

»Mladi za napredek Maribora 2017«

34. srečanje

Sistem skladiščenja z razvojno ploščo Arduino in čitalcem črtnih kod

Raziskovalno področje: ELEKTROTEHNIKA, ELEKTRONIKA

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor:	NEJC GROBELNIK, JURE GOLOB
Mentor:	MILAN IVIČ
Šola:	SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA MARIBOR

Maribor, januar 2017

Kazalo vsebine

1.	POVZETEK	4
2.	ZAHVALA.....	4
3.	VSEBINSKI DEL.....	4
3.1.	Uvod.....	4
3.2.	Napredni sistemi skladiščenj.....	4
3.3.	Razvojna plošča Arduino.....	7
3.4.	Serijski monitor	8
3.5.	Črtne kode.....	11
3.6.	Čitalci črtnih kod.....	12
3.7.	Opis našega skladiščnega sistema.....	13
3.8.	Časovne prekinitev in opis programske kode	19
4.	DRUŽBENA ODGOVORNOST	26
4.1.	Realna praksa problema	25
5.	ZAKLJUČEK	26
6.	VIRI	27
6.1.	Viri vsebine.....	27
6.2.	Viri slik.....	27

Kazalo slik

Slika 1:	Skladiščni roboti v skladišču spletne trgovine Amazon (vir: Amazon).....	5
Slika 2:	Sistem skladiščenja s pametnimi očali (vir: Picavi).....	6
Slika 3:	Razvojna plošča arduino UNO (vir: Avtor naloge)	7
Slika 4:	Programsko okolje Arduino IDE (vir: Avtor naloge)	7
Slika 5:	Serijski monitor (vir: Avtor naloge)	8
Slika 6:	Izpisovanje podatkov na serijskem monitorju (vir: Avtor naloge)	9
Slika 7:	Vpisovanje vrednosti v serijski monitor (vir: Avtor naloge).....	10
Slika 8:	Primer kode EAN13 (vir: Wikipedija)	11
Slika 9:	TF (Interleaved 2 of 5) (vir: Computalabel).....	11
Slika 10:	Shema delovanja čitalca črtnih kod (vir: explainthatstuff)	13

Slika 11: Vezava gumbov za izbiro funkcije (vir: Avtor naloge).	13
Slika 12: Odlagalna mesta v skladišču (vir: Avtor naloge).....	14
Slika 13: Dodeljevanje črtnih kod izdelkom ob prihodu v skladišče (vir Avtor naloge).	15
Slika 14: Vnašanje črtnih kod v program s pomočjo čitalca (vir: Avtor naloge).	15
Slika 15: Vnos paketov v skladiščni sistem (vir: Avtor naloge).	16
Slika 16: Iskanje tovara v skladišču (vir: Avtor naloge)	17
Slika 17: Odjavljanje tovara v sistemu (vir: Avtor naloge).	18
Slika 18: Izpis tovara v skladišču (vir: Avtor naloge).	19
Slika 19: Prikaz delovanja prekinitev (vir: Avtor naloge).....	20
Slika 20: Biti v registru TCR2B (vir: Avtor naloge).....	20
Slika 21: Štetje timerja2 ob nastavitvi registra TCNT2 na 130 (vir: Avtor naloge).....	21

1. POVZETEK

V tej raziskovalni nalogi smo preučili razne sisteme skladiščenj, prav tako pa smo ustvarili svojega, s katerim smo se skušali čim bolj približati najsodobnejšim sistemom.

Naš sistem je zasnovan tako, da se lahko poveže s katerim koli osebnim računalnikom, preko katerega lahko podatke sprejemamo oziroma vnašamo s pomočjo čitalca črtnih kod. V prostoru za shranjevanje oziroma v skladišču smo odlagalna mesta razdelili v sistem, kot je pri potapljanju ladjic (tabela s številkami in črkami). Vsakemu odlagalnemu mestu smo dodelili številsko vrednost v obliki črtnih kod (vrednosti od ena do šest), ki jo lahko preberemo s čitalcem in informacijo posredujemo osebnemu računalniku. Prav tako smo izdelali črne kode (te imajo prav tako številsko vrednost, vendar od sedem do dvanajst), ki se naključno dodelijo tovoru, ki pride v skladišče, da se ga kasneje lažje najde in identificira. Prav tako pa smo v maketi skladišča pod vsa odlagalna mesta namestila fotoupore, ki delujejo kot senzorji za nadzor tovara na odlagalnem mestu.

2. ZAHVALA

Zahvaljujemo se mentorju za podporo in pomoč pri izdelovanju te raziskovalne naloge. Zahvaljujemo se tudi sošolcem in prijateljem za razne predloge.

3. VSEBINSKI DEL

3.1.Uvod

Za raziskovalno nalogo smo se odločili zato, ker nas je zanimalo, kako delujejo razni skladiščni sistemi. Naš namen je bil spoznati razne sisteme shranjevanja oz. skladiščenja in ustvariti podoben skladiščni sistem. Sistemi, ki nam pomagajo stvari shraniti oziroma skladiščiti imajo dandanes izjemen pomen, saj si ne moremo privoščiti dolgotrajnega iskanja artiklov po skladišču. S tem prihranimo na času, mnogim podjetjem pa pripomoremo k zmanjšanju stroškov.

3.2. Napredni sistemi skladiščenj

Zaradi hitrega napredka tehnologije, se je tehnološko razvila tudi logistika in skladiščenje. Dandanes je zato delo v skladiščih lažje za delavca, saj so se razvile različne elektronske naprave, ki omogočajo hitrejšo delo z manj napora.

Sistem skladiščenja z roboti

Spletna trgovina Amazon, ki je med največjimi spletnimi trgovinami na svetu, v svojem skladišču uporablja sistem, ki je namenjen predvsem večjim skladiščem. Je zelo hiter in ergonomičen za zaposlene. Sistem predstavljajo roboti in odlagalne police, ki jih roboti premikajo po skladišču. Na tleh pa so nalepljene posebne **QR kode**¹, ki robotu povedo, kje se nahaja. Tovoru oziroma predmetu, ki pride v skladišče, se določi mesto na odlagalni polici. Nato delavec, s pomočjo čitalca črtnih kod vnese podatke v sistem, ki povedo, na kateri odlagalni polici se tovor nahaja. Roboti odnesejo odlagalno polico v skladišče na pristoječe mesto. Robot oziroma sistem si mesto, kjer je polica postavljena, zapomni s pomočjo QR kod, ki jo robot prebere. Sčasoma sistem analizira podatke ter ustvari seznam tovara, ki najpogosteje odhaja iz skladišča. Roboti sortirajo odlagalne police po skladišču tako, da police s tovorom, ki najpogosteje odhajajo iz skladišča, postavi bližje izhodu.

MNENJE: Sistem je zelo hiter, vendar je slabost tega sistema ta, da ima službo zaradi robotov vse manj ljudi.



Slika 1: Skladiščni roboti v skladišču spletne trgovine Amazon (vir: Amazon)

Sistem skladiščenja s pametnimi očali

Ta skladiščni sistem uporablja najsodobnejšo tehnologijo (očala s posebno lečo, ki služi kot prikazovalnik, prav tako je v očala vgrajena kamera, ki služi kot čitalec črtnih kod). Sistem deluje tako, da delavec v skladišču s pomočjo kamere, vgrajene v očala, prebere kodo naročila. Na leči se izpiše črna koda tovara oziroma paketa, ki ga mora delavec najti in ga s

¹ QR koda – vrsta dvodimenzionalne (vertikalno-horizontalne) črtne kode

čitalcem črtnih kod prebrati. V primeru, da delavec prebere napačno kodo, se mu na leči izpiše, da je prebral napačni tovor, v obratnem primeru lahko delavec pobere tovor oziroma paket in ga pripravi za odhod.

MNENJE: Sistem je sicer zelo napreden, kljub temu ohranja delovna mesta, prav tako pripomore k temu, da ima delavec ves čas proste roke. Slabost, ki sva jo našla pri tem sistemu je ta, da delavec izgubi veliko časa med iskanjem tovara, saj mora pravo črtno kodo poiskati sam.



Slika 2: Sistem skladiščenja s pametnimi očali (vir: Picavi).

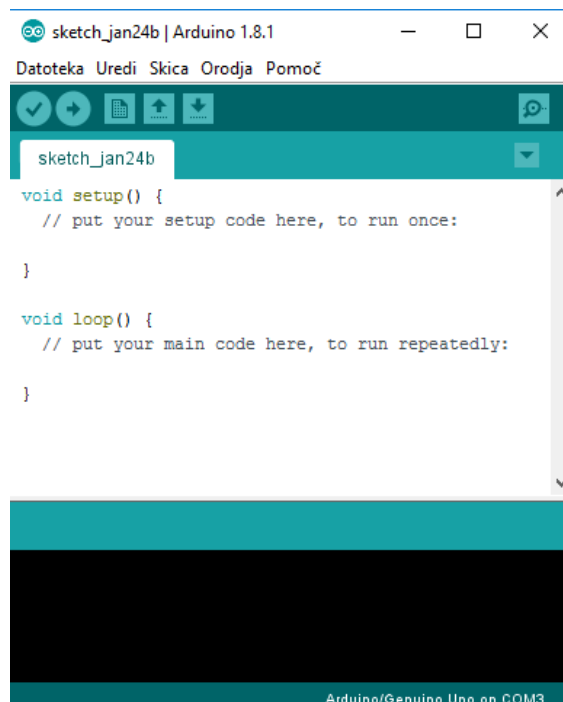
3.3. Razvojna plošča Arduino

Odločili smo se, da bomo pri našem projektu uporabili razvojno ploščico Arduino UNO. Arduino je podjetje, ki izdeluje razvojne plošče, ki temeljijo na mikrokrmilnikih Atmel. Najpogostejše uporabljene razvojne ploščice so Arduino UNO, Mega in Nano.



Slika 3: Razvojna plošča arduino UNO (vir: Avtor naloge)

V tej raziskovalni nalogi smo uporabili razvojno ploščico Arduino UNO, ki je primerna predvsem za manjše elektronske projekte. Arduino UNO temelji na 8-bitnem mikrokrmilniku Atmel Atmega328P. Ima 14 Digitalnih vhodno/izhodnih priključkov, prav tako pa ima 6 analognih vhodnih priključkov. Takt mikrokontrolerja, ki ga narekuje oscilator, je 16 MHz, delovna napetost pa znaša 5 V.

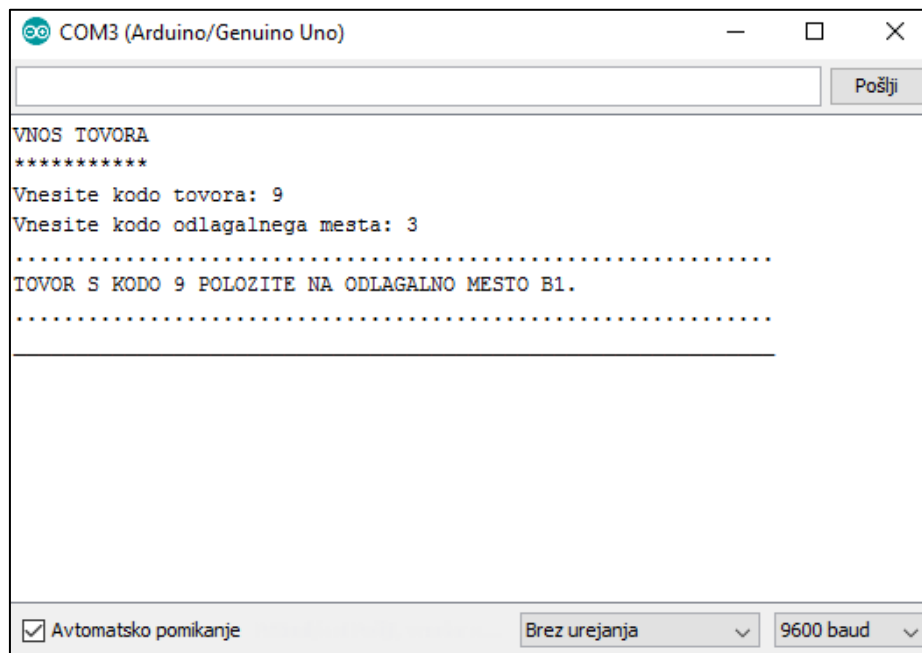


Slika 4: Programsko okolje Arduino IDE (vir: Avtor naloge)

Program, za katerega želimo, da ga razvojna ploščica izvede, napišemo v programskem okolju Arduino IDE. Sestavljen je iz dveh delov, funkcij, void setup() in void loop(). V prvem delu kode, void setup(), nastavimo vse potrebno za delovanje priključenih elementov, določimo vhodne in izhodne priključke. V drugem delu programske kode, void loop(), pa zapišemo program, ki se neprekinjeno ponavlja.

3.4. Serijski monitor

Serijski monitor je okno, s pomočjo katerega lahko komuniciramo z Arduino ploščico prek serijskih vrat. V serijski monitor lahko podatke vnašamo ali pa jih sprejemamo. Serijski monitor bomo v tej raziskovalni nalogi uporabili tako, da bomo lahko podatke vnašali in brali. Vnašali bomo podatke, na katerem odlagalnem mestu se nahaja paket oziroma tovor. Na našo zahtevo se bodo podatki na serijski monitor izpisovali, če bomo želeli priti do informacije, na katerem odlagalnem mestu se nahaja paket oziroma tovor.



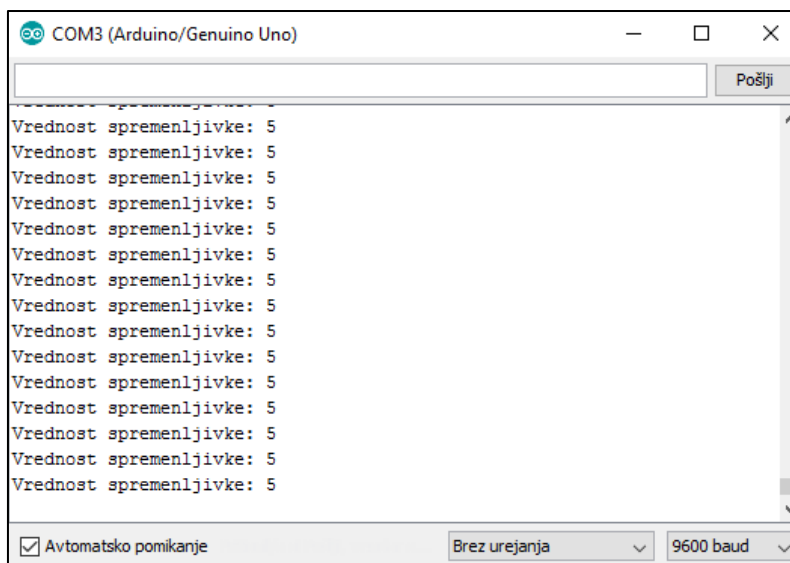
Slika 5: Serijski monitor (vir: Avtor naloge)

Da vzpostavimo komunikacijsko zvezo med računalnikom in Arduino ploščico, uporabimo operator `Serial.begin()`, ki ga vstavimo v prvi del kode void setup(). Med oklepaje operatorja nastavimo hitrost prenosa podatkov. Najpogostejše hitrost prenosa podatkov nastavimo na vrednost 9600 Baudov oziroma bitov na sekundo.

```
void setup() {  
  Serial.begin(9600);  
}
```


Za izpisovanje podatkov na serijskem monitorju uporabimo operator `Serial.print()`. Tako se nam podatki na serijskem monitorju izpisujejo v isti vrsti, če pa želimo da se podatki izpisujejo eden pod drugim oziroma vsak v svoji vrsti, pa uporabimo operator `Serial.println()`. Podatek, za katerega želimo da se izpiše na serijskem monitorju, vpišemo med oklepajema. Če želimo da je izpisan podatek poljubno besedilo, med oklepaje operatorja postavimo besedilo med narekovaje. V primeru, da želimo da se izpisuje vrednost neke spremenljivke, pa med oklepaje postavimo samo ime spremenljivke.

```
void loop () {  
    int Vrednost = 5;  
  
    Serial.print ("Vrednost spremenljivke: ");  
    Serial.print (Vrednost);  
}
```



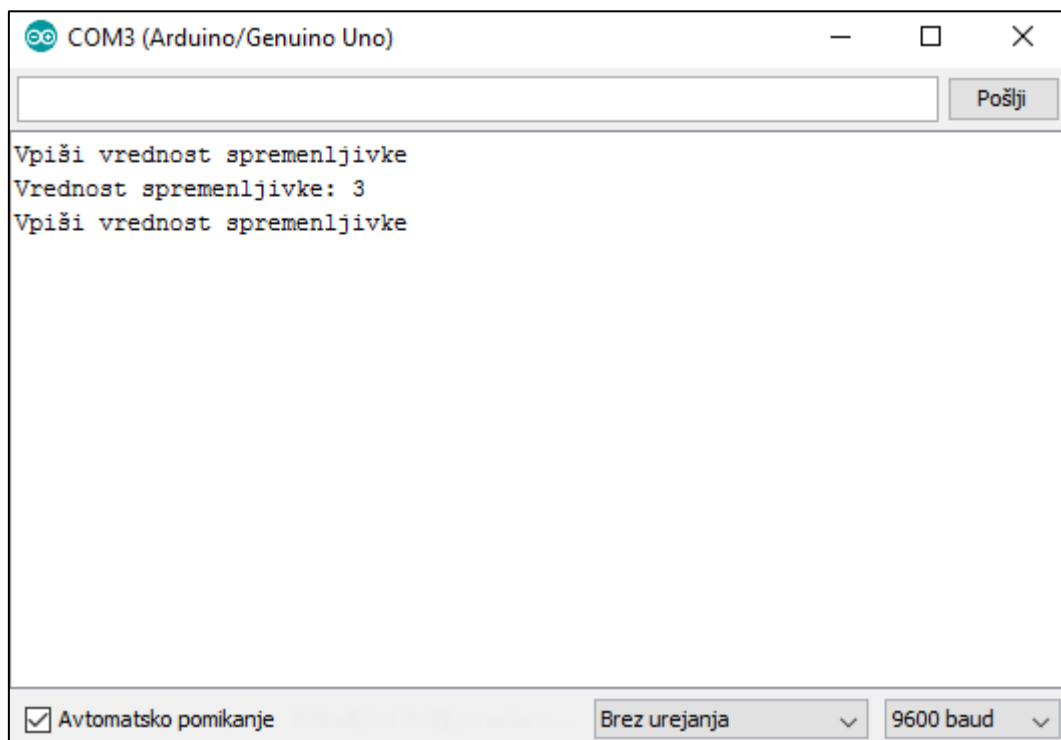
Slika 6: Izpisovanje podatkov na serijskem monitorju (vir: Avtor naloge)

Prav tako lahko v serijski monitor podatke tudi vnašamo, na primer lahko določimo vrednost spremenljivk. To izvedemo tako, kot kaže primer spodaj. Medtem ko vpisujemo vrednost spremenljivke v polje za vpisovanje, mora program počakati, zato uporabimo zanko **while()**, kar v angleškem jeziku pomeni medtem.

Med navadne oklepaje zanke **while()** vstavimo pogoj, med zavite oklepaje, pa programsko kodo, ki se izvaja tako dolgo, dokler je vpisani pogoj resničen. V našem primeru bo pogoj (**Serial.available == 0**), kar pomeni, dokler ni nič vpisano izvedi sledeči program, program ki

pa ga želimo, da se med tem časom izvaja pa je nič, zato prostor med zavirami oklepaji pustimo prazne. Torej, dokler v polje za vpisovanje ne vpišemo ničesar, se program ne bo izvajal, vendar ko pa vpišemo vrednost, v našem primeru število, pa se program nadaljuje. Nato moramo Arduino povedati, kateri spremenljivki smo dodelili vrednost, kar pa izvedemo z operatorjem **Serial.parseInt()**.

```
void loop() {  
  int Vrednost = 0;  
  
  Serial.println ("Vpiši vrednost spremenljivke ");  
  while(Serial.available()==0){ }  
  Vrednost = Serial.parseInt();  
  
  Serial.print ("Vrednost spremenljivke: ");  
  Serial.println (Vrednost);  
}
```



Slika 7: Vpisovanje vrednosti v serijski monitor (vir: Avtor naloge)

3.5.Črtne kode

Črna koda je kombinacija različno debelih temnih črt in prav tako različno širokih svetlih presledkov. Črna barva se uporablja za črte in bela barva za presledke. Najosnovnejša lastnost črtnih kod je ta, da vsebujejo nek podatek, ki je najpogosteje zapisan v številkah, črkah ali celo v nekaterih znakih.

Ideja črtne kode prihaja iz ZDA, kjer se je porodila dvema ameriškima inženirjema že leta 1948. Že v začetku so se črtne kode uporabljale za določanje artiklov v trgovinah (od tam jih še danes najbolj poznamo). Njihova prednost je nezmotljivost – človek se, ko npr. vtipka neko število, zlahka zmoti, se zatipka. Ideja se je v ZDA precej obnesla in sistem s črtnimi kodami ter čitalci črtnih kod se je širil po svetu, kmalu tudi v Evropo.

S tehnološkim razvojem so se razvile različne vrste, različni tipi črtnih kod od katerih ima vsaka neko prednost:

EAN-13 (European Article Numbering)

Koda EAN13 se je uveljavila na evropskem tržišču za označevanje artiklov in proizvodov kot koda, ki točno določa državo in podjetje nastanka. Običajno prvi trije znaki definirajo državo oz. nacionalno organizacijo, ki je izdala številko, naslednjih štiri, pet ali šest mest pove proizvajalca artikla preostalih pet, štiri ali tri mesta dodeli artiklom proizvajalec, 13-ti znak je kontrolni znak, ki se po posebnem algoritmu izračuna na osnovi predhodnih dvanajsti števil in preverja točnost celotne številke izdelka. Koda EAN13 se je uveljavila kot standard pri označevanju različnih publikacij pod imenom ISBN koda.



Slika 8: Primer koda EAN13 (vir: Wikipedija)

TF (Interleaved 2 of 5 – prekrivajoča koda 2 od 5)

Gre za numerično kodo višje gostote, ki se največkrat uporablja v skladiščih in v aplikacijah v težki industriji. Zaradi možnosti napačnega odčitavanja se okrog kode dostikrat uporablja črni okvir, ki preprečuje, da bi laserski žarek pri delnem preletu kode prenesel napačne podatke v aplikacijo.



Slika 9: TF (Interleaved 2 of 5) (vir: Computalabel)

RSS simbolika

Skrčena simbolika (RSS) je družina linearnih simbolik, ki omogoča kodiranje 14-mestnih števil globalne trgovinske identifikacije (GTIN) EAN UCC. Z RSS naj bi izkoristili prednosti popolne identifikacije proizvodov in drugih aplikacij za preskrbovalne verige, v situacijah s prostorskimi omejitvami, kjer ne moremo normalno uporabiti že obstoječih linearnih simbolik.

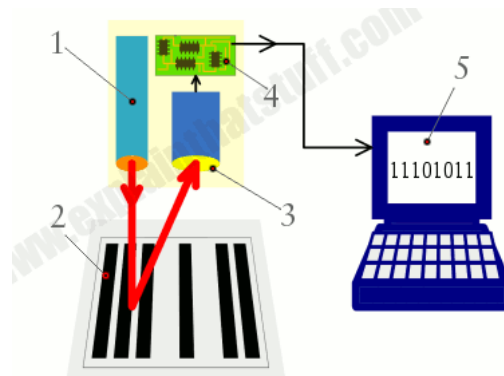
Koda 128

V njej lahko kodirajo tako števila kot tudi črke in določeni simboli. Prednost kode pred drugimi je tudi izredna zgoščenost, s čimer lahko na majhni površini zapišemo veliko število podatkov. Zaradi svojih lastnosti omogoča čitalnikom črtne kode hitro in zanesljivo odčitavanje.

Kontrolni znak (Checksum) je dodaten znak pripet črtni kodi, ki zagotavlja njeno pravilno odčitavanje. Ni nujno prisoten pri vseh črtnih kodah. Pri nekaterih kodah kot je na primer ITF koda je nujen, saj je tam nagnjenost k napakam pri odčitavanju velika. Medtem ko ima koda 128 že sama po sebi vgrajen kontrolni znak in dodatnega preverjanja običajno ne potrebuje.

3.6.Čitalci črtnih kod

Črtne kode, ki jih imamo npr. natisnjene na nekem artiklu moramo "prebrati", jih pretvoriti v nam razumljivo obliko (črke, številke, znaki). Vrednost na črtni kodi je zapisana v obliki binarne kode, temni deli predstavljajo 1, beli pa 0, binarna mesta pa določajo debeline črt. Da lahko preberemo binarno vrednost črtne kode, uporabljamo čitalce črtnih kod. Čitalci črtnih kod temeljijo na laserskem žarku, ki se premika levo in desno ter se ob tem ko sveti na svetlo podlago oz. temno črto različno močno odbije nazaj proti čitalcu. To razliko v odbijanju pa zaznavamo v notranjosti čitalca tik ob izvoru svetlobe. Čitalec nato prevedeno črtno kodo, (uporabljajo se bralniki, ki imajo vgrajen samo program za dešifriranje črtne kode) pošlje vmesniku ali na terminal računalnika zaporedno s tipkovnico, tako da program v računalniku ne opazi razlike med ročno vnesenimi podatki in tistimi, ki so prišli skozi čitalnik, v našem primeru dobi ta podatek računalnik po kablu.



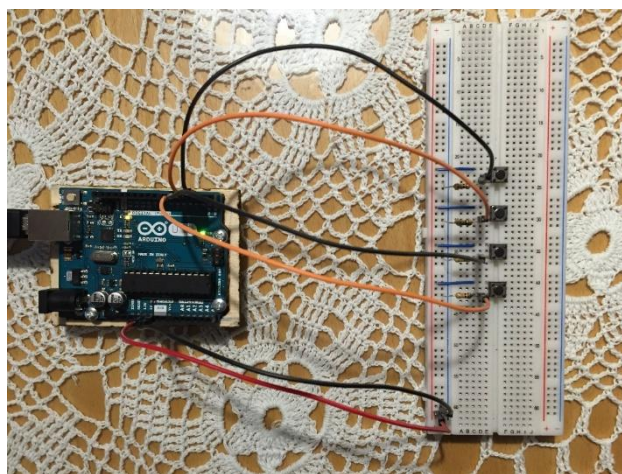
Slika 10: Shema delovanja čitalca črtnih kod (vir: explainthatstuff)

Primer v knjižnicah

V knjižnicah prejme danes vsak član identifikacijsko kartico, na kateri je poleg vseh podatkov tudi črna koda. Ker je s črtno kodo označeno tudi vse gradivo, ki je članom na voljo, je s takšno kartico zagotovljena avtomatizirana izposoja gradiva. Uporaba papirja v knjižnicah odpade (od fizične evidence na karticah, ki smo jim bili priča še pred kakšnim desetletjem, je evidenca zdaj ves čas dostopna v računalniku, celo na spletu). V nekaterih primerih se za članske izkaznice uporablja tudi magnetni zapis, ki se odčita s čitalniki magnetnih zapisov. Zaradi vsak dan večje zbirke knjižnega in časopisnega gradiva, obstaja zanje poseben sistem enotnega številčenja.

3.7. Opis našega skladiščnega sistema

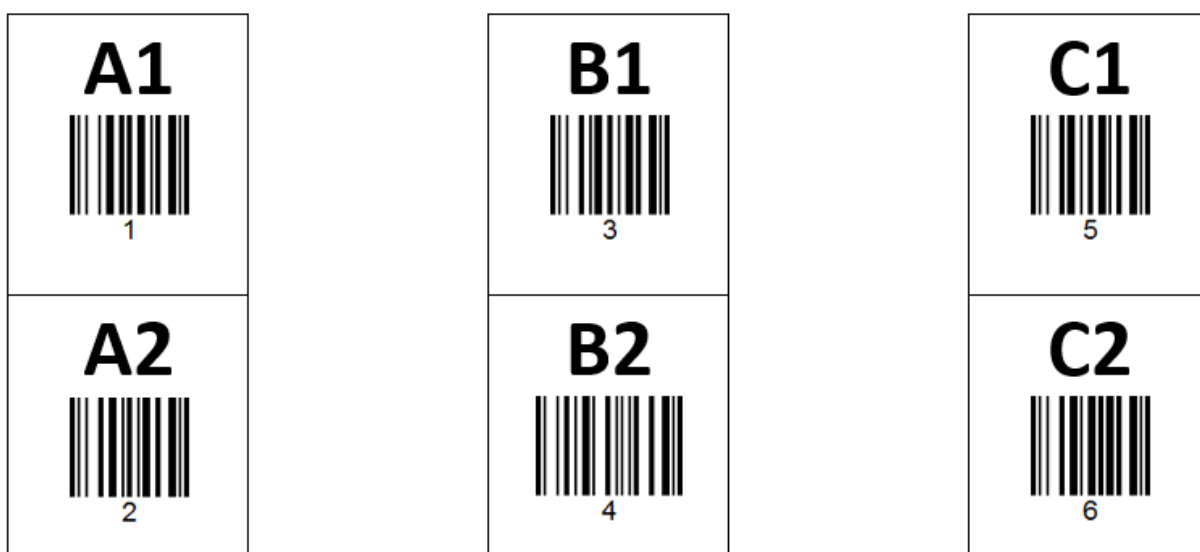
S sistemom za skladiščenje, ki smo ga izdelali, smo se želeli čim bolj približati najsodobnejšim skladiščnim sistemom, torej želeli smo izdelati takšnega, ki je čim lažji za uporabo in hkrati hiter. Uporabili smo štiri gumbе, za izbiranje funkcij v programu, v sistem pa lahko vnesemo največ šest tovorov oziroma paketov.



Slika 11: Vezava gumbov za izbiro funkcije (vir: Avtor naloge).

Odlagalna mesta v skladišču

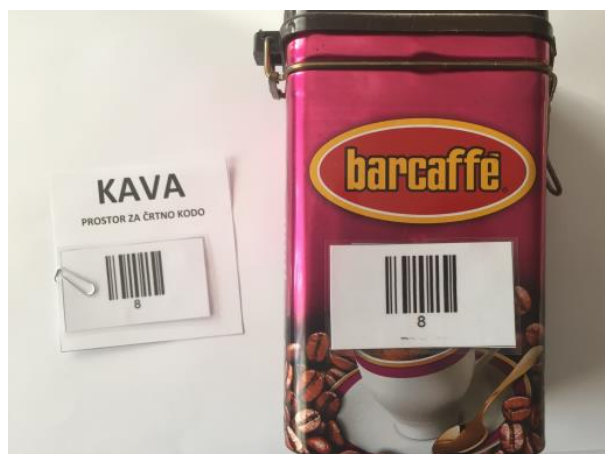
V skladišču smo določili šest odlagalnih mest, saj prav tako v naš sistem lahko vnesemo samo šest tovorov. Odlagalnim mestom v skladišču smo vsakemu posebej dodelili unikatno številsko vrednost od ena do šest, ki je prav tako zapisana v obliki črtne kode, ki smo jih izdelali s pomočjo generatorja črtnih kod na spletni strani: <http://barcode.tec-it.com/en>. Prav tako smo odlagalna mesta razdelili v tri sektorje: A, B in C, v vsakem sektorju pa sta dve odlagalni mesti. Tako pridemo do sistema, kot je pri potapljanju ladjic, ki je bolj nadzoren.



Slika 12: Odlagalna mesta v skladišču (vir: Avtor naloge)

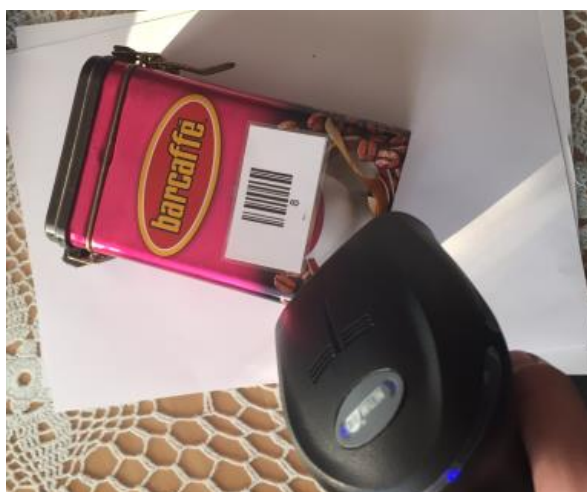
Vnašanje tovara v skladišče

Za predstavo delovanja našega sistema si predstavljamo, da bomo v tem skladišču shranjevali izbrane izdelke: Kava, sol, čaj in bonbone. Da jim bomo lažje sledili, jim bomo ob prihodu v skladišče dodelili unikatne črtne kode (vsakemu izdelku svojo). Črtne kode za označevanje tovorov smo prav tako izdelali s pomočjo generatorja črtnih kod, vendar smo jim določili številске vrednosti od sedem do dvanajst, tako da niso enake tistim na odlagalnih mestih, da ne bi prišlo do pomote pri odčitavanju. Vsaka črna koda za označevanje izdelkov v skladišču je podvojena, saj ob prihodu tovara v skladišče eno črtno kodo dodelimo izdelku samemu, drugo pa na dokument tega tovara, tako da kasneje vemo, kateremu izdelku smo dodelili neko črtno kodo in obratno (glej sliko 13).



Slika 13: Dodeljevanje črtnih kod izdelkom ob prihodu v skladišče (vir: Avtor naloge).

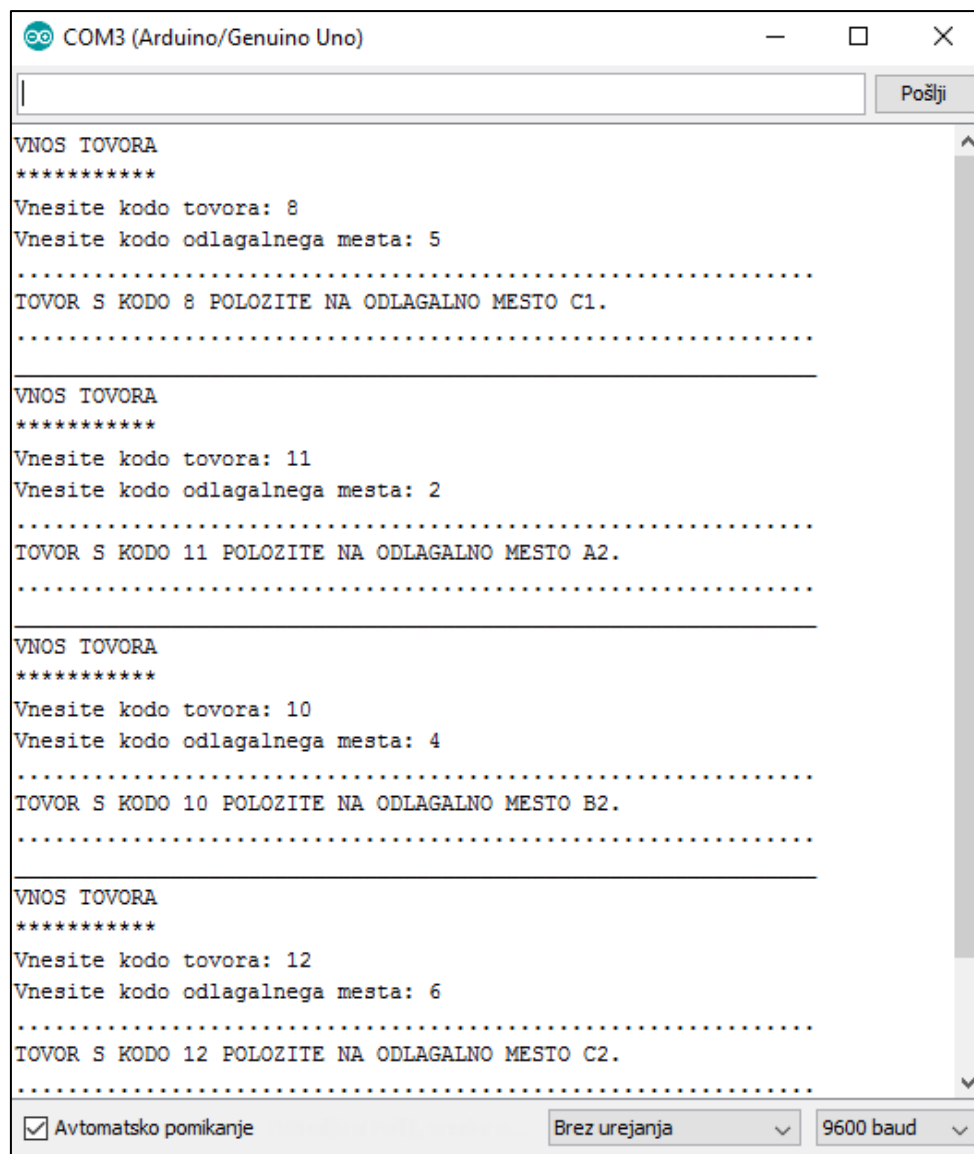
Nato skladiščnik določi, na katero prosto odlagalno mesto bo postavil ta izdelek. Da to mesto lažje najdemo, smo v skladišče namestili senzorje (fotoupore), ki nadzirajo zasedenost odlagalnih mest, nad odlagalna mesta pa smo nastavili rdeče LED diode, ki svetijo takrat, kadar je odlagalno mesto zasedeno. Ko najdemo prosto odlagalno mesto, pritisnemo enega izmed gumbov za izbiro funkcije, in sicer tistega, ki je predstavlja vnos tovara. Nato se nam na serijskem monitorju izpiše besedilo, ki nam pove, da moramo vnesti črtno kodo izdelka. Če preberemo napačno kodo, na primer črtno kodo nekega odlagalnega mesta (ena do šest), bo program javil napako, v nasprotnem primeru, torej če je vrednost črtne kode med sedem in dvanajst, pa bomo lahko z vnašanjem nadaljevali in program bo zahteval naj vnesemo črtno kodo odlagalnega mesta, na katerega bomo položili ta izdelek. Ko to storimo, si program zabeleži vnesene podatke.



Slika 14: Vnašanje črtnih kod v program s pomočjo čitalca (vir: Avtor naloge).

Za prikaz bomo v program vnesli štiri izdelke: **Kavo**, ki smo ji dodelili črtno kodo z vrednostjo osem in jo bomo položili na odlagalno mesto C1, **Čaj**, kateremu smo dodelili črtno kodo

enajst in ga bomo položili na odlagalno mesto A2, **Bonbone**, ki smo jim dodelili črtno kodo z vrednostjo deset in jih bomo položili na odlagalno mesto B2, prav tako pa bomo skladiščili še **Sol**, ki pa smo ji dodelili črtno kodo z vrednostjo dvanajst, skladiščena pa bo na mestu C2. Primer vnosa tovora prikazuje spodnja slika, slika 15.



Slika 15: Vnos paketov v skladiščni sistem (vir: Avtor naloge).

Iskanje tovora

Kadar želimo tovor v skladišču poiskati, pritisnemo gumb, ki sproži program za iskanje tovora v skladišču. Serijski monitor izpiše, naj vnesemo črtno kodo tovora oziroma paketa, ki jo imamo pripeto na dokumentu tega paketa. Ko to opravimo, se nam izpiše besedilo, ki nam

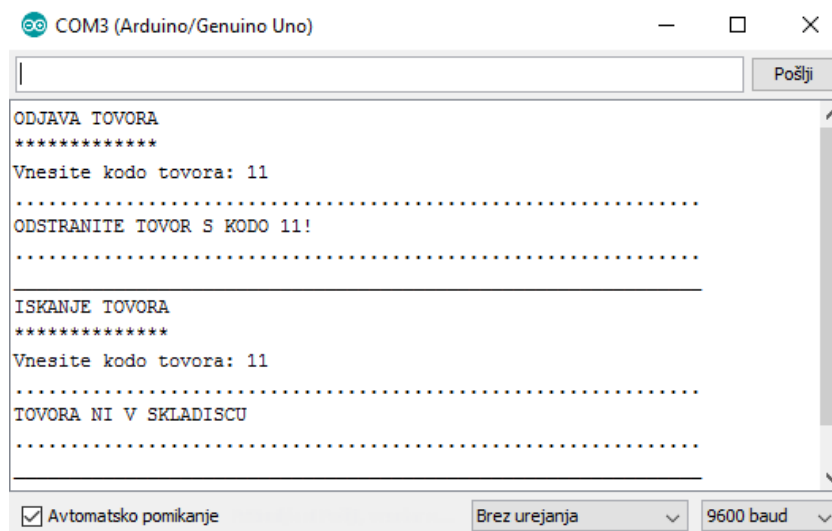
pove, na katerem odlagalnem mestu se nahaja iskani tovor. Če na primer želimo poiskati sol, s čitalcem preberemo črtno kodo, ki smo jo pritrdili na dokument tovora soli.



Slika 16: Iskanje tovora v skladišču (vir: Avtor naloge)

Odjava tovora

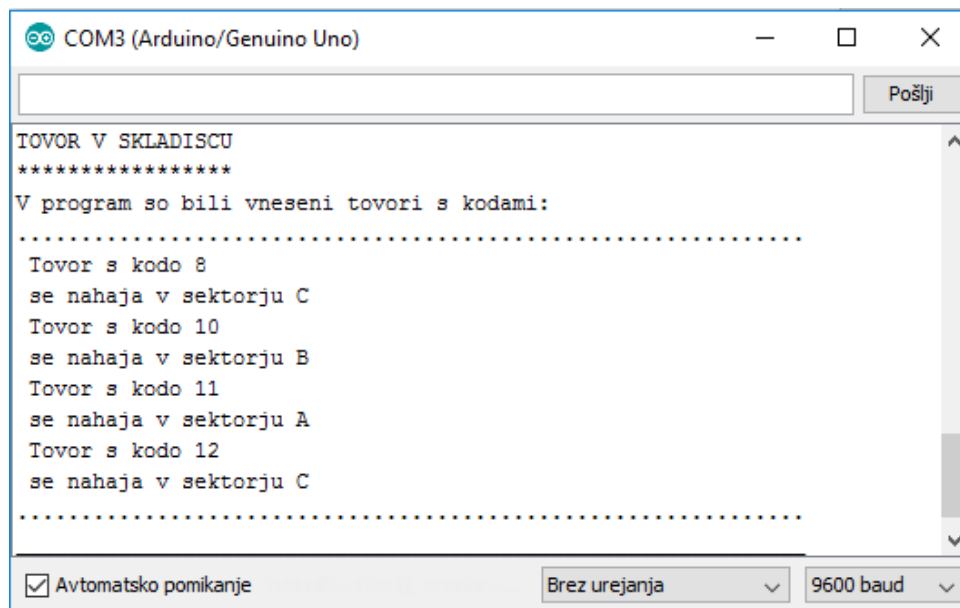
Kadar nekega tovora več ne želimo skladiščiti, ga moramo v programu odjaviti. To storimo tako, da pritisnemo gumb za odjavo tovora, serijski monitor pa nas vpraša, kateri tovor želimo odjaviti. Črtno kodo tega izdelka oziroma tovora enostavno preberemo s čitalcem, tako si program zabeleži, da tega izdelka ni več v skladišču. Črtno kodo, ki jo je imel ta izdelek, pa že lahko dodelimo naslednjemu, ki pride v skladišče. V našem primeru želimo odjaviti na primer čaj, tako preberemo njegovo črtno kodo, in če kasneje iščemo črtno kodo tega izdelka, nam bo serijski monitor podal podatek, da v skladišču tega tovora več ni.



Slika 17: Odjavljanje tovara v sistemu (vir: Avtor naloge).

Izpis tovara v skladišču

Ker velikokrat v skladiščih pride do požara ali katere druge katastrofe in je treba hitro oceniti škodo, smo se odločili, da sistemu dodamo gumb, s katerim sprožimo izpis kode tovorov na serijskem monitorju, ki so vnesene v sistem. Prav tako pa se ob vsaki kodi tudi izpiše podatek, v katerem sektorju se nahaja (sektorji v skladišču: A, B in C). Torej če bi v našem skladišču prišlo do katastrofe in bi se poškodoval sektor C, v programu odpremo *Izpis tovara v skladišču* in pogledamo, katere kode tovorov se nahajajo v tem sektorju. Vidimo, da se v sektorju C nahajajo tovari s kodami osem in dvanajst, nato pogledamo dokumente tovorov, in tako ugotovimo, katerim tovorom smo dodelili te kode. Na teh mestih sta bila sol in kava.

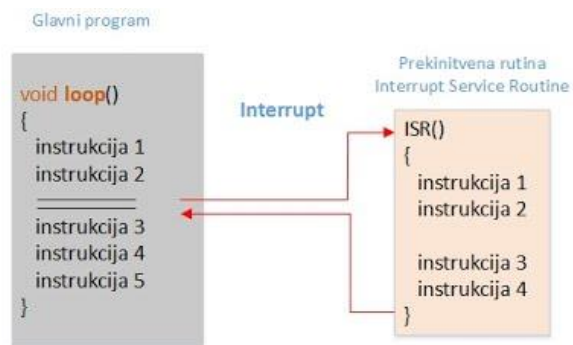


Slika 18: Izpis tovora v skladišču (vir: Avtor naloge).

3.8.Časovne prekinitve in opis programske kode

Naš sistem opravlja dve opravili, to je nadziranje prostih odlagalnih mest in program, za vnašanje in iskanje tovora v skladišču, vendar pa naša razvojna ploščica Arduino lahko v momentu izvaja samo en program. Ta problem smo rešili s časovnimi prekinitvami (Interrupt). S tem lahko dosežemo, da vsake toliko časa program skoči v prekinitveno rutino in ko jo izvede se vrne nazaj v glavni program točno tja, kjer je bil pred tem.

Časovne prekinitve lahko prožimo s časovniki (timerji). Arduino UNO vsebuje tri timerje: Timer0, Timer1 in Timer2. Timer0 se privzeto uporablja za zakasnitve – delay(), millis(), micros(), Timer1 se uporablja za krmiljenje servo motorčkov, Timer2 pa se uporablja za generiranje tonov. V našem primeru bomo uporabili Timer2. Ob prekinitvi mikrokontroler preneha izvajati glavni program (void loop()) in preide v prekinitveno rutino (ISR – Interrupt service routine), nato se vrne v glavni program in mikrokontroler nadaljuje program, kjer ga je pred prekinitvijo končal.



Slika 19: Prikaz delovanja prekinitev (vir: Avtor naloge).

Timer2 je 8 bitni števec, njegovo delovanje pa nadzira več registrov. Ob vsakem strojnem taktu se mu vrednost poveča za 1. Ker je Timer2 8-bitni pomeni, da lahko šteje do 255, nato pride do prekoračitve (overflow), vrednost je znova 0, timer pa ob tem sproži prekinitev, če smo jo dovolili. Strojni takt določa 16 MHz oscilator, iz tega lahko izračunamo čas trajanja ene periode, ko se timerjeva vrednost poveča za 1.

$$t_{\text{strojnega takta}} = \frac{1}{f} = \frac{1}{16 \text{ MHz}} = 0,0625 \mu\text{s}$$

Čas enega strojnega takta znaša 0,0625 μs .

$$t_{\text{prekinitve}} = 256 * t_{\text{strojnega takta}} = 256 * 0,0625 \mu\text{s} = 15,9375 \mu\text{s}$$

Timer2 doseže maksimalno vrednost vsakih 15,9375 μs . Povečevanje vrednosti Timerja2 lahko upočasnimo s preddelitvijo.

Če hočemo Timer2 uporabiti, ga moramo pravilno nastaviti. Timer2 vsebuje številne registre s katerimi nastavimo (določimo) njegovo delovanje. Dva od teh registrov, s katerima določimo njegovo delovanje sta TCCR2A in TCCR2B. S preddelilnikom lahko dosežemo, da se Timer2 poveča za 1 vsakih 8, 32, 64, 128, 256 ali 1024 strojnih ciklov. Preddelitev določimo s prvimi tremi biti v registru TCCR2B (CS20, CS21, CS22), ki je sicer 8 bitni.

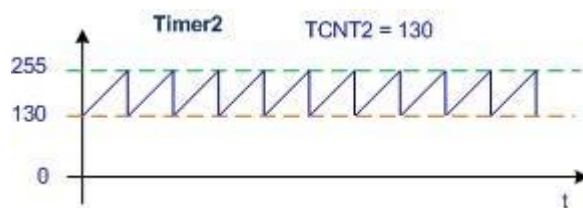


Slika 20: Biti v registru TCR2B (vir: Avtor naloge).

Tabela 1: Vloga bitov CS za preddelitev.

CS20	CS21	CS22	PREDDELITEV
0	0	0	Timer2 onemogočen
0	0	1	1:1
0	1	0	1:8
0	1	1	1:32
1	0	0	1:64
1	0	1	1:128
1	1	0	1:256
1	1	1	1:1024

S timerjevim 8-bitnim registrom TCNT2 lahko začetno vrednost Timerja2 nastavimo na vrednost, različno od 0. Če nastavimo njegovo vrednost na 130, bo Timer2 začel povečevati svojo vrednost od 130 naprej in vsakič, ko bo timer2 prekoračil vrednost 255, se bo vrnil na vrednost 130. V tem primeru bo Timer2 sprožil prekinitve po vsakih 125 strojnih ciklih ($255 - 130 = 125$).



Slika 21: Štetje timerja2 ob nastavitvi registra TCNT2 na 130 (vir: Avtor naloge).

Izračunajmo čas, v katerem se bo časovnik Timer2 povečal od vrednosti 130 do 255:

$$t = (255 - 130) * 0,0625 \mu s = 7,8125 \mu s$$

Če uporabimo še preddelilnik, tako da bita CS20 in CS22 postavimo na 1. S tem dosežemo preddelitev 1:128. To pomeni, da se bo vrednost časovnika Timer2 povečala za 1 po vsakih 128-ih strojnih ciklih. S tem smo povečali čas med prekinitvami na 1 ms:

$$t_1 = (255 - 130) * 128 * 0,0625 \mu s = 1 \text{ ms} \quad (f_1 = 1/t_1 = 1 \text{ kHz})$$

Timer2 se bo povečal od vrednosti 130 do 255 vsako milisekundo. Sedaj se bo prekinitev na časovniku Timer2 izvedla vsako milisekundo.

Opisane nastavitve časovnika Timer2 določimo na sledeči način:

```
TCCR2B = 0x00;    //Inicializacija, vse bite registra TCCR2B postavimo na 0. Operator 0x
                  //pomeni, da je število za njim (00) šestnajstiško.
                  // (00(16)) = 00000000(2)).

TCNT2 = 130;

TIFR2 = 0x00;    //Timer2 Interrupt Flag Register => pobrišemo zastavico prekinitve, ki
                  //se je postavila na 1, ko se je sprožila prekinitve.

TCCR2B = 0x05;    //Nastavimo preddelitev 1:128 (05(16)) = 00000101(2)) => CS20 in CS22
                  //bita v registru TCCR2B postavimo na 1.
```

Takšnemu načinu delovanja časovnika Timer2 pravimo normalni (Normal) način oziroma Normal timer mode.

OPIS PROGRAMSKE KODE za nadzor odlagalnih mest v skladišču

Najprej deklariramo spremenljivke, določimo jim imena in jim dodelimo vrednosti.

```
int StanjeM1 = A0;
int StanjeM2 = A1;
int StanjeM3 = A2;
int StanjeM4 = A3;
int StanjeM5 = A4;
int StanjeM6 = A5;

int Led1 = 6;
int Led2 = 7;
int Led3 = 8;
int Led4 = 9;
int Led5 = 10;
int Led6 = 11;
```

Nato v prvem delu kode – void setup() določimo vhodne in izhodne priključke z operatorjem pinMode().

```
void setup() {  
  pinMode(StanjeM1, INPUT);  
  pinMode(StanjeM2, INPUT);  
  pinMode(StanjeM3, INPUT);  
  pinMode(StanjeM4, INPUT);  
  pinMode(StanjeM5, INPUT);  
  pinMode(StanjeM6, INPUT);  
  
  pinMode(Led1, OUTPUT);  
  pinMode(Led2, OUTPUT);  
  pinMode(Led3, OUTPUT);  
  pinMode(Led4, OUTPUT);  
  pinMode(Led5, OUTPUT);  
  pinMode(Led6, OUTPUT);  
  
  noInterrupts()  
    TCCR2B = 0x00;  
    TCNT2 = 130;  
    TIFR2 = 0x00;  
    TIMSK2 = 0x01;  
    TCCR2A = 0x00;  
    TCCR2B = 0x05;  
}
```

Ker v času nastavitve časovnika ne sme priti do prekinitve, uporabimo funkcijo noInterrupts(). Nato vse bite registra TCCR2B postavimo na 0 (0x00 – šestnajstiški zapis). Vrednost Registra TCNT2 nastavimo na 130, tako da Timer2 začne naraščati od vrednosti 130 naprej. Prvi bit registra TIMSK2 postavimo na 1. S tem omogočimo overflow prekinitve na časovniku Timer2. Vsakokrat ko do Timer2 prekoračil svojo vrednost, se bo sprožila prekinitve. Vse bite registra TCCR2A postavimo na 0. Nato nastavimo preddelitev 1:128 tako, da prvi in tretji bit registra TCCR2B postavimo na 1.

```

void loop()
{
    glavna koda
}
ISR(TIMER2_OVF_vect)
{
    if(analogRead(StanjeM1)>900)
    {
        digitalWrite(Led1, HIGH)
    }
    else
    {
        digitalWrite(Led2, LOW)
    }
    if(analogRead(StanjeM2)>900)
    {
        digitalWrite(Led2, HIGH)
    }
    else
    {
        digitalWrite(Led2, LOW)
    }
    if(analogRead(StanjeM3)>900)
    {
        digitalWrite(Led3, HIGH)
    }

    else
    {
        digitalWrite(Led3, LOW)
    }
    if(analogRead(StanjeM4)>900)
    {
        digitalWrite(Led4, HIGH)
    }
    else
    {
        digitalWrite(Led4, LOW)
    }
}

```



```

    }
    if(analogRead(StanjeM5)>900)
    {
        digitalWrite(Led5, HIGH)
    }
    else
    {
        digitalWrite(Led5, LOW)
    }
    if(analogRead(StanjeM6)>900)
    {
        digitalWrite(Led6, HIGH)
    }
    else
    {
        digitalWrite(Led6, LOW)
    }
}

```

Na koncu kode, pod glavno zanko void loop(), vstavimo prekinitveni program (ISR), ki se bo izvedel vsakokrat, ko bo sprožena prekinitev. V našem primeru je to program, ki nadzira stanje na odlagalnih mestih. Kot je že omenjeno, odlagalna mesta nadzirajo fotoupori in če vrednost nekega fotoupora preseže 900, rdeča LED dioda nad tistim odlagalnim mestom zasveti.

Realna praksa problema

Zamislimo si proizvodnjo v kateri izdelujejo končne izdelke za prodajo. Za izdelavo delavci potrebujejo posebna ročna orodja (npr. za oblikovanje posebnih oblik), brez tega orodja proizvodnja ne steče do konca. Naše skladišče je kot nalašč za ta primer: ta posebna orodja bi opremili s črtnimi kodami, prav tako tudi prostore, kamor pospravimo orodje, ko ga ne potrebujemo.

V podjetje smo dobili naročilo za neke izdelke, ki jih nimamo na zalogi. Preden začnemo s proizvodnjo zahtevanega artikla preverimo še, če za ta izdelek potrebujemo orodje ter ali je to orodje sploh na voljo, preverimo ali je na svojem mestu, da ni mogoče že v uporabi za kaj drugega ali pa npr. na popravilu. Pred seboj bi imeli v tem primeru seznam orodij, ob vsakem

imenu pa še črtno kodo določenega, nam potrebnega orodja. V našem sistemu uporabimo funkcijo "ISKANJE TOVORA", s čitalcem preberemo črtno kodo iz dokumenta, sistem pa nam sam pove ali je in kje je to orodje v skladišču orodja pospravljeno. Če so vsa potrebna orodja v skladišču, lahko začnemo s proizvodnjo.

Funkcijo "ODJAVA TOVORA" uporabimo, ko orodje vzamemo iz skladišča ter ga začnemo uporabljati. Funkcijo "VNOS TOVORA" pa uporabimo takrat, ko orodje vrnemo na svoje mesto v skladišču.

4. DRUŽBENA ODGOVORNOST

Ta raziskovalna naloga je že v osnovi zasnovana na primeru, ki se stalno uporablja v industrijski proizvodnji, logistiki, trgovini ... Sistem skladiščenja s črtnimi kodami ter čitalci črtnih kod nam, če z njim zamenjamo star sistem (npr. pisanje na papir), prihrani veliko časa, delo je manj naporno, sistem je moderniziran. Žal ni povsod na Zemlji tako, nekateri namreč še niso videli niti črtne kode, pri nas v Evropi pa je povprečje in kakovost tehnologije veliko višja. Med potekom raziskovanja je debata nanese tudi na to temo, da bi bilo dobro, da bi stanje čim hitreje postalo vsaj podobno ali enako našim razmeram. Ta raziskovalna naloga nam vsem, ki se zanimamo za to smer tehnologije (elektronika, sistemi, programiranje, elektrotehnika) pomaga razumeti, kako je svet okoli nas kompleksen in hkrati, če ga razumemo, izredno enostaven (z nekaj zelo osnovnega predznanja). Eden izmed glavnih ciljev je bil namreč razvozlati veliko zanko, čeprav preprostega sistema samo z nekaj tovari v skladišču.

5. ZAKLJUČEK

Na začetku, ko smo se odločili za raziskovalno nalogo, nismo imeli pretirano velikega znanja o stroki. Ne moremo trditi, da smo sedaj profesionalci vendar lahko potrdimo, da smo napredovali v osnovnem splošnem znanju o sistemih v skladiščih. Na začetku smo predvsem iskali že znane informacije o skladiščih, predvsem zanimivi so se nam zdeli podatki o naprednih sistemih skladiščenja, saj vsako raziskovanje pomeni nekoleten napredek in razčlenitev že znanega o izbrani temi. Raziskovali smo tudi črtne kode, ki smo jih morali uporabiti, sami smo jih tudi natisnili s pomočjo generatorja črtnih kod na spletni strani: <http://barcode.tec-it.com/en>. Največ časa nam je vzelo izdelovanje programske kode. Mislimo da smo izbrali dobro temo, ker nam je uspelo rešiti zastavljeno nalogo tj.

poenostaviti sistem preprostega skladišča s črtnimi kodami tako, da ga lahko razume vsakdo.

6. VIRI

Viri vsebine

<http://www.explainthatstuff.com/barcodescanners.html> (10. oktober 2016)

Generator črtnih kod, <http://barcode.tec-it.com/en> (25. november 2016)

<https://en.wikipedia.org/wiki/Barcode> (28. november 2016)

<http://www.gs1si.org/EAN> (30. december 2016)

http://leoss.si/strokovnjak_svetuje/24/kaj_je_crtna_koda/ (30. december 2016)

<http://www.identicus.si/EAN-crtne-kode.html> (30. december 2016)

<http://picavi.com/> (3. januar 2017)

<https://www.amazonrobotics.com/#/vision> (5. januar 2017)

Viri slik

Sistem skladiščenja s pametnimi očali, <https://www.youtube.com/watch?v=B6zPnVGS0VI> (slika ekrana).

Skladiščni roboti v skladišču spletne trgovine Amazon, <http://www.mmdonline.com/dc-and-warehouse-operations/robots-running-amazon-dcs-136934/>

Primer kode EAN13, <https://en.wikipedia.org/wiki/File:EAN13.png> (12. januar 2017)

TF (Interleaved 2 of 5), <http://www.computalabel.com/aboutitf.htm>

Shema delovanja čitalca črtnih kod, <http://www.explainthatstuff.com/barcodescanners.html>