

Kristijan Čavić

0036473375

Jelena Ivanković

0036461986

Kristina Ledić

0036457705

Filip Ložić

0023094754

SEMINARSKI RAD - SPVP



Fakultet elektrotehnike i računarstva, Sveučilišta u Zagrebu

Zavod za elektroničke sustave i obradbu informacija

Sveučilište u Zagrebu

Detektor

prisustva osoba



- Δ Svim zainteresiranim osobama
- Δ Osnove programiranja
- Δ Spajanje senzora s Arduinom
- Δ Opis realizacije sustava

Sažetak

Sustav za detekciju prisustva osoba dio je projekta „pametna kuća“. Pomoću njega možemo detektirati prisutnost i aktivnosti osoba u određenoj prostoriji te predvidjeti trend njihovog rada. Sustav je ostvaren pomoću raznih senzora postavljenih na različita mesta u spavaćoj sobi i spojenih na Arduino. Prednost sustava je to što otvara mogućnost za spajanje na druge sustave zadužene za automatsko podešavanje okoline na osnovu detekcije senzora. Nedostaci su potencijalno krivo interpretiranje određenih radnji. Koristi od sustava će imati svi oni koji žele imati nadzor nad aktivnostima u željenoj prostoriji.

Sadržaj

1. UVOD	3
2. OPIS SUSTAVA	4
2.1. Evidencija o broju osoba (Kristijan Čavić).....	4
2.2. Detekcija rada i oblačenja (Kristina Ledić).....	5
2.3. Detekcija opuštanja (Filip Lozić).....	5
2.4. Detekcija spavanja (Jelena Ivanković)	6
3. REALIZACIJA SUSTAVA.....	7
3.1. Fotootpornik	7
3.2. PIR senzor.....	7
3.3. Senzor sile	8
3.4. Arduino Mega	9
3.5. DS1302	9
3.6. Senzor zvuka.....	9
3.7. Programski kod.....	10
4. ZAKLJUČAK.....	19
5. LITERATURA.....	20
6. POJMOVNIK	21

Ovaj seminarski rad je izrađen u okviru predmeta „Sustavi za praćenje i vođenje procesa“ na Zavodu za elektroničke sisteme 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.

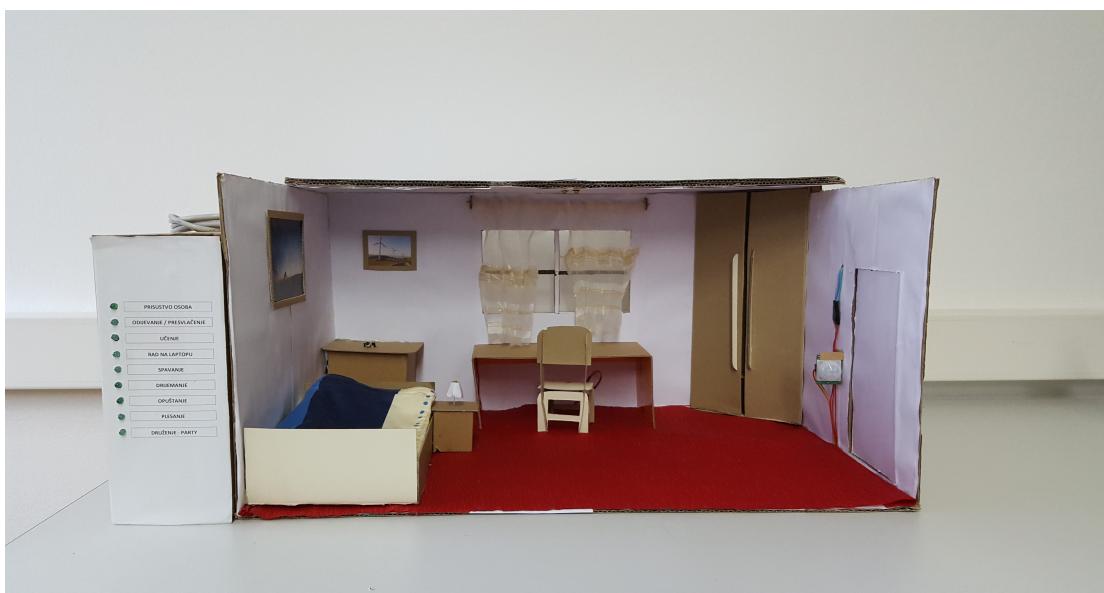
1. Uvod

Tehnologije vezane uz pametne kuće sve više se koriste za opremanje doma prema potrebama stanara i njihovim zahtjevima za sigurnost, udobnost i ekonomičnost. Ideja sustava koji će biti prezentiran u nastavku rada je da detektira prisutnost i aktivnost osoba u prostoriji koristeći različite senzorske uređaje te predvidi trend njihovog rada. Ove parametre moguće je jednostavno pratiti i pomoći kamere. Međutim, taj pristup nam daje samo uvid u trenutno stanje u prostoriji, dok nam sustav temeljen na senzorima otvara mogućnost za povezivanje s drugim sustavima.

Podaci dobiveni od strane senzora mogu se iskoristiti za ostvarenje složenijeg sustava kojim bi se ostvarila interakcija između ljudi i uređaja. Takav sustav prilagođavao bi se čovjekovim potrebama odnosno omogućio određivanje načina na koji će se ambijent prilagođavati. To je posebno korisno osobama kojima je zbog određenih invaliditeta ručno izvođenje tih radnji otežano. Također je namijenjeno svima onima koji žele automatizirati prilagođavanje različitih parametara za ugodniji boravak u prostoriji.

2. Opis sustava

Za potrebe ovog projekta izrađena je maketa spavaće sobe te su na mjestima od interesa postavljeni odgovarajući senzori. Pomoću senzora pritiska, PIR senzora, fotootpornika, RTC-a mikrofona i Arduina ostvaren je sustav koji detektira prisustvo i aktivnosti osobe u spavaćoj sobi.



Slika 1. Maketa sobe s postavljenim senzorima

2.1. Evidencija o broju osoba (Kristijan Čavić)

Detekcija prisutnosti osoba i evidencija o njihovom broju ostvarena je tako da je PIR senzor postavljen s vanjske i unutrašnje strane vrata kako bi se detektiralo je li osoba ušla ili izšla iz prostorije. Ako se najprije aktivirao senzor s vanjske, a zatim s unutrašnje strane znamo da je osoba ušla u prostoriju, dok je u obrnutom slučaju izšla.

RTC senzor je korišten kako bi se utvrdilo doba dana u kojem se odvija određena radnja. Na osnovu njega se, u slučaju da je u trenutku ulaska osobe u sobu bila noć, automatski pali svjetlo.

Više uzastopnih jednakih detekcija nam govori da je u prostoriju ušlo nekoliko osoba. U tom slučaju možemo pretpostaviti kako se u sobi odvija druženje te se tu ostavlja mogućnost za povezivanje sa sustavom koji bi automatski prilagodio ambijent, što bi u ovom slučaju moglo značiti da upali glazbu ili izvrši neku drugu radnju prema prethodno definiranim željama korisnika.

Zbog karakteristika PIR senzora, odnosno vremena potrebnog za njegovu reakciju, može doći do propusta pri detekciji osobe ako u isto vrijeme, ili u malim vremenskim razmacima, uđe više osoba. Do neispravne detekcije također može doći i u slučaju ulaska kućnog ljubimca u sobu.

2.2. Detekcija rada i oblačenja (Kristina Ledić)

Detekcija oblačenja vrši se pomoću fotootpornika koji je postavljen u ormara. Prilikom otvaranja vrata ormara, mijenja se količina svjetlosti pa se senzor aktivira i na osnovu toga predviđamo da će se osoba oblačiti te je vrlo vjerojatno da će potom napustiti prostoriju.

Drugi fotootpornik nalazi se na stolici za radnim stolom. Na radnom stolu također imamo postavljen gumb koji signalizira rad na laptopu. Za ovu svrhu prikladniji bi bio senzor pritiska, no zbog nedostatka istih odabrani su ovi senzori.

Ako je pomoću fotootpornika detektirano da osoba sjedi na stolici za radnim stolom, a nije aktivna signalizacija za rad na laptopu, zaključujemo da će osoba učiti. Ako pak imamo detekciju rada na laptopu i detekciju glazbe pretpostavlja se da osoba koristi računalo za razonodu.

2.3. Detekcija opuštanja (Filip Lozić)

Za potrebe detekcije radnji čija je svrha opuštanje koristi se kombinacija senzora. Fotootpornik je postavljen na stolicu koja se nalazi za radnim stolom i njegova uloga je detektirati da je osoba sjela za radni stol. Mikrofon ima ulogu da detektira sviranje glazbe dok PIR detektira prisutnost ljudi. Senzor sile na krevetu detektira ležanje.

Ako je zabilježena detekcija zvuka na mikrofonu i aktivan je senzor pritiska na krevetu, zaključujemo da će se osoba opuštati. Također smatramo da se osoba opušta, ako je uz sviranje glazbe prisutna i detekcija sjedenja za radnim stolom. U tim slučajevima nema potreba za prilagođavanjem ambijenta budući da je već prilagođen. No, ako se u prostoriji nalazi više ljudi, a mikrofonom nije utvrđeno sviranje glazbe, mogla bi se ostvariti komunikacija sa sustavom koji bi automatski puštao glazbu te na tako prilagodio ambijent.

2.4. Detekcija spavanja (Jelena Ivanković)

Detekcija spavanja vrši se pomoću senzora pritiska koji je postavljen na krevet. Ovisno o dobi dana i trajanju djelovanja sile na ovaj senzor, predviđa se aktivnost osobe. Za određivanje dobi dana koristi se RTC, a na osnovu trajanja detekcije predviđa se koja se radnja odvija.

Ako je detekcija senzora duža od 5 minuta i na osnovu RTC-a se ustanovi da je noć, zaključuje se da će osoba spavati i sustav gasi svjetlo ako je bilo upaljeno. Ako je pak ustanovljeno da je dan i detekcija je duža od 5 min te nema detekcije na preostalim senzorima, zaključuje se da osoba rijemeta. Tu se ostavlja mogućnost spajanja sa sustavom za dodatnu prilagodbu ambijenta ovim aktivnostima. Primjerice, ako osoba rijemeta tijekom dana bilo bi praktično povezati se sa sustavom zaduženim za kontrolu roleta čime bi se zamračila soba i slično.

3. Realizacija sustava

3.1. Fotootpornik

Fotootpornik (*eng. photoresistor ili light dependent resistor - LDR*) je otpornik kojemu se otpor mijenja ovisno o količini upadne svjetlosti tako da mu se električni otpor smanjuje ako se količina upadne svjetlosti povećava.

Izrađuje se od poluvodiča s velikim električnim otporom. U mraku može imati otpor i do nekoliko $M\Omega$), a ako je izložen svjetlosti tada mu se otpor može smanjiti na nekoliko stotina oma. Ako je frekvencija upadne svjetlosti dovoljno velika, odnosno veća od granične frekvencije materijala od kojeg je napravljen poluvodič, fotoni apsorbirani od strane poluvodiča predaju energiju elektronima koji potom prelaze iz valentnog u vodljivi pojas stvarajući time električnu struju. Posljedica toga je smanjenje otpora fotootpornika.



Slika 2. Fotoopornik

3.2. PIR senzor

Najčešće korišten senzor za detekciju prisustva osoba je PIR (*engl. Passive InfraRed*) senzor. Senzor se sastoji od piroelektričnog elementa koji ima svojstvo da generira električni naboј uslijed promjene toplinskog toka odnosno radijacije. Svi objekti zrače s određenim stupnjem radijacije, a to je zračenje jače ako je objekt topliji. Stoga je ovaj senzor prikladan za detekciju živih bića.

Navedeni piroelektrični element sastoji se od dva dijela sačinjena od materijala osjetljivog na infracrveno zračenje. Kada je senzor u mirovanju (*eng.idle*) oba dijela detektiraju jednaku količinu zračenja. Međutim, kada se u području djelovanja senzora nađe čovjek, toplina koju on zrači najprije je detektirana od strane prvog dijela te se javlja pozitivni

diferencijalni napon između dvaju dijelova piroelektričnog elementa. Prilikom izlaska čovjeka iz dometa senzora se pak javlja negativni diferencijalni napon. Navedene promjene napona omogućavaju detekciju.

Drugi važni dio ovog senzora Fresnelova leća čija je uloga podjela područja detekcije na što manje dijelove s ciljem povećavanja dometa senzora.

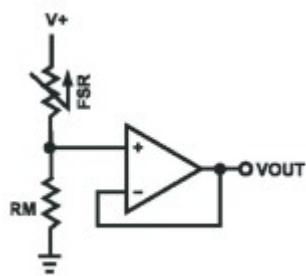


Slika 3. PIR senzor

3.3. Senzor sile

Otpornici osjetljivi na silu (*eng. Force Sensing Resistors*) su otpornici čija se otpornost mijenja ako se silom djeluje na njihovu površinu. Što je primjenjena sila veća, to je njihova otpornost manja. Najčešće se spajaju u naponsko djelilo s mjernim otpornikom fiksne vrijednosti. Izlazni napon u tom slučaju se mijenja prema sljedećoj jednadžbi:

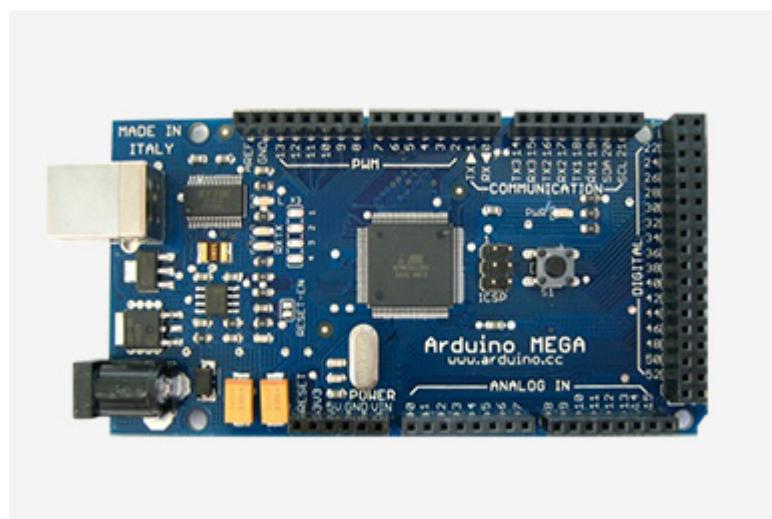
$$V_{out} = \frac{R_m V}{R_m + R_{fsr}}$$



Slika 4. Način spajanja FSR-a

3.4. Arduino Mega

Arduino Mega je mikrokontroler temeljen na ATmega2560 mikroprocesoru. Ima 54 digitalna ulazno-izlazna pina (od kojih se 15 može koristiti kao PWM izlazi), 4 UART porta, 16 analognih ulaza, 16 MHz kristalni oscilator, ulaz za napajanje, ICSP konektor i gumb za reset. Na računalo se povezuje pomoću USB kabela. Ovaj mikrokontroler radi na naponu od 5 V te ima EEPROM memoriju veličine 4 KB, SRAM memoriju veličine 8 KB i FLASH memoriju veličine 256 KB. Arduino mikrokontroler je otvorenog koda te podržava spajanje raznih periferija.



Slika 5. Arduino Mega

3.5. DS1302

DS1302 je čip koji sadrži sat koji daje trenutno vrijeme (Real Time Clock). Statički RAM mu je veličine 31 bajt. Komunicira s mikroprocesorom preko jednostavnog serijskog sučelja. Pruža informacije o sekundama, minutama, satima, datumu, mjesecu, godini. Koristi 32.768kHz vanjski oscilator.

3.6. Senzor zvuka

Pomoću ovog senzor vrši se detekcija glazbe. Koristi se digitalni izlaz, odnosno bilježi se samo je li zvuk prisutan ili ne, što je prikladno zbog

nenušavanja privatnosti. Prag okidanja može biti podešen pomoću potenciometra.

3.7. Programski kod

```

***** DETEKCIJA PRISUTSTVA *****
***** Autori: Kristijan Čavić, Jelena Ivanković, Kristina Ledić, Filip Ložić *****
// Fotootpornici
const byte pinPhotores_wardrobe = A1; // Senzor u ormaru
const byte pinPhotores_chair = A2; // Senzor na stolici (zamjena za senzora
// sile)
// Mikrofon modul
const byte pinMic = 15; // Senzor glazbe/TV-a
// Senzori sile
const byte pinForce_bed = A0; // Senzor sile na krevetu
// Tipka
const byte pinForce_laptop = 14; // Senzor na radnom stolu (zamjena za
senzor sile)
// PIR
int PIR_IN = 11; // PIR senzor ispred sobe
int PIR_OUT = 12; // PIR senzor u sobi

***** Pinovi za spajanje LED indikatora radnji *****
const byte pinLED_wall = 16; // Stropna lampa
const byte pinLED_person = 2; // Osoba je u sobi
const byte pinLED_wardrobe = 3; // Osoba se odijeva
const byte pinLED_learn = 4; // Osoba uči
const byte pinLED_laptop = 5; // Osoba radi na laptopu
const byte pinLED_bed_sleep = 6; // Osoba spava
const byte pinLED_bed_relax = 7; // Osoba drijema
const byte pinLED_mic_relax = 8; // Osoba se opušta
const byte pinLED_mic_dance = 9; // Osoba pleše
const byte pinLED_party = 10; // Više osoba u sobi, druženje
const byte pinLED_bed_light = 13; // Lampa pokraj kreveta

***** REAL TIME CLOCK *****
// Pinovi za spajanje RTC-a
#define DS1302_SCLK_PIN 19
#define DS1302_IO_PIN 20
#define DS1302_CE_PIN 21
#define DS1302_CLOCK_BURST_READ 0xBF
// Funkcije za izračun sata, minute i sekunde
#define bcd2bin(h,l) (((h)*10) + (l))
#define bin2bcd_h(x) ((x)/10)
#define bin2bcd_l(x) ((x)%10)
//
typedef struct {
    uint8_t Seconds:4; // Donje decimalne znamenke 0-9
    uint8_t Seconds10:3; // Gornje decimalne znamenke 0-5
    uint8_t CH:1; // CH = Clock Halt
    uint8_t Minutes:4;
    uint8_t Minutes10:3;
    uint8_t reserved1:1;
    union {
        struct {
            uint8_t Hour:4;
            uint8_t Hour10:2;
            uint8_t reserved2:1;
            uint8_t hour_12_24:1; // Upisati 0 za 24-satni format
        } h24;
    };
}

```

Detektor prisustva osoba



```
    uint8_t Hour:4;
    uint8_t Hour10:1;
    uint8_t AM_PM:1;      // Upisati 0 za AM, 1 za PM
    uint8_t reserved2:1;
    uint8_t hour_12_24:1; // Upisati 1 za 12-satni format
    } h12;
};

uint8_t Date:4;           // Dan u mjesecu, 1 = prvi dan
uint8_t Date10:2;
uint8_t reserved3:2;
uint8_t Month:4;          // Mjesec, 1 = Siječanj
uint8_t Month10:1;
uint8_t reserved4:3;
uint8_t Day:3;             // Dan u tjednu, 1 = prvi dan (Nedjelja)
uint8_t reserved5:5;
uint8_t Year:4;            // Godina, 0 = godina 2000.
uint8_t Year10:4;
uint8_t reserved6:7;
uint8_t WP:1;               // WP = Write Protect
} ds1302_struct;

//***** VARIJABLE ZA UPIS VRIJEDNOSTI SENZORA *****
int rtc_time=0;           // Varijabla za upis očitanog sata s RTC
int Photores_wardrobe_value = 0; // Fotootpornik u ormaru
int Photores_chair_value = 0; // Fotootpornik na stolici
int Force_laptop_value = 0; // Tipka na radnom stolu
int Force_bed_value = 0; // Senzor sile na krevetu

//***** GRANIČNE VRIJEDNOSTI ZA ODREĐIVANJE RADNJE *****
int Wardrobe_threshold = 15; // Fotootpornik u ormaru
int Chair_threshold = 1; // Fotootpornik na stolici
int Bed_threshold = 100; // Senzor sile na krevetu
const unsigned int Sleep_threshold = 5*1000UL; // 5 sekundi

// Brojači uzastopnih očitavanja koja su prešla granične vrijednosti
short c_wardrobe = 0;
short c_chair = 0;
int c_PIR = 0;
short c_laptop = 0;
unsigned long c_bed = 0;

// Broj uzastopnih očitavanja dovoljnih za potvrdu radnje
short N = 100;

// Početak detekcije senzora sile na krevetu iznad granične vrijednosti
unsigned long t_bed = 0;

// Vremena unutar kojih se treba dogoditi detekcija na PIR senzorima
int t_pir_in_start = 0;
int t_pir_out_start = 0;
int t_pir_in_now;
int t_pir_out_now;

//***** ZASTAVICE ZA DETEKCIJU RADNJE *****
short f_PIR = 0;
short f_PIR2 = 0;
short f_wardrobe = 0;
short f_chair = 0;
short f_bed_sleep = 0;
short f_bed_relax = 0;
short f_laptop = 0;
short f_mic = 0;
short f_night = 0;
short f_light = 0;
```

```
int PIR_IN_state = LOW; // Početno stanje PIR-a za ulaz, nema detekcije
int PIR_OUT_state = LOW; // Početno stanje PIR-a za izlaz

//*****SETUP*****
void setup() {
    ds1302_struct rtc; // Definicija varijable za očitavanje stanja RTC-a
    // Definicije pinova senzora
    pinMode(pinPhotores_wardrobe, INPUT);
    pinMode(pinPhotores_chair, INPUT);
    pinMode(pinForce_bed, INPUT);
    pinMode(pinForce_laptop, INPUT);
    pinMode(PIR_IN, INPUT);
    pinMode(PIR_OUT, INPUT);
    pinMode(pinMic, INPUT);
    // Definicije pinova LED indikatora
    pinMode(pinLED_wall, OUTPUT);
    pinMode(pinLED_person, OUTPUT);
    pinMode(pinLED_wardrobe, OUTPUT);
    pinMode(pinLED_learn, OUTPUT);
    pinMode(pinLED_bed_sleep, OUTPUT);
    pinMode(pinLED_bed_relax, OUTPUT);
    pinMode(pinLED_bed_light, OUTPUT);
    pinMode(pinLED_laptop, OUTPUT);
    pinMode(pinLED_mic_dance, OUTPUT);
    pinMode(pinLED_mic_relax, OUTPUT);
    pinMode(pinLED_party, OUTPUT);
    // Ispis broja ljudi u sobi koje su PIR senzori detektirali
    Serial.begin(9600);
    Serial.print("Broj ljudi u sobi:");
    Serial.print("\t");
    Serial.println(c_PIR);
}
//*****GLAVNI PROGRAM*****
void loop() {
    // Poziv funkcija koje očitavaju vrijednosti senzora i postavljaju zastavice
    // ovisno o tome je li i koja je radnja detektirana
    readPIR();
    readPhotoresWardrobe();
    readPhotoresChair();
    readForceBed();
    readForceLaptop();
    readMic();
    // Poziv funkcija koje pale LED indikatore ovisno o postavljenim zastavicama
    LEDwall();
    LEDwardrobe();
    LEDsitting();
    LEDsleep();
    LEDmic();
    LEDparty();
}
//*****FUNKCIJE ZA OČITAVANJE VRIJEDNOSTI SENZORA*****
//**
void readPIR(){
    if(digitalRead(PIR_IN)){
        t_pir_out_now = millis();
        if(PIR_OUT_state == HIGH && c_PIR > 0 && f_PIR == 0){
            if((t_pir_out_now-t_pir_out_start) < 100){
                if((t_pir_out_now-t_pir_out_start) < 100){
```

```

        c_PIR--;
        f_PIR = 1;
        f_PIR2 = 1;
        Serial.println("Motion detected out!");
        Serial.print("Broj ljudi u sobi:");
        Serial.print("\t");
        Serial.println(c_PIR);
        PIR_OUT_state = LOW;
        if(c_PIR == 0) f_PIR = 0;
    } else {
        PIR_OUT_state = LOW;
        t_pir_out_start = millis();
        Serial.println("Preveliki razmak između signala.");
    }
} else {
    PIR_IN_state = HIGH;
    t_pir_in_start = millis();
}
}
if(digitalRead(PIR_OUT)){
    t_pir_in_now = millis();
    if(PIR_IN_state == HIGH && f_PIR == 0){
        if((t_pir_in_now-t_pir_in_start) < 1000){
            c_PIR++;
            f_PIR = 1;
            f_PIR2 = 1;
            Serial.println("Motion detected in!");
            Serial.print("Broj ljudi u sobi:");
            Serial.print("\t");
            Serial.println(c_PIR);
            PIR_IN_state = LOW;
            if(c_PIR > 0) f_PIR = 1;
        } else{
            PIR_IN_state = LOW;
            t_pir_in_start = millis();
            Serial.println("Preveliki razmak između signala.");
        }
    } else {
        PIR_OUT_state = HIGH;
        t_pir_out_start = millis();
    }
}
if (digitalRead(PIR_IN) == 0 && digitalRead(PIR_OUT) == 0 && f_PIR2 == 1) {
    //ako su oba pira prestala dobivat signal vraca zastavicu u 0
    f_PIR = 0;
    PIR_IN_state = LOW;
    PIR_OUT_state = LOW;
    f_PIR2 = 0;
    Serial.println("Motion ended");
}
}

// Čita vrijednost fotootpornika u ormaru. Ako je nekoliko uzastopnih
// vrijednosti veće od granične postavlja se zastavica, što znači da je ormar
// otvoren.
void readPhotoresWardrobe(){
    Photores_wardrobe_value = analogRead(pinPhotores_wardrobe);
    //Serial.println(Photores_wardrobe_value);
    if (Photores_wardrobe_value > Wardrobe_threshold) {
        if(c_wardrobe < N) c_wardrobe++;
        if(c_wardrobe == N){
            f_wardrobe = 1;
            c_wardrobe = 0;
        }
    }
}

```

```
        }
    } else {
        f_wardrobe = 0;
        c_wardrobe = 0;
    }
}

// Čita vrijednost fotootpornika na stolici. Ako je nekoliko uzastopnih
// vrijednosti veće od granične postavlja se zastavica, što znači da netko
// sjedi.
void readPhotoresChair(){
    Photores_chair_value = analogRead(pinPhotores_chair);
    //Serial.print("stolica:");
    //Serial.println(Photores_chair_value);
    if (Photores_chair_value < Chair_threshold){
        if (c_chair < N) c_chair++;
        if (c_chair == N){
            f_chair = 1;
            c_chair = 0;
        }
    } else{
        f_chair = 0;
        c_chair = 0;
    }
}

// Čita vrijednost senzora sile na krevetu. Ako se uzastopno očitavaju
// vrijednosti (veće od granične) duže od vremenske granice za krevet (npr. 5
// minuta, ovdje u kodu 5 sekundi), provjerava se doba dana i postavljaju
// zastavice. Ako je noć, osoba spava. Ako je dan, osoba drijema. Osim toga,
// ako je noć svjetlo kraj kreveta se automatski upali kad
// se očita vrijednost veća od granične i, nakon što se ustanovi da osoba
// spava, svjetlo se ugasi.
void readForceBed(){
    Force_bed_value = analogRead(pinForce_bed);
    //Serial.println(Force_bed_value);
    if (Force_bed_value > Bed_threshold){
        c_bed++;
        ds1302_struct rtc;
        DS1302_clock_burst_read( (uint8_t *) &rtc );
        //Serial.println(rtc.h24.Hour10);
        rtc_time = bcd2bin(rtc.h24.Hour10, rtc.h24.Hour);
        //Serial.print("RTC:");
        //Serial.println(rtc_time);
        if (rtc_time > 21 || rtc_time < 7 || rtc_time == 0) f_night = 1;
        else f_night = 0;
        if(c_bed == 1){
            t_bed = millis();
            f_light = 1;
        }
        //if(c_bed >= 2) c_bed = 2;
        if((millis()-t_bed)>=Sleep_threshold){
            if (f_night) f_bed_sleep = 1;
            else f_bed_relax = 1;
            c_bed = 0;
            f_light = 0;
        }
    }
    else {
        f_bed_sleep = 0;
        f_bed_relax = 0;
        f_light = 0;
        c_bed = 0;
    }
}
```

Detektor prisustva osoba



```
}

// Čita vrijednost senzora pritiska (tipka) sa laptopa, ako je iznad granične
// vrijednosti, postavlja zastavicu što znači da osoba radi na laptopu.
void readForceLaptop(){
    //Force_laptop_value = analogRead(pinForce_laptop);
    if (digitalRead(pinForce_laptop) == HIGH) f_laptop = 1;
    else f_laptop = 0;
}
// Čita vrijednost s mikrofona. Ako mikrofon detektira zvuk, postavljaju se
// zastavice kojima se dalje provjeravaju radnje.
void readMic(){
    if(digitalRead(pinMic) == HIGH) f_mic = 1;
    else f_mic = 0;
    //Serial.println(digitalRead(pinMic));
}
//***** FUNKCIJE ZA ČITAVANJE RTC-a *****
//***** Preuzeto s https://playground.arduino.cc/Main/DS1302
void DS1302_clock_burst_read(uint8_t *p){
    int i;
    digitalWrite(DS1302_CE_PIN, LOW); // default, not enabled
    pinMode(DS1302_CE_PIN, OUTPUT);
    digitalWrite(DS1302_SCLK_PIN, LOW); // default, clock low
    pinMode(DS1302_SCLK_PIN, OUTPUT);
    pinMode(DS1302_IO_PIN, OUTPUT);
    digitalWrite(DS1302_CE_PIN, HIGH); // start the session
    delayMicroseconds(4); // tCC = 4us

    DS1302_togglewrite(DS1302_CLOCK_BURST_READ, true);
    for( i=0; i<8; i++){*p++ = _DS1302_toggleread();}
    digitalWrite(DS1302_CE_PIN, LOW); // Set CE low
    delayMicroseconds(4); // tCWH = 4us
}
// -----
uint8_t _DS1302_toggleread( void){
    uint8_t i, data;
    data = 0;
    for( i = 0; i <= 7; i++){
        digitalWrite( DS1302_SCLK_PIN, HIGH);
        delayMicroseconds( 1);
        digitalWrite( DS1302_SCLK_PIN, LOW);
        delayMicroseconds( 1); // tCL=1000ns, tCDD=800ns
        bitWrite( data, i, digitalRead( DS1302_IO_PIN)); // read bit, and set it
                                                       //in place in 'data' variable
    }
    return( data);
}
// -----
void _DS1302_togglewrite( uint8_t data, uint8_t release){
    int i;
    for( i = 0; i <= 7; i++){
        digitalWrite( DS1302_IO_PIN, bitRead(data, i));
        delayMicroseconds( 1); // tDC = 200ns
        digitalWrite( DS1302_SCLK_PIN, HIGH);
        delayMicroseconds( 1); // tCH = 1000ns, tCDH = 800ns
        if( release && i == 7) pinMode( DS1302_IO_PIN, INPUT);
        else {
            digitalWrite( DS1302_SCLK_PIN, LOW);
            delayMicroseconds( 1); // tCL=1000ns, tCDD=800ns
        }
    }
}
```

```
        }
    }
//***** FUNKCIJE ZA PALJENJE LED INDIKATOR *****
//*****
// Primarna funkcija: upali se LED kad osoba uđe u sobu te stropna lampa kad
// je osoba u sobi i noć je (za ostvarivanje ove funkcije treba odkomentirati
// kod koji je trenutno
// zakomentiran, a onaj koji je aktivan treba zakomentirati. Sekundarna
// funkcija (s trenutno
// aktivnim kodom) je da se upali LED kad osoba uđe u sobu te stropna lampa
// ako je ušlo više
// osoba.
void LEDwall(){
    if( c_PIR > 0 ) {
        digitalWrite(pinLED_person, HIGH);
        digitalWrite(pinLED_wall, HIGH);
    } else {
        digitalWrite(pinLED_person, LOW);
        digitalWrite(pinLED_wall, LOW);
    }
    // ds1302_struct rtc;
    // DS1302_clock_burst_read( uint8_t * ) &rtc);
    // rtc_time = bcd2bin( rtc.h24.Hour10, rtc.h24.Hour);
    // if (rtc_time > 22 || rtc_time < 8 || rtc_time == 0){
    //     if(f_PIR == 1) digitalWrite(pinLED_wall, HIGH);
    //     else digitalWrite(pinLED_wall, LOW);
    // }
    // else digitalWrite(pinLED_wall, LOW);
}

// Ako je ormar otvoren, upali LED koja označava da se osoba odijeva.
void LEDwardrobe(){
    if(f_wardrobe == 1) digitalWrite(pinLED_wardrobe, HIGH);
    else digitalWrite(pinLED_wardrobe, LOW);
}

// Ako je postavljena samo zastavica da je netko na stolici, upali LED da
// osoba uči.
// Ako su postavljene zastavice za stolicu i rad na laptopu, upali LED da
// osoba radi na laptopu.
void LEDsitting(){
    if (f_chair == 1){
        if (f_laptop == 1){
            digitalWrite(pinLED_laptop, HIGH);
            digitalWrite(pinLED_learn, LOW);
        } else{
            digitalWrite(pinLED_learn, HIGH);
            digitalWrite(pinLED_laptop, LOW);
        }
    } else {
        digitalWrite(pinLED_laptop, LOW);
        digitalWrite(pinLED_learn, LOW);
    }
}

// Ako je detektirano da je osoba na krevetu, ovisno o dobu dana pali se LED
// za drijemanje ili za
// spavanje. Ako je noć i detektirano je da je osoba na krevetu, upali se
// automatski lampa kraj kreveta
// i gasi se automatski ako je osoba na krevetu duže vrijeme.
void LEDsleep() {
    //if(c_bed > 0){
    if(f_bed_sleep == 1) digitalWrite(pinLED_bed_sleep, HIGH);
```

Detektor prisustva osoba



```
if(f_bed_relax == 1){
    digitalWrite(pinLED_bed_relax, HIGH);
    digitalWrite(pinLED_bed_light, HIGH);
} else{
    digitalWrite(pinLED_bed_sleep, LOW);
    digitalWrite(pinLED_bed_relax, LOW);
    digitalWrite(pinLED_bed_light, LOW);
}
// if(f_light == 1 && f_bed_sleep == 0){
// digitalWrite(pinLED_bed_sleep, LOW);
// digitalWrite(pinLED_bed_light, HIGH);
// } else if (f_light == 1 && f_bed_sleep == 1){
// digitalWrite(pinLED_bed_sleep, HIGH);
// digitalWrite(pinLED_bed_light, LOW);
// } else if (f_bed_relax == 1) digitalWrite(pinLED_bed_relax, HIGH);
// else digitalWrite(pinLED_bed_relax, LOW);
// } else{
// digitalWrite(pinLED_bed_sleep, LOW);
// digitalWrite(pinLED_bed_light, LOW);
// digitalWrite(pinLED_bed_relax, LOW);
// }
// }
}

// Ako je postavljena samo zastavica za mikrofon, a ostale nisu, upali LED da
// osoba pleše.
// Ako je postavljena zastavica za mikrofon i krevet/laptop, upali LED da se
// osoba opušta.
void LEDmic(){
if (f_mic == 1){
    if (c_PIR > 0 && f_bed_relax == 0) digitalWrite(pinLED_mic_dance, HIGH);
    if (f_bed_relax || f_chair || f_laptop) digitalWrite(pinLED_mic_relax,
HIGH);
} else {
    digitalWrite(pinLED_mic_dance, LOW);
    digitalWrite(pinLED_mic_relax, LOW);
}
} else {
    digitalWrite(pinLED_mic_dance, LOW);
    digitalWrite(pinLED_mic_relax, LOW);
}
}

// Ako je u sobi više od jedne osobe, upali LED za druženje.
void LEDparty(){
if (c_PIR >= 2 && f_mic == 1) digitalWrite(pinLED_party, HIGH);
else digitalWrite(pinLED_party, LOW);
}
//*****
```

4. Zaključak

Ostvarenjem opisanog sustava dobivamo uvid u trenutno stanje prostorije. Na osnovu podataka sa senzora možemo zaključiti hoće li osoba spavati, učiti, oblačiti se ili pak odmarati te poduzeti neke radnje kao što je paljenje/gašenje svjetla.

Ovi podaci mogu se iskoristiti kao ulazni parametri složenijeg sustava koji bi na osnovu njih poduzimao dodatne akcije. Primjerice, ako osoba sjedne za radni stol s namjerom da uči, sustav bi ugasio glazbu ako je svirala i slično. Time bi se stvorio ambijent koji bi se prilagođavao potrebama i željama korisnika.

5. Literatura

- [1] <http://www.resistorguide.com/photoresistor/>
- [2] <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work>
- [3] https://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf
- [4] <https://www.arduino.cc/en/main/arduinoBoardMega2560>

6. Pojmovnik

Pojam	Kratko objašnjenje	Više informacija potražite na
Arduino Mega	Mikrokontroler koji je osnova sustava.	https://www.arduino.cc/en/main/arduinoBoardMega2560
FSR	Otpornik čija se otpornost mijenja ovisno o primijenjenoj sili.	https://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf
Fotootpornik	Otpornik čiji se otpor mijenja ovisno o količini svjetlosti.	http://www.resistorguide.com/photoresistor/
PIR	Senzor koji mjeri koičinu infracrvenog zračenje u svojoj okolini.	https://learn.adafruit.com/assets/35647
RTC	Modul koji daje informacije o trenutnom vremenu.	https://datasheets.maximintegrated.com/en/ds/DS1302.pdf