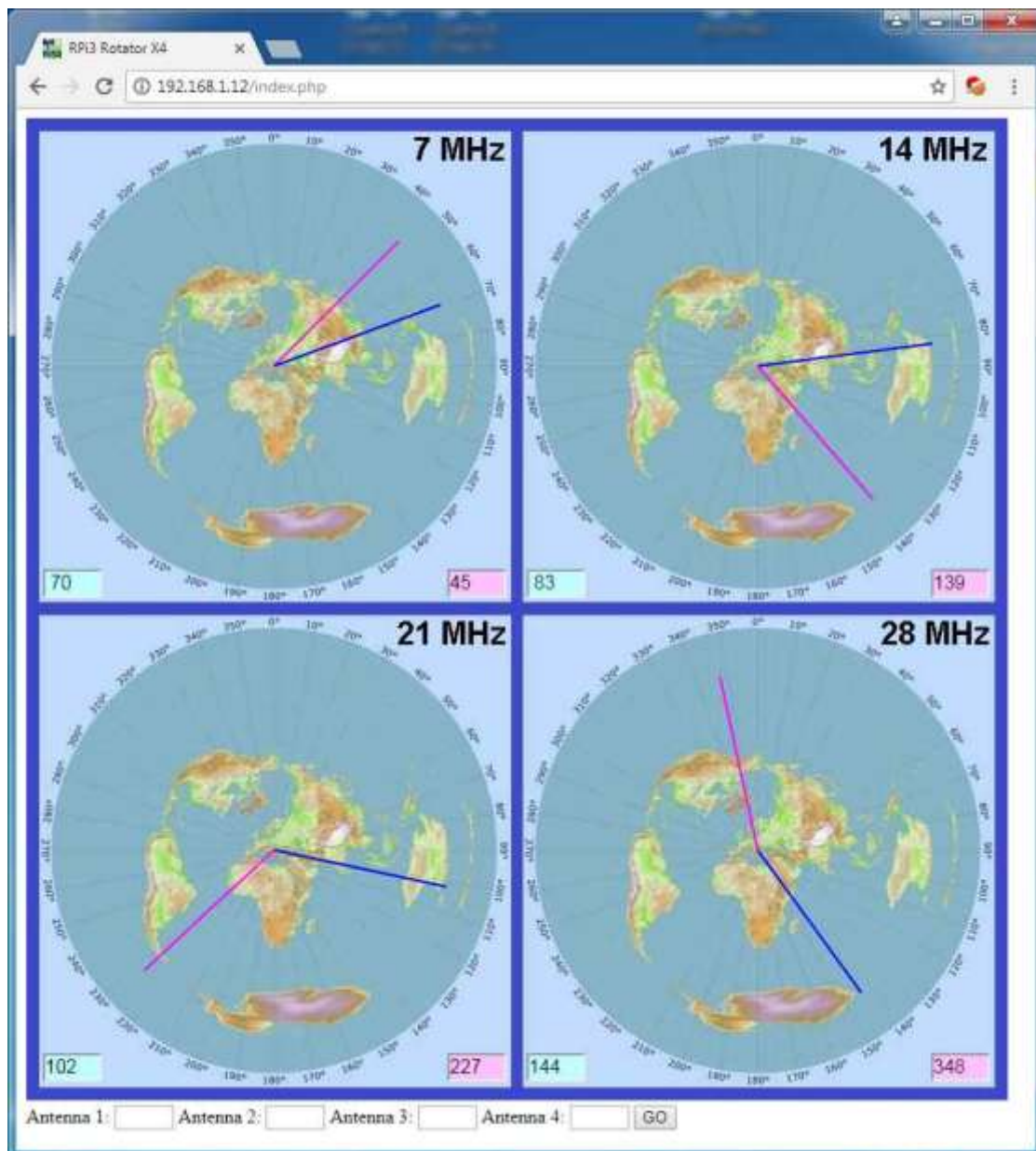
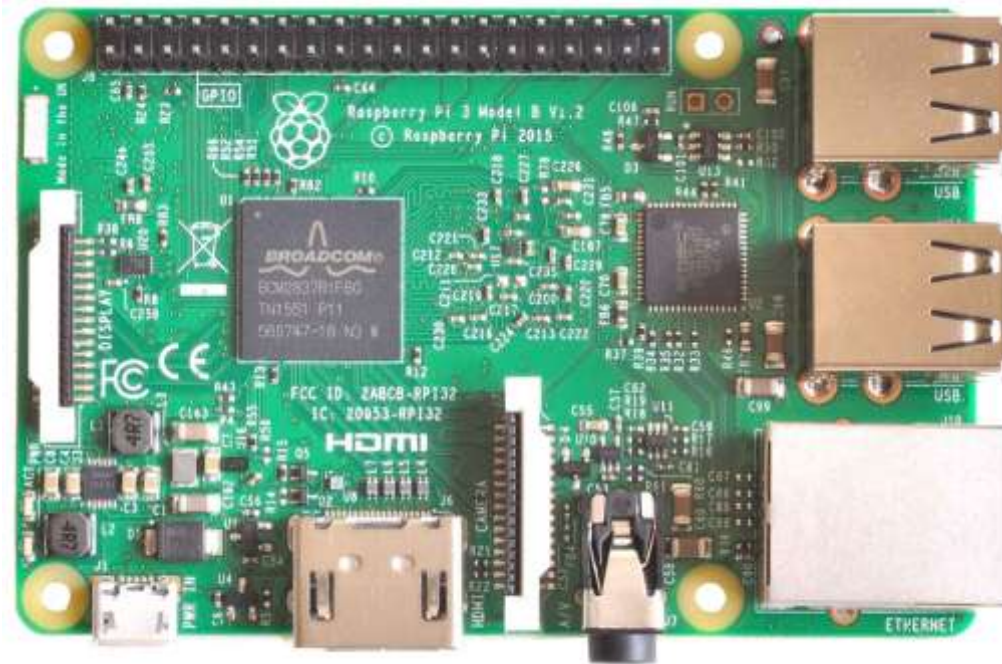


RPI3 Rotator X4

by Goran Stankovic dip.ing.el. – YT2FSG – Date: 04.01.2018.

email: goranstank@gmail.com, web: <http://www.gsl.net/yt2fsg/>





LINUX – najčešće komande

```

$ sudo ime_kom      substitute user do (omogućuje izvršavanje komandi kao drugi korisnik)
$ pwd              prikaži ime radnog direktorijuma
$ cd ime_dir       promeni direktorijum
$ ls               lista fajlova u direktorijumu
$ ls -altr        lista fajlova u direktorijumu, poredaj po vremenu formiranja fajla
$ mkdir ime_dir   kreiraj novi direktorijum
$ rmdir ime_dir   izbriši direktorijum
$ rm ime          izbriši fajl
$ rm *.txt        izbriši fajlove sa ekstenzijom txt
$ touch ime       kreiraj tekst fajl u tekućem direktorijumu
$ mv ime1 ime2    promeni ime fajla
$ cp ime /ime_dir kopiraj fajl u direktorijum
$ less ime        aplikacija za pregledanja fajlova (prekid pritiskom na Q)
$ nano ime        tekst editor
$ ./ime_prg       izvrši program
$ exit           izlazak iz shell-a ako smo logovani kao root ili neki drugi korisnik
$ man ime_kom    upustvo za komande
$ df             prikaži prostor na disku
$ free          prikaži prostor RAM memorije
$ ps aux        lista svih procesa
$ top           zauzeće procesora
$ tar cvf ime.tar /ime_dir   pakovanje fajlova bez kompresije sa imenom direktorija
$ tar xvf ime.tar            raspakivanje fajlova
$ tar cvf ime.tgz /ime_dir   kompresija fajlova sa imenom direktorija
$ tar xvf ime.tgz           dekompresija fajlova
$ zip ime.zip *.*           kompresija svih fajlova u direktorijumu
$ unzip -La ime.zip        dekompresija fajlova
$ chown ime               promena vlasništva fajla ili direktorijuma
$ chmod ime              promena dozvola na fajlu ili folderu, prvi broj predstavlja kreatora, drugi grupu i tereći bilo koga.
                          (read = 4, write = 2, exec = 1)
$ chmod 000             nije dozvoljeno ništa
$ chmod 644            najčešće dozvole
$ chmod 754            korisnik – može sve, grupa – da čita i izvršava, i bilo ko – može samo da čita
$ chmod 755            korisnik – može sve, grupa – da čita i izvršava, i bilo ko – da čita i izvršava
$ id                   identifikacija korisnika
$ date                 prikaži sistemsko vreme

$ sudo shutdown -h now isključi računar (RPI)

```

nano - tekst editor komande:

Ctrl+G = HELP
 Ctrl+K = CAT LINE
 Alt+6 = COPY LINE
 Ctrl+U = PASTE LINE
 Ctrl+O = SAVE FILE
 Ctrl+X = QUIT
 Ctrl+C = Cursor position
 Ctrl+W = Search text
 Alt+W = Replace
 Alt+R = Search and replace

Instalacija operativnog sistema

Sa zvaničnog sajta: <https://www.raspberrypi.org/downloads/raspbian/> skinuti zadnju verziju RASPBIAN YYYY-mm-dd-raspbian-stretch-lite, raspakovati sa 7-Zip, snimiti iso file na micro SD karticu od 4GB pomocu Win32 Disk Imager.

User: pi

Password: raspberry

```
$ df -h          prikaži prostor na disku
$ free -h       prikaži prostor RAM memorije
$ hostname -I   prikaži net adresu RPI-ja
```

Promenite password

```
$ passwd
```

Instalacija SSH telnet

```
$ sudo systemctl enable ssh
$ sudo systemctl start ssh
```

Instalacija Apache servera i PHP:

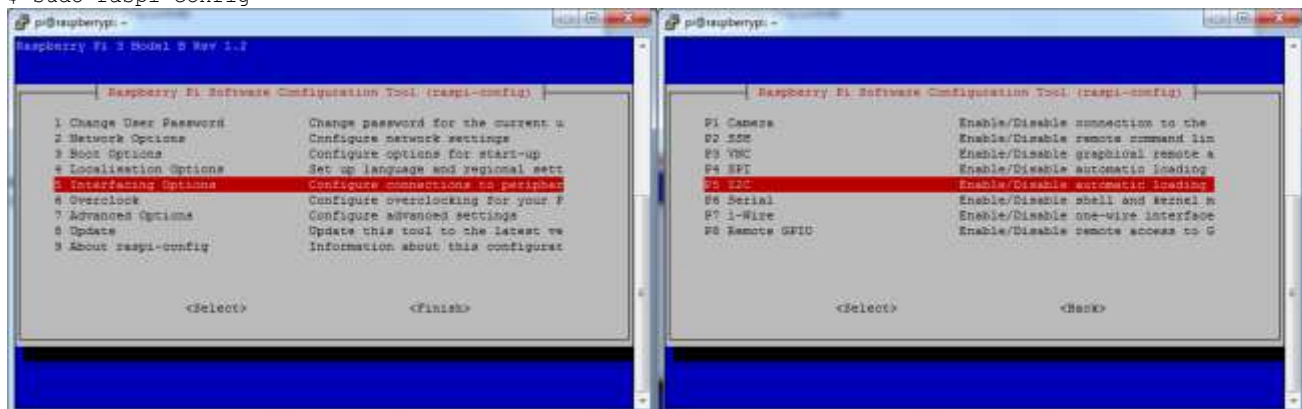
```
$ sudo apt-get install apache2 -y
$ sudo apt-get install php libapache2-mod-php -y
```

Instalacija FTP servera:

```
$ sudo apt-get install pure-ftpd
$ sudo groupadd ftpgroup
$ sudo useradd ftpuser -g ftpgroup -s /sbin/nologin -d /dev/null
$ sudo mkdir /home/pi/FTP
$ sudo chown -R ftpuser:ftpgroup /home/pi/FTP
$ sudo pure-pw useradd upload -u ftpuser -g ftpgroup -d /home/pi/FTP -m
$ sudo pure-pw mkdb
$ sudo ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/60puredb
$ sudo service pure-ftpd restart
```

Configuracija SPI, I2C, 1-Wire

```
$ sudo raspi-config
```



Enable P4 SPI, P5 I2C; Disable P7 1-Wire

```
$ sudo apt-get install i2c-tools
$ sudo apt-get install python-smbus
```

Instalacija WiringPi:

```
$ sudo apt-get install git-core
$ cd
$ git clone git://git.drogon.net/wiringPi
$ cd ~/wiringPi
$ git pull origin
$ ./build
$ gpio -v
```

Formiranje RAM DISK

```
$ sudo mkdir /var/tmp1
u File /etc/fstab dodati:
$ sudo nano /etc/fstab
.
.
.
tmpfs /var/tmp1 tmpfs nodev,nosuid,size=1M 0 0
$ sudo mount -a
$ df
```

Instalacija software RPI3 Rotator X4

Kompajlirajte softver pomoću gcc:

```
$ gcc -Wall -o rotator3x4 rotator3x4.c -lwiringPi
```

Testirajte softver pomoću:

```
$ sudo ./rotator3x4
```

U fajlu /etc/rc.local:

```
$ sudo nano /etc/rc.local
```

Dodati liniju:

```
sudo /home/pi/rotator3x4
```

Raspakujte fajl rpi3_rotator_x4.tar u direktorijum /var/www/html

```
$ tar xvf rpi3_rotator_x4.tar
```

```
$ ls /var/www/html
index.php          styles.css        yu5r_360_4.jpg   yu5r.ico
pos_set           position         set
```

```
$ sudo chmod 777 /var/www/html/set
$ sudo chmod 777 /var/www/html/pos_set
```

Razmena podataka izmedju /var/www/html/index.php i programa /home/rpi/rotator3x4 je pomoću fajlova:

Podaci o trenutnom položaju antena upisuju se svake sekunde i nalaze se na RAM-DISK-u (rotator3x4 upisuje, PHP čita):

```
File: /var/tmp1/position
228,164,135,116
```

Podaci o zadatom položaju antena nalaze se na FLASH-DISK-u (PHP upisuje, rotator3x4 čita):

```
File: /var/www/html/pos_set
90,179,270,330
```

Podatak o komandi **GO** za pokretanje motora rotatora nalazi se na FLASH-DISK-u (PHP upisuje(1), rotator3x4 čita - upisuje(0)):

```
File: /var/www/html/set
0
```

Parametre koristi program /home/rpi/rotator3x4 nalaze se na FLASH-DISK-u:

Parametri za kalibraciju pokazivanja trenutnog položaja antena 0° i 360° su u binarnim jedinicama, vrednost koju čita ADC MCP3008 :

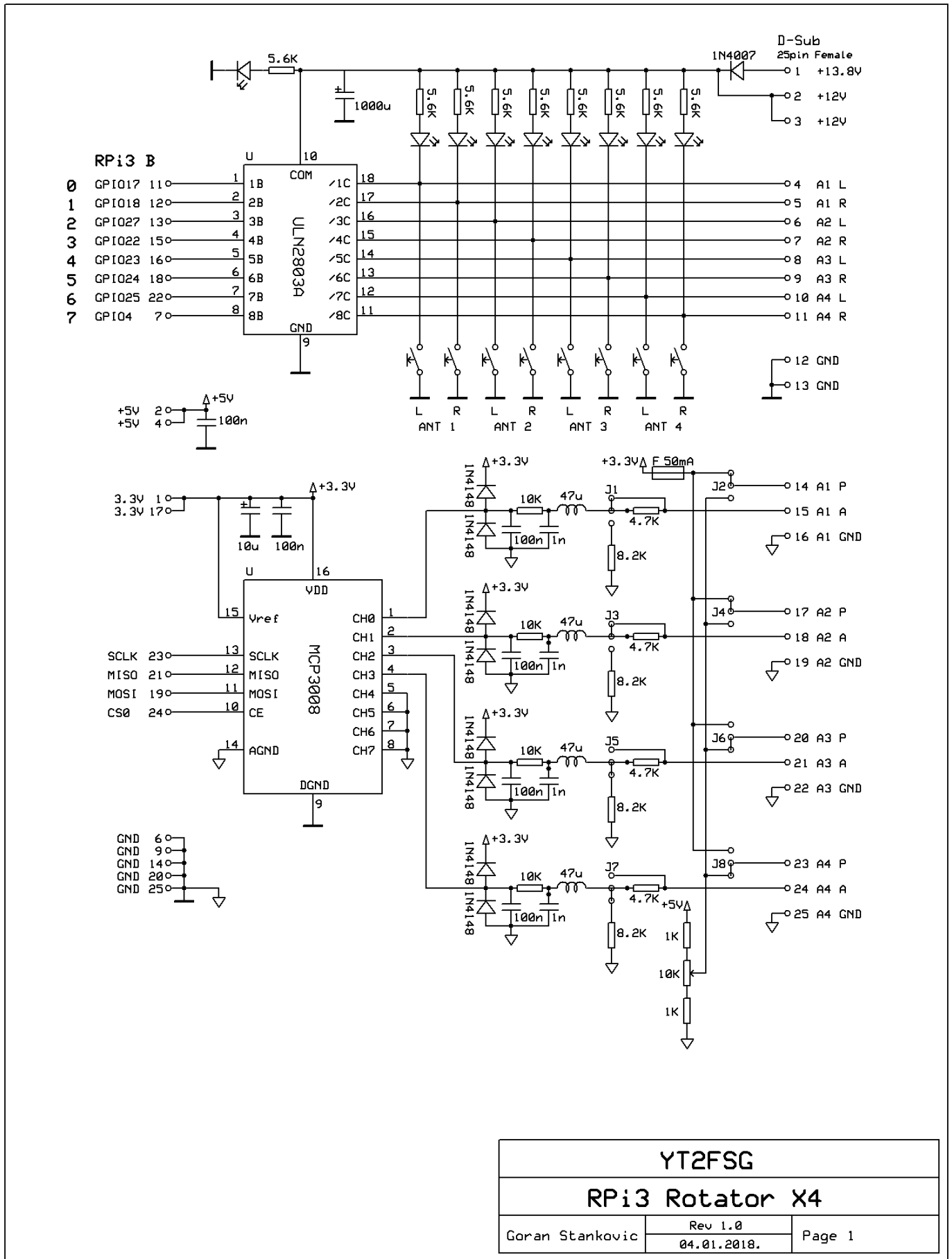
pmin1, pmax1, pmin2, pmax2, pmin3, pmax3, pmin4, pmax4;

```
File: /home/rpi/calibration
50,1000,50,1000,50,1000,50,1000
```

Parametri maksimalno vreme rada motora rotatora posle komande **GO** su: time_max1, time_max2, time_max3, time_max4;

```
File: /home/rpi/time_max
```

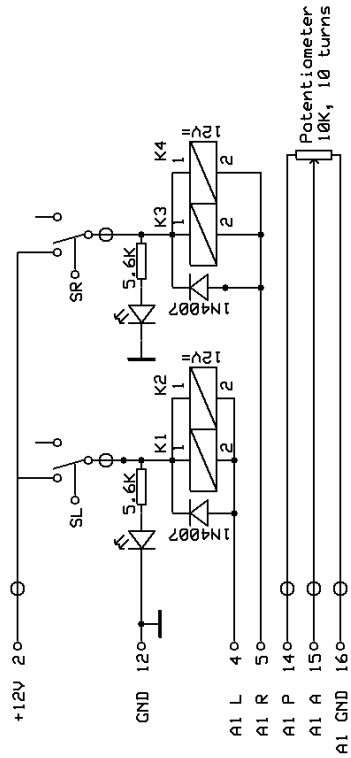
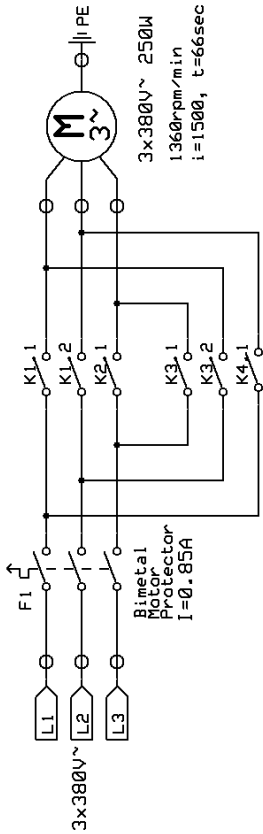
80, 80, 120, 180



YT2FSG		
RPi3 Rotator X4		
Goran Stankovic	Rev 1.0 04.01.2018.	Page 1

YU5R Antenna Rotators

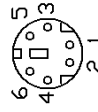
ANTENNA 1 = 7MHZ



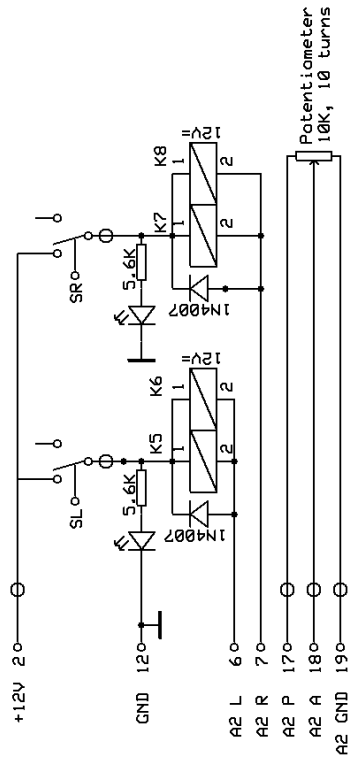
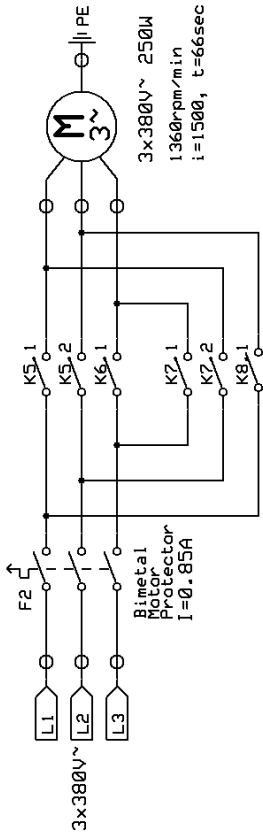
ANTENNA 3 = 21MHZ

YAESU G-800DX, G-1000DX

miniDIN connector



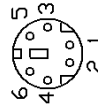
ANTENNA 2 = 14MHZ



ANTENNA 4 = 28MHZ

YAESU G-800DX, G-1000DX

miniDIN connector



File: /var/www/html/index.php

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="author" content="YT2FSG - Goran Stankovic dipl.ing.el.">
<title>Rpi3 Rotator X4</title>
<link rel="icon" href="yu5r.ico" type="image/ico" />

<script language="JavaScript">
function getDetails(obj)
{
    var clickX = window.event.x-obj.offsetLeft;
    var clickY = window.event.y-obj.offsetTop;

    if (clickX > 10 && clickX < 410 && clickY > 10 && clickY < 410) {
        clickX=210-clickX;
        clickY=clickY-210;
        var azi = clickY / clickX;
        azi = Math.atan(azi);
        azi=parseInt(90-57.29*azi);
        if (clickX >= 0)
        {
            azi=180+azi;
        }
        document.getElementById("p_az1").value = azi;

    } else if (clickX > 420 && clickX < 820 && clickY > 10 && clickY < 410) {
        clickX=620-clickX;
        clickY=clickY-210;
        var azi = clickY / clickX;
        azi = Math.atan(azi);
        azi=parseInt(90-57.29*azi);
        if (clickX >= 0)
        {
            azi=180+azi;
        }
        document.getElementById("p_az2").value = azi;

    } else if (clickX > 10 && clickX < 410 && clickY > 420 && clickY < 820) {
        clickX=210-clickX;
        clickY=clickY-620;
        var azi = clickY / clickX;
        azi = Math.atan(azi);
        azi=parseInt(90-57.29*azi);
        if (clickX >= 0)
        {
            azi=180+azi;
        }
        document.getElementById("p_az3").value = azi;

    } else if (clickX > 420 && clickX < 820 && clickY > 420 && clickY < 820) {
        clickX=620-clickX;
        clickY=clickY-620;
        var azi = clickY / clickX;
        azi = Math.atan(azi);
        azi=parseInt(90-57.29*azi);
        if (clickX >= 0)
        {
            azi=180+azi;
        }
        document.getElementById("p_az4").value = azi;
    }
}

function draw()
{
    var canvas = document.getElementById('cav1');

    if (canvas.getContext)
    {
        var ctx = canvas.getContext('2d');

        ctx.drawImage(mapa_4, 0, 0);

        ctx.beginPath();
        ctx.moveTo(210, 210);
        ctx.lineTo(a_x1, a_y1);
        ctx.strokeStyle = '#FF00FF';
        ctx.lineWidth = 2;
        ctx.stroke();
        ctx.beginPath();
    }
}
```

```

ctx.moveTo(210, 210);
ctx.lineTo(p_x1, p_y1);
ctx.strokeStyle = '#0000FF';
ctx.lineWidth = 2;
ctx.stroke();

ctx.beginPath();
ctx.moveTo(620, 210);
ctx.lineTo(a_x2, a_y2);
ctx.strokeStyle = '#FF00FF';
ctx.lineWidth = 2;
ctx.stroke();
ctx.beginPath();
ctx.moveTo(620, 210);
ctx.lineTo(p_x2, p_y2);
ctx.strokeStyle = '#0000FF';
ctx.lineWidth = 2;
ctx.stroke();

ctx.beginPath();
ctx.moveTo(210, 620);
ctx.lineTo(a_x3, a_y3);
ctx.strokeStyle = '#FF00FF';
ctx.lineWidth = 2;
ctx.stroke();
ctx.beginPath();
ctx.moveTo(210, 620);
ctx.lineTo(p_x3, p_y3);
ctx.strokeStyle = '#0000FF';
ctx.lineWidth = 2;
ctx.stroke();

ctx.beginPath();
ctx.moveTo(620, 620);
ctx.lineTo(a_x4, a_y4);
ctx.strokeStyle = '#FF00FF';
ctx.lineWidth = 2;
ctx.stroke();
ctx.beginPath();
ctx.moveTo(620, 620);
ctx.lineTo(p_x4, p_y4);
ctx.strokeStyle = '#0000FF';
ctx.lineWidth = 2;
ctx.stroke();

}
else
{
// make the element with the no-canvas message visible
document.getElementById('no_cav').style.display = 'block';
}
}
</script>
</head>

<body onload="draw()">

<canvas width="832" height="832" id="cav1" onclick="getDetails(this)">
  
</canvas>
<p id="no_cav" style="display:none;">You need a Canvas-compatible browser to view canvas area.</p>

<?php
echo '<link rel="stylesheet" href="styles.css">';

//echo "<br>";

$filename = "/var/tmp1/position";
//$filename = "position";
$handle = fopen($filename, "r");
$azim_f = fread($handle, filesize($filename));
fclose($handle);

$azim_f1=$azim_f;
$p1 = substr($azim_f1, 0, strpos($azim_f1, ','));
$azim_f1 = substr($azim_f1, strpos($azim_f1, ',')+1 );
$p2 = substr($azim_f1, 0, strpos($azim_f1, ','));
$azim_f1 = substr($azim_f1, strpos($azim_f1, ',')+1 );
$p3 = substr($azim_f1, 0, strpos($azim_f1, ','));
$p4 = substr($azim_f1, strpos($azim_f1, ',')+1 );

```



```

$filename = "pos_set";
$handle = fopen($filename, "r");
$azim_f = fread($handle, filesize($filename));
fclose($handle);

$azim_f1=$azim_f;
$a1 = substr($azim_f1, 0, strpos($azim_f1, ','));
$azim_f1 = substr($azim_f1, strpos($azim_f1, ',')+1 );
$a2 = substr($azim_f1, 0, strpos($azim_f1, ','));
$azim_f1 = substr($azim_f1, strpos($azim_f1, ',')+1 );
$a3 = substr($azim_f1, 0, strpos($azim_f1, ','));
$a4 = substr($azim_f1, strpos($azim_f1, ',')+1 );

//print("$p1 $p2 $p3 $p4");

echo '<input class="p1" type="text" value="'. $p1 .' " name="p1" maxlength="4">';
echo '<input class="p2" type="text" value="'. $p2 .' " name="p2" maxlength="4">';
echo '<input class="p3" type="text" value="'. $p3 .' " name="p3" maxlength="4">';
echo '<input class="p4" type="text" value="'. $p4 .' " name="p4" maxlength="4">';
echo '<input class="a1" type="text" value="'. $a1 .' " name="a1" maxlength="4">';
echo '<input class="a2" type="text" value="'. $a2 .' " name="a2" maxlength="4">';
echo '<input class="a3" type="text" value="'. $a3 .' " name="a3" maxlength="4">';
echo '<input class="a4" type="text" value="'. $a4 .' " name="a4" maxlength="4">';

$az1=(180-$a1)*0.01744;
$az_x1=150*(sin($az1))+210;
$az_y1=150*(cos($az1))+210;
$az_x1=(int)$az_x1;
$az_y1=(int)$az_y1;

$pz1=(180-$p1)*0.01744;
$pz_x1=150*(sin($pz1))+210;
$pz_y1=150*(cos($pz1))+210;
$pz_x1=(int)$pz_x1;
$pz_y1=(int)$pz_y1;

$az2=(180-$a2)*0.01744;
$az_x2=150*(sin($az2))+620;
$az_y2=150*(cos($az2))+210;
$az_x2=(int)$az_x2;
$az_y2=(int)$az_y2;

$pz2=(180-$p2)*0.01744;
$pz_x2=150*(sin($pz2))+620;
$pz_y2=150*(cos($pz2))+210;
$pz_x2=(int)$pz_x2;
$pz_y2=(int)$pz_y2;

$az3=(180-$a3)*0.01744;
$az_x3=150*(sin($az3))+210;
$az_y3=150*(cos($az3))+620;
$az_x3=(int)$az_x3;
$az_y3=(int)$az_y3;

$pz3=(180-$p3)*0.01744;
$pz_x3=150*(sin($pz3))+210;
$pz_y3=150*(cos($pz3))+620;
$pz_x3=(int)$pz_x3;
$pz_y3=(int)$pz_y3;

$az4=(180-$a4)*0.01744;
$az_x4=150*(sin($az4))+620;
$az_y4=150*(cos($az4))+620;
$az_x4=(int)$az_x4;
$az_y4=(int)$az_y4;

$pz4=(180-$p4)*0.01744;
$pz_x4=150*(sin($pz4))+620;
$pz_y4=150*(cos($pz4))+620;
$pz_x4=(int)$pz_x4;
$pz_y4=(int)$pz_y4;

echo '
<script type="text/javascript">
// <![CDATA[
    a_x1 = " " . $az_x1 . " ";
    a_y1 = " " . $az_y1 . " ";
    p_x1 = " " . $pz_x1 . " ";
    p_y1 = " " . $pz_y1 . " ";
    a_x2 = " " . $az_x2 . " ";

```

```

a_y2 = " " . $az_y2 . " ";
p_x2 = " " . $pz_x2 . " ";
p_y2 = " " . $pz_y2 . " ";
a_x3 = " " . $az_x3 . " ";
a_y3 = " " . $az_y3 . " ";
p_x3 = " " . $pz_x3 . " ";
p_y3 = " " . $pz_y3 . " ";
a_x4 = " " . $az_x4 . " ";
a_y4 = " " . $az_y4 . " ";
p_x4 = " " . $pz_x4 . " ";
p_y4 = " " . $pz_y4 . " ";
// ]]>
</script>';

echo "<br>";

if(!isset($_REQUEST['p_az1']))
{
?>
<form method="get" name="input">
  Antenna 1: <input id="p_az1" name="p_az1" value="" style="width: 50px;" maxlength="3">
  Antenna 2: <input id="p_az2" name="p_az2" value="" style="width: 50px;" maxlength="3">
  Antenna 3: <input id="p_az3" name="p_az3" value="" style="width: 50px;" maxlength="3">
  Antenna 4: <input id="p_az4" name="p_az4" value="" style="width: 50px;" maxlength="3">
  <input type="submit" value="GO" >
</form>
<?php
  exit();
}

$azim_1 = $_REQUEST['p_az1'];
$azim_2 = $_REQUEST['p_az2'];
$azim_3 = $_REQUEST['p_az3'];
$azim_4 = $_REQUEST['p_az4'];

if (strlen($azim_1)<1) { $azim_1=$p1;}
if (strlen($azim_2)<1) { $azim_2=$p2;}
if (strlen($azim_3)<1) { $azim_3=$p3;}
if (strlen($azim_4)<1) { $azim_4=$p4;}

print("Antenna 1 = $azim_1, Antenna 2 = $azim_2, Antenna 3 = $azim_3, Antenna 4 = $azim_4");
echo "<br>";

$filename = "pos_set";
$handle = fopen($filename, "w+");
fwrite($handle, $azim_1);
fwrite($handle, ",");
fwrite($handle, $azim_2);
fwrite($handle, ",");
fwrite($handle, $azim_3);
fwrite($handle, ",");
fwrite($handle, $azim_4);
fclose($handle);

$filename = "set";
$handle = fopen($filename, "w+");
fwrite($handle, '1');
fclose($handle);

echo "<br>";
?>
<meta http-equiv="refresh" content="1;url=index.php">
</body>
</html>

```

File: /home/rpi/rotator3x4.c

```

// RPi3 Rotator X4 with MCP3008
// 03.01.2018. Author: YT2FSG - Goran Stankovic dipl.ing.el.
// email: goranstank@gmail.com, web: http://www.qsl.net/yt2fsg
//
// gcc -Wall -o rotator3x4 rotator3x4.c -l wiringPi
// sudo ./rotator3x4

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <stdint.h>
#include <string.h>
#include <errno.h>

#include <wiringPi.h>
#include <wiringPiSPI.h>

#define TRUE          (1==1)
#define FALSE        (!TRUE)
#define CHAN_CONFIG_SINGLE 8
#define CHAN_CONFIG_DIFF 0

static int myFd ;

int main(void)
{
    FILE * fp;
    int spiChannel=0;

    if ((myFd = wiringPiSPISetup (spiChannel, 1000000)) < 0)
    {
        fprintf (stderr, "Can't open the SPI bus: %s\n", strerror (errno)) ;
        exit (EXIT_FAILURE) ;
    }

    wiringPiSetup () ;
    pinMode (0, OUTPUT) ; // 17
    pinMode (1, OUTPUT) ; // 18
    pinMode (2, OUTPUT) ; // 27
    pinMode (3, OUTPUT) ; // 22
    pinMode (4, OUTPUT) ; // 23
    pinMode (5, OUTPUT) ; // 24
    pinMode (6, OUTPUT) ; // 25
    pinMode (7, OUTPUT) ; // 4

    unsigned char buf[] = { 0x01, 0x80, 0x00 }; // Data to send
    int adc1, adc2, adc3, adc4;
    int p1, p2, p3, p4;
    int set=1; // Start pozicioniranja
    int delta=2; // min ugao za koji rotator ne reaguje
    int out1=0, out2=0, out3=0, out4=0;
    int ps1=100, ps2=200, ps3=300, ps4=340;
    int tt=4; // 250ms
    int time1=0, time2=0, time3=0, time4=0;
    int time_max1=60, time_max2=60, time_max3=60, time_max4=60;
    int pmin1=0, pmax1=1023, pmin2=0, pmax2=1023, pmin3=0, pmax3=1023, pmin4=0, pmax4=1023;
    int op1, op2, op3, op4;
    unsigned char cc=0;
    char c;
    char buffer[100];
//    long length;
    int i=0, j=0;
    int buffer2[16];
// -----
    printf("*** RPi3 Rotator for X4 antennas system ***\n");
    printf("by: YT2FSG - Goran Stankovic dipl.ing.el. (goranstank@gmail.com)\n\n");
    printf("Parameters: Ant1 Ant2 Ant3 Ant4\n");
// Ucitavanje parametara iz file 'time_max'
    if ((fp = fopen("time_max","r")) == NULL)
    {
        printf("file 'time_max' not found! \n");
    }
    else
    {
        while(1)
        {
            c = fgetc(fp);
            if (feof(fp)) { break; }
            if (c==',') { buffer[i] = 0; i = 0; buffer2[j] = atoi(buffer); j++; }

```

```

    else { buffer[i] = c; i++; }
}
buffer[i] = 0;
buffer2[j] = atoi(buffer);

    printf("time_max (sec): %3d %3d %3d %3d \n", buffer2[0], buffer2[1], buffer2[2], buffer2[3]);

time_max1 = buffer2[0];
time_max2 = buffer2[1];
time_max3 = buffer2[2];
time_max4 = buffer2[3];
}
fclose(fp);
// -----
// Ucitavanje parametara iz file 'calibration'
i=0; j=0;
if ((fp = fopen("calibration","r")) == NULL)
{
    printf("file 'calibration' not found! \n");
}
else
{
    while(1)
    {
        c = fgetc(fp);
        if (feof(fp)) { break; }
        if (c==',') { buffer[i] = 0; i = 0; buffer2[j] = atoi(buffer); j++; }
        else { buffer[i] = c; i++; }
    }
    buffer[i] = 0;
    buffer2[j] = atoi(buffer);

        printf("cal_min  ( ): %3d %3d %3d %3d \n", buffer2[0], buffer2[2], buffer2[4], buffer2[6]);
        printf("cal_max  ( ): %3d %3d %3d %3d \n", buffer2[1], buffer2[3], buffer2[5], buffer2[7]);

        pmin1 = buffer2[0]; pmax1 = buffer2[1];
        pmin2 = buffer2[2]; pmax2 = buffer2[3];
        pmin3 = buffer2[4]; pmax3 = buffer2[5];
        pmin4 = buffer2[6]; pmax4 = buffer2[7];
    }
    fclose(fp);
//    exit(0);
// -----
while (1)
{
    // Citanje pozicije antena - potencijometri (MCP3008)
    delay(50); // delay 50ms
    // CH0
    buf[0] = 0x01; buf[1] = 0x80; buf[2] = 0x00;
    wiringPiSPIDataRW(spiChannel, buf, 3);
    adc1 = ( (buf[1] & 3) << 8 ) + buf[2];

    delay(50); // delay 50ms
    // CH1
    buf[0] = 0x01; buf[1] = 0x90; buf[2] = 0x00;
    wiringPiSPIDataRW(spiChannel, buf, 3);
    adc2 = ( (buf[1] & 3) << 8 ) + buf[2];

    delay(50); // delay 50ms
    // CH2
    buf[0] = 0x01; buf[1] = 0xA0; buf[2] = 0x00;
    wiringPiSPIDataRW(spiChannel, buf, 3);
    adc3 = ( (buf[1] & 3) << 8 ) + buf[2];

    delay(50); // delay 50ms
    // CH3
    buf[0] = 0x01; buf[1] = 0xB0; buf[2] = 0x00;
    wiringPiSPIDataRW(spiChannel, buf, 3);
    adc4 = ( (buf[1] & 3) << 8 ) + buf[2];

// -----
// Calibration
op1 = pmax1 - pmin1;
    if (op1 < 1 ) { p1 = adc1 * 360 / 1023; }
    else { p1 = (adc1 - pmin1) * 360 / op1; }
op2 = pmax2 - pmin2;
    if (op2 < 1 ) { p2 = adc2 * 360 / 1023; }
    else { p2 = (adc2 - pmin2) * 360 / op2; }
op3 = pmax3 - pmin3;

```

```

    if (op3 < 1 ) { p3 = adc3 * 360 / 1023; }
    else { p3 = (adc3 - pmin3) * 360 / op3; }
op4 = pmax4 - pmin4;
    if (op4 < 1 ) { p4 = adc4 * 360 / 1023; }
    else { p4 = (adc4 - pmin4) * 360 / op4; }

//   printf("Position: %d=%d %d=%d %d=%d %d=%d \n", adc1, p1, adc2, p2, adc3, p3, adc4, p4);

// -----
// Ucitavanje parametara iz FLASH-DISK file 'pos_set'
i=0; j=0;
if ((fp = fopen("/var/www/html/pos_set","r")) == NULL)
{
    printf("file '/var/www/html/pos_set' not found! \n");
}
else
{
    while(1)
    {
        c = fgetc(fp);
        if (feof(fp)) { break; }
        if (c==',') { buffer[i] = 0; i = 0;  buffer2[j] = atoi(buffer); j++; }
        else { buffer[i] = c; i++; }
    }
    buffer[i] = 0;
    buffer2[j] = atoi(buffer);

//   printf("pos_set : %3d %3d %3d %3d \n", buffer2[0], buffer2[1], buffer2[2], buffer2[3]);

    ps1 = buffer2[0];
    ps2 = buffer2[1];
    ps3 = buffer2[2];
    ps4 = buffer2[3];
}
fclose(fp);
//   exit(0);
// -----
// Ucitavanje parametara iz FLASH-DISK file 'set'
i=0; j=0;
if ((fp = fopen("/var/www/html/set","r")) == NULL)
{
    printf("file '/var/www/html/set' not found! \n");
}
else
{
    while(1)
    {
        c = fgetc(fp);
        if (feof(fp)) { break; }
        buffer[i] = c; i++;
    }
    buffer[1] = 0;
    set = atoi(buffer);
//   printf("set : %d \n", set);
}
fclose(fp);

//   exit(0);
// -----

// Obrada - pozicioniranje antena na zadatu poziciju

if (set==1)
{
    set = 0;
    cc = 8;

    // -----
//Upisi '0' u FLASH-DISK file 'set'
if ((fp = fopen("/var/www/html/set","w")) == NULL)
{
    printf("file '/var/www/html/set' not open! \n");
}
else
{
    fputc( '0', fp);
}
fclose(fp);

```

```

if (p1 < ps1 - delta) { out1 = 1; }
else if (p1 > ps1 + delta) { out1 = 2; }
else { out1 = 0; }

if (p2 < ps2 - delta) { out2 = 1; }
else if (p2 > ps2 + delta) { out2 = 2; }
else { out2 = 0; }

if (p3 < ps3 - delta) { out3 = 1; }
else if (p3 > ps3 + delta) { out3 = 2; }
else { out3 = 0; }

if (p4 < ps4 - delta) { out4 = 1; }
else if (p4 > ps4 + delta) { out4 = 2; }
else { out4 = 0; }
}
else
{
if ((p1 >= ps1) && (out1==1)) { out1 = 0; }
if ((p1 <= ps1) && (out1==2)) { out1 = 0; }
if (out1 !=0)
{
time1++;
if ((time1/tt) > time_max1) { out1 = 0; }
}

if ((p2 >= ps2) && (out2==1)) { out2 = 0; }
if ((p2 <= ps2) && (out2==2)) { out2 = 0; }
if (out2 !=0)
{
time2++;
if ((time2/tt) > time_max2) { out2 = 0; }
}

if ((p3 >= ps3) && (out3==1)) { out3 = 0; }
if ((p3 <= ps3) && (out3==2)) { out3 = 0; }
if (out3 !=0)
{
time3++;
if ((time3/tt) > time_max3) { out3 = 0; }
}

if ((p4 >= ps4) && (out4==1)) { out4 = 0; }
if ((p4 <= ps4) && (out4==2)) { out4 = 0; }
if (out4 !=0)
{
time4++;
if ((time4/tt) > time_max4) { out4 = 0; }
}
}
}

// -----
// Postavnjanje izlaza
if (out1 == 1)
{
digitalWrite(0,0);
digitalWrite(1,1);
}
else if (out1 == 2)
{
digitalWrite(0,1);
digitalWrite(1,0);
}
else
{
time1 = 0;
digitalWrite(0,0);
digitalWrite(1,0);
}

if (out2 == 1)
{
digitalWrite(2,0);
digitalWrite(3,1);
}
else if (out2 == 2)
{
digitalWrite(2,1);
digitalWrite(3,0);
}
else

```

```

    {
        time2 = 0;
        digitalWrite(2,0);
        digitalWrite(3,0);
    }

    if (out3 == 1)
    {
        digitalWrite(4,0);
        digitalWrite(5,1);
    }
    else if (out3 == 2)
    {
        digitalWrite(4,1);
        digitalWrite(5,0);
    }
    else
    {
        time3 = 0;
        digitalWrite(4,0);
        digitalWrite(5,0);
    }

    if (out4 == 1)
    {
        digitalWrite(6,0);
        digitalWrite(7,1);
    }
    else if (out4 == 2)
    {
        digitalWrite(6,1);
        digitalWrite(7,0);
    }
    else
    {
        time4 = 0;
        digitalWrite(6,0);
        digitalWrite(7,0);
    }
// -----
//Upisi podatke u RAM-DISK file 'position'
if ((fp = fopen("/var/tmp/position","w")) == NULL)
{
    printf("file '/var/tmp/position' not open! \n");
}
else
{
    fprintf(fp,"%3d,%3d,%3d,%3d\n",p1,p2,p3,p4);
}
fclose(fp);
// -----
printf("%d # A1: %3d=%3d %d-%03d # A2: %3d=%3d %d-%03d # A3: %3d=%3d %d-%03d # A4: %3d=%3d %d-%03d # \n",
        cc,p1,ps1,out1,time1/tt,p2,ps2,out2,time2/tt,p3,ps3,out3,time3/tt,p4,ps4,out4,time4/tt);

    cc++;
    if (cc > (tt-1)) { cc=0; }

    delay(50); // delay 4*50+50=250ms => tt=4
}
close (myFd) ;
return 0;
}

```

REFERENCES:

<http://www.digistore.rs/>
<http://www.raspberrypi.org/>
<http://wiringpi.com/>
<http://www.microchip.com/wwwproducts/en/MCP3008>
<http://www.ti.com/lit/ds/symlink/uln2803a.pdf>
<http://poincare.matf.bg.ac.rs/~maricm/komande.html>