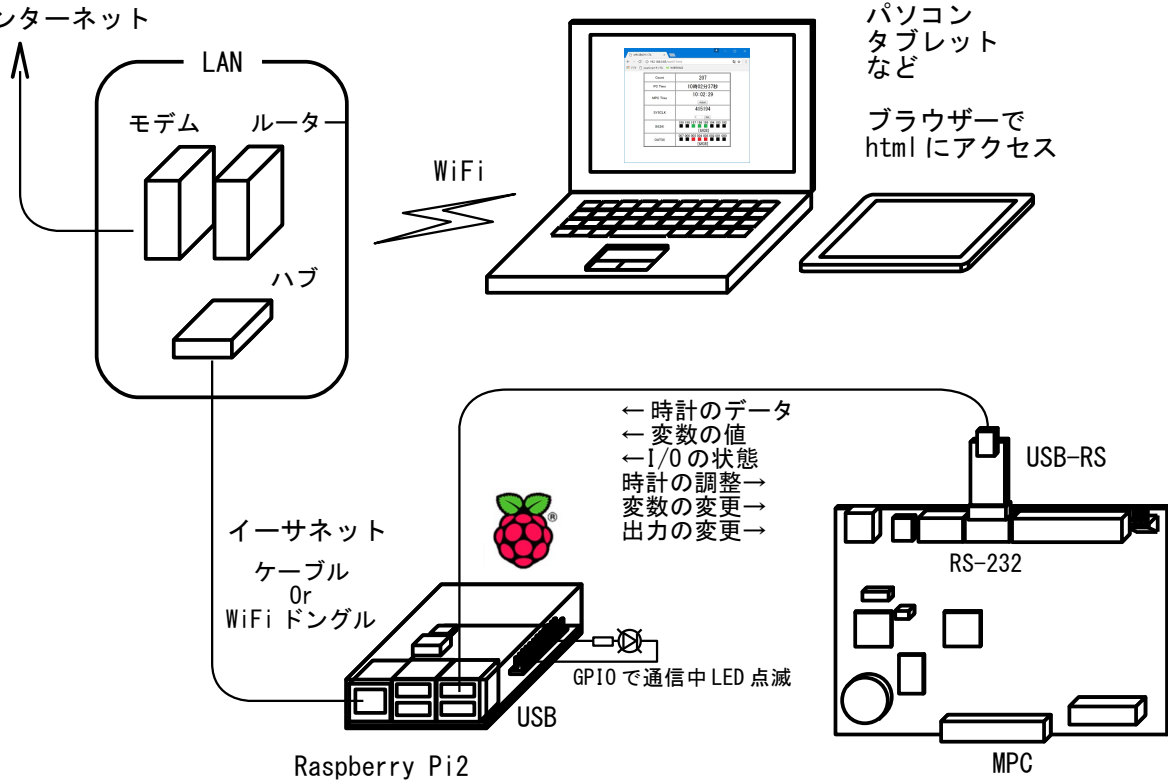


Application Note		Ref No: an2k-050	Last Modify 161129
テーマ	Raspberry Pi を使ってネットワークから MPC をみる		
使用機器	MPC-2000 シリーズ, USB-RS, Raspberry Pi2		

イメージ

名刺サイズのコンピュータ Raspberry Pi に Web サーバーを乗せて MPC の状態を取得・変更します。小型で安価な Raspberry Pi は IoT のソリューションのひとつになる?。

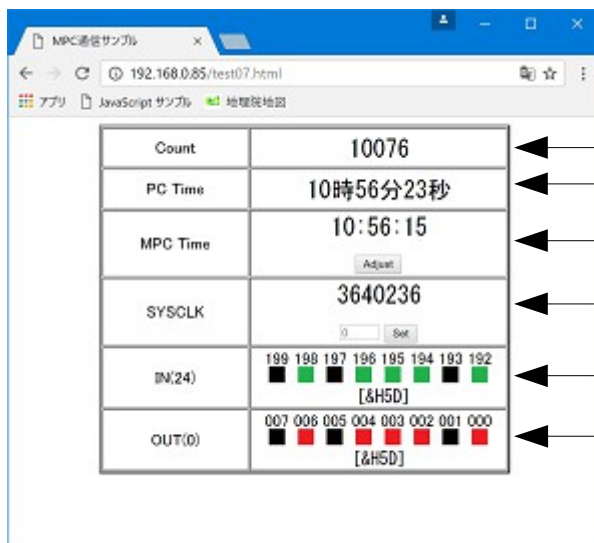
インターネット



OS は RASPBIAN (Linux raspberrypi 4.1.13)。
Web サーバーは Apache2、CGI は python で作成。
この OS には最初から FTDI のデバドラが入っている。

RS-232 で Raspberry Pi と通信。
USB-RS は FTDI のチップを搭載している。
スレーブ通信プログラムを記述。

html 画面 (Google Chrome)



- 実行回数
- パソコンの時計表示
- MPC の時計表示と調整 (PC の時計に合わせる)
- MPC の変数 (SYSCLK) 表示と設定
- MPC の入力ポートの状態表示
- MPC の出力ポートの状態表示と変更 (赤ボタンのクリックで On/Off)

(実験環境での実行速度は約 3 回/sec)

html ソース

```
<HTML>
<HEAD>
<META charset="cp932" />
<TITLE>MPC 通信サンプル</TITLE>
<SCRIPT src="http://code.jquery.com/jquery-1.11.1.min.js"></SCRIPT>
<SCRIPT src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.18/jquery.min.js"></SCRIPT>
<SCRIPT type="text/JavaScript">

var inonImg   = "../image/kaku_g.jpg";
var inoffImg  = "../image/kaku_b.jpg";
var outonImg  = "../image/kaku_r.jpg";
var outoffImg = "../image/kaku_b.jpg";
var outstat = new Array(8);
for(i = 0 ; i < 8 ; i++){ outstat[i]=0;}
var cnt = 0;
var flg = 0;
var oldsysclk;

function main() {
    now = new Date();
    str = addZero(now.getHours())+"時";
    str+= addZero(now.getMinutes())+"分";
    str+= addZero(now.getSeconds())+"秒";
    document.form3.textArea2.value=str;
    send();
    setTimeout("main()",10);
}

function addZero(i) {
    if (i < 10) {
        i = "0" + i;
    }
    return i;
}

function callback(json) {
    if( flg == 1)return;

    if(json.hoge2==oldsysclk) return;
    oldsysclk=json.hoge2;
    document.form3.textArea1.value= String(cnt);
    cnt += 1;

    $("#result1").html(json.hoge1);           // MPC Time
    $("#result2").html(json.hoge2);           // SYSCLK

    var d=parseInt(json.hoge3, 10);           // dec
    var b=d.toString(2);                      // binary string
    var h=d.toString(16);                     // hex string
    if( b.length < 8){
        for( i = 0 ; 8 - b.length ; i++){
            b= "0"+ b ;
        }
    }
    if( h.length < 2){
        h= "0"+ h ;
    }
    h = " [H" + h.toUpperCase() + "]" ;
    $("#result3").html(h);                   // IN(0)

    if( (d & 0x80) != 0){ document.lamp80.src = inonImg; }else{ document.lamp80.src = inoffImg; }
    if( (d & 0x40) != 0){ document.lamp40.src = inonImg; }else{ document.lamp40.src = inoffImg; }
    if( (d & 0x20) != 0){ document.lamp20.src = inonImg; }else{ document.lamp20.src = inoffImg; }
    if( (d & 0x10) != 0){ document.lamp10.src = inonImg; }else{ document.lamp10.src = inoffImg; }
    if( (d & 0x08) != 0){ document.lamp08.src = inonImg; }else{ document.lamp08.src = inoffImg; }
    if( (d & 0x04) != 0){ document.lamp04.src = inonImg; }else{ document.lamp04.src = inoffImg; }
```

```

if( (d & 0x02) != 0) { document.lamp02.src = inonImg; }else{ document.lamp02.src = inoffImg; }
if( (d & 0x01) != 0) { document.lamp01.src = inonImg; }else{ document.lamp01.src = inoffImg; }

var outd=parseInt(json.hoge4,10); // dec
$("#result4").html("&H"+("0"+(outd.toString(16)).toUpperCase()).substr(-2)+""); // IN(0)
for( i=0 ; i < 8 ; i++){
    outstat[i]=(outd >> i) & 0x01;
}
outdisp();
}

function send() {
    var s = document.createElement('script');
    s.src = '/cgi-bin/test07.py';
    document.body.appendChild(s);
    flg = 0;
    return true;
}

function send1() {
    flg = 1;
    var s = document.createElement('script');
    var now = new Date();
    var str = addZero(now.getHours())+" ";
    str+= addZero(now.getMinutes())+" ";
    str+= addZero(now.getSeconds())+" ";
    s.src = '/cgi-bin/test07.py?-f '+ str
    document.body.appendChild(s);
    return false;
}

function send2() {
    flg = 1;
    var s = document.createElement('script');
    var t = document.form3.textArea3.value;
    s.src = '/cgi-bin/test07.py?-d '+ t
    document.body.appendChild(s);
    return false;
}

function outchange(j) {
    flg = 1;
    if( outstat[j] ==0 ){
        outstat[j] = 1 ;
    }else{
        outstat[j] = 0 ;
    }
    var outdata = 0;
    for(j=0 ; j<8 ; j++){
        outdata = outdata | (outstat[j] << j);
    }
    var s = document.createElement('script');
    s.src = '/cgi-bin/test07.py?-o '+ outdata;
    document.body.appendChild(s);
    return false;
}

function outdisp() {
    for(i=0 ; i < 8 ; i++){
        if( outstat[i] ==1 ){
            document.getElementById("out" + i).src=outonImg;
        }else{
            document.getElementById("out" + i).src=outoffImg;
        }
    }
}
}

</SCRIPT>
</HEAD>

```

```

<BODY onLoad="main();" >
<FORM name="form3">
  <CENTER>
    <TABLE cellspacing="1" cellpadding="2" border="2">
      <TBODY>
        <TR>
          <TD align="center" width="140" height="40">
            Count
          </TD>
          <TD align="center" width="180" height="40" >
            <TEXTAREA style="border:none;resize:none;font-size:24;text-align:center" rows="1"
            cols="20" name="textArea1" ></TEXTAREA>
          </TD>
        </TR>
        <TR>
          <TD align="center" width="140" height="40">
            PC Time
          </TD>
          <TD align="center" width="180" height="40" >
            <TEXTAREA style="border:none;resize:none;font-size:24;text-align:center" rows="1"
            cols="20" name="textArea2"></TEXTAREA>
          </TD>
        </TR>
        <TR>
          <TD align="center" width="140" height="40">
            MPC Time
          </TD>
          <TD align="center" width="180" height="40" >
            <TEXTAREA style="border:none;resize:none;font-size:24;text-align:center" rows="1"
            cols="20" name="textArea5" id="result1" ></TEXTAREA>
            <p></p>
            <input type="button" name="btn" value="Adjust" onclick="return send1();" />
          </TD>
        </TR>
        <TR>
          <TD align="center" width="140" height="40">
            SYSCLK
          </TD>
          <TD align="center" width="180" height="40" >
            <TEXTAREA style="border:none;resize:none;font-size:24;text-align:center" rows="1"
            cols="20" name="textArea6" id="result2"></TEXTAREA>
            <p></p>
            <div style="display:inline-flex">
              <form><TEXTAREA style="resize:none" rows="1" cols="5" name="textArea3"
              ></TEXTAREA></form>
            </div>
            <input type="button" name="btn" value="Set" onclick="return send2();" />
          </TD>
        </TR>
        <TR>
          <TD align="center" width="140" height="60">
            IN(24)
          </TD>
          <TD align="center" width="180" height="60" >
            &nbsp;:199 198 197 196 195 194 193 192
            <br>
            
            
            
            
            
            
            
            
            <TEXTAREA style="border:none;resize:none;font-size:18;text-align:center" rows="1"

```

```

cols="20" name="textArea7" id="result3" ></TEXTAREA>
</TD>
</TR>
<TR>
<TD align="center" width="140" height="60">
OUT (0)
</TD>
<TD align="center" width="180" height="60" >
 nbsp;nbsp;007 006 005 004 003 002 001 000
<br>








<TEXTAREA style="border:none;resize:none;font-size:18;text-align:center" rows="1"
cols="20" name="textArea8" id="result4" ></TEXTAREA>
</TD>
</TR>
</BODY>
</TABLE>
</CENTER>
</FORM>
</BODY>
</HTML>

```

Python CGI スクリプト

```

#!/usr/bin/python
# -*- coding: cp932 -*-

import os
import argparse
import sys
import serial

args = sys.argv
argc = len(args)

# コマンドライン引数を格納したリストの取得
# 引数の個数

parser = argparse.ArgumentParser() # parser を作る
parser.add_argument('-d', type=str) # オプション追加
parser.add_argument('-f', type=str)
parser.add_argument('-s', type=str)
parser.add_argument('-o', type=str)
args = parser.parse_args() # コマンドラインの引数を解釈

print "Content-Type:text/javascript"
print

con = serial.Serial('/dev/ttyUSB0', 38400)

cmdline = 'sudo python /usr/lib/cgi-bin/gpio25.py'
os.system(cmdline)

if argc == 1 :
    con.write('B\r') # オプション無
    # Get TIME$
    rcv1 = con.readline()
    rcv1 = rcv1.strip()
    con.write('C\r') # Get SYSCLK
    rcv2 = con.readline()
    rcv2 = rcv2.strip()

```

```

con.write('E¥r')          # Get IN(24)
rcv3 = con.readline()
rcv3 = rcv3.strip()
con.write('G¥r')          # Get IN(0)
rcv4 = con.readline()
rcv4 = rcv4.strip()

.., 'hoge4' : "" + rcv4 + "" }):"
print str

else:
if args.d :
con.write('D' + args.d + '¥r')    # Set SYSCLK
rcv2 = con.readline()
rcv2 = rcv2.strip()
if args.f :
con.write('F' + args.f + '¥r')    # Adjust RTC
rcv2 = con.readline()
rcv2 = rcv2.strip()
if args.s :
os.system('sudo shutdown -h now')
if args.o :
con.write('O' + args.o + '¥r')    # Set Output
rcv2 = con.readline()
rcv2 = rcv2.strip()

```

MPC プログラム

```

QUIT_FORK 1 *main
END

*main
QUIT_FORK 2 *IO

rsch_=2 /* RS-232 Port No.
GNFG# rsch_ "38400b8pns1NONE"
INPUT# rsch_ CLR_BUF

DO
INPUT# rsch_ rcv$ /* Raspberry Pi から受信
PR rcv$
ptr=rcv$
arg$=PTR$(1) /* 先頭の文字を取得
SELECT_CASE arg$
CASE "A"
send$=DATE$(1) /* 年月日
CASE "B"
send$=TIME$(1) /* 時分秒
CASE "C"
send$=STR$(SYSCLK) /* SYSCLK 値
CASE "D"
SYSCLK=VAL(rcv$) /* SYSCLK 設定
send$=""
CASE "E"
send$=STR$(IN(24)) /* 入力の状態
CASE "F"
SET_RTC VAL(rcv$) VAL(0) VAL(0) /* 時分秒 設定
send$=""
CASE "G"
send$=STR$(IN(0)) /* 出力の状態
CASE "O"
OUT VAL(rcv$) 0 /* 出力設定
send$=""
CASE_ELSE
send$=""
END_SELECT
PR send$

```

```
PRINT# rsch_send$ "%r%n" /* Raspberry Pi に返信
LOOP
*10                                /* 出力のダミー動作
DO
  FOR i_=0 TO 255
    OUT i_ 0
    TIME 500
  NEXT
LOOP
```