



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu
Zavod za elektroničke sustave i obradu informacija
Sveučilište u Zagrebu

sustav za provjeru aktivnosti kuhinjskih uređaja



- Δ Dio projekta „Pametna kuća“
- Δ Ugradbeni sustavi, programiranje
- Δ Korištenje wifi modula
- Δ Wifi provjera aktivnosti uređaja

Sažetak

Sustav omogućuje daljinsko provjeravanje aktivnosti toplinskih uređaja u kućanstvu preko web servera, gdje dobivamo uvid u temperaturu oko određenog uređaja te estimaciju aktivnosti. Do sada su razvijene razne utičnice koje na sličan način pristupaju internetu te daju uvid u aktivnost uređaja. Temeljna ideja je olakšati uvid u aktivnost uređaja te riješiti problematiku sigurnosti kuće od požara. Prednosti su brz i jednostavan pristup, a mane potencijalna provala privatnosti i moguća pogrešna estimacija. Koristi mogu imati stanari koji se ne nalaze u blizini kuće, a nemaju vremena vratiti se i osobno provjeriti stanje uređaja.

Sadržaj

1. UVOD.....	3
2. KONTROLER (PIC16F628A)	3
3. WIFI MODUL ESP8266.....	3
4. TEMPERATURNI SENZOR DS18B20	5
5. IDEJA I OPIS SUSTAVA.....	6
5.1. Opis rada sustava	7
6. ZAKLJUČAK.....	11
7. LITERATURA.....	12
8. POJMOVNIK	13

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.

1. Uvod

Ovaj rad prikazuje jednu od mogućih izvedbi provjere aktivnosti toplinskih uređaja te je jedan u nizu projekata „pametna kuća“. U današnje užurbano vrijeme rutinske radnje često nas mogu dovesti do preispitivanja jesmo li ili nismo nešto napravili te su većinom nesvjesne radnje.

Nadzor nad toplinskim uređajima važan je zbog sigurnosti doma i prevencije požara. Pristup internetu u današnje vrijeme moguć je bilo gdje i skoro sa svih uređaja te su nam online podaci uvijek dostup.

Ukoliko se ovaj sustav postavi u kući, korisnik može iz bilo kojeg dijela svijeta imati nadzor nad toplinskim uređajima. To ne moraju biti samo kuhinjski uređaji, već bilo koji uređaj koji emitira toplinu.

U razvoju ovog projekta predložene su dvije inačice, prva koja daje samo uvid u aktivnost i estimira aktivnost uređaja i druga u kojoj bi postojala opcija gašenja i paljenja uređaja, ali zbog sigurnosnih razloga na internetu odabrana je prva.

2. Kontroler (PIC16F628A)

PIC16F628A je 8-bitni mikrokontroler proizvođača Microchip. Dolazi u standardnom DIP18 kućištu. Kontroler je izabran zbog broja izlaznih pinova, unutarnjeg oscilatora te UART sučelja.

3. Wifi modul ESP8266

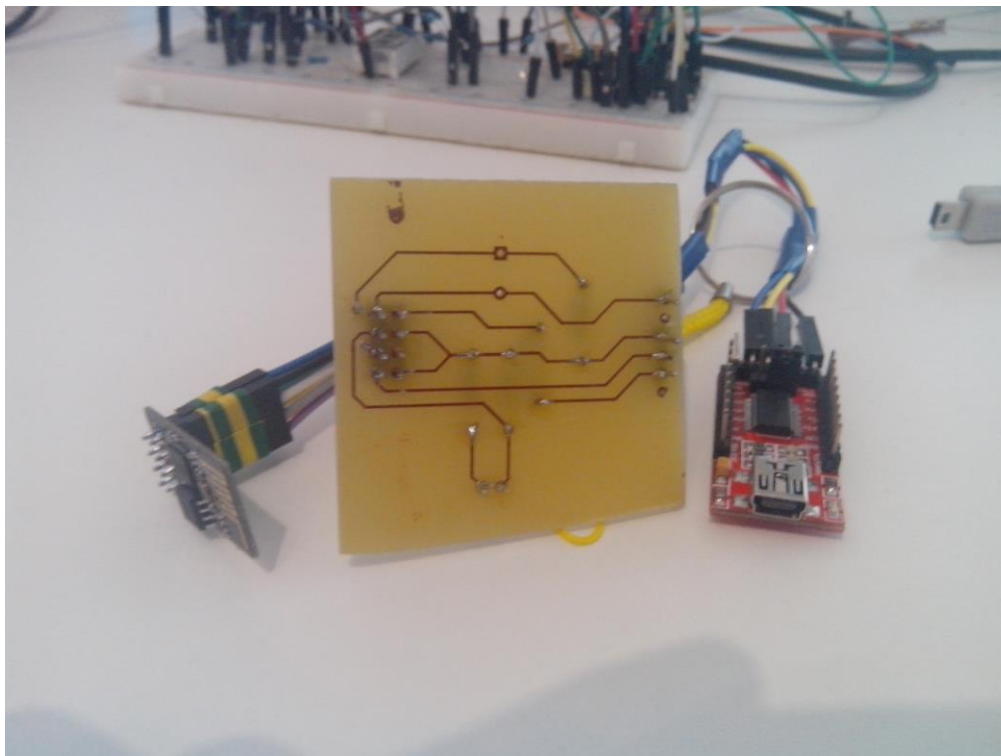
Modul ESP8266 je izabran zbog niske cijene te mogućnosti postavljanja web servera. Prije korištenja modula preporuča se nadograditi firmware. Pri paljenju, status pinova odredit će u koji mod ESP ulazi.

ESP8266 Boot Options

GPIO15 (MTDO)	GPIO0	GPIO2	Mode	Comments
L	H	H	Flash	Boot from SPI Flash (Normal running)
L	L	H	UART	Program via UART (TX/RX)
H	x (not care)	x (not care)	SDIO	Boot from SD-card

Slika 1. Modovi ESP8266 modula

Za potrebe bržeg prototipiranja sustava i konstantnog odspajanja i spajanja modula radi programiranja, napravljen je PCB koji spaja ESP i FTDI232R ploču koja povezuje modul s osobnim računalom.



Slika 2. PCB za spajanje ESP i FTDI232R

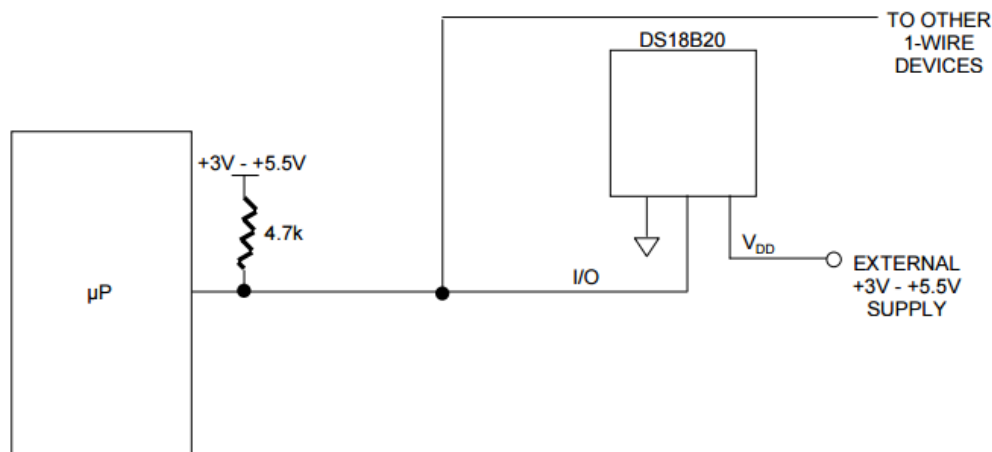
Između FTDI232R i ESP-a nije korišten regulator napona, o kojemu će biti riječi kasnije. Razlog tome je mala izlazna struja koju FTDI232R može dati te korištenje jumper-a na 3.3V neće biti dovoljno za pokretanje ESP-a.

Iako se ne preporuča upotreba napona iznad 3.3V, koristi se jumper na 5V koji može dati dovoljnu snagu ESP-u za normalan rad.

ESP je kodiran u ESPlorer razvojnom okruženju na kojemu je napravljen web server te zadaci koje ESP mora obavljati.

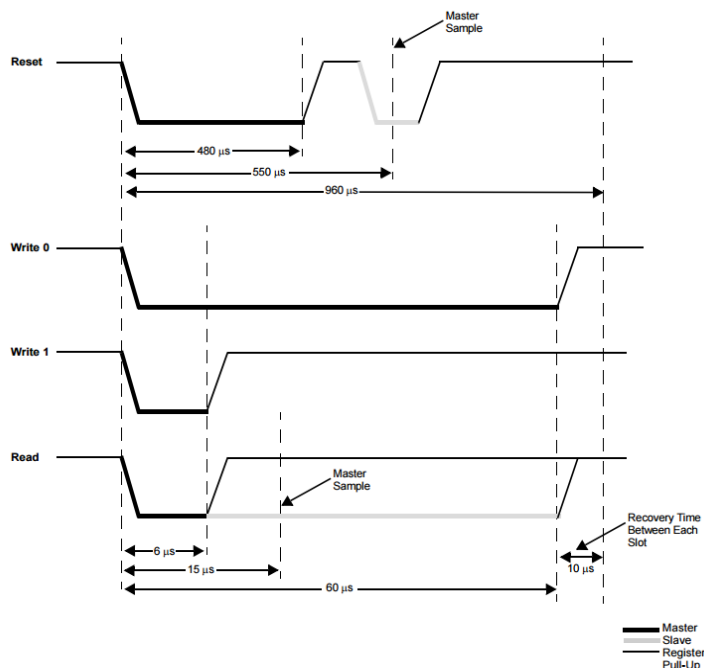
4. Temperaturni senzor DS18B20

Zbog jednostavnosti protokola izabran je temperaturni senzor DS18B20 proizvođača Dallas semiconductor.



Slika 3. Način spajanja temperaturnog senzora

One-Wire protokol služi za komunikaciju između senzora i kontrolera. Postavljanjem kontrolera u ulogu mastera te promjenom stanja kontrolnog pina, senzor koji je slave, šalje nam povratnu informaciju na kontrolni pin.



Slika 4. One-Wire protokol

Funkcije za one-wire protokol pisane su u MPLAB razvojnom okruženju. Komunikacijska sekvenca počinje reset pulsom, nakon toga se write naredbama zahtijeva željena radnja od senzora te se read naredbom čita informacija. Prototipi funkcija dani su u nastavku.

```
unsigned short Ow_Reset(unsigned short *port, unsigned short pin);

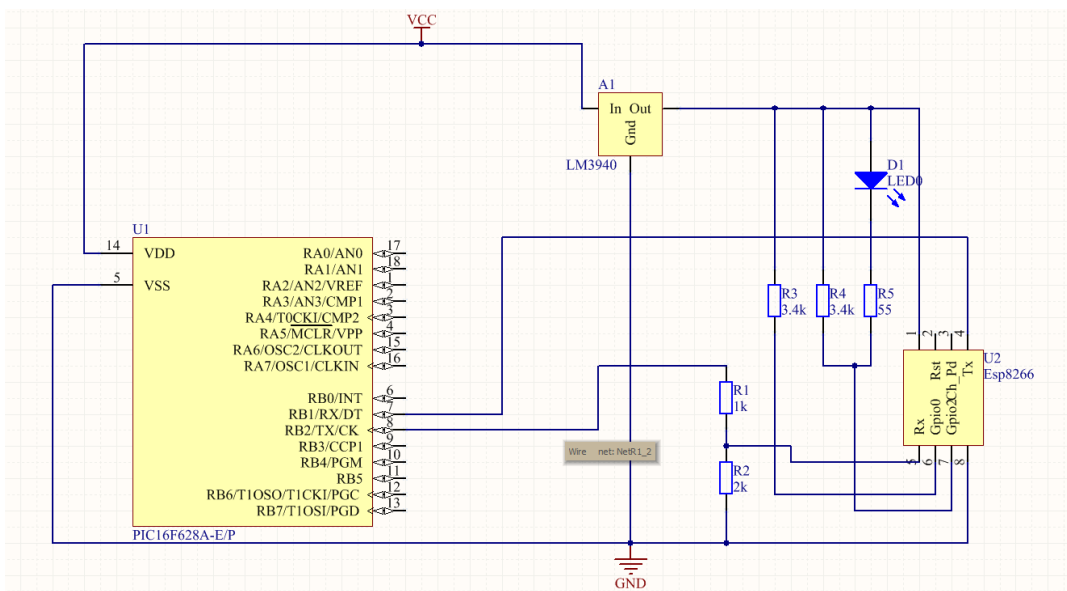
unsigned short Ow_Read(unsigned short *port, unsigned short pin);

void Ow_Write(unsigned short *port, unsigned short pin, unsigned short par);
```

5. Ideja i opis sustava

Sustav je zamišljen tako da prikupljanje i obradu podataka vrši mikrokontroler koji podatke šalje na esp8266 na kojemu se vrti web server.

Esp8266 ima pristup internetu preko kućnog rutera te je port forwardan tako da mu se može pristupiti s bilo kojeg mjesta. Zbog različitog napona napajanja mikrokontrolera i wifi modula, koristi se naponski regulator LM3940IT koji spušta napon s 5V na 3,3V te može dati 1A.

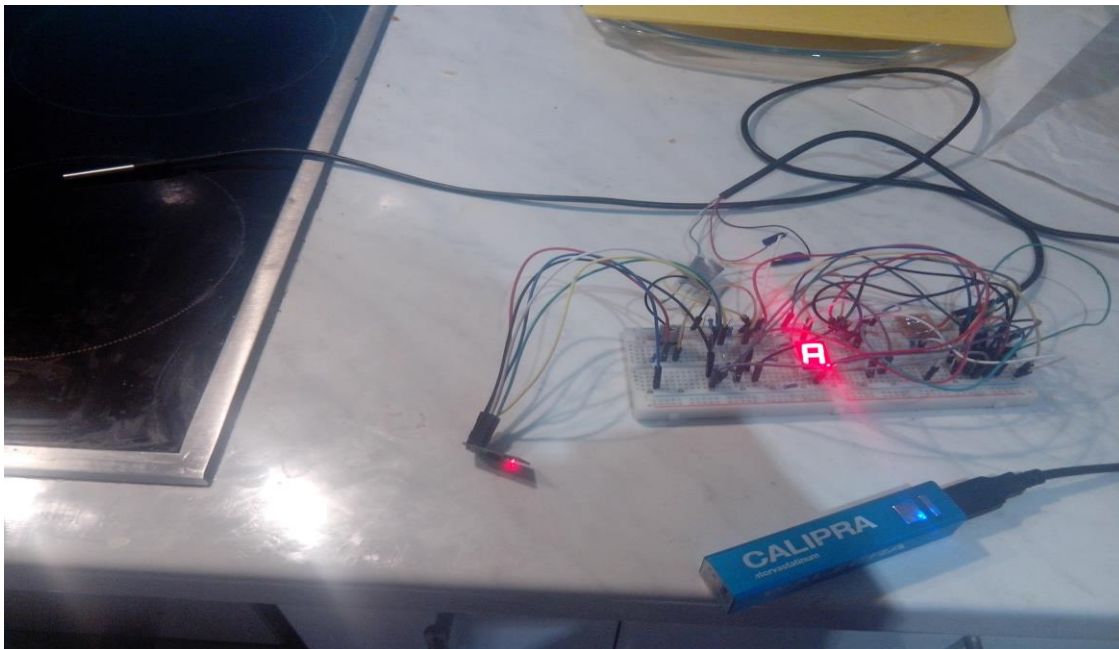


Slika 5. Shema spajanja

Za lakše praćenje ispravnosti koda mikrokontrolera nadoan je 7-segmentni display na koji mikrokontroler šalje dvoznamenkasti prikaz temperature.

5.1. Opis rada sustava

Nakon uspostave komunikacije između mikrokontrolera i temperaturnog senzora, kontroler obrađuje podatke s preciznošću 1°C te ih ispisuje na 7-segmentni prikaznik. Podatke na server šalje na zahtjev servera odnosno korisnika. Za slanje podataka između kontrolera i ESP-a koristi se UART sučelje te se temperatura ispisuje na serveru. Za estimaciju aktivnosti uređaja koristi se granična temperatura, ukoliko se ona prijeđe/prekorači, server javlja upaljenost uređaja.



Slika 6. Prikaz sustava za provjeru aktivnosti toplinskih uređaja

Senzor se može držati na samoj ploči štednjaka. Prednost DS18B20 je vodootpornost, ali može smetati pri uporabi uređaja. Druga opcija je držati ga iznad štednjaka zalijepljenog za napu. Žarulja koja radi dok je napa uključena emitira toplinu te je senzor očitava. Kao najbolji oblik postavljanja pokazalo se postaviti senzor između nape i štednjaka. Na taj način ne smeta u radu s uređajem i žarulja ne utječe toliko na samo očitavanje.

Na serveru pritiskom gumba refresh osvježavamo stanje i učitavamo trenutnu temperaturu s kontrolera. Učestalim stiskanjem refresh gumba može doći do zatrpavanja UART prekida te pogrešnog rada kontrolera.

ESP8266 Temperature Server

GPIO2 **Uređaj je isključen****Temp =26 C**

ESP8266 Temperature Server

GPIO2 **Uređaj je uključen****Temp =43 C**

Slika 7. Prikaz aktivnosti uređaja na web-serveru

Kao što je i prije u tekstu spomenuto, ovaj sustav omogućava samo uvid u aktivnost odnosno temperaturu uređaja, ali ne i opciju gašenja i paljenja uređaja. Razlog tome je taj da pristup serveru ima bilo tko te time može doći do opasnosti. Implementacija navedene opcije sa hardverske strane je jednostavna, u sustav bi se nadodao tranzistor koji bi s aktivnosti GPIO1 pina ESP-a regulirao protok struje, ali zbog neznanja HTML-a i web developmenta nije ostvaren.

6. Programaska podrška

Program za mikrokontroler pisan je u MPLAB i MikroC razvojnim okruženjima.

U glavnom programu vrti se funkcija očitavanja temperature sa senzora te preslagivanje podataka.

```
void main() {
    TRISB = 0;
    TRISA.F1 = 0;
    TRISA.F2 = 0;
    TRISA.F3 = 0;

    CMCON = 7;
    UART1_init(9600);
    PIR1.RCIF = 0;
    PIE1.RCIE = 1;
    INTCON.GIE = 1;
    INTCON.PEIE = 1;
    TRISA.F1 = 0;
    PORTA.F0 = 0;

    delay_ms(1000);

    while(1)
    {
        Ow_Reset(&PORTA, 0);
        Ow_Write(&PORTA,0, 0xCC); //write_1w();
        Ow_Write(&PORTA,0, 0x44);
        Delay_ms(750);

        Ow_Reset(&PORTA, 0);
        Ow_Write(&PORTA,0, 0xCC);
        Ow_Write(&PORTA,0, 0xBE);

        tempL = Ow_Read(&PORTA,0); //read_1w();
        tempH = Ow_Read(&PORTA,0);
        tempL >>= 4;
        tempH <<= 4;
        tempH += tempL;
        tempUsart2 = (tempH/10);
        tempUsart1 = (tempH%10);
        sevenseg(tempH); //7s call
    }
}
```

Gdje su `Ow_` funkcije reseta, čitanja i pisanja one-wire protokola, a `sevenseg` poziv funkcije za ispis podatka na 7-segmentni pokaznik, na kojoj se nalazi i funkcija estimacije aktivnosti.

Prekidna rutina se izvršava dizanjem UART receive interrupt flag (PIR1.RCIF) te se ulazi u prekid.

```

void interrupt()
{
  if(PIR1.RCIF)
  {
    rx_read = UART1_Read();
    if(rx_read == 0x31)
    {
      UART1_Write('=');
      UART1_Write((tempUsart2+48));
      UART1_Write((tempUsart1+48));
      UART1_Write('\n\r');
    }
    PIR1.RCIF = 0;
  }      // reset receive interrupt flag
}

```

U prekidnoj rutini provjerava se poziv. Ako je server poslao 1 (0x31), mikrokontroler odgovara s temperaturom. Tom IF naredbom olakšavamo mikrokontroleru situaciju učestalih poziva odnosno dojavu greške od strane ESP-a.

Kod za ESP pisan je u ESPlorer v0.2. Definirani su parametri konekcije, uarta i pinova, te server sluša port 8366 koji je dodjeljen ESP-u.

```

wifi.setmode(wifi.STATION)
wifi.sta.config("SSID","PW")
gpio.mode(3,gpio.INPUT,gpio.FLOAT)
gpio.mode(4, gpio.OUTPUT)
uart.setup(0, 9600,8,0,1)
srv=net.createServer(net.TCP)
srv:listen(8366,function(conn) --Listening port 8366
.
.
.
buf = buf.."<h1> ESP8266 Temperature Server</h1>"
buf = buf.."<p>GPIO2 <a href=\"?pin=OFF\"><button>Refresh</button></a></p>"
client:send(buf)
local _on,_off = "",""
  if(_GET.pin == "OFF")then
    uart.write(0, "1")
    if(gpio.read(3) == gpio.HIGH)then
      gpio.write(led2, gpio.LOW)
      conn:send("<h3> Uređaj je uključen</h3>")
    else
      gpio.write(led2, gpio.HIGH)
      conn:send("<h3> Uređaj je isključen</h3>")
    end
  end
uart.on("data",4,function(data)
  if(data == nil) then
    conn:send("<h2> Temp je nedostupna </h2>")
  else
    temp = data
    conn:send("<h2>Temp "..temp.." C</p1>")
    end
    uart.on("data")
  end)
end)

```

7. Zaključak

Razvijeni sustav opisan ovom dokumentacijom nudi nam nova rješenja i time možemo poboljšati sigurnost doma. Nadogradnjom sustava moglo bi se ostvariti daljinsko paljenje i gašenje uređaja te na taj način izbjeći povratak u dom u slučaju aktivnog uređaja.

Posebnu korist imali bi zaboravniji, užurbani i preopterećeni ljudi koji svoje rutinske radnje ne registriraju. Ako se dogodi takva situacija, ostaje samo vratiti se u dom i provjeriti ili konstantno sumnjati i preispitivati svoje prošle radnje.

Sljedeći korak bi mogao biti mreža provjere aktivnosti u domu, koja bi pazila na više uređaja te u jednostavnoj aplikaciji pokazivala aktivnost određenog uređaja. Kao i u ovom slučaju najvažnija je sigurnost preko interneta i mogućnost zlouporabe.

Uz ovu bi aplikaciju osobe mogle lagodnije i bezbrižnije napuštati dom te imati mogućnost uvida u njega čak i dok nisu u njemu.

8. Literatura

- [1] ESP8266 web server tutorial URL: <http://randomnerdtutorials.com/esp8266-web-server>
- [2] NewBoston HTML, CSS tutorial URL: https://www.youtube.com/watch?v=cqszz_OfAFQ&list=PLC1322B5A0180C946
- [3] NodeMCU Lua URL: http://www.electrodragon.com/w/ESP8266_NodeMCU_Lua
- [4] MicroChip PIC16F627A/628A/648A Datasheet

