

Dokumentáció

Ötlet:

Ötletem, egy az okos otthon témából kiragadott darab, interneten keresztül kapcsolgatható lámpa. A megvalósítás során azonban akadt egy kis probléma az internetszolgáltatómmal, így sajnos csak helyi hálózaton belül működik. Egyrészt egy webes felületen keresztül, másrészt pedig egy saját androidos alkalmazás segítségével kapcsolhatjuk ki/be a lámpát.

A megvalósítás:

- Először is le kellett tölteni a PUTTY nevű programot, melynek segítségével SSH-n keresztül elérhetővé vált a Raspberry.
- Ezután a grafikus felület érdekében az xtightvncserver-t telepítettem.
- Webszervernek az Apache-ot választottam, adatbázis-kezelőnek pedig a MySQL-t, illetve hozzá a phpMyAdmin-t.
- Ezek telepítése után létrehoztam egy adatbázist a felhasználóknak, akik hozzáférhetnek a lámpához.
- Ezután jött a PHP fájlok elkészítése:
2 fájl van, az egyik csak egy szimpla bejelentkező oldal (index.php), a másikon pedig bejelentkezés után kapcsolgatható a lámpa. (Itt egy kis gond akadt a scriptek futtatásával.)
- Miután tudtam már, hogy a Raspberryn melyik pin legyen a relé kapcsolgató, és melyik legyen az „érzékelő” (később bővebben), megterveztem a kapcsolást, majd megépítettem.
- Végül pedig miután a kapcsolás és a webszerver is működött, megírtam hozzá az androidos alkalmazást.

Források:

- Putty letöltés - <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>
- xtightvncserver letöltés, telepítés - <https://www.raspberrypi.org/documentation/remote-access/vnc/>
- Apache, PHP, MySQL telepítés - <http://www.instructables.com/id/Raspberry-Pi-Web-Server/>
- phpMyAdmin telepítés - <https://www.stewright.me/2012/09/tutorial-install-phpmyadmin-on-your-raspberry-pi/>
- HTML - <http://pcsysak.uw.hu/html2.html>
- CSS - <http://htmlkezdoknek.blogspot.hu/2011/11/css-alapok.html>
- PHP - <http://php.net/manual/en/>
- MySQL kezelése PHP-ben - <http://php.net/manual/en/book.mysql.php>
- GPIO kezelése PHP-ben (python scriptekkel):
- Python script futtatása - <http://www.knight-of-pi.org/apache-web-server-with-cgi-for-python/> (+ addgroup gpio)

- Kimenet kezelése - <http://www.raspberry-pi-geek.com/Archive/2014/07/PHP-on-Raspberry-Pi>
- Bemenet kezelése - <https://www.youtube.com/watch?v=lOb4G8Le9fQ>
- Android alapok:
 - http://www.tankonyvtar.hu/en/tartalom/tamop412A/2011-0013_vizvari_programozas_android_i/index.html
 - https://www.youtube.com/playlist?list=PLDLHEyX_XL1MkhJsSB2SCy8lveNP2Gvb2
- Android kliens– webszerver kommunikáció:
 - <https://www.simplifiedcoding.net/android-user-registration-tutorial/>
 - <https://www.youtube.com/watch?v=eldh8l8yPew>

Config.php:

Megjelenés:

```
echo "<body>";  
echo "<link type='text/css' href='./src/style.css' rel='Stylesheet'>";  
echo "</body>";
```

\$_SESSION kezelés inntől kezdve, s a kijelentkezés mevalósítása:

```
session_start();  
if(isset($_POST["logout"])) { if(isset($_SESSION["usr"])) { session_destroy(); header("Location:  
config.php"); } }
```

Ha nincs még bejelentkezve, akkor a következőképpen jelentkezik be:

```
if(!isset($_SESSION["usr"]))  
{  
    if(isset($_POST["name"])) //Ha kapott nevet  
    {  
        $n = $_POST["name"]; //eltároljuk,  
        if(isset($_POST["psw"])) //s ha kapott jelszót  
        {  
            $p = $_POST["psw"]; //akkor azt is eltároljuk,  
            $connect = mysql_connect("localhost","root","91827355"); //s csatlakozhatunk az adatbázishoz.  
            if($connect) //Ha sikerült, akkor  
            {  
                mysql_select_db("users"); //kiválasztjuk az adatbázist,  
                $adatok = mysql_query("select * from `family` where name='$n'"); //lekérjük az adatokat,  
                $sor = mysql_fetch_assoc($adatok); //eltároljuk őket egy tömbben,  
                $dbname = $sor["name"]; //majd kiszedjük változóba a nevet,  
                $dbpsw = $sor["psw"]; //és a jelszót.  
                if(($n == $dbname) && ($n != "") && ($p == $dbpsw) && ($p != "")) //Ha adatok stimmelnek
```

```
{
  echo "Bejelentkezve mint $dbname";
  $_SESSION["usr"] = $dbname; //létrehozzuk a $_SESSION-t, hogy bejelentkeztünk.
  //Ha telefonról küldtünk relé ki/be kapcsolást akkor elvégezzük a műveletet
  if(isset($_POST["rbe"])) { $_SESSION["rele"] = 1; exec("python ./src/be.py"); }
  if(isset($_POST["rki"])) { $_SESSION["rele"] = 0; exec("python ./src/ki.py"); }
  }else echo "Hibas felhasznalonev/jelszo!"; //Ha az adatok nem stimmelnek
  }else echo "Nem lehet csatlakozni az adatbazishoz!"; //Ha nem sikerült csatlakozni
  }else echo "Az oldal hasznalatahoz legyen szives es jelentkezzen be!<br/><a
href='index.php'><button type='button'>Foldalra</button></a>"; //Ha nem vagyunk bejelentkezve
  }else echo "Az oldal hasznalatahoz legyen szives es jelentkezzen be!<br/><a
href='index.php'><button type='button'>Foldalra</button></a>";
}
```

Ha bejelentkezett beolvassa a kapcsoló értékét, inicializálja a relét, ha éppen most jelentkezett be és kiszámolja a lámpaállapotát (A 2 kapcsoló kizáróvagy kapcsolata):

```
if(isset($_SESSION["usr"]))
{
  //Kapcsoló állapotának beolvasása
  $result = exec("python ./src/olvas.py");
  $resultdata = json_decode($result, true);
  $kapcsolo[0] = $resultdata["kapcsolo"];
  //Hibakezelés
  if(isset($kapcsolo))
  {
    if(isset($kapcsolo[0]))
      echo "Kapcsoló=".$kapcsolo[0];
    else
    {
      echo "'kapcsoló' is nem tomb!!<br />";
    }
  }
  else
  {
    echo "'kapcsoló' nem letezik!<br />";
  }
  echo "<br />";

  //Relé inicializálása
```

```
if(!isset($_SESSION["rele"]))
{
    $_SESSION["rele"] = 0;
    exec("python ./src/ki.py");
}

echo "Rele=".$_SESSION["rele"];
echo "<br/>";

//Lámpa állapotának megállapítása
$power = (($kapcsoló[0]) xor ($_SESSION["rele"]));
if($power != 1)
{
    $power = 0;
}
echo "Power=".$power;
```

A lámpa állapotától függő form elkészítése:

```
echo "<form action='config.php' method='post'>";
if($power == 1)
{
    echo "<input type='submit' name='ki' value='Kikapcsolás'><br/>";
}
else if($power == 0)
{
    echo "<input type='submit' name='be' value='Bekapcsolás'><br/>";
}
echo "<input type='submit' name='logout' value='Kijelentkezés'>";
echo "</form>";
```

A forrás többi része pedig a lámpa kapcsolgatása a relé és a lámpa állapotától függően.

Az androidos alkalmazás:

Több részből épül fel:

- Az activity grafikus felületéből
- Az activity „tevékenységéből” - főosztály
- Háttérben dolgozó osztályból

Grafikus felület:

Különböző elrendezéskezelőkből illetve beviteli mezőkből és gombokból áll össze a következőképpen:

- RelativeLayout //Viszonyításos elrendezéskezelő, gyakorlatilag bárhova lerakhatunk bármit
- scrollView //Ezen belül egy görgethető rész
- LinearLayout (vertical) //Ezen belül függőlegesen „pakoló” elrendezéskezelő
- textView //Ez a felhasználónév felirat
- UsernameID_et //Felhasználónév beviteli mező
- textView2 //Jelszó felirat
- PasswordID_et //Jelszó beviteli mező
- textView3 //Szerver IP felirat
- IPID_et //Szerver IP beviteli mező
- LinearLayout (horizontal) //Vízszintesen „pakoló” elrendezéskezelő
- LoginID_on_btn //Relé bekapcsoló gomb
- LoginID_off_btn //Relé kikapcsoló gomb
- RelayID_tv //Relé állapotjelző felirat
- (colors.xml – színek beállítása)
- (strings.xml – feliratok beállítása)
- (AndroidManifest.xml – Alkalmazás alapbeállításai, pl.: jogok)

A főosztály:

A java nyelv csomagokban dolgozik:

```
package com.dokitech.netlampclient;
```

Szükséges osztályok importálása:

```
import...
```

A főosztály:

```
public class MainActivity extends AppCompatActivity {
```

Ezen „változókkó” segítségével kapcsoljuk össze az activity grafikus felületét a forrással:

```
EditText Username_et, Password_et, IP_et;  
TextView Relay_tv;
```

Itt tesszük láthatóvá az activityt és összekapcsoljuk a grafikus felületet a forráskóddal:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Username_et = (EditText) findViewById(R.id.UsernameID_et);
    Password_et = (EditText) findViewById(R.id.PasswordID_et);
    IP_et = (EditText) findViewById(R.id.IPID_et);
    Relay_tv = (TextView) findViewById(R.id.RelayID_tv);
}
```

Attó függően, hogy melyik gombot nyomtuk le meghívódik a háttérben dolgozó osztály, ami a kapcsolgatást végzi:

```
public void onLogin(View view){
    int gomb = view.getId(); //Melyik gomb?
    String powerState = ""; //Ki/be?
    switch (gomb){
        case R.id.LoginID_on_btn:
            powerState = "rbe"; //Be
            break;
        case R.id.LoginID_off_btn:
            powerState = "rki"; //Ki
            break;
    }
    String username = Username_et.getText().toString(); //Felhasználónév a bejelentkezéshez
    String password = Password_et.getText().toString(); //Jelszó a bejelentkezéshez
    String type = "login"; //Bejelentkezést akarunk végrehajtani
    String ip = IP_et.getText().toString(); //Szerver IP címe ahova bejelentkezünk
    BackgroundWorker backgroundWorker = new BackgroundWorker(this); //Háttérosztály
    példányosítása
    backgroundWorker.execute(type, ip, username, password, powerState); //Háttérfolyamat
    (bejelentkezés és ki/be kapcsolás) elindítása
    if(powerState.equals("rbe")) Relay_tv.setText("Relé: BE"); //Visszajelzések
    if(powerState.equals("rki")) Relay_tv.setText("Relé: KI");
}
```

A háttérben dolgozó osztály:

```
public class BackgroundWorker extends AsyncTask<String, Void, String>{
    Context context; //A konstruktor beállítja a contextet, azaz, hogy mivel van összekapcsolva ez az osztály.
    BackgroundWorker(Context ctx){
        context = ctx;
    }

    @Override
    protected String doInBackground(String... params) { //Az execute hatására fut le
        String type = params[0]; //Paraméterben megkapta a tpiust (bejelentkezés)
        String ipaddress = params[1]; //A szerver IP-t
        String login_url = "http://" + ipaddress + "/config.php"; //Melyet a bejelentkező URL-hez használ fel.

        if(type.equals("login")){ //Ha a típus bejelentkezés („login”)
            try {
                String user = params[2]; // Átvesszük a felhasználónév,
                String pass = params[3]; //A jelszó
                String powerState = params[4]; //És a ki/be kapcsolás paramétert
                URL url = new URL(login_url); //Létrehozuk a csatlakozáshoz szükséges URL-t
                //Létrehozuk a kapcsolatot
                HttpURLConnection httpURLConnection = (HttpURLConnection) url.openConnection();
                if(httpURLConnection!=null) {
                    httpURLConnection.setRequestMethod("POST"); //POST-tal kommunikálunk
                    httpURLConnection.setDoOutput(true); //A ki- és a
                    httpURLConnection.setDoInput(true); //bemenetet engedélyezzük

                    OutputStream outputStream = httpURLConnection.getOutputStream(); //Kimenet
                    BufferedWriter bufferedWriter = new BufferedWriter(new
                    OutputStreamWriter(outputStream, "UTF-8")); //Kimenethez buffer, ebbe kerül az üzenet
                    String post_data = URLEncoder.encode("name", "UTF-8") + "=" +
                    URLEncoder.encode(user, "UTF-8") + "&"
                        + URLEncoder.encode("psw", "UTF-8") + "=" + URLEncoder.encode(pass, "UTF-8") + "&" + URLEncoder.encode(powerState, "UTF-8"); //Ez a hosszú az üzenet (bejelentkezik
                    //illetve elküldi, hogy ki vagy bekapcsolni szeretnénk a relét
                    bufferedWriter.write(post_data); //Elküldi a szervernek
                    bufferedWriter.flush(); //Felszabadítja illetve bezárja a kimenetet
                    bufferedWriter.close();
                }
            }
        }
    }
}
```

```
        outputStream.close();
        //Ez pedig a bemenet, amit a szerver üzen vissza (echo-k és printf-ek), de ez nem kell
csak majd a további fejlesztéshez
        InputStream inputStream = httpURLConnection.getInputStream();
        BufferedReader bufferedReader = new BufferedReader(new
InputStreamReader(inputStream, "iso-8859-1"));
        String result = "";
        String line;
        while ((line = bufferedReader.readLine()) != null) {
            result = line;
        }
        bufferedReader.close();
        inputStream.close();
        httpURLConnection.disconnect();
        return result;
    }
    //Hibakezelés
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    }
    return null;
}
}
```