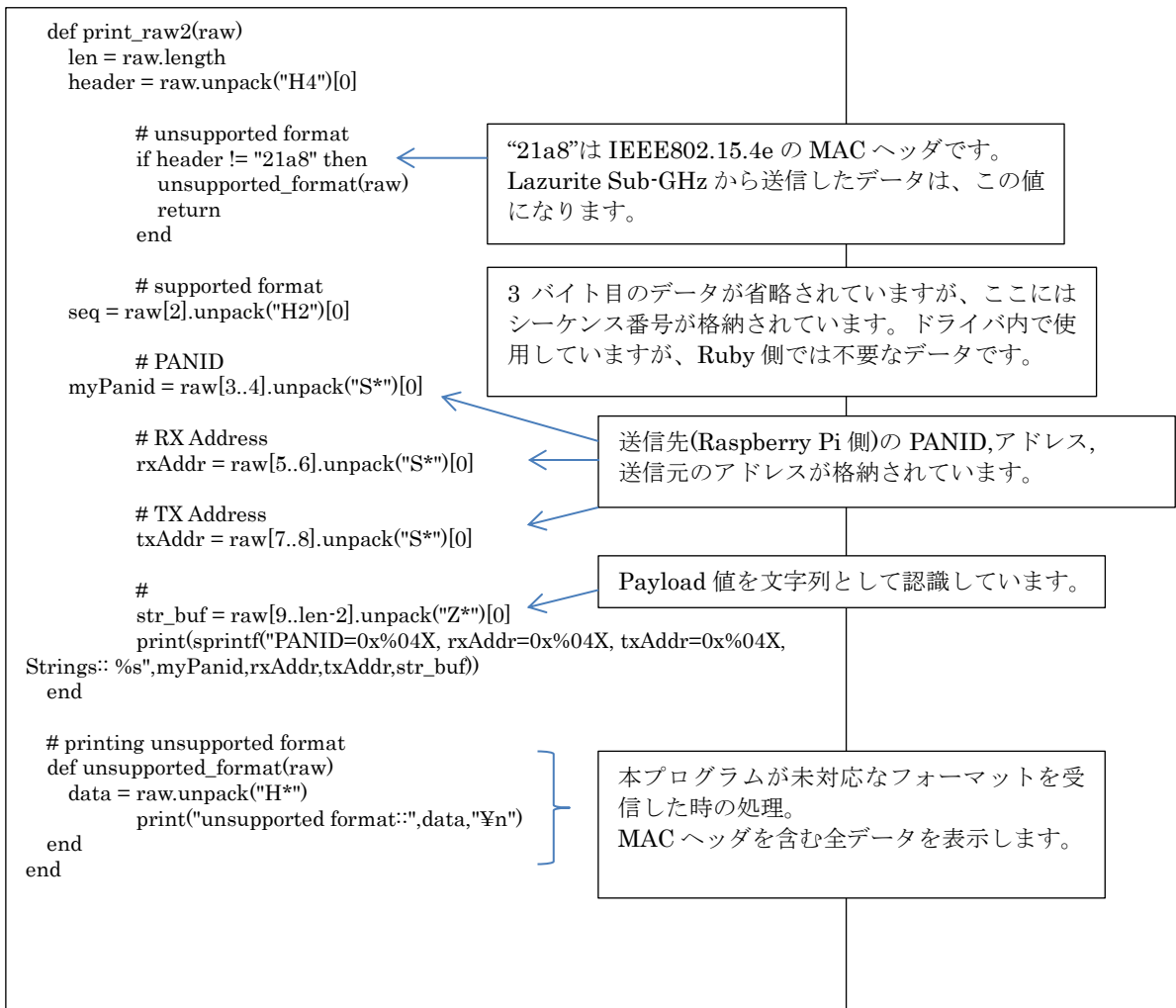
Ruby のコードであることの宣言

Time out 待ち時間を設定

ドライバの仕様
2 バイトの read を行くと、受信したデータ長を返します。

受信していなかったら終了

受信したデータを取得



```
##
# Main function
##
class MainFunction
  ### Variable definition
  bp3596_dev = "/dev/bp3596" # Receiving pipe
  finish_flag = 0           # Finish flag

  # Process at the time of SIGINT Receiving
  Signal.trap(:INT){
    finish_flag=1
  }

  # Open the Receiving pipe
  bp3596_fp = open(bp3596_dev, "rb")

  bp_api = BP3596PipeApi.new

  # Loop until it receives a SIGINT
  while finish_flag==0 do
    # Read device
    recv_buf = bp_api.read_device(bp3596_fp)
    if recv_buf == -1
      next
    end
    # Create a transmit buffer from the receive buffer
    bp_api.print_raw2(recv_buf)
  end
  # Close the Receiving pipe
  bp3596_fp.close
end
```

デバイスファイルの指定

Ctrl+C を押したときの処理

デバイスドライバを開く

ドライバから raw データを取得

Raw データをモニターに出力