

Mario Šimunić
2401028282

Marko Kovačić
0036465943

SEMINARSKI RAD - SPVP



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za električne sustave i obradbu informacija
Sveučilište u Zagrebu

Pametna rasvjeta



- △ Projekt „Pametna kuća“
- △ Pametna rasvijeta
- △ Aplikacija za udaljeno upravljanje
- △ Onion Omega, node.js, Arduino
- △ Blynk, mobilna aplikacija

Sažetak

Pametna rasvjeta je sustav koji korisniku omogućuje udaljeno upravljanje rasvjetnim tijelom prema zadanim parametrima. Dosadašnja rješenja masovno koriste rudimentarnu radio-komunikaciju ili vrlo jednostavne radio-komunikacijske protokole koji su slabi u smislu informacijske sigurnosti. Osnovna ideja ovog rješenja je sustav za upravljanje nekim vanjskim uređajem koji bi trebao podržavati kompleksne komunikacijske protokole i razinu sigurnosti tipičnu za uređaje koji se spajaju na Internet. Sustav mora biti moguće izgraditi u kratkom vremenu i za relativno nizak trošak komponenti.

Servis Blynk omogućuje brzu izgradnju mobilne aplikacije i jednostavno povezivanje uređaja na server kroz sigurnu TCP/IP konekciju. Postoje relativno dobri primjeri sličnih rješenja za platformu Arduino. Nedostatak ovakvog pristupa na platformi Onion Omega je potreba za naprednim korištenjem Linux operacijskog sustava, programiranje uređaja u skriptnom jeziku node.js i nedostatak primjera za node.js.

Koristi od ovakvog sustava mogu imati rezidenti koji aktivno koriste pametni telefon kao sučelje prema internetu stvari. Upravljanje rasvjetom pomoći Onion Omega pločice i Blynk aplikacije može biti osobito zanimljivo korisnicima vještim u programiranju u jeziku JavaScript jer im omogućuje daljnji razvoj funkcionalnosti. Za sve ostale manje zahtjevne korisnike pametna rasvjeta može biti isporučena takva kakav jest, odnosno kao sustav koji omogućuje udaljeno paljenje i gašenje rasvjetnog tijela te uvid u trenutno stanje rasvjetnog tijela s mogućnosti za automatsko paljenje/gašenje rasvjete u zadano vrijeme ili na neki vanjski podražaj poput prisutnosti u prostoru.

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sustave i obradbu informacija, Fakulteta elektrotehnike i računarstva, Sveučilišta u Zagrebu.

Sadržaj ovog rada može se slobodno koristiti, umnožavati i distribuirati djelomično ili u cijelosti, uz uvjet da je uvijek naveden izvor dokumenta i autor, te da se time ne ostvaruje materijalna korist, a rezultirajuće djelo daje na korištenje pod istim ili sličnim ovakvim uvjetima.

Sadržaj

1. UVOD.....	4
2. OPIS SUSTAVA (MARKO KOVAČIĆ).....	5
2.1 Shema sustava.....	5
3. BLYNK APLIKACIJA	6
4. ARDUINO.....	6
5. PROTOBOARD I SVJETLEĆA DIODA.....	7
6. REALIZACIJA CIJELOG SUSTAVA.....	8
7. PROGRAMSKA REALIZACIJA SUSTAVA.....	10
8. OPIS SUSTAVA (MARIO ŠIMUNIĆ).....	11
8.1 Shema sustava.....	11
9. BLYNK APLIKACIJA.....	12
10. ONION OMEGA2 PLATFORMA.....	12
11. PIR SENZOR.....	13
12. SVJETLEĆA TRAKA.....	14
13 PROGRAMSKA REALIZACIJA SUSTAVA.....	15
14. REALIZACIJA CIJELOG SUSTAVA.....	18
15. ZAKLJUČAK.....	19
16. LITERATURA.....	20
17. POJMOVNIK.....	21

1. Uvod

Projekt Pametna rasvjeta omogućavati će ljudima upravljanje rasvjetnim objetima na velikim udaljenostima pomoću vrlo jednostavne aplikacije Blynk. Radom ovog projekta ponuđena su veoma jednostavna i sigurna rješenja toga problema dok dosadašnja istraživanja nisu omogućavala korisnicima sigurnu komunikaciju između njih i rasvjetnog tijela .

Napredne mogućnosti uspješne izvedbe ovog projekta pružati će korisnicima upravljanje rasvjetom u fiksno vrijeme ili služiti kao sustav detekcije prisutnosti objekta u prostoru.

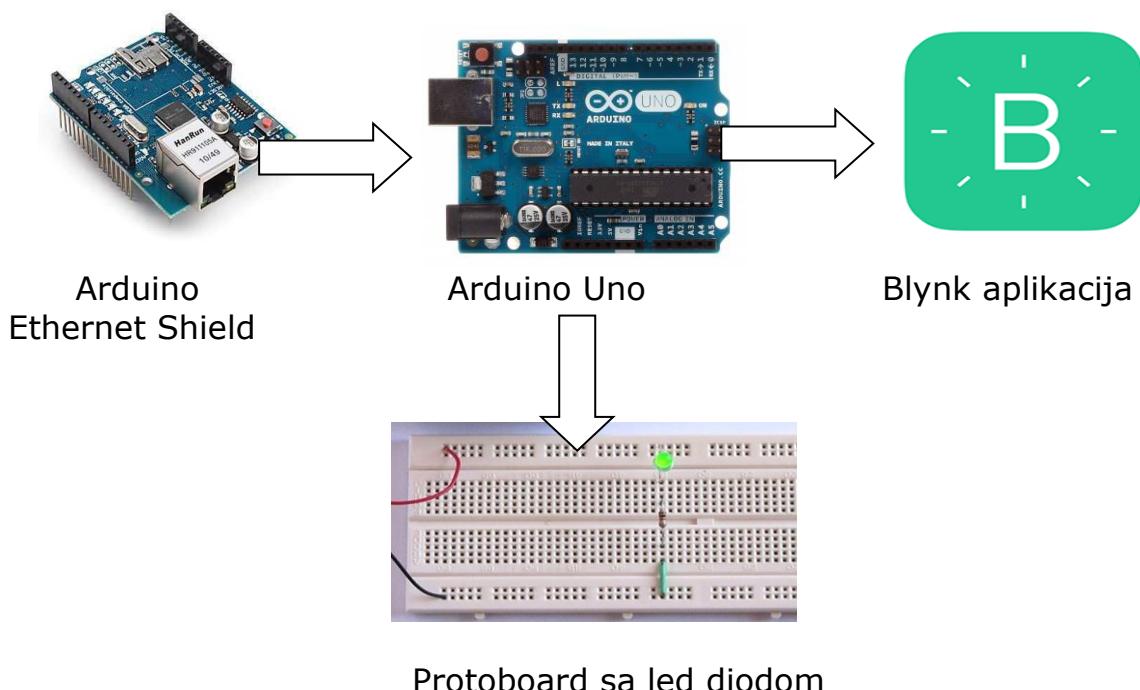
U ovome radu napravljena je prvo izvedba sa Oniom Omega pločicom koja komunicira preko Blynk servisa no tokom rada projekta ustanovljen je nedostatak izvedbe zbog potrebe za naprednim znanjem o Linux operacijskom sustavu i skriptnim jezikom node.js.

Sljedeća izvedba projekta s Arduino Uno pločicom je pokazala dobra rješenja za problem pametne rasvjete preko Blynk servisa.

2. Opis sustava (Marko Kovačić)

Arduino Uno je spojen preko USB kabela na računalo, dok je Arduino Ethernet Shield spojen mrežnim kabelom na lokalnu mrežu preko koje se obavlja komunikacija. Pomoću Blynk aplikacije na pametnom telefonu napravljen je gumb imena BUTTON koji ima funkciju ON/OFF pritiskom na njega. Nakon uspostave konekcije Arduina i Blynk aplikacije signalizira se u aplikaciji je ili nije uspješno spojena na ethernet lokalnu mrežu. Nakon što je signal na Blynk aplikaciji potvrđio uspostavu veze s Arduinom gumbom ON/OFF uključujemo/isključujemo led diodu te se time izvršava zadatak koji je postaljen ovim dijelom projekta .

2.1 Shema sustava



Slika 1. Grafički opis izvedenog sustava

3. Blynk aplikacija

Blynk je aplikacija koja omogućuje povezivanje između fizičke pločice (Arduino Uno i ostale vrste Arduina , Onion Omega, Croduino NOVA) i mobilne aplikacije. Putem mobilne aplikacije tada može se upravljati pločicom (npr. kontrolirati rasvjetu) ili očitavati podatke s nje (npr. temperaturu). Postoji mogućnost upravljanja motorom koji se spojen na pločicu putem Blynk aplikacije no to nije predviđeno radom ovog dijela projekta.

4. Arduino

Arduino je ime za otvorenu računalnu i softversku platformu koja omogućava dizajnerima i konstruktorima stvaranje uređaja i naprava koje omogućuju spajanje računala s fizičkim svijetom. Arduino je stvorila talijanska tvrtka SmartProject 2005. rabeći 8-bitne mikrokontrolere Atmel AVR, da bi stvorili jednostavnu, malu i jeftinu platformu s kojom bi mogli lakše povezivati računala s fizičkim svijetom. Dizajneri su izabrali ime Arduino po imenu kafića u kojem su se sastajali kada su stvarali projekt.

Arduino Uno ima 14 digitalnih I/O pinova, 6 analognih ulaza i kristalni oscialtor od 16 mHz, Usb vezu ,tipku za reset i napajanje.

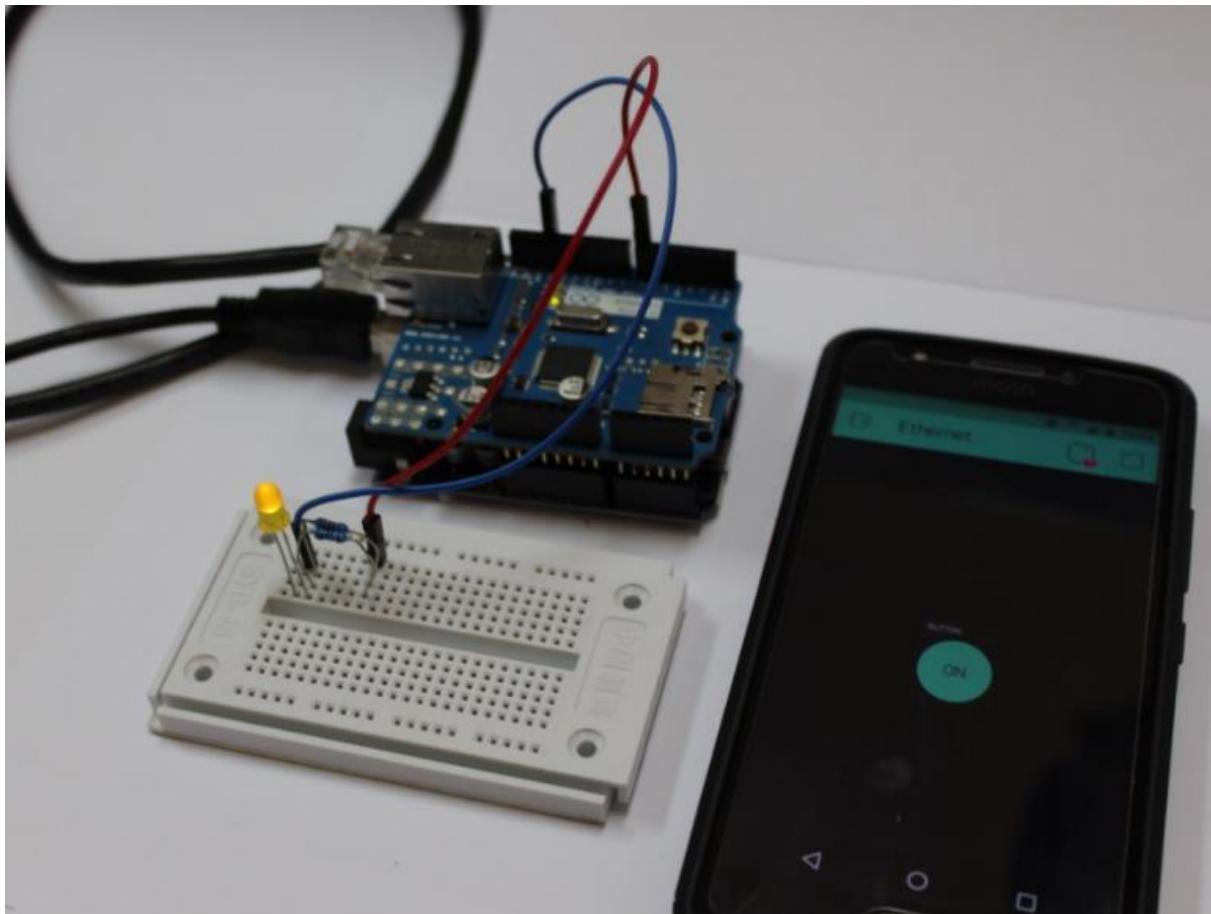
Arduino Ethernet Shiled omogućuje nekoj od Arduino pločica spajanje na lokalnu mrežu. Temelji se na Wiznet W5100 Ethernet čipu. Omogućuje čak i mikro SD kartice.

5. Protoboard i svjetleća dioda

Protoborad je dio sustava projekta na kojoj se realizira proces na kojem je napravljena električna mreža svjetleće diode i spoja dva paralelna otpornika od 470 Ohma. Light emitting diode tkz. svjetleća dioda je komponenta pomoću koje signaliziramo da li je rasvjeta upaljena ili nije tj. signalizira da li sve radi kako treba.

Pmjenuje se najčešće kao indikator na komandnim i signalnim pločama uređaja i strojeva ili kao alfanumerički pokazivač na zaslonima džepnih kalkulatora, zatim za ukrasno osvjetljenje te u industriji zabave, za signalnu rasvjetu za bicikle, automobilска svjetla, kao dijelovi za daljinski prijenos signala u upravljačkim krugovima, kod televizorskih daljinskih upravljača. Veliko područje primjene imaju i u optičkim komunikacijama, gdje služe za prijenos podataka na kraće udaljenosti multimodnim optičkim vlaknom.

6. Realizacija cijelog sustava



Slika 2. Blynk aplikacija i Arduino sustav

Nabavom potrebnih komponenti počinje se realizacija projekta. Prvo se spaja Arduino Uno sa Arduino Ethernet Shieldom. Pri tome Arduino Uno spaja se USB kabelom na računalo dok Arduino Ethernet Shield spajamo kabelom na lokalnu mrežu . Na protoboard spajamo led diodu i dva paralno spojena otpornika radi njene strujne zaštite te s dvije žice preko protoboarda dovodimo na digitalni pin 7 i pin GND na Arduino Ethernet Shield.

Na pametnom telefonu u Blynk aplikaciji napravimo projekt zvan Ethernet te u njemu dodamo gumb/button pomoću kojeg ćemo pokretati led

diodu. Korak prije toga isporgamira se dio koda za Arduino koji pokreće ledicu. Nakon provjere ispravnosti koda možemo pokretati Blynk aplikaciju pomoću tipke BUTTON. Ako je sve u redu led dioda će se paliti /gasiti na dodir tipke BUTTON. Time je ovaj veoma jednostavan dio projekta kompletiran.

7. Programska realizacija sustava

```
#define BLYNK_PRINT Serial

#include <SPI.h>
#include <Ethernet.h>
#include <BlynkSimpleEthernet.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "7d2cd570d5764097b62a2514d82daabe";

#define W5100_CS 10
#define SDCARD_CS 4

const int ledPin = 7; // LED pin
int ledState = HIGH; // LED inicijalno stanje

// Postavimo LED u stanje ON
//digitalWrite(ledPin, ledState);

//BLYNK_WRITE(V0) {
//    ledState = param.asInt();
//    digitalWrite(ledPin, ledState);
//}

void setup()
{
    //SEtup LED pin
    pinMode(ledPin, OUTPUT);
    digitalWrite(ledPin, ledState);

    // Debug console
    Serial.begin(9600);

    pinMode(SDCARD_CS, OUTPUT);
    digitalWrite(SDCARD_CS, HIGH); // Deselect the SD card

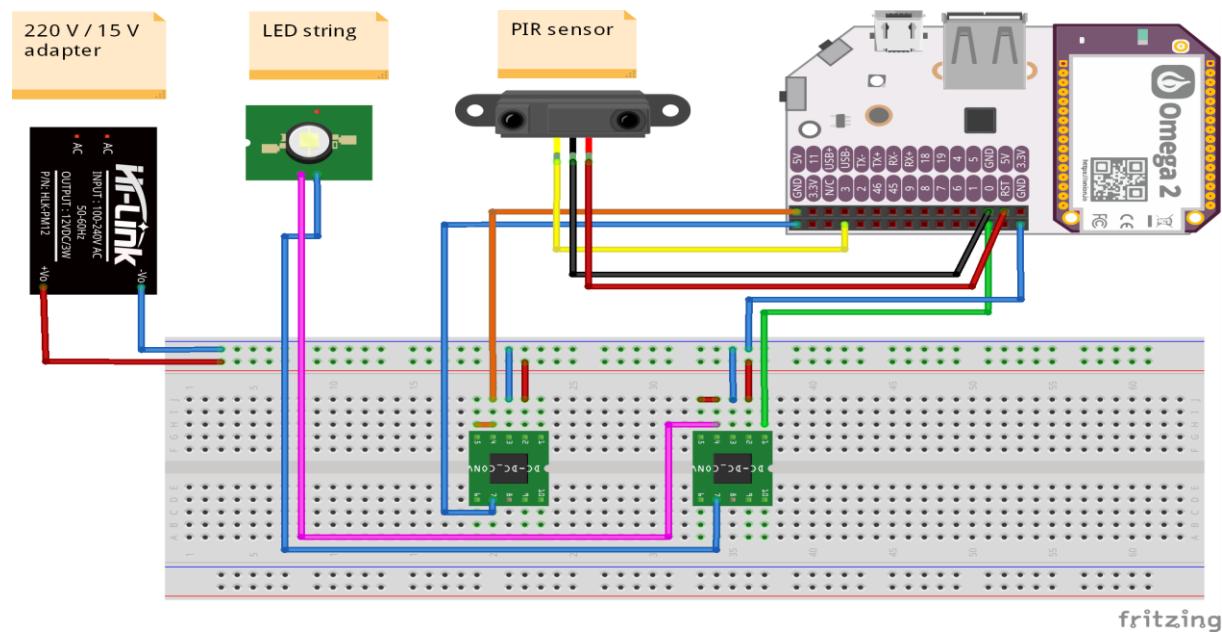
    Blynk.begin(auth);
    // You can also specify server:
    //Blynk.begin(auth, "blynk-cloud.com", 80);
    //Blynk.begin(auth, IPAddress(192,168,1,100), 8080);
    // For more options, see Boards_Ethernet/Arduino_Ethernet_Manual
example
}

void loop()
{
    Blynk.run();
}
```

8. Opis sustava (Mario Šimunić)

Onion Omega platforma spojena je preko USB kabela, dok je na njen Ethernet Shield spojen mrežni kabel na lokalnu mrežu preko koje se obavlja komunikacija. Pomoću Blynk aplikacije na pametnom telefonu napravljen je gumb imena BUTTON koji ima funkciju ON/OFF pritiskom na njega. Nakon uspostave konekcije Onion Omega pločice i Blynk aplikacije signalizira se u aplikaciji je ili nije uspješno spojena na ethernet lokalnu mrežu. Nakon što je signal na Blynk aplikaciji potvrđio uspostavu veze s Onion Omega gumbom ON/OFF uključujemo/isključujemo led diodu te se time izvršava zadatak koji je postaljen ovim dijelom projekta. Dodatni dio koji je napravljen u ovom složenijem dijelu projekta je detekcija pomaka pomoću PIR senzora.

8.1 Shema sustava



Slika 3. Grafički opis izvedenog sustava kao Fritzing shema u attach alatu

9. Blynk aplikacija

Tijekom izrade aplikacije za Onion Omega pločicu koja se povezuje sa Blynk serverom bilo je mnogo složenija realizacija nego sa Arduino platformom. Na to je najviše utjecalo dobro poznavanje Linux operacijskog sustava i programiranje u jeziku JavaScript.

Blynk je aplikacija koja omogućuje povezivanje između fizičke pločice, Onion Omega i mobilne aplikacije. Putem mobilne aplikacije tada može se upravljati pločicom (npr. kontrolirati rasvjetu) ili očitavati podatke s nje (npr. temperaturu).

10. Onion Omega2 platforma

Onion Omega2 je novija generacija platforme Omega koje proizvodi kanadska kompanija iz Toronto zvana Onion.

Izvorni Omega temelji se na Qualcomm Atheros AR9331 (MIPS arhitekturi) SoC koje pokreće Linux operacijski sustav dizajniran za uradbene računalne sustave. Kompanija je prekinula proizvoditi Omega platformu i zamijenila je s nasljednikom Omega2 koristeći drugačiji SoC čip Mediatek MT7688 kojim je smanjena cijena pločice.

Omega2 dolazi u dvije verzije. Prva verzija je osnovna Omega2, a druga boljih specifikacija je Omega2 Plus. Omega2 CPU se temelji na MIPS arhitekturi koja radi brzinom od 580 MHz opremljenom s 64 MB RAM-a i 16 MB flash memorije.

11. PIR senzor

PIR (engl. Passive InfraRed) je senzor koji nam omogućava detekciju pokreta ljudskog tijela u području njegovog doseg-a. PIR senzori su lako dostupni za kupnju, imaju dug vijek trajanja . Ove specifikacije omogućavaju PIR senzorima da se koriste za detekciju ljudske ili druge prisutnosti ili u upotrebi alarma.



Slika 4. PIR senzor

Osnovni dio PIR senzora je piroelektrični element koji omogućuje detekciju infracrvenog zračenja. Princip rada temelji se na detekciji promjene primljenog infracrvenog zračenja na jednom od dva dijela koja čine piroelektrični element. Ukoliko jedna od njih primi više zračenja od druge, senzor će reagirati na tu promjenu stoga je moguće detektirati pokrete živih bića jer ona zrače.

12. Svjetleća traka



Slika 5. Svjetleća traka

Prikazana je LED svjetleća traka koja se koristi za realizaciju rasvjete u drugom dijelu projekta. Napravljena je od niza led dioda za malim padovima napona između njih jer su spojeni otpornici kojima se postiže regulacija struje .

13. Programska realizacija sustava

```
/* GLOBAL VARIABLES */
var child, hostname, text;
var apcliConnectionStatus;
var timeString;
var alarmStartSeconds;
var alarmStopSeconds;
var alarmDays, alarmDaysLength;
var alarmEvent, alarmState;
var alarmEnabled;

var lightState, lightDuty;
var lightChanged;

var alarmActivationTime;
var alarmActionDelay;           // seconds
var alarmDelayEnabled;

var inputAState, inputBState;
var inputAChange, inputBChange;

var blinkingTaskID;
var alarmTaskID;
var alarmTaskEnabled;
var iftttTaskID;
var timeCnt;

var presenceState, inBedState, alarmActionEvent;
var blinkingEnabled, timeoutEnable;
var inBedTriggerEnabled, presenceTriggerEnabled;

var digpotValue;    // present value of digital potentiometer

/* PHERIPHERALS*/
var gpioLight = new Gpio(0, 'out');
var CS = new Gpio(6, 'high');    //!< 'high' = output with logic "1" as
initial state
var UD = new Gpio(18, 'high');
var INC = new Gpio(19, 'high');  //!< active on falling edge
var gpioInputA = new Gpio(3, 'in', 'rising', {debounceTimeout: 10});
    // PIR sensor
var gpioInputB = new Gpio(2, 'in', 'both', {debounceTimeout: 10});
    // Something else
```

```

/* BLYNK */
// Blynk
var blynk = new BlynkLib.Blynk(auth);
var buttonLight = new blynk.VirtualPin(0); //!< Blynk Button for
Light
var sliderLight = new blynk.VirtualPin(1); //!< Blynk Slider for
Light Dimming
//var terminal = new blynk.VirtualPin(3); //!< Terminal widget
var timeInput = new blynk.VirtualPin(5); //!< Blynk Time Input

/* GPIO inputs*/
gpioInputA.watch(function (err, value) {
    inputAState = value;
    inputAChange = 1;
});

gpioInputB.watch(function (err, value) {
    inputBState = value;
    inputBChange = 1;
});

function digpotSet(value){
    var i, diff;
    if (value <= 100 && value >= 0 ) {
        if (value < digpotValue){
            UD.writeSync(0);
            diff = digpotValue - value;
        } else{
            UD.writeSync(1);
            diff = value - digpotValue;
        }
        CS.writeSync(0);
        for (i = 0; i < diff+1; i++){
            INC.writeSync(0);
            a = setTimeout(function(){}{},1);
            INC.writeSync(1);
            clearTimeout(a);
        }
        CS.writeSync(1);
        digpotValue = value;
    }
    UD.writeSync(1);
}

function digpotInit(){
    var i, a;
    CS.writeSync(0); // select digpot
    for(i=0;i<101;i++){
        INC.writeSync(0);
        a = setTimeout(function(){}{},1);
        INC.writeSync(1);
        clearTimeout(a);
    }
    CS.writeSync(1); // select digpot
    digpotValue = i-1;
}

```

```
/***
 * Reads host name
 */
function readHostname(){
    var result = require('child_process').execSync('uci get
system.@system[0].hostname').toString();
    return result;
}

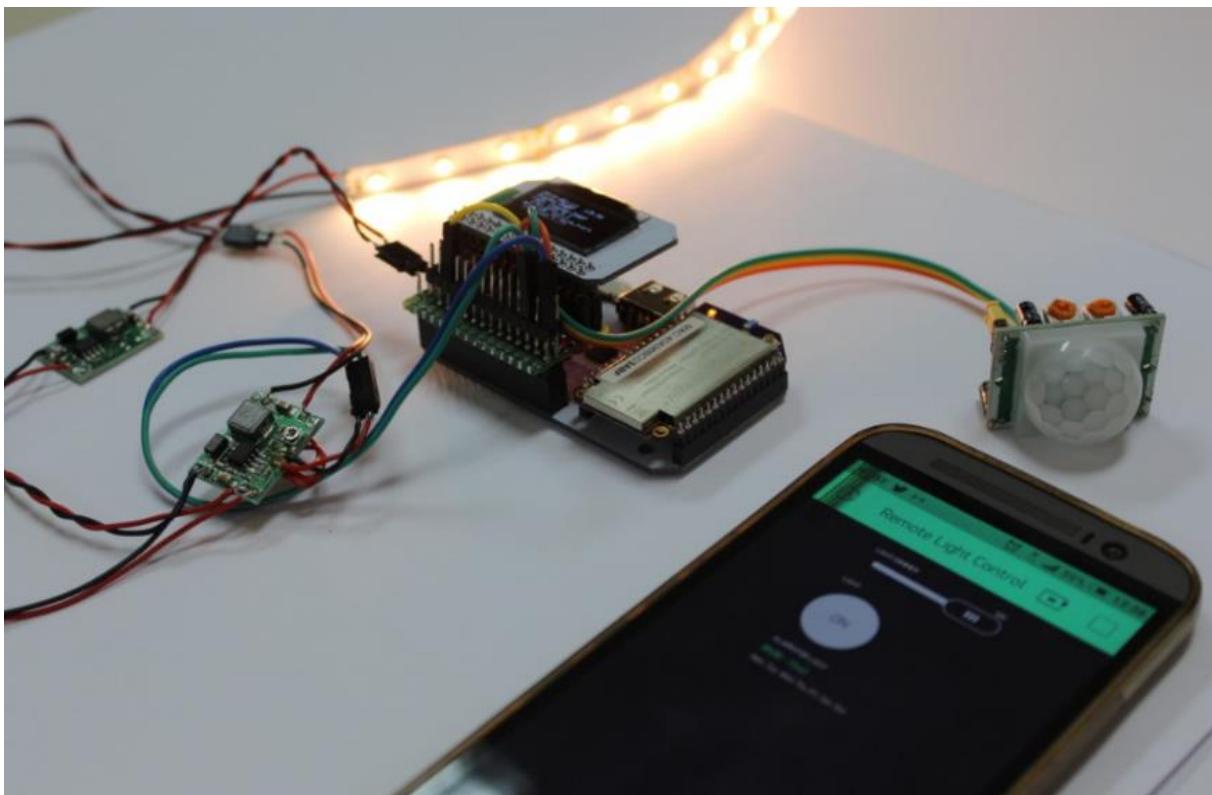
/***
 * Reads status of Omega's acces-point
 */
function readStatusWlan(){
    var result = require('child_process').execSync('uci get
wireless.ap.ifname').toString().trim();
    var tmp = result;
    result = require('child_process').execSync('uci get
wireless.ap.ssid').toString().trim();
    return (tmp + ":" + result);
}

/***
 * Reads IP Address of Omega's acces-point
 */
function readAddressWlan(){
    var result = require('child_process').execSync('ifconfig | grep -i
wlan -A 1 | grep -i inet').toString().trim();
    result = result.substr(result.indexOf(':') + 1);
    var addr = result.substr(0,result.indexOf(' '));
    return (" " + addr);
}

/***
 * Reads SSID of wlan to which Omega is connected
 */
function getSSIDApcli0(){
    var text = require('child_process').execSync('iwconfig apcli0 | grep -i
ssid').toString().trim();
    var start = text.indexOf('ESSID:');
    text = text.substr(start+7);
    var stop = text.indexOf("'");
    text = text.substr(0,stop);
//    fs.writeFileSync('/root/ssid-connected.log', stop.toString());
    if ( stop === 0 ){
        apcli0ConnectionStatus = 0 ;
    } else {
        apcli0ConnectionStatus = 1;
    }
    return ("apcli0: " + text);
}

/***
 * Reads IP Address of wlan to which Omega is connected
 */
return (" " + addr);
}
```

14. Realizacija cijelog sustava



Slika 6. Blynk aplikacija i Onion Omega sustav

Sustav se temelji na Onion Omega pločici koja komunicira sa Blynk serverom i ako je uspješno uspotavljenja komunikacija svjetleće traka se pali/gasi na pritisak gumba BUTTON u Blynk aplikaciji. Na slici se vidi još PIR senzor koji detektira pokret ljudskog tijela.

15. Zaključak

Sustav koji je ostvaren omogućiti će ljudima jednostavnije, sigurnije praćenje detekcije prisutnosti nekog objekta u prostoru ili što je najjednostavnije paljenje/gašenje lampice na neki zadatak koji će korisnici postaviti. Blynk aplikacija omogućava prije svega jednostavnost i sigurnost ovog sustava i svima je dostupna preko pametnih telefona stoga je korisnici mogu veoma brzo i lako instalirati na svojim uređajima i pratiti ili upravljati neki sustavom .

16. Literatura

- [1] Arduino
<https://www.arduino.cc> (2018-06-08)
- [2] Blynk servis
<https://www.blynk.cc> (2018-06-08)
- [3] Onion Omega
<https://www.onion.io> (2018-06-08)

17. Pojmovnik

--	--	--