

Seminarska naloga pri predmetu računalniški sistemi in  
omrežja

## Razvoj avtomata za pripravo kave

Mentorica: Maja Azarov Domajnko, prof.

Avtor: Jakob Hostnik, G 3. A

Ljubljana, marec 2016

# Zahvala

Hvala profesorici Maji Azarov Domajnko za mentorstvo, Marku Uhanu in Anžetu Žaberlu za pomoč pri izdelavi vezja in hvala tudi Martinu Hostniku za pomoč, pri zasnovi sistemov.

# Povzetek

Pri predmetu računalniški sistemi in omrežja sem izdelal kavni aparat. V seminarski nalogi je opisan način njegovega delovanja in sestavnih sistemov. Predstavljen je program, izdelava vezja, izdelava ohišja in sistemov za delovanje. Opisane so težave, ki sem jih imel pri razvoju sistemov in razlaga zakaj kavni aparat trenutno ne more delovati popolnoma pravilno. V bližnji prihodnosti pa sledijo izboljšave, da bo kavomat deloval popolnoma brez težav, ki so se pojavile na strojnem področju.

# Kazalo

## Kazalo vsebine

|     |  |    |
|-----|--|----|
| 1.  | Uvod .....                               | 6  |
| 2.  | Komponente .....                         | 7  |
| 3.  | Sistemi.....                             | 8  |
| 3.1 | Sistem za preverjanje denarja .....      | 8  |
| 3.2 | Sistem za vračanje denarja.....          | 9  |
| 3.3 | Sistem za dovajanje sestavin.....        | 9  |
| 3.4 | Sistem za gretje in dovajanje vode ..... | 11 |
| 4.  | Program.....                             | 13 |
| 5.  | Vezje .....                              | 15 |
| 6.  | Ohišje .....                             | 16 |
| 7.  | Zaključek .....                          | 17 |
| 8.  | Viri .....                               | 18 |

## Kazalo slik

|  |    |
|--|----|
| Slika 1: Bralec denarja - spredaj .....              | 9  |
| Slika 2: Bralec denarja - s strani .....             | 9  |
| Slika 3: Izdelava sistema za dovajanje praškov ..... | 10 |
| Slika 4: Elektromagnetni ventil .....                | 11 |
| Slika 5: Aluminijast grelnik .....                   | 13 |
| Slika 6: Gumbi za izbiro .....                       | 14 |
| Slika 7: Izbiranje sladkorja .....                   | 15 |
| Slika 8: Risba vezja - grafično.....                 | 15 |
| Slika 9: Risba vezja - shematsko .....               | 16 |
| Slika 10: Izdelava vezja .....                       | 16 |
| Slika 11: Ohišje - ekran z gumbi.....                | 17 |
| Slika 12: Ohišje - celotno .....                     | 17 |

# 1. Uvod

Kavni aparati so zelo dragi in zasedejo veliko prostora. Kavomati, kakršen je moj, stanejo med 1 000 in 2 000 evrov. Menim, da lahko celoten sistem izdelam bistveno ceneje. Izdelal bom tudi sistem za preverjanje in vračanje denarja, sistem za gretje vode in sistem za dodajanje sestavin. Preventivno nisem podrobno pogledal delovanja današnjih kavomatov, saj mislim, da je njihova proizvodnja predraga in lahko z razvojem svojega sistema to ceno bistveno zmanjšam. Takšno razmišljanje se je kasneje izkazalo za napačno, saj sem imel velike težave, ker so se sistemi, ki sem jih razvil, v praksi obnašali zelo drugače, kot sem si jih zamislil v teoriji.

Pri izdelovanju kavomata sem se soočil z štirimi strokami. To so računalništvo (programski del), elektrotehnika (izdelava vezja), strojništvo (sistemi) in mizarstvo (izdelava ohišja). Največ težav sem imel z strojništvom in delovanjem sistemov v praksi.

## 2. Komponente

Pri svojem delu sem potreboval kar nekaj delov in stvari, ki jih nisem mogel izdelati sam in sem jih moral kupiti.

| Komponenta                       | Cena     |
|----------------------------------|----------|
| 15x servomotor                   | 23 €     |
| Elektromagnetni ventil           | 4 €      |
| 4x mikrokrmilnik Arduino         | 8 €      |
| 3x ploščica za vezje             | 3 €      |
| Cevi                             | 2 €      |
| Nastavki za priklop cevi         | 5 €      |
| Uravnalec tlaka                  | 15 €     |
| Potenciometri                    | 1 €      |
| Zaslon s tekočimi kristali (LCD) | 2 €      |
| Napajalnik                       | 15 €     |
| Grelnik za spajkalnik            | 10 €     |
| Aluminij                         | 1 €      |
| Tipke                            | 1 €      |
| Podaljšek za vtičnico            | 5 €      |
| Lijak                            | 5 €      |
| Vijaki                           | 1 €      |
| Lesonit                          | 10 €     |
| Plastenke                        | /        |
| Ostalo                           | 15 €     |
| Skupaj                           | 126,00 € |

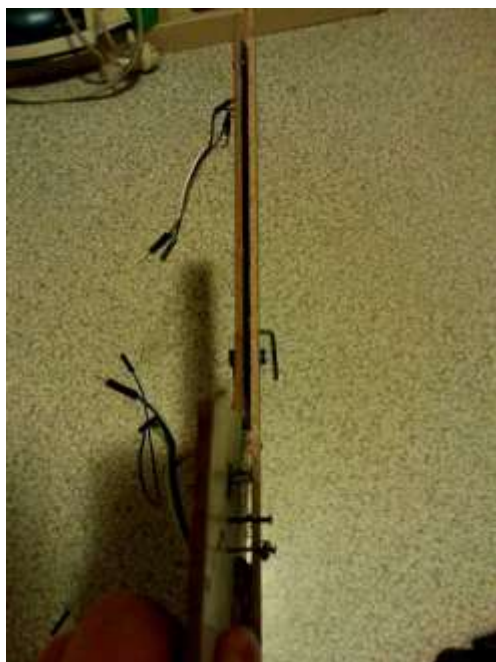
### 3. Sistemi

Ko sem izdeloval napravo sem moral praktično na novo iznajti nekaj sistemov, kot na primer sistem za preverjanje denarja, sistem za vračanje denarja, za dovajanje sestavin in sistem za gretje in dovajanje vode. Vsak sistem sem razvil sam, brez pomoči obstoječih sistemov, kar se je tudi izkazalo za napako, saj sem na ta način izgubil veliko časa. Zaradi tega se kavomat na preizkusih ni odrezal najbolje.

#### 3.1 Sistem za preverjanje denarja

Kavomat denar določi na podlagi njegove teže in velikosti. Kovanec se zakotali po ozki poti, pri tem premakne vzvod pri potenciometru, ki odčita njegovo velikost in s svojo težo premakne vzvod, potenciometer na katerega je pritrjen, pa to odčita. Nekaj težav sem imel pri razvoju sistema, saj so si teže in velikosti kovancev med seboj zelo podobne. Vendar sem po nekaj truda uspel kovance razlikovati. Sistem sem izdelal, a ga zaradi pomanjkanja časa nisem vgradil v napravo. Takšen sistem je zelo neodporen proti preverjanju ponarejenih kovancev.





Slika 1: Bralec denarja - spredaj



Slika 2: Bralec denarja - s strani

### 3.2 Sistem za vračanje denarja

Denar za vračanje je že vnaprej pripravljen in zložen v tulcih za vsako od velikosti med 2 evroma in 10 centi evropskih kovancev. S servomotorjem porine spodnji kovanec iz tulca v režo, kjer uporabnik prejme ostanek. Sistema prav tako zaradi pomanjkanja časa nisem uspel izdelati, ga pa imam namen vgraditi v kavomat s sistemom za odčitavanje vrednosti kovanca v bližnji prihodnosti.

### 3.3 Sistem za dovajanje sestavin

Vsaka od netekočih sestavin kave je shranjena v navrtani plastenki, luknjo pokriva servomotor in s svojim obračanjem prepušča sestavino skozi luknjo.

Vse plastenke so pritrjene nad lijakom, ki potem vse usmeri proti kozarcu s kavo.

Sestavine, ki so v plastenkah, so:

- Dve vrsti kave
- Sladkor
- Mleko v prahu
- Kapučino
- Kakav



Slika 3: Izdelava sistema za dovajanje praškov

### 3.4 Sistem za gretje in dovajanje vode

Za dovajanje vode so ključne tri komponente. Elektromagnetni ventil, aluminijast grelnik in uravnalec tlaka.

#### **Elektromagnetni ventil**

Ventil prepušča vodo tlaka med 0.2 bara in 8 barov. Deluje na 12 voltih, torej sem za njegovo krmiljenje moral uporabiti električno stikalo (rele). Ko je ventil pod napetostjo, v notranjosti odpre majhno luknjico za pretok. V celoti pa ventil odpre šele sila vode. Za odprtje ventil ustvari okrog sebe močno magnetno polje, kar lahko zmede delovanje mikrokrmilnika.



Slika 4: Elektromagnetni ventil

## **Uravnalec tlaka**

Vodovodni tlak je 5 barov. Kar pa je preveč, kot sem ugotovil po testih, saj voda skozi ventil teče prehitro in se ne bi mogla dovolj segreti, lonček z kavo pa bi se napolnil prehitro. Zato sem kupil uravnalec tlaka, ki je tlak zmanjšal na pol bara, kar je bilo dovolj, da je voda počasi tekla skozi cevi.

## **Grelnik**

Eno izmed večjih težav je predstavljal grelnik. Izdelal sem ga sam iz kosa aluminija in grelnika za spajkalnik. V aluminij sem zavrtal 2 luknji eno za cev z vodo drugo za grelnik, okrog katerega sem dal termalno pasto, da bi se aluminij hitreje segrel. Namestil sem še kabel z električnim stikalom, da sem krmilil njegov vklop in diodo, da sem pretvoril izmenično napetost v enosmerno.

Namen je bil, da bi kos aluminija z maso enega kilograma ohranjal na stalni temperaturi, saj bi zaradi velike mase potreboval dolgo časa, da se segreje. Zaradi tega pa sem tudi pričakoval, da se bo ohlajal počasi.

V praksi pa se je stvar obnašala popolnoma drugače od mojih pričakovanj, saj je grelnik potreboval 40 minut časa, da se je segrel na 100 °C, ko pa sem čez njega spustil en deciliter vode pa se je shladil na 30 °C. Največji šok pa je bil, ko je bila voda, ki je pritekla iz grelca, na otip mrzla. Če bi izdelal vezje, ki bi mi pretvoril izmenično napetost v enosmerno brez intervalov, bi grelnik po mojem mnenju grel še enkrat hitreje, saj je sedaj grel v zelo majhnih intervalih. Ko gre napetost v nasprotno smer jo dioda blokira. Tako je grelnik pod napetostjo samo polovico časa. Tudi, če bi grelnik grel še enkrat hitreje, bi bilo to še vedno prepočasi. Zaenkrat

kavomat pripravlja hladno kavo, dokler ne vgradim močnejšega grelca.



Slika 5: Aluminijast grelnik

## 4. Program

Zaradi pomanjkanja izhodov sem moral uporabiti dva mikrokontroler, ki sem ju med seboj povezal, podatke pa si med seboj izmenjujeta s serijsko povezavo. Tako sem moral napisati dva programa. Vsak od mikrokontrolerov ima svoje naloge. Prvi izpisuje na ekran in bere uporabnikovo izbiro, drugi pa nam nameša želeno kavo. Zato sem prvi program poimenoval reader, drugega pa coffeMaker. Lotil sem se

objektnega programiranja, kar naredi kodo bolj berljivo. Napisal sem približno 350 vrstic kode. Vso kodo pa sem pisal v angleščini, saj tako lahko kodo uporabijo in razumejo ljudje po celem svetu.

V programu coffeMaker sem za objekte določil sestavine v obliki praškov in vodo. Tako sem lahko pri pisanju receptov za posamezno kavo preprosto določal katero sestavino in koliko sestavine naj spusti v lonček. Objekt sem napisal tako, da to naredim z ukazom `addition.openForUnit(numberOfUnits)`. Spremenljivka `addition` nam pove, katero sestavino naj dodamo. Parameter `numberOfUnits` pa nam pove, za koliko prej podanih časovnih enot naj sestavino dodajamo v lonček.

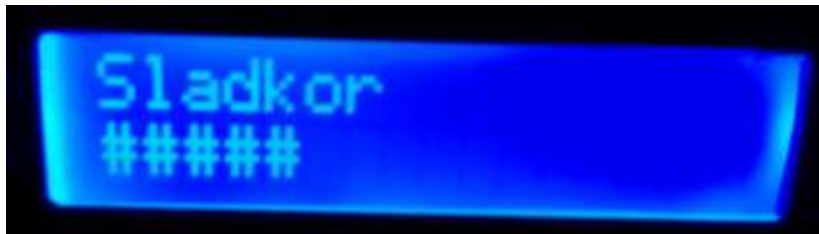
V programu reader pa sem napisal objekte za gumbе in sladkor. Gumbi z metodo `read()` preverjajo, kdaj je bil kateri gumb pritisnjen.



Slika 6: Gumbi za izbiro

Objekt za sladkor pa nam pomaga določiti količino sladkorja, ki ga želimo imeti v kavi. Mi določimo koliko enot sladkorja bomo imeli v kavi od 0 do 5. Ko izberemo

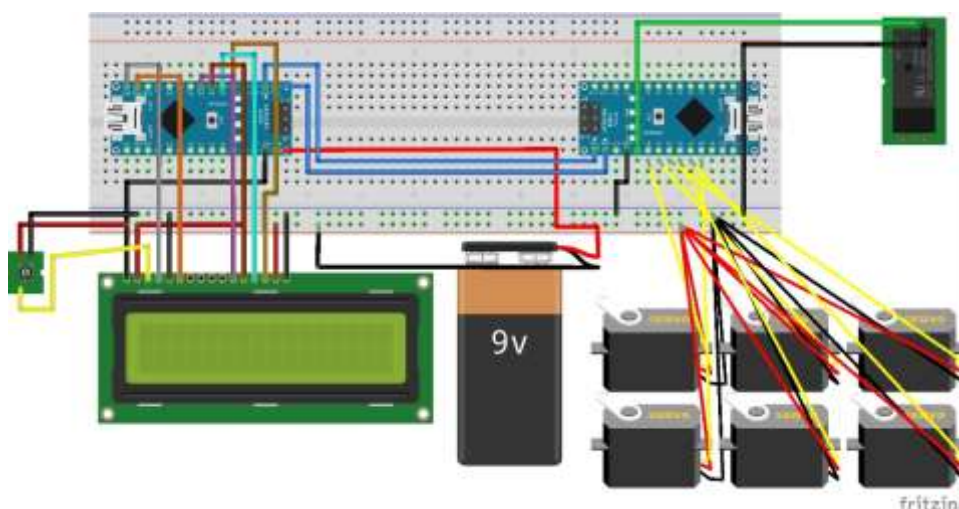
količino sladkorja in izberemo kavo, program pošlje podatke drugemu mikrokrmilniku, ki jih uporabi in nam pripravi kavo.



Slika 7: Izbiranje sladkorja

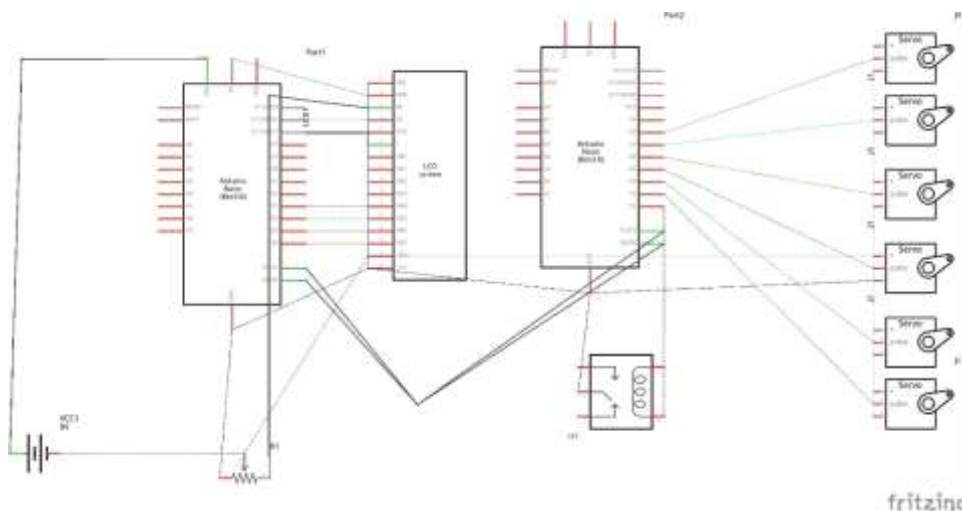
## 5. Vezje

Preden sem se lotil izdelave vezja, sem v programu Fritzing narisal približen načrt. Skica v programu zaradi premajhnega števila komponent v programu ne vsebuje vseh elementov. Namesto baterije sem uporabil računalniški napajalnik, ki ima več kablov z različnimi napetostmi. Potreboval sem 5 voltov za napajanje servomotorjev in 12 voltov za napajanje mikrokrmilnikov in odpiranje ventila. 220 voltov za grelnik pa sem dobil iz vtičnice. Shema ne vključuje grelnika, gumbov in napajalnika.

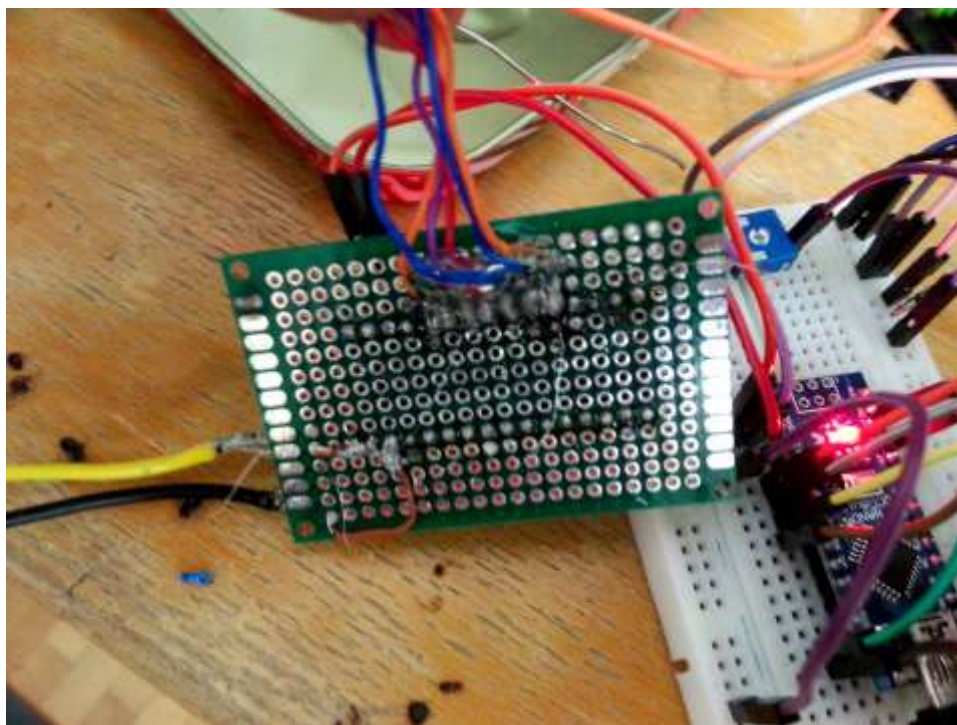


Slika 8: Risba vezja - grafično





Slika 9: Risba vezja - shematsko



Slika 10: Izdelava vezja

## 6. Ohišje

Ohišje sem naredil iz lesonita. Oblikoval pa sem ga z vbodno žago. S tem, da sem



ohišje izdelal sam, sem zagotovil dovolj prostora vsebini. Za enkrat ohišje še nima pokrova, z namenom, da lahko popravljam in spreminjam postavitev komponent. V ohišje sem naredil odprtini za gumbe in zaslon. Na koncu pa sem še dodal nalepke z oznakami.



Slika 11: Ohišje - ekran z gumbi



Slika 13: Ohišje - celotno

## 7. Zaključek

Kavomat sem z mnogo truda uspel izdelati in tudi cenovno je izdelava bistveno cenejša od konkurenčne. Moral pa sem vložiti kar nekaj časa, truda, potrpljenja in denarja. Kljub vsemu kar sem vložil, pa zaradi moje neizkušenosti aparat še vedno ne deluje popolnoma pravilno in ima kar nekaj fizičnih pomanjkljivosti.

Če bi se lahko še enkrat odločal o naslovu seminarske naloge, ne bi naredil kavnega aparata. V primeru ponovne izdelave, pa zagotovo ne bi razvijal vseh sistemov sam, ampak bi uporabil že kakšnega od obstoječih.

## 8. Viri

Uradni list Republike Slovenije (2012), Pridobljeno 20. 2. 15 s strani:

<https://www.uradni-list.si/1/content?id=107922>

Arduino – Libraries, Pridobljeno 10. 3. 15 s strani:

<https://www.arduino.cc/en/hacking/libraries>

Arduino Playground – Library tutorial, Pridobljeno 10. 3. 15 s strani:

<https://playground.arduino.cc/Code/Library>

Arduino – Servo, Pridobljeno 10. 3. 15 s strani:

<https://www.arduino.cc/en/Reference/Servo>

Arduino – Serial, Pridobljeno 10. 3. 15 s strani:

<https://www.arduino.cc/en/Reference/Serial>

# PRILOGE

## Koda

coffeMaker.ino

```
#include "Servo.h";
```

```
#include "Addition.h"
```

```
#include "Water.h"
```

```
/*define defaults*/
```

```
#define ms 500 //time default unit milliseconds
```

```
#define warmerPin 13
```

```
/*define pins*/
```

```
/*static variables*/
```

```
Water water(2,ms);
```

```
Addition sugar(3,250);
```

```
Addition milk(4, ms);
```

```
Addition coffe1(5, ms);
```

```
Addition coffe2(6, ms);
```

```
Addition capuccino(7, ms);
```

```
Addition cacao(8, ms);
```

```
/*code*/
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    Serial.write(2);
```

```
    /*Attach servos*/
```

```

sugar.start();
coffe1.start();
milk.start();
coffe2.start();
capuccino.start();
cacao.start();
pinMode(warmerPin,OUTPUT);
}
void loop() {
    //Control temperature
    float temp=(analogRead(A0)/1023.0)*500;
    if(temp>100)
        digitalWrite(warmerPin,0);
    else if(temp<95)
        digitalWrite(warmerPin,1);
    /*start checking*/
    if(Serial.available()){
        if(Serial.read()=='S'){ //Start
            char mode;
            int sUnit;
            do{
                mode=Serial.read();
            }while(mode===-1);
            do{
                sUnit=(int)Serial.read();//sugar, from 1 to 5
            }while(sUnit===-1);
            /*make coffe*/

```

```

    Serial.print(mode);
    Serial.print(sUnit);
    switch(mode){
        case 'A': makeCoffe1(); break;
        case 'B': makeCoffe2(); break;
        case 'C': makeCacao(); break;
        case 'D': makeCapuccino(); break;
        case 'E': makeWhiteCoffe(); break;
        case 'F': makeCoffeWithMilk(); break;
        case 'G': makeCoffeWithCacao(); break;
    }
    /*sugar*/
    if(sUnit!=0){
        sugar.openForUnit(sUnit);
    }
    Serial.write('E');//end signal
    }
}
}

/*recepts*/
void makeCoffe1(){
    coffe1.openForUnit(2);
    water.openForUnit(4);
}

void makeCoffe2(){
    coffe2.openForUnit(2);
    water.openForUnit(5);
}

```

```

}

void makeCacao(){
    cacao.openForUnit(2);
    milk.openForUnit(3);
    water.openForUnit(4);
}

void makeCapuccino(){
    capuccino.openForUnit(2);
    water.openForUnit(4);
}

void makeWhiteCoffe(){
    coffe1.openForUnit(2);
    milk.openForUnit(4);
    water.openForUnit(4);
}

void makeCoffeWithMilk(){
    coffe1.openForUnit(2);
    milk.openForUnit(2);
    water.openForUnit(4);
}

void makeCoffeWithCacao(){
    cacao.openForUnit(2);
    coffe1.openForUnit(2);
    milk.openForUnit(2);
    water.openForUnit(4);
}

```

Addition.cpp

```
#include "Arduino.h"
```

```
#include "Servo.h"
```

```
#include "Addition.h"
```

```
Addition::Addition(int pin, int unit){
```

```
    this->pin=pin;
```

```
    this->unit=unit;
```

```
}
```

```
void Addition::start(){
```

```
    s.attach(pin);
```

```
    close();
```

```
}
```

```
void Addition::open(){
```

```
    s.write(90);
```

```
    delay(100);
```

```
}
```

```
void Addition::close(){
```

```
    s.write(180);
```

```
    delay(100);
```

```
}
```

```
void Addition::openFor(int milliseconds){
```

```
    open();
```

```
    delay(milliseconds);
```

```
    close();
```

```
}
```

```
void Addition::openForUnit(int numberOfUnits){
```

```
    open();
```

```

    delay(numberOfUnits*unit);
    close();
}

```

Addition.h

```

#ifndef Addition_h
#define Addition_h
#include "Servo.h"
class Addition{
public:
    Addition(int pin, int unit);
    void open();
    void close();
    void start();
    void openFor(int milliseconds);
    void openForUnit(int numberOfUnits);
private:
    Servo s;
    int pin;
    int unit;
};
#endif

```

Water.cpp

```

#include "Arduino.h"
#include "Water.h"
Water::Water(int pin, int unit){

```



```

    this->pin=pin;
    this->unit=unit;
    pinMode(pin,OUTPUT);
}
void Water::open(){
    digitalWrite(pin,HIGH);
}
void Water::close(){
    digitalWrite(pin,LOW);
}
void Water::openFor(int milliseconds){
    open();
    delay(milliseconds);
    close();
}
void Water::openForUnit(int numberOfUnits){
    open();
    delay(numberOfUnits*unit);
    close();
}

```

Water.h

```

#ifndef Water_h
#define Water_h
class Water{
public:
    Water(int pin, int unit);

```

```

    void open();
    void close();
    void openFor(int milliseconds);
    void openForUnit(int numberOfUnits);
private:
    int pin;
    int unit;
};
#endif

```

reader.ino

```

#include <LiquidCrystal.h>
#include "Button.h"
#include "Sugar.h"
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
Button coffe1(A0);
Button coffe2(A1);
Button cacao(A2);
Button capuccino(A3);
Button whiteCoffe(A4);
Button coffeWithMilk(6);
Button coffeWithCacao(9);
Sugar sugar(7,8);
bool sugarF=false;
unsigned long m;
void setup() {
    m=millis();

```

```

    lcd.begin(16, 2);
    lcd.print("Pozdravljeni!");
    lcd.setCursor(0, 1);
    lcd.print("Dolocite sladkor");
    Serial.begin(9600);
}

void loop() {
    if(coffe1.read())sendData('A');
    if(coffe2.read())sendData('B');
    if(cacao.read())sendData('C');
    if(capuccino.read())sendData('D');
    if(whiteCoffe.read())sendData('E');
    if(coffeWithMilk.read())sendData('F');
    if(coffeWithCacao.read())sendData('G');

    if(millis()-m>2000&&ugarF){
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Izberite napitek");
        ugarF=false;
    }

    if(sugar.isChanged()){
        sugarPrint();
        ugarF=true;
        m=millis();
    }
}

```

```

void sendData(char mode){
    Serial.write('S');
    Serial.write(mode);
    Serial.write(sugar.sugar);
    String t;
    switch(mode){
        case 'A': t="Kava 1"; break;
        case 'B': t="Kava 2"; break;
        case 'C': t="Kakav"; break;
        case 'D': t="Kapucino"; break;
        case 'E': t="Bela kava"; break;
        case 'F': t="Kava z mlekom"; break;
        case 'G': t="Cokoladna kava"; break;
    }
    byte p=0;
    while(Serial.read()!='E'){
        if(millis()-m>500){
            m=millis();
            lcd.clear();
            lcd.setCursor(0,0);
            lcd.print("Priprava");
            for(byte i=0;i<p;i++)
                lcd.print(".");
            p=(p+1)%4;
            lcd.setCursor(0,1);
            lcd.print(t);
            m=millis();
        }
    }
}

```

```

    }
}

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Koncano!");

lcd.setCursor(0,1);

lcd.print("Postrezite si!");

delay(5000);

sugar.sugar=3;

sugarF=false;

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Pozdravljeni!");

lcd.setCursor(0, 1);

lcd.print("Dolocite sladkor");
}

void sugarPrint(){
    lcd.clear();

    lcd.setCursor(0, 0);

    lcd.print("Sladkor");

    lcd.setCursor(0,1);

    for(byte i =0;i<sugar.sugar;i++)
        lcd.print("#");
}

```

Button.cpp

```
#include "Arduino.h"
```

```

#include "Button.h"

Button::Button(int pin){
    this->pin=pin;
    pinMode(pin,INPUT_PULLUP);
}

boolean Button::read(){
    boolean curr=!digitalRead(pin);
    if(!prev&&curr){//previous is 0 current is 1
        prev=curr;
        return true;
    }
    prev=curr;
    return false;
}

```

```

Button.h

#ifndef Button_h
#define Button_h

class Button{
    public:
        Button(int pin);
        boolean read();
    private:
        int pin;
        int prev;
};

#endif

```

Sugar.cpp

```
#include "Arduino.h"
```

```
#include "Sugar.h"
```

```
#include "Button.h"
```

```
Sugar::Sugar(int pin0,int pin1):add(pin1),rem(pin0){}
```

```
boolean Sugar::isChanged(){
```

```
    if(add.read()){
```

```
        sAdd();
```

```
        return true;
```

```
    }
```

```
    if(rem.read()){
```

```
        sRemove();
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

```
void Sugar::sAdd(){
```

```
    if(sugar<5)sugar++;
```

```
}
```

```
void Sugar::sRemove(){
```

```
    if(sugar>0)
```

```
        sugar--;
```

```
}
```

Sugar.h

```
#ifndef Sugar_h
```

```
#define Sugar_h
#include "Button.h"
class Sugar{
    public:
        Sugar(int pin0,int pin1);
        boolean isChanged();
        byte sugar=3;
    private:
        Button add,rem;
        void sAdd();
        void sRemove();
};
#endif
```



# Urnik dela

| Datum            | Vsebina   | Porabljen čas v urah |
|------------------|---|----------------------|
| 6. 3. 2016       | Razvoj sistema za odčitavanje denarja           | 5                    |
| 7. 3. 2016       | Testiranje opreme                               | 1                    |
| 12. 3. 2016      | Nakup delov                                     | 1                    |
| 12. 3. 2016      | Razvoj in izdelava sistema za dovajanje praškov | 3                    |
| 12. 3. 2016      | Testiranje                                      | 0.5                  |
| 13. 3. 2016      | Izdelava ohišja                                 | 1.5                  |
| 18. 3. 2016      | Izdelava ohišja                                 | 2                    |
| 19. 3. 2016      | Izdelava sistema za dotok vode                  | 1                    |
| 19. 3. 2016      | Izdelava grelnika                               | 2                    |
| 21. 3. 2016      | Prilagajanje sistema za dotok in gretje vode    | 3                    |
| 22. 3. 2016      | Pisanje programa za pripravo kave               | 5                    |
| 23. 3. 2016      | Pisanje programa za določanje kave              | 5                    |
| 29. 3. 2016      | Izdelava vezja                                  | 6                    |
| 30. 3. 2016      | Sestavljanje komponent in prilagajanje          | 5                    |
| 30. 3. 2016      | Pisanje dokumentacije, komentiranje kode        | 2                    |
| 31. 3. 2016      | Pisanje dokumentacije, komentiranje kode        | 6                    |
| <b>Skupaj ur</b> |   | <b>58</b>            |



# Izjava o avtorstvu

Izjavljam, da je seminarska naloga Razvoj avtomata za kavo v celoti moje avtorsko delo, ki sem ga izdelal samostojno s pomočjo navedene literature in pod vodstvom mentorja.

10. 4. 2016

Jakob Hostnik