

Rad sa PHP serverom

U ovoj vežbi ćemo se upoznati na osnovnom nivou sa PHP serverom i upotrebom *PHPlot* biblioteke radi prikaza merenih podataka (npr. vreme i temperatura) u formi grafika.

Da bi bilo moguće koristiti PHP server i biblioteku *PHPlot*, najpre je potrebno instalirati HTTP server, zatim PHP server i na kraju kopirati biblioteku *PHPlot* u željeni direktorijum (bilo koji) odakle će se kasnije moći koristiti.

15.1. HTTP server

HTTP server koji je korišćen u ovom projektu je *apache2 web-server*, koji se često koristi kao *http server*. Za njegovu instalaciju je potrebno u terminalu *Raspbian linux*-a na RPi izvršiti sledeću komandu:

```
sudo apt-get install apache2
```

Zatim nastaviti sa instalacijom i kada se instalira, može se proveriti da li server radi tako što se u *web* pretraživaču na *Raspbian linux*-u na RPi upiše *localhost* i tada bi trebalo da se pojavi stranica sa naslovom: *It works!* Da bi ovo radilo neophodno je isključiti podešavanja za proxy u internet pretraživaču! Ista stranica bi trebalo da se pojavi i ako se u *web* pretraživaču upiše IP adresa RPi. (Ovo se može uraditi sa bilo kojeg računara, koji je povezan na istu lokalnu mrežu kao RPi.)

Apache2 web-server se može prekinuti komandom:

```
sudo /etc/init.d/apache2 stop
```

A, takođe, može se pokrenuti komandom:

```
sudo /etc/init.d/apache2 start
```

Stranica sa naslovom *IT works!* se nalazi u direktorijumu: `/var/www/html`

Njeno ime je: `index.html`

Ova stranica se uvek otvara kada se sa bilo kojeg računara u *web* pretraživaču upiše IP adresa RPi i ta stranica mora imati baš to ime! Ovu stranicu je potrebno promeniti u skladu sa potrebama.

15.2. PHP server

PHP server omogućava pokretanje PHP skripti na *web*-serveru. Verzija PHP servera koju ćemo koristiti je PHP5. Potrebno je imati ovu verziju, jer je sa njom moguće koristiti biblioteku *PHPlot*. Za instalaciju ove verzije i za njeno povezivanje sa prethodno instaliranim *apache2* serverom, potrebno je u terminalu *Raspbian linux*-a na RPi izvršiti sledeću komandu:

```
sudo apt-get install php5 libapache2-mod-php5 php5-gd
```

Zatim nastaviti sa instalacijom i kada se instalira, potrebno je restartovati *apache2 web-server* komandom:

```
sudo /etc/init.d/apache2 restart
```

Može se proveriti da li server radi tako što se napravi test PHP datoteka komandom:

```
sudo touch /var/www/html/test.php
```

I otvori se u editoru komandom:

```
sudo nano /var/www/html/test.php
```

Upisati u editoru sledeću test PHP skriptu:

```
<?php phpinfo(); ?>
```

I sačuvati datoteku.

Zatim se u *web* pretraživaču na *Raspbian linux*-u na RPi upiše `localhost/test.php` i tada bi trebalo da se pojavi stranica u kojoj su ispisani podaci o *web*-serveru. Takođe se može otvoriti ova datoteka i iz *web* pretraživača bilo kog drugog računara u mreži upisivanjem IP adrese RPi. Nakon provere poželjno je obrisati datoteku `test.php`, jer nije potrebno da se nalazi na *web*-serveru.

15.3. *PHPlot* biblioteka

Potrebno je skinuti sa interneta *PHPlot* biblioteku [36], i potom raspakovati u željeni direktorijum (npr. `/home/Pi`). Potrebno je zapamtiti putanju do tog direktorijuma. Uputstvo za *PHPlot* biblioteku može se naći na [37].

15.4. Primeri

15.4.1. Primeri iscrtavanja grafika

Uneti sledeći kod u tekst editoru i sačuvati ga kao g1.php.

```
<?php
//Include the code
include('<putanja do direktorijuma>/phplot.php');

//Define the object
$graph =& new PHPlot();

//Define some data
$example_data = array(
    array('a',3),
    array('b',5),
    array('c',7),
    array('d',8),
    array('e',2),
    array('f',6),
    array('g',7)
);
$graph->SetDataValues($example_data);

//Draw it
$graph->DrawGraph();
?>
```

Nakon toga u terminalu ukucati: php g1.php > g1.png

Pogledati datoteku g1.png koja je dobijena u folderu.

Uneti sledeći kod u tekst editoru i sačuvati ga kao g2.php

```
<?php
//phplot.php
include('<putanja do direktorijuma>/phplot.php');

//definicija objekta
$graph =& new PHPlot(300,250);

// naslovi
$graph->SetTitle("Title\n\rSubtitle");
$graph->SetXTitle('X data');
$graph->SetYTitle('Y data');
```

```

//podaci
$example_data = array(
    array('a',3,4,2),
    array('b',5,'',1), // ovde namerno nedostaje
                        // jedna tačka
    array('c',7,2,6),
    array('d',8,1,4),
    array('e',2,4,6),
    array('f',6,4,5),
    array('g',7,2,3)
);
$graph->SetDataValues($example_data);

//Draw it
$graph->DrawGraph();
?>

```

Nakon toga u terminalu ukucati: `php g2.php > g2.png`

Pogledati datoteku `g2.png` koja je dobijena u folderu.

15.4.2. Primer iscrtavanja grafika u internet pretraživaču

U tekst editoru kopirati sledeći kod i snimiti ga pod imenom `g3.html`

```

<html>
<head>
<title></title>
</head>
<body>

<img src= 'g1.png' >

</body>
</html>

```

Kao rezultat bi trebalo da bude u internet pretraživaču iscrtan grafik iz prvog primera iscrtavanja grafika.

15.4.3. Primer osvežavanja html stranice

Sledeći kod omogućava da se *html* stranica koja poziva *php* skriptu osvežava na svakih 5 sekundi. Datoteku snimiti sa ekstenzijom *php*.

```
<html><head>
<title>PHP refresh</title>

<meta http-equiv="refresh" content="5">
</head></html>
<?php
echo "Hello world!";
echo "<br>";
echo 'Time: ' . date('H:i:s');
?>
```

Isto ovo se može uraditi i korišćenjem PHP metode što se smatra boljim načinom:

```
<?php
header("Refresh:5");
echo "Hello world!";
echo "<br>";
echo 'Time: ' . date('H:i:s');
?>
```

15.4.4. Toggle pina

Ako želimo da menjamo stanje nekog pina sa visokog na niski nivo i obrnuto koristeći PHP možemo iskoristiti HTML Forme. Na primer, možemo koristiti dugme da menjemo GPIO pin 7, a u tekst polju ispisujemo ON ili OFF kao stanje izlaza. Ovakva forma je prikazana na slici 15.1.

RaspberryPi GPIO Toggler

Pin 7:

Slika 15.1. Izgled stranice u internet pretraživaču

```
<!DOCTYPE html>
<html>
<head><title>GPIO Tog</title></head>
<h1>RaspberryPi GPIO Toggler</h1>
<body>
<form method="POST" action="">
  <input type="submit"
name="btnToggle" value="TOGGLE!"><br><br>
</form>

<?php
# set pin 7 mode to output
exec("sudo gpio mode 7 out");
if ( isset( $_POST["btnToggle"] ) ) {
  # GPIO pin 7 toggles state
  # If error codes are not 0, you have a problem
  # read pin level
  exec ("sudo gpio read 7", $rReturn, $errcode );
  if ( $rReturn[0] == "0" ) {
    $rReturn[0] = "1";
  }
  else {
    $rReturn[0] = "0";
  }
  exec("sudo gpio write 7 ".$rReturn[0] );
}
# read pin
exec ("sudo gpio read 7", $rReturn, $errcode );
if ( $rReturn[0] == "0" ) { ?>
  Pin 7: <input type="text" name="Pin7" value=<?php echo
"OFF";?> />
<?php
}
else { ?>
  Pin 7: <input type="text" name="Pin7" value=<?php echo
"ON";?> />
  <?php
}
?>
</body></html>
```

Kada se datoteka otvori u internet pretraživaču, postavlja se pin 7 kao izlazni i čita se status pina što određuje da li treba ispisati ON ili OFF. Pritiskom na dugme možemo menjati stanje. Ovo radi lepo osim za slučaj ako pin 7 još neko drugi kontroliše. Tada je potrebno koristiti periodično osvežavanje objašnjeno ranije.

15.5. Zadaci

15.5.1. Zadatak 1

Potrebno je na RPi dodati sat realnog vremena (ili koristiti DVK512) i temperaturni senzor DS18B20 i potom napisati program u C-u koji će na svakih 5 sekundi upisivati podatke u log datoteku. Grafički prikaz realizovati koristeći *PHPlot* biblioteku tako što će se prikazivati u internet pretraživaču promena temperature u zavisnosti od vremena čitanjem podataka iz datoteke. Koristiti *wiringPi* biblioteku.

15.5.2. Zadatak 2

Ponoviti prethodni zadatak ali bez upisa u datoteku već dinamičkim osvežavanjem i direktnim čitanjem.