



Univerza v Ljubljani
Fakulteta *za računalništvo*
in informatiko

Letno poročilo

Vhodno izhodne naprave

2021/2022

Mentor:
Robert Rozman

Študent:
Žan Juvan

Kazalo:

Uvod	3
Izbira elementov.....	3
Vezava	4
Programiranje.....	5
Modeliranje	6
Zaključek.....	7
Povezave na portale	8

Kazalo slik:

Slika 1: Testiranje servo motroja	3
Slika 2: Vezava v ThinkerCad-u	4
Slika 3: Shema v ThinkerCad-u	4
Slika 4: Spredaj stari model, zadaj novi model	6
Slika 5: Priprava na tiskanje.....	6
Slika 6: Zaključen projekt.....	7

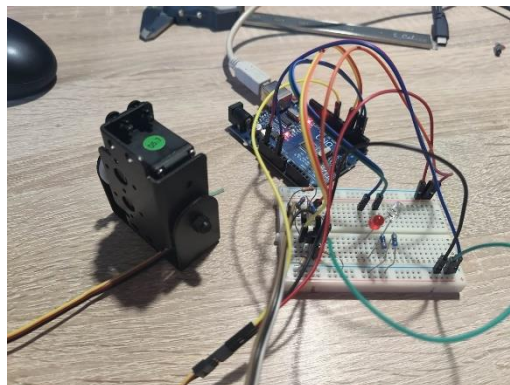
Uvod

Ta projekt sem si izbral, saj sem že pred nekaj časa videl podobno zasnovo kako jo aktivno uporabljajo med poplavami v tujih državah, kjer so poplave dosti pogostejše kot pri nas. Sam se bil že prisoten na velikih poplavah kot gasilec in vem da bi lahko takšna vrata preprečila velike katastrofe že v našem okraju.

Izbira elementov

Zasnoval sem projekt na plošči Arduino Uno, saj sem pred izbiro testiral STM32 a mi je bila njegova logika nekako pretežka. Naslednja najtežja izbira je bila pri izbiri vodila, kako naj dvignim vrata. Vem da bi bila najboljša izbira električni cilinder, a ga na žalost nisem imel zato sem se odločil za motor. Projekt sem začel zasnovati z upanjem, da bom lahko uporabil navaden motor z veliko hitrostjo. Hitro sem ugotovil, da ga ne bom mogel dosti upočasniti niti z uporabo zobnikov. Tu sem se obrnil na mentorja, ki mi je dal servo motor Hitec HS-422.

Da nisem uporabljal posebnega senzorja za pregledovanje nivoja vode sem uporabil dva tranzistorja (BC-337) in pet uporov. Da pa smo lahko videli, da voda narašča in da je dosegla kritičen nivo sem uporabil še dve led diodi in dva upora.

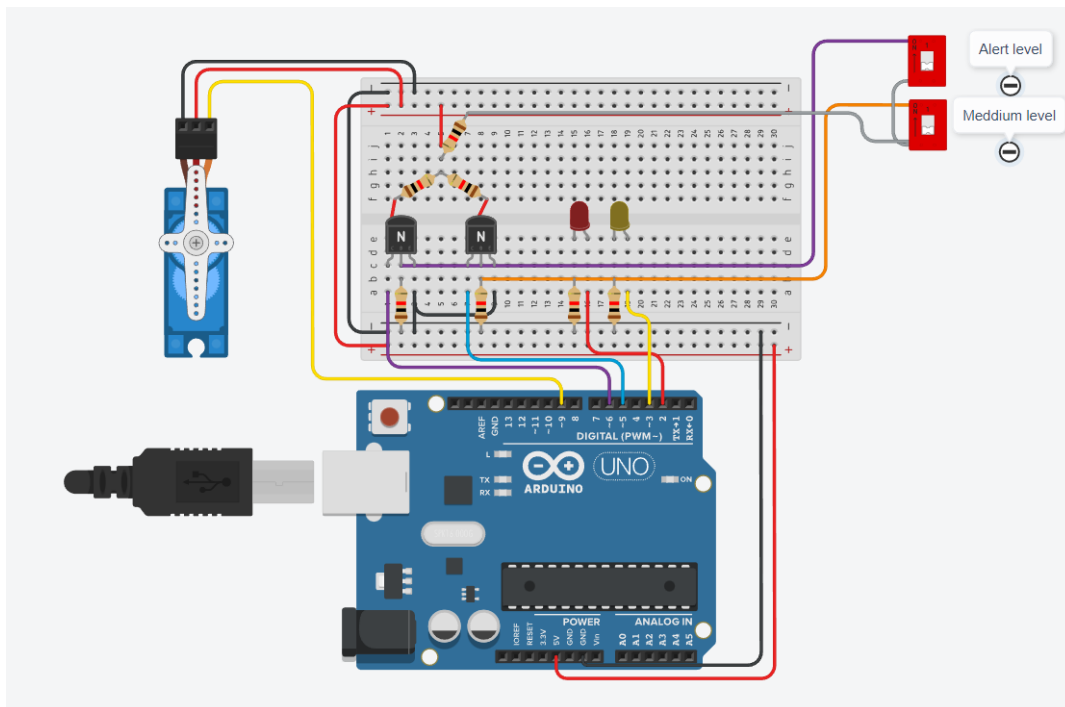


Slika 1: Testiranje servo motroja

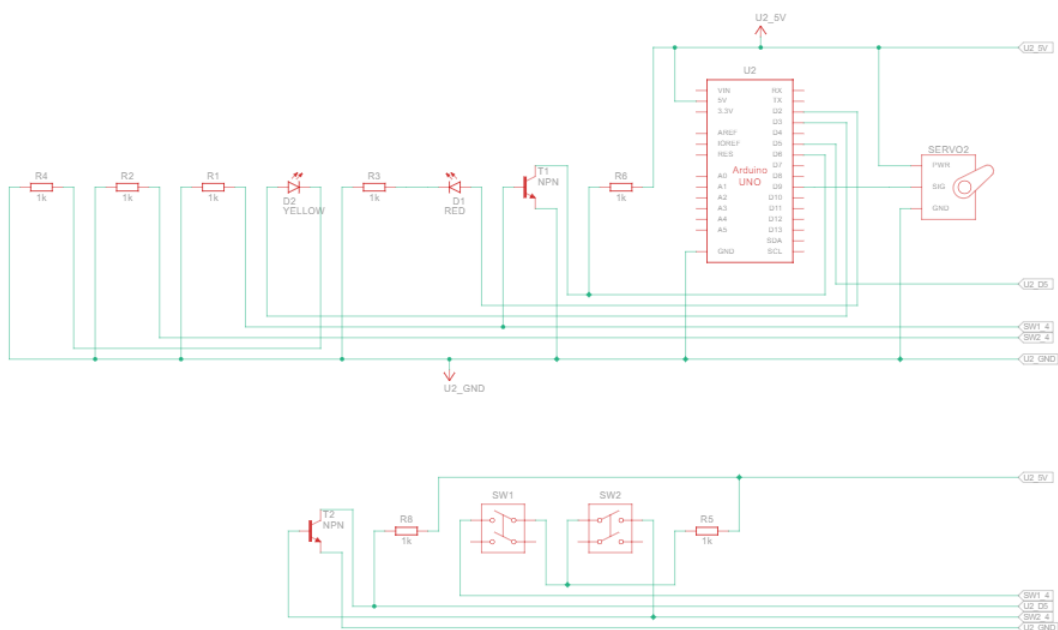
Vezava

Nejprej sem priklopil motor in poskusil ga uporabljati in mu pošiljati različne pozicije. Sama vezava motorja je preprosta saj priklopimo napajanje, vzemljitev in kontrolno. Po kontrolni sporčamo samo signale za pozicijo.

Nato sem se lotil narediti logiko za spremljanje nivoja vode. Tu sem si pomagal s spletom, kjer sem našel [članek](#) o spremljanju nivoja vode. Tranzistor sem uporabil, da lahko varno testiram če pride do stika med nivoji vode.



Slika 2: Vezava v ThinkerCad-u



Slika 3: Shema v ThinkerCad-u

Programiranje

Na začetku sem začel z osnovo, prioriteri mi je bila upravljanje servomotorja. Ker sem že v preteklosti delal na projektu z robotom mi to ni bil pretežak iziv saj smo že tem uporabljali podoben servo motor. Lahko bi uporabil knjižnico za servo motor, a sem zaradi načel predmeta raje šel po poti brez nje. Zato sem z malo pomoči spleta napisal svojo funkcijo za premikanje motorja.

Naslednji korak je bil branje nivoja vode. Tudi to ni nič težjega saj moramo na vhidih le brati ali dobivo visoko ali nisko stanje. Tako kot smo zvezali nam je visoko stanje pomenilo, da sta žički v stiku oziroma da je voda na tistem nivoju. Nisko stanje pa je pomenilo razklenjene žičke, kar pomeni da voda še ni segal do tistega nivoja. Da lahko uporabnik že na daleč vidi nivo vode sem uredil tudi dve led diodi. Rumena led dioda z počasnejšim utripanjem pomeni, da je korito napol polno. To služi kot opozorilo, da ne smemo voziti čez vrata, saj se lahko zaradi dotoka vode dvignejo. Nato je tu še rdeča led dioda, ki pa opozarja na kritičen nivo vode v koritu. Ko obe začnejo utripati pomeni, da se bodo vrata začela zapirati. Da sem se prepričal, da koda res deluje sem jo najprej poskusil na breadboardu. Tako da sem enostavno vse tri žičke vezal skupaj in naredil zanko. Ker me je zanimalo kako bo to delovalo v vodi sem v kozarec natočil vodo in najprej potopil dve žički in nato še zadnjo za kritično stanje. Tako sem se prepričal, da vezava in koda delujeta tako kot morata.

Da je vsa koda dosti bolj berljiva smo jo postavili v različne metode. Metoda RED pomeni kritično stanje. Led diodi začneta utripati zelo hitro. Metoda ORANGE pomeni, da je voda prišla, do srednjega nivoja in led dioda utripa počasneje. In še za konec metoda GATE, ta metoda nam kontrolira servo motor. Če je aktivirano kritično stanje se bodo vrata zaprla, drugače se bodo odprla. Pri tej metodi uporabljamo tudi spremenljivko za shranjevanje ali so vrata zaprta ali odprta, da nebi nepotrebno obremenjevali servo motorja.

Na koncu vse delčke zložimo v metodo, ki se bo konstantno izvajala. Ta metoda bo vedno brala stanje nivoja vode in klicala metode glede same stanja.

```
int gate = LOW;
int waterH = 6;
int waterM = 5;
int redLED = 2;
int redState = LOW;
int oraLED = 3;
int oraState = LOW;
int openGate = 48;
int closeGate = 150;

void setup() {
  pinMode(waterH, INPUT);
  pinMode(waterM, INPUT);
  pinMode(redLED, OUTPUT);
  pinMode(oraLED, OUTPUT);
  pinMode(9, OUTPUT);
  for (byte i = openGate; i < closeGate; i++) {
    servo_pwm(i,9);
  }
  digitalWrite(9, HIGH);
}

void ORANGE() {
  oraState = HIGH;
  digitalWrite(oraLED, HIGH);
  delay(500);
  digitalWrite(oraLED, LOW);
  delay(500);
}
```

```
void loop() {
  if (!digitalRead(waterH) == HIGH) {
    RED();
  } else if (!digitalRead(waterM) == HIGH) {
    redState = LOW;
    ORANGE();
  } else {
    redState = LOW;
    oraState = LOW;
  }
  GATE();
}

void GATE() {
  if (redState == HIGH && gate == LOW) {
    gate = HIGH;
    for (byte i = closeGate; i > openGate; i--) {
      servo_pwm(i,9);
    }
  } else if (redState == LOW && gate == HIGH) {
    gate = LOW;
    for (byte i = openGate; i < closeGate; i++) {
      servo_pwm(i,9);
    }
  }
  digitalWrite(redLED, LOW);
  digitalWrite(oraLED, LOW);
}
```

```
void RED() {
  redState = HIGH;
  digitalWrite(redLED, HIGH);
  digitalWrite(oraLED, HIGH);
  delay(150);
  digitalWrite(redLED, LOW);
  digitalWrite(oraLED, LOW);
  delay(150);
}

void servo_pwm(int x, int pin){
  if( x%14 == 0){
    digitalWrite(redLED, HIGH);
    digitalWrite(oraLED, HIGH);
  }
  else if( x%7 == 0){
    digitalWrite(redLED, LOW);
    digitalWrite(oraLED, LOW);
  }
  int val = (x*10.25)+500;
  digitalWrite(pin,HIGH);
  delayMicroseconds(val);
  digitalWrite(pin,LOW);
  delay(20);
}
```

Modeliranje

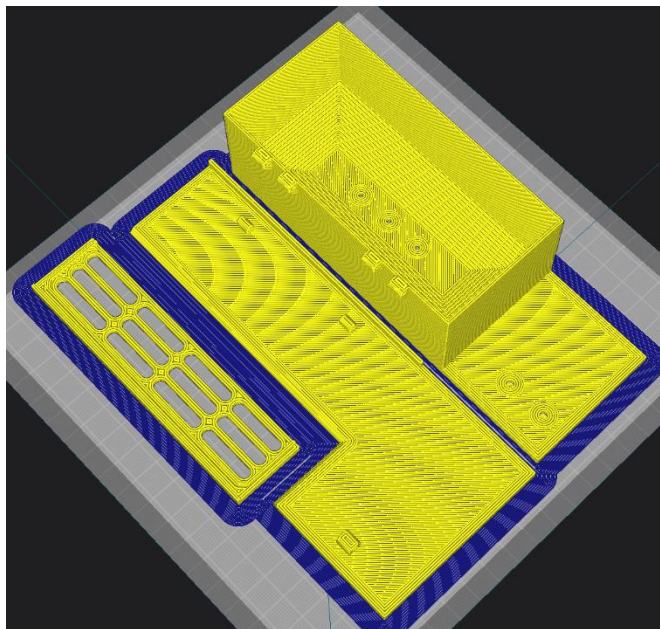
Sedveda nisem mogel iti mimo tega, da bi samo gladal kako se motor premika in led diode utripajo med tem ko jaz skupaj devam dve ali tri žičke za nivo. Najprej sem pomislil da bi uporabil kozarec za prikaz delovanja samo nisem vedel kako bi naredil odpiranje vrat.

Ker imam doma 3D printer sem se odločil zmodelirati korito in vrata. Pri modeliranju korita nisem imel večjih težav, ko pa sem prišel do vrat sem ugotovil, da bom mogel narediti tudi nekakšne rešetke, ki bodo omogočale pritok vode v korito. Za lažjo predstavbo v kakem merilu bi naredil korito sem si rahlo pomagal z igračo avtomobila. Korito sem naredil 12cm dolgo in 6cm široko. Da sem lahko nekam pritrdil motor sem dno korita iztegnil in naredil dodaten prostor za motor. Korito je bilo opremljeno z tremi luknami za iztekanje vode in rahlo klančino, ki omogoča lažje odtekanje vode. Vrh korita bo polovico prekrivala rešetka in polovica vrata. Na vratih sem naredil luknje za montažo na okvir motorja. Ko sem natisnil prvi model sem ugotovil, da se varta preveč skrijejo za korito. Zato sem model popravil in naredil tečaj kot ga lahko vidimo pri različnih skrinjah za shranjevanje. Vrata sem nato na motor pritrdil kar z gumico.



Slika 4: Spredaj stari model, zadaj novi model

Tako je bilo večino dela končanega, prevrtati sem moral samo še luknje za žičke, ki bodo merile nivo vode. To sem naredil z biaksom in pištolo za vroče lepljeneje. Lepilo je lepo zatesnilo luknje in pine, ki so gledali ven iz korita.



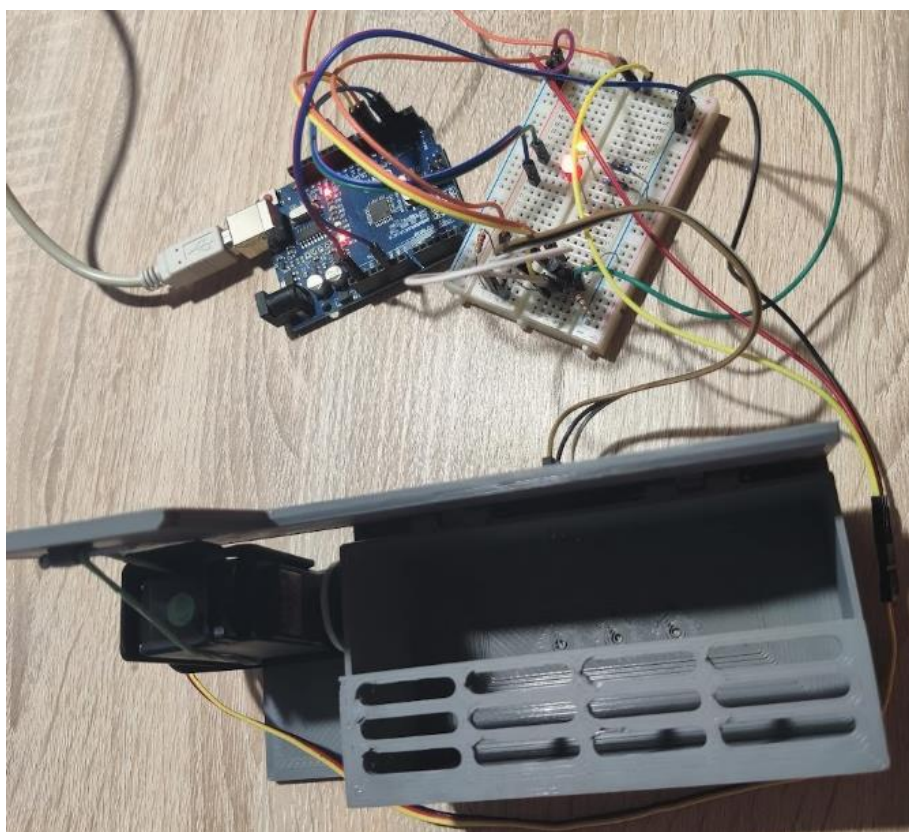
Slika 5: Priprava na tiskanje

Zaključek

Projekt je bil uspešno zaključen saj bi nas tak koncept varoval pred poplavami. Sem pa opazil nekaj podrobnosti med dolom na tem projektu. Vsaka stvar ima prednosti in slabosti. Opazil sem, da je knjižnica *servo* boljša za dvigovanje. Knjižnica drži pri meru motor z pravilnim pwm-jem. Kar je v sami kodi kar težko narediti. Je pa moja implementacija boljša za zapiranje saj ko motor izklopimo se lahko sam prilagodi in ga ne silimo v neki poziciji.

Če bi model hoteli uporabljati na prostem in ob poplavah bi tudi morali dodati baterijo, ki bi služila za napajanje sistem. V tem primeru pa bi bilo optimalno vgašanje arduino kontrolerja. To bi lahko naredili z dodatnim modulom, ki pošilja signale na določenem intervalu. Tako bi lahko na vsake pet sekund zbujali arduino samo, da bi odčital stanje in če bi bilo stanje na srednjem nivoju se enostavno nebi ugasnil, dokler nebi voda uplahnila tako bi lahko zagotovili konstantnost in vseeno prihranili pri energiji. Kot pa sem že omenil bi bilo v realnosti boljše uporabiti elektronske cilindre, saj so v primerjavi z servo motorjem bolj nadzorljivi in močnejši.

Za testiranje sem tudi pripravil način za testiranje na spletni strani [thinkercad.com](https://www.thinkercad.com), na [gitHubu](https://github.com) lahko najdete kodo in pa video predstavitev na [YouTub-u](https://www.youtube.com).



Slika 6: Zaključen projekt

Povezave na portale

GitHub – <https://github.com/ZanJuvan/floodGates>

YouTube – <https://youtu.be/El3Q7Rk2JAE>

ThinkerCad - <https://www.tinkercad.com/things/jE8gs1S2fXO?sharecode=5jYKY0pRq9Eeay-WjD405uC-mTn2bVBv0Gq23O8FCFE>