

»Mladi za napredek Maribora 2017«

34. srečanje

## **Ponovna oživitev didaktičnega kompleta Lego Mindstorms s prenovo “pametne kocke”**

Raziskovalno področje: PROIZVODNO-TEHNIČNO

Inovacijski predlog

Avtor:	JAKA ZUPANC, MATIJA LIKAR, LEV ŠPIČKA
Mentor:	BORIS VOLARIČ
Šola:	OŠ BRATOV POLANČIČEV MARIBOR

Maribor, februar 2017

## KAZALO

Povzetek.....	4
Zahvala .....	5
1 Uvod.....	6
2 Teoretske osnove .....	7
2.1 Mikrokontrolnik.....	7
2.1.1 Vrste procesorjev in programiranje.....	7
2.1.2 Vhodne in izhodne enote.....	7
2.2 Lego Mindstorms.....	10
2.2.1 Novejše različice.....	10
2.2.2 Zgodovinski pregled .....	11
2.2.3 Elektromotorji.....	13
2.2.4 Senzorji.....	13
2.3 3D modeliranje in 3D tisk.....	14
2.3.1 3D modeliranje.....	14
2.3.2 3D tisk .....	14
3 Izbira ustreznih tehnologij.....	17
3.1 Strojna oprema.....	17
3.1.1 Določitev kriterijev.....	17
3.1.2 Primerni mikrokontrolerji .....	18
3.1.3 Primerjava .....	20
3.2 Programska oprema .....	22
3.2.1 Določitev kriterijev.....	22
3.2.2 Primerna programska oprema.....	23

3.2.3	Primerjava .....	26
4	Metodologija dela.....	27
4.1	Modeliranje ohišja.....	27
4.2	3D tisk ohišja .....	28
4.3	Priprava programskega okolja.....	29
4.4	Sestavljanje vezja .....	32
4.5	Testiranje s skupino učencev .....	34
5	Družbena odgovornost .....	35
5.1	Spletna stran .....	35
5.2	GitHub .....	36
5.3	Facebook stran .....	37
6	Viri.....	38
7	Kazalo slik.....	41
8	Kazalo tabel.....	41
9	Kazalo shem.....	41
10	Priloge .....	42
10.1	Tehniška risba pokrova.....	42
10.2	Tehniška risba ohišja .....	42
10.3	Izpis programa znotraj razširitve za motorje ter senzorje .....	42
10.4	Izpis programa znotraj razširitve za zaslon .....	42
10.5	Navodila za preizkus s šestošolci.....	42

## POVZETEK

Didaktični kompleti Lego Mindstorms na naši šoli imajo še vse sestavne dele, senzorje in motorje, a zastarele "pametne kocke" RCX. Programska oprema na sodobnih operacijskih sistemih ne deluje več, na trgu pa sta že nova generacija Lego Mindstorms EV3 ter vmesna generacija NXT, katerih programska oprema je drugačna, zato ne moremo delati z vsemi kompleti enako.

V našem inovacijskem predlogu smo razvili "nove možgane" za Lego Mindstorms. Nova "pametna kocka" ima iste dimenzije in priklone kot obstoječi RCX, tako da lahko pri gradnji lahko uporabimo originalne načrte in ostale dele.

Model za ohišje smo narisali s programom za 3D modeliranje SketchUp in ga natisnili s 3D tiskalnikom. Izbrali smo mikrokontroler Arduino Nano in ustrezne razširitvene kartice za "možgane" "pametne kocke" ter odprtokodni uporabniški vmesnik mBlock, ki omogoča vizualno programiranje in programiranje v naprednejših programskih jezikih, s čimer je izdelek primeren tudi za učence, ki v programiranju niso veščji.

## ZAHVALA

Radi bi se zahvalili ekipi mariborske odprte delavnice KreatorLab za tehnično pomoč, še posebej Katji za lekturo in šestošolcem naše šole za sodelovanje pri našem preizkusu.

Iskrene zahvale bi še posvetili našemu mentorju za moralno podporo, spodbudo in nasvete.

# 1 UVOD

Na šoli že vrsto let uporabljamo komplet Lego Mindstorms RCX za učenje robotike, čeprav je program LEGO Education nehajl podpirati RCX platformo že leta 2009. Zaradi tega je programska oprema za izdelovanje programov za LEGO RCX kompatibilna le s starejšimi operacijskimi sistemi Windows (32-bitni Windows 7 ali starejši).

Odločili smo se, da bomo prenovili pametno kocko tako, da bomo zamenjali računalnik znotraj nje. S programom SketchUp 2017 bomo izdelali model novega ohišja za mikrokontroler in ga natisnili s 3D tiskalnikom. Ohišje bo imelo iste dimenzije, kot že obstoječa kocka, tako bo kompatibilno z lego projekti. Na ohišju bodo tudi ustrezni lego konektorji (trije konektorji za motorje in trije za senzorje), tako da bodo lahko učenci nanj povezali komponente (motorje, senzorje ...), ki so že v kompletih.

Kocko bo lahko učenec programiral z orodji na mikrokontrolerju, ki jih bomo vgradili. Na kocki bo nameščen zaslon, ki bo prikazoval programe, naložene na kocko, in ali se ti izvajajo. Poleg zaslona bodo, tako kot pri Lego RCX-u nameščene tipke. Kocka bo preko brezžične tehnologije povezana z računalnikom, na katerem bo učenec znotraj programskega okolja izdeloval svoj program. Programsko okolje se bo izvajalo na operacijskem sistemu Windows 7, tako da predhodna namestitev programa ne bo potrebna. Na mikrokontroler bo torej nameščen zaslon (verjetno 2-palčni) ter razširitvena kartica za Wi-Fi ali BlueTooth. Kocka se bo napajala preko vgrajene baterije. Za izdelavo programa bodo učenci uporabljali grafični programski jezik, ki je podoben programskemu jeziku za RCX, ki že obstaja, in zahtevnejše programske jezike (basic, python, c++ ...).

## 2 TEORETSKE OSNOVE

### 2.1 Mikrokrmilnik

Mikrokrmilnik je majhen računalnik z enim samim integriranim vezjem, ki vsebuje procesor, pomnilnik, vhode ter izhode, ki jih je mogoče programirati. Prve mikrokontrolerje je izdelal Intel leta 1971. Mikrokontrolerji imajo tudi majhno količino RAM-a. Srečamo jih v večini modernih elektronskih naprav.

#### 2.1.1 Vrste procesorjev in programiranje

Nekateri mikrokontrolerji uporabljajo 4 bitne procesorje, ki delujejo na nižjih frekvencah (4 kHz) in porabijo bistveno manj energije. Takšne procesorje uporabljamo za izvajanje enostavnejših programov. Na drugi strani imamo zmogljivejše in zanesljivejše procesorje. Taki uporabljajo signalne procesorje (DSP) ter delujejo na višjih frekvencah in porabijo več energije. Uporabljajo se za zahtevnejše programe.

Za programiranje mikrokontrolerjev se najpogosteje uporabljajo jeziki, kot so C, Python ali JavaScript. (1)

#### 2.1.2 Vhodne in izhodne enote

Poznamo analogne in digitalne vhode ter izhode mikrokontrolerjev. Največja razlika med temi je, da lahko analogni vhodi/izhodi odčitajo ali oddajo katerokoli napetost znotraj določenega spektra, medtem ko lahko digitalni vhodi/izhodi odčitajo ali oddajo le določeno nizko napetost (LOW) in visoko napetost (HIGH) (ponekod tudi več vrednosti). (2)

##### 2.1.2.1 Vhodne enote

GPIO pri vhidih najpogosteje uporabljajo načine visoka impedanca, pull up/down upor, repeater mode ter histerezo.

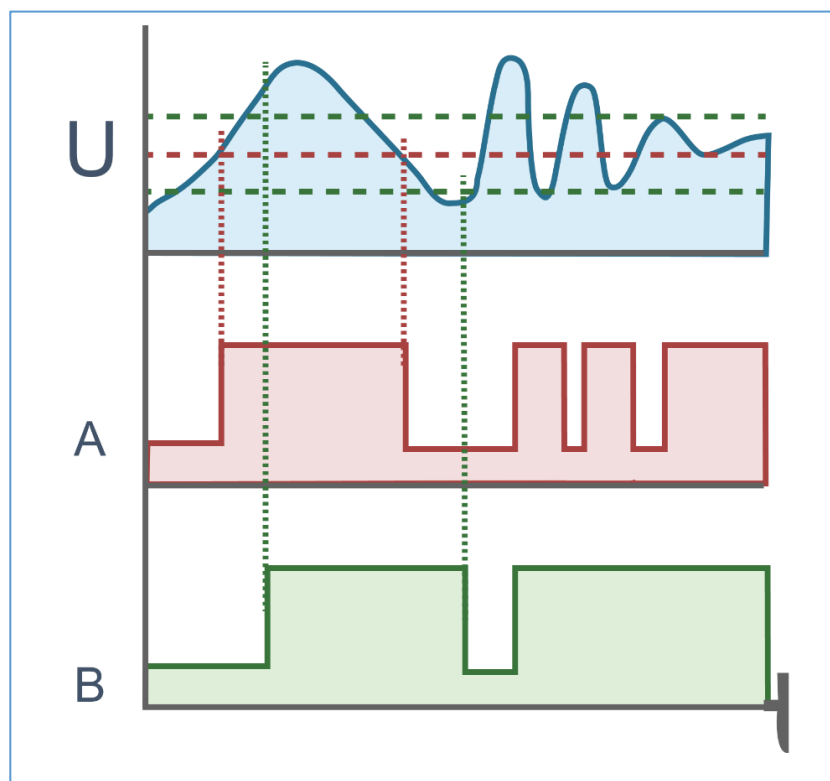
Način visoke impedance je digitalni vhod, katerega vrednost ni definirana, če ne dobi signala iz zunanjega vira, torej visoka ali nizka. Imenujejo se tudi tristopenjski vhodi.

**Pull up** so vhodi, katerih privzeto stanje je »visoko«, razen ko ga zunanji vir ne spremeni na »nizko«. Za pull down vhode velja obratno.

**Repeater** je vhod, ki deluje podobno kot pull up/down upor, a bo ohranil zadnje stanje, na katerega je bil naravnano, če v vhod ne prihaja tok.

**Histereza** je princip, s katerim odpravimo lažne spremembe ob spreminjanju napetosti, ki prihaja na vhod. Kot se pri vseh brez histereze spremeni vrednost, ko tok pade pod ali se dvigne čez 66 % napajalne napetosti (kot je že zgoraj omenjeno se odstotki pri različnih primerih razlikujejo), se pri histerezi spremeni vrednost glede na dve vnaprej definirani vrednosti: visoko vhodno napetost » $V_{IH}$ « ter nizko vhodno napetost » $V_{IL}$ «. Stanje vhoda se bo spremenilo na »visoko«, ko se bo napetost na njem dvignila nad » $V_{IH}$ «. Vrednost bo takšna tudi ostala, dokler napetost ne pade pod » $V_{IL}$ «. Tako bo stanje vhoda nizko, dokler se spet napetost ne dvigne nad » $V_{IH}$ «. (3)

Spodnji graf prikazuje razliko med odčitanimi vrednostmi na vhodu brez histereze (A) ter vhodom z histerezo (B), glede na spreminjanje napetosti (U). (3)



Shema 1: Razlika med digitalnimi vhodi s histerezo in brez nje (3)

### 2.1.2.2 Izhodne enote

Najpogostejši tipi izhodnih enot so line driver, odprti kolektor, push-pull izhod ter high-drive izhod.

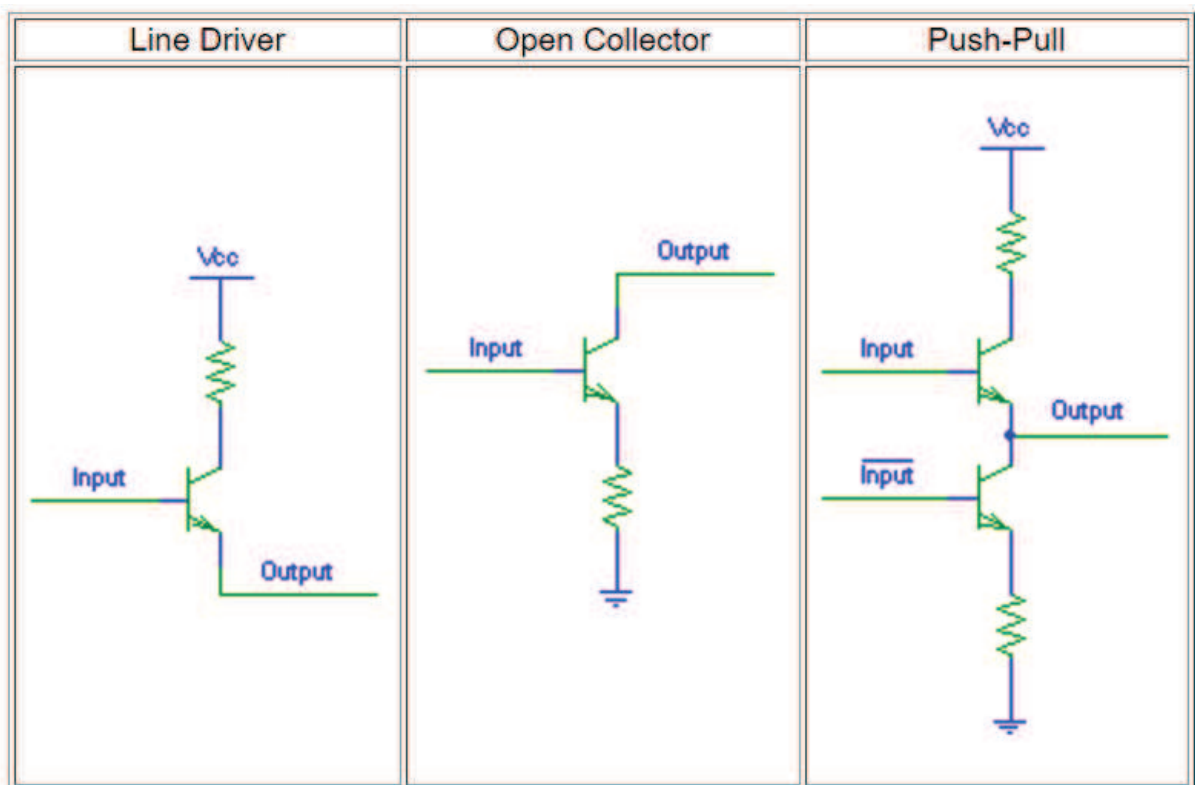


**Line driver** je tip izhoda, ki, ko je aktiviran, s tranzistorjem poveže izhod na določen tok (njegova napetost je odvisna od modela mikrokrmilnika). Če izhod ni aktiviran, se tranzistor zapre in je izhod v nedefiniranem stanju.

**Odpri kolektor** (Open drain) je izhod, podoben line driverju, vendar v tem primeru s tranzistorjem poveže izhod na ground. V primeru da je izhod ugasnjen, je tako kot line driver v nedefiniranem stanju.

**Push-pull** izhod je kombinacija line driverja ter odprtega kolektorja. Ta lahko s kombinacijo dveh tranzistorjev izhod poveže ali na ground ali na napetost.

Spodnja shema prikazuje razliko med zgoraj navedenimi tipi izhodnih enot mikrokrmilnikov.



Shema 2 Tipi izhodnih enot mikrokrmilnikov (4)

**High-drive** izhod je push/pull izhod, ki vzdrži delo pri višjem toku. Medtem ko lahko push-pull izhod vzdrži tok do okoli 8 mA, lahko high-drive izhod vzdrži tok vse do okoli 40 mA. (3) (4)

**Pulse Width Modulation (PWM)** je tehnika, s katero z digitalnimi izhodi simuliramo analogni signal. To dosežemo z izmenjavanjem visokega ter nizkega toka v konstantnih intervalih.

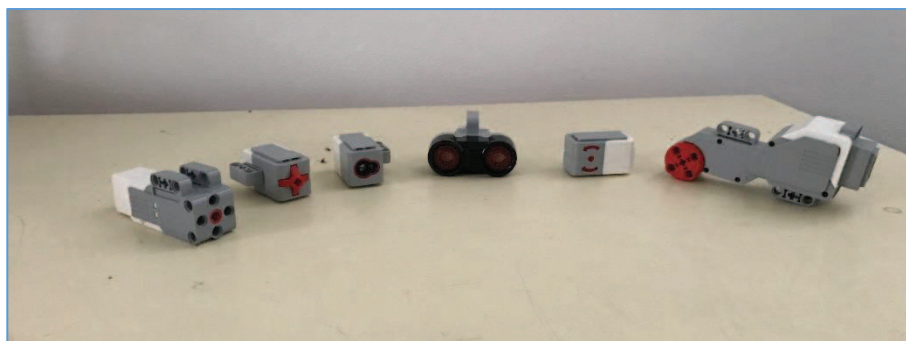
Razmerje med intervalom visokega toka in intervalom nizkega toka nam da navidezno napetost, ki ima lahko vrednosti med visoko ter nizko napetostjo. (5)

## 2.2 Lego Mindsorms

Lego Mindstorms je serija kompletov programske in strojne opreme podjetja Lego v okviru programa Lego Education. Komplet vsebuje »pametno kocko«, na katero se povežejo kompletu priloženi senzori in motorji. Kocko lahko brezžično ali s kablom povežemo z računalnikom, na katerem izdelamo program v grafičnem programskem jeziku in ga zaženemo na kocki. (6)

### 2.2.1 Novejše različice

Na trgu obstaja najnovejša različica iz LEGO Education programa, Lego Mindstorms EV3, ter malenkost starejša različica, Lego Mindstorms NXT. Te različice imajo drugačne grafične programske jezike in drugačne priključke. Lego Mindstorms EV3 se na uradni Lego spletni trgovini prodaja za ceno \$ 350 (332 €)<sup>1</sup>, Lego Mindstorms NXT pa za ceno \$ 540 (512 €). Pri nakupu rabljenih kompletov od posameznikov se cena spusti za EV3 do \$ 250 (237 €) ter za NXT do \$ 220 (209 €), vendar so cene neprimerne za nakup več kompletov.



*Slika 1: Elektromotorji in senzori EV3*

---

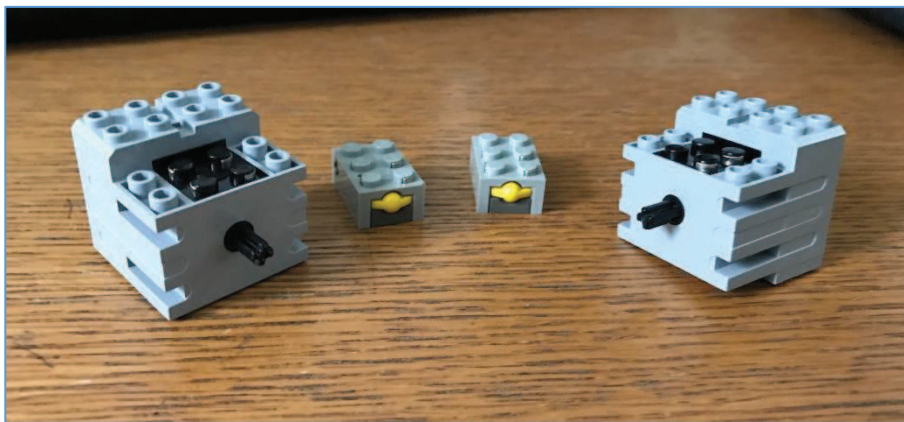
<sup>1</sup> Valute so pretvorjene po menjalnem tečaju USD–EUR dne 7. 1. 2017.

## 2.2.2 Zgodovinski pregled

**1986:** Izidejo prvi računalniško krmiljeni LEGO izdelki na trgu.

**1988:** Sodelovanje med The LEGO Group in Massachusetts Institute of Technology začne razvoj "inteligentne kocke", ki bo prinesla LEGO kreacijam življenje preko računalniškega programiranja.

**Januar 1998:** The LEGO MINDSTORMS RCX Pametna kocka in Robotics Invention System sta bila predstavljena medijem v Muzeju moderne umetnosti v Londonu.



*Slika 2: Elektromotorji in senzorji Lego RCX*

**September 1998:** The Robotics Invention System se začne istočasno prodajati v ZDA in v Veliki Britaniji. Na voljo sta tudi dva razširitveni kompleta – RoboSports in Extreme Creatures.

**November 1998:** Lastnik LEGO Group, Kjeld Kirk Kristiansen, in sloviti izumitelj, Dean Kamen, dobita partnerja za začetek FIRST LEGO League (PRVE LEGO Lige). To je tekmovanje iz robotike za srednješolce, na katerem se uporablja LEGO Mindstorm. Prvi turnir je potekal v Muzeju znanosti in industrije v Chicagu z 200 študentskimi ekipami.

**September 1999:** The Robotics Discovery Set, Ultimate Accessory Set, Droid Developer Kit in Robotics Invention sistem 1.5 izidejo v Združenih državah Amerike. Droid Developer Kit in Robotics Invention sistem 1.5 izidejo v Evropi in Aziji, in Robotics Discovery Set in Robotics Invention sistem 1.5 se začne izdelovati v Združenem kraljestvu.

**April 2005:** Prvič v zgodovini organizirajo FIRST LEGO League. Svetovno prvenstvo je potekalo v Atlanti.

**Avgust 2006:** LEGO MINDSTORMS NXT izide v Združenih državah Amerike; mednarodna razpoložljivost sledi.

**April 2007:** FIRST LEGO League prvič presega 100.000 udeležencev.

**Maj 2008:** LEGO MINDSTORMS je rekrut na Carnegie Mellon University Robot Hall of Fame.

**Avgust 2009:** Ponovno izide LEGO MINDSTORMS NXT 2.0 platforma.

**Januar 2013:** LEGO MINDSTORMS praznuje 15. obletnico, na mednarodnem Consumer Electronics Show je predstavljena platforma naslednje generacije – LEGO MINDSTORMS EV3.

**September 2013:** Tretja inkarnacija LEGO robotike, LEGO MINDSTORMS EV3, ki se je začela po vsem svetu. (7)

## 2.2.3 Elektromotorji

RCX:

- Tehnic motor 9 V
- Tehnic micromotor 9 V

EV3:

- Velik servo motorja
- Srednji servo motor

*Tabela 1 Specifikacije motorjev v kompletih Mindstorms (8)*

KATEGORIJE/MOTORJI	EV3 V. servo motor	EV3 S. servo motor	RCX Tehnic motor	RCX Tehnic micromotor
<b>OBRATOV NA MINUTO</b>	Med 24 in 153 rpm.	Med 24 in 249 rpm.	Med 580 in 3300 rpm.	Med 50 in 333 rpm.
<b>PRIKJUČEK</b>	Podoben tistemu za ethernet.	Podoben tistemu za ethernet.	V obliki lego kocke.	V obliki lego kocke
<b>NAPETOST</b>	Preizkušen od 4.5 V do 12 V.	Preizkušen od 4.5 V do 12 V.	Preizkušen od 4.5 V do 12 V.	Preizkušen od 4.5 V do 12 V.
<b>NAVOR za ročno ustavljanje</b>	43 N.cm	15 N.cm	0.85 N.cm	5.5 N.cm
<b>NAVOR za električno ustavljanje</b>	1.8 A	780 mA	700 mA	340 mA

## 2.2.4 Senzorji

- 1 ultrasonični senzor
- 2 senzorja na dotik
- 1 barvni senzor
- 1 žiroskop

## 2.3 3D modeliranje in 3D tisk

### 2.3.1 3D modeliranje

3D modeliranje je proces oblikovanja matematične predstavitve tridimenzionalnega objekta. Cilj tega postopka je oblikovanje 3D modela. Skozi proces 3D upodabljanja lahko dobimo 3D slike 2D modelov iz določenega vidika ali za upodobitev 3D modela v realnem času pri grafični simulaciji.

Poligonsko modeliranje – točke v 3D prostoru, ki se imenujejo vezje, so povezane z daljico, da tvorijo mnogokotno mrežo. Velika večina 3D modelov danes je zgrajenih kot teksturirani poligonski model, saj so prilagodljivi in jih lahko računalniki takoj naredijo. Vendar pa so poligoni ravninski in lahko samo približno narišejo ukrivljene površine s pomočjo različnih poligonov. (9)

### 2.3.2 3D tisk

3D tisk je ena izmed najsodobnejših tehnologij, ki je v zadnjih letih postala tudi širše zanimiva za uporabo na področjih razvoja in raziskav, konceptnega oblikovanja in predstavitvenih tehnik, arhitekture, urbanizma, konstruiranja, kulturne dediščine ter protetike, medicine in zobozdravstva. 3D tiskanje je omogočilo razvoj novih materialov, ki so namenjeni posebej za 3D tiskalnice. Pri 3D tisku se uporabljajo različni materiali, kot so plastika, kovine, keramika, vosek itd. Med seboj se razlikujejo predvsem po vrsti, minimalni debelini, teksturi in seveda ceni. Materiali so običajno predelani v vrvično obliko, ki se prodaja po kolutih, standardna preseka pa sta 1.75 in 3 mm, zaradi dimenzij šob, skozi katere se material topi. Poraba materiala je odvisna od zapoljenosti objekta ter sten. Material strokovno imenujemo tudi filament. Našteli bomo samo nekaj materialov, ki se uporabljajo pri različnih tehnologijah 3D tiskanja, saj na trg vedno znova prihajajo novi materiali s še bolj izpopolnjenimi karakteristikami.

Pri **FDM in FFF, tehnologijah** iztiskanja oziroma nalaganja materiala po plasteh, se uporabljajo naslednji materiali:

**PLA (Polylactic Acid)** je eden izmed največkrat uporabljenih materialov. Izdelan je na osnovi obnovljivih virov, koruznega škroba ter sladkornega trsa. Je razgradljiv in se ga lahko reciklira. Objekti so trdni, ne pa odporni na visoke temperature ter močne udarce. Iz PLA materiala se izdelujejo predvsem okrasni izdelki.

**ABS (Acrylonitrile Butadiene Styrene)** je alternativa materialu PLA. Odporen je na udarce, pri višjih temperaturah ne zbledi ali se zmežča, ima dolgo življenjsko dobo. Material žal ni okolju prijazen, saj je narejen na osnovi nafte, vendar se ga da reciklirati. Temperatura posteljice mora biti ogreta vsaj na 100 stopinj Celzija, v nasprotnem primeru s tem materialom ni možno tiskati.

**PVA material** se uporablja pri izdelkih, ki potrebujejo podporno strukturo med tiskanjem. Model lahko po tiskanju postavimo v vodo, kjer se podporna PVA struktura popolnoma raztopi. Material se uporablja pri 3D tiskalnikih, ki imajo več glav.

Ostali materiali, ki se uporabljajo, so najlon, kovine (zlato, srebro, bron, baker), guma, porcelan, keramika.

Na trgu najdemo več tipov tiskalnikov. Naj naštejemo le nekaj tipov:

**SLA 3D tiskalniki** uporabljajo fotopolimerizacijo za izdelavo trdnega modela iz tekočih materialov s pomočjo ultravijoličnih laserskih žarkov. Za vsako plast laserski žarek sledi prečnemu prerezu površine tekočega materiala. Izpostavljenost površine ultravijoličnemu žarku laserja povzroči trdenje materiala ter spajanje s prejšnjo plastjo. Po vzorcu sledenja se SLA plošča tiskalnika spušča za debelino enega sloja, kar znaša približno 0.05 do 0.15 mm. Postopek se ponavlja tako dolgo, dokler celoten 3D model ni izdelan. Stereolitografija zahteva uporabo podporne strukture, ki služi pritrjevanju modela na SLA ploščo. Tehniko je izumil Charles Hull leta 1986.

**FDM ali FFM 3D tiskalniki** uporabljajo plastično nitko ali kovinsko žico, ki se odvija iz tuljave in je namenjena dovajanju materiala do brizgalne šobe, ki lahko dovajanje vklaplja in izklaplja. Šoba je segreti do temperature, ki material topi in se lahko premika horizontalno ter vertikalno s pomočjo številnih krmilnih mehanizmov, ki so vodeni s pomočjo CAM programskega paketa. Model je izdelan na principu iztiskanja tekočega materiala skozi šobo za izdelavo plasti, ki se takoj po iztisku strdijo. FDM tehnologijo je izumil Scott Crump v poznih

80-ih letih prejšnjega stoletja. Programski paket, ki je dostavljen s FDM 3D tiskalnikom, avtomatično izdeluje podporne elemente pri modelu, če so potrebni. Tiskalnik iztiska dva različna materiala, enega za izdelavo modela, drugega pa za izdelavo podpornih elementov.

**SLS 3D tiskalniki** uporabljajo laserje visokih moči za zlivanje delcev plastičnega prahu, kovine, keramike ali stekla v maso, ki ima željeno tridimenzionalno obliko. Laser selektivno zliwa material iz prahu s predhodnim skeniranjem plasti, ki ga izdelava 3D modelirni program. Po vsakem prečnem skeniranju se posteljica prahu pomakne navzdol za debelino ene plasti, nato je nova plast prahu uporabljena na površini. Proces se ponavlja tako dolgo, dokler model ni izdelan. Ves nedotaknjen prašni material postane podporni element, kar je tudi velika prednost pred SLA 3D tiskalniki. Neuporabljen prah se namreč lahko uporabi tudi za naslednje 3D tiskanje. SLS tehnologijo je razvil in patentiral Dr. Carl Deckard v srednjih 80-ih letih prejšnjega stoletja. (10)

Pri pripravi tridimenzionalne g-kode lahko izbiramo med več programi. Naj navedemo le nekaj izmed njih:

**Cura** pripravi model za 3D tiskanje. Novincem omogoča, da dobijo odlične rezultate. Za strokovnjake obstaja več kot 200 nastavitev za prilagoditev potrebam.

**Slic3r** – program, primerljiv programu Cura, ki pa ni tako prijazen za uporabo in je primeren za napredne uporabnike. V primerjavi s programom Cura pripravi učinkovitejšo kodo, kar omogoči hitrejši tisk



## 3 IZBIRA USTREZNIH TEHNOLOGIJ

### 3.1 Strojna oprema

#### 3.1.1 Določitev kriterijev

Kriteriji za mikrokontroler, ki jih bomo uporabili v prenovljeni različici RCX pametne kocke:

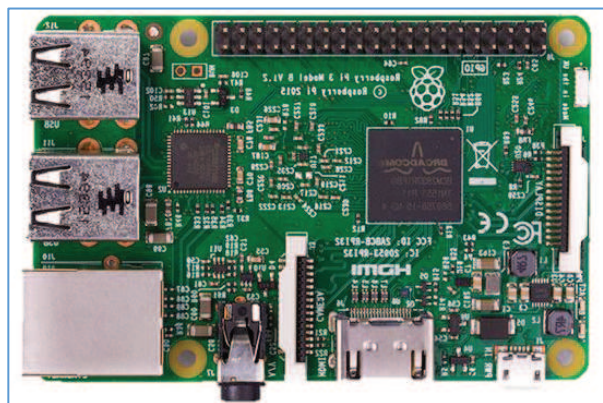
- Mikrokontroler je zmožen izmenjavati podatke z brezžično tehnologijo (tudi preko razširitvene kartice, če je potrebna).
- Ima vsaj 3 analogne vhode in 9 izhodov (od tega vsaj 3 PWM) za senzorje in motorje.
- Ima 5 preostalih digitalnih izhodov za zaslon, 1 analogni vhod ali 4 digitalne za tipke ter 2 izhoda za Bluetooth kartico (če je le-ta potrebna).
- Ima relativno nizko porabo.
- Je zmožen delovati na 9 V (mikrokrmilnike bo napajala 9 V baterija).
- Ima izhodno napetost 9 V.
- Lahko odda 0.7 A toka (tolikšen je tok, ki je potreben, da napajamo 1 velik motor iz kompleta pri napetosti 9 V).
- Njegove dimenzije (skupaj z baterijo in morebitnimi »shieldi« ter drugimi razširitvami) so manjše od 89 mm \* 57.5 mm \* 29 mm.
- Ima dovolj pomnilnika za hrambo več programov.
- Je zmožen izvajati programe, izdelane na računalniku.

## 3.1.2 Primerni mikrokontrolerji

Najprimernejši mikrokontrolerji za naš projekt so Raspberry Pi 3 B, Arduino Nano ter Arduino Uno.

### 3.1.2.1 Raspberry Pi 3 Model B

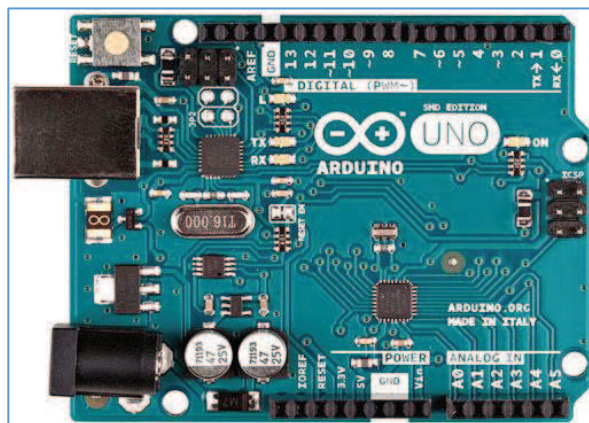
Raspberry Pi 3 Model B je zmogljivejši mikrokontroler v velikosti kreditne kartice, ki lahko za svoj operacijski sistem uporablja tudi Linux. Raspberry Pi mikrokontrolerji so priljubljeni predvsem v namene izobraževanja v osnovni ter srednji šoli. Raspberry Pi 3 B ima 4 USB vhode, Ethernet vhod, USB Micro vhod za napajanje, vhod za SD kartico, HDMI izhod ter 3.5 mm AUX vhod. Zaradi njegove relativno visoke zmogljivosti ga lahko uporabljamo kot vsak drug namizni računalnik. GPIO vodila delujejo na 3.3 V in imajo maksimalni tok 50 mA. Njegove dimenzije so 85 mm \* 56 mm. Masa mikrokontrolnika je 45 gramov. Raspberry Pi mikrokontrolnik lahko uporabljamo kot samostojen računalnik. (11)



Slika 3 Raspberry Pi 3 Model B (11)

### 3.1.2.2 Arduino Uno

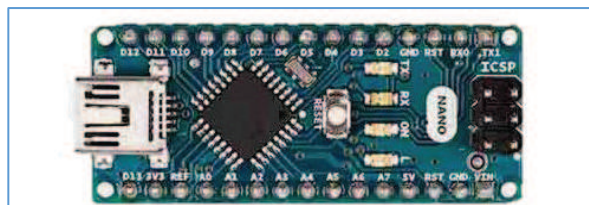
Arduino Uno je mikrokrmilnik, ki temelji na ATmega328P. Ima 14 digitalnih vhodno-izhodnih pinov (od katerih se jih 6 lahko uporablja kot PWM izhodov), 6 analognih vhodov, 16 MHz keramični resonator, USB povezavo, vhod za napajanje in gumb za ponastavitev. Arduino deluje na 5 V, ima 32 kB spomina, 2 kB SRAMA. I/O izhodni pini imajo tok do 20 mA, medtem ko imajo 3.3 V pini tok do 50 mA. Dimenzije so 68.6 mm \* 53.4 mm. Mikrokontroler tehta 25 gramov. Programe zanj izdelujemo v programskem okolju Arduino IDE na drugem računalniku, na katerega Arduino povežemo preko USB kabla. (12)



Slika 4 Arduino Uno (12)

### 3.1.2.3 Arduino Nano 3.0

Arduino Nano je mikrokrmilnik, ki temelji na ATmega328. Prav tako kot Arduino Uno ima 14 digitalnih vhodno-izhodnih pinov (od katerih se jih 6 lahko uporablja kot PWM izhodov), 6 analognih vhodov, 16 MHz keramični resonator, USB povezavo, vhod za napajanje in gumb za ponastavitev. Arduino deluje na 5 V, ima 32 kB ali 16 kB spomina, 2kB ali 1 kB SRAMA. I/O izhodni pini imajo tok do 40 mA. Dimenzije mikrokontrolerja so 45 mm \* 18 mm in tehta 5 gramov. Potek izdelave programov je enak kot pri Arduino Uno. (13)



Slika 5 Arduino Nano 3.0 (13)

### 3.1.3 Primerjava

Izmed treh mikrokontrolerjev ima samo Raspberry Pi 3 B vgrajeno kartico za WLAN in Bluetooth 4.0. Nakup kompatibilne Bluetooth 4.0 kartice za Arduino bi stal okoli 20 €. Vsi mikrokrmilniki imajo dovolj digitalnih izhodov, vendar imata le Arduino mikrokrmilnika dovolj analognih vhodov ter PWM izhodov. Kot vidimo, ima Raspberry Pi precej večjo porabo kot Arduino (z neobremenjenimi izhodi). (14) (15) (13)

Vsi mikrokrmilniki lahko delujejo na 9 V, vendar bi Raspberry Pi potreboval regulator napetosti, saj lahko deluje samo na napetosti 5 V. Ker izhodi mikrokrmilnikov niso zmožni oddati napetosti 2.1 A toka (Raspberry Pi ne more oddati zadostne napetosti), bi za naš projekt potrebovali H-mostove, s katerimi bi lahko z mikrokrmilnikom krmilili hitrost ter smer rotacije motorjev, ki bi dobili tok neposredno iz baterij.

Če bi kocko napajali z 6 AA alkalnimi baterijami (17) (glej dokumentacijo), bi ob 1 delujoči maksimalni obremenitvi velikih motorjev baterijo izpraznili v dveh urah (pri tem zanemarimo porabo mikrokrmilnikov). Ob samem delovanju mikrokrmilnika, brez priključenih senzorjev ali motorjev, bi Arduino Uno deloval 150 ur, Arduino Nano pa kar 300 ur, medtem ko bi Raspberry Pi 3 Model B izpraznil baterije v pribl. 100 minutah.

Zaradi baterij, ki zasedejo večino prostora, ima le Arduino Nano dimenzije, ki ustrezajo naši prenovljeni Lego RCX kocki. Spomin Arduino Nano mikrokontrolerja je enak spominu obstoječe Lego RCX pametne kocke (32 kB). Vsi mikrokontrolerji imajo relativno zmogljive procesorje, ki bodo zmožni izvajati enostavne programe na osnovnošolski ravni.

Zgoraj navedene specifikacije mikrokrmilnikov so zbrane iz sledečih strani: (12) (13) (11) (18)

Na podlagi dejstev navedenih v tem poglavju je jasno razvidno, da je najprimernejši mikrokrmilnik za naš projekt Arduino Nano.

Tabela 2 Primerjava mikrokrmilnikov<sup>2</sup>

Mikrokontroler	Raspberry Pi 3 B	Arduino Uno	Arduino Nano 3.0
Vhodi/Izhodi	40 GPIO pinov	14 digital I/O, 6 analog input pinov	14 digital I/O pinov, 8 analog pinov
Od tega PWM pinov	1	6	6
Napetost izhodov	3.3V	0V - 5V	0V - 5V
Maksimalni tok izhodov	50 mA	I/O pini 20 mA in 3,3 V pini 50 mA	I/O 40 mA
Procesor	quad-core ARM Cortex-A53 CPU	ATmega328P	ATmega328
Takt Procesorja	1,2 GHz	16 MHz	16 MHz
Tip procesorja	64 -bit	8-bit	8-bit
Zmožnost procesorja	2441 MIPS @ 1.2GHz	8-11 MIPS @ 16MHz	8-11 MIPS @ 16MHz
Brezžična tehnologija	802.11n WLAN, Bluetooth 4.1	Brez	Brez
Spomin	Odvisno od kapacitete SD kartice	32 kB	32 kB
Bralno pisalni pomnilnik	1 GB RAM	2 KB SRAM, 1KB EEPROM	2 KB SRAM, 1KB EEPROM
Dimenzije	85 mm * 65 mm * 16mm	86,5 mm * 53,4 mm * 14mm	45 mm * 18mm * 7,3mm (brez pinov)
Poraba	800 mA * 5V = 4W	26mA * 9V = 234 mW	19mA * 9V = 171 mW
Cena	<b>35.76 €</b>	<b>20.00 €</b>	<b>28.00 €</b>

<sup>2</sup> Pri podatkih o porabi posameznih mikrokrmilnikov navedenih v preglednici lahko pride do odstopanj.

## 3.2 Programska oprema

### 3.2.1 Določitev kriterijev

Kriteriji za razvojno okolje, ki ga bomo uporabili za izdelavo programov, ki se bodo izvajali na pametni kocki:

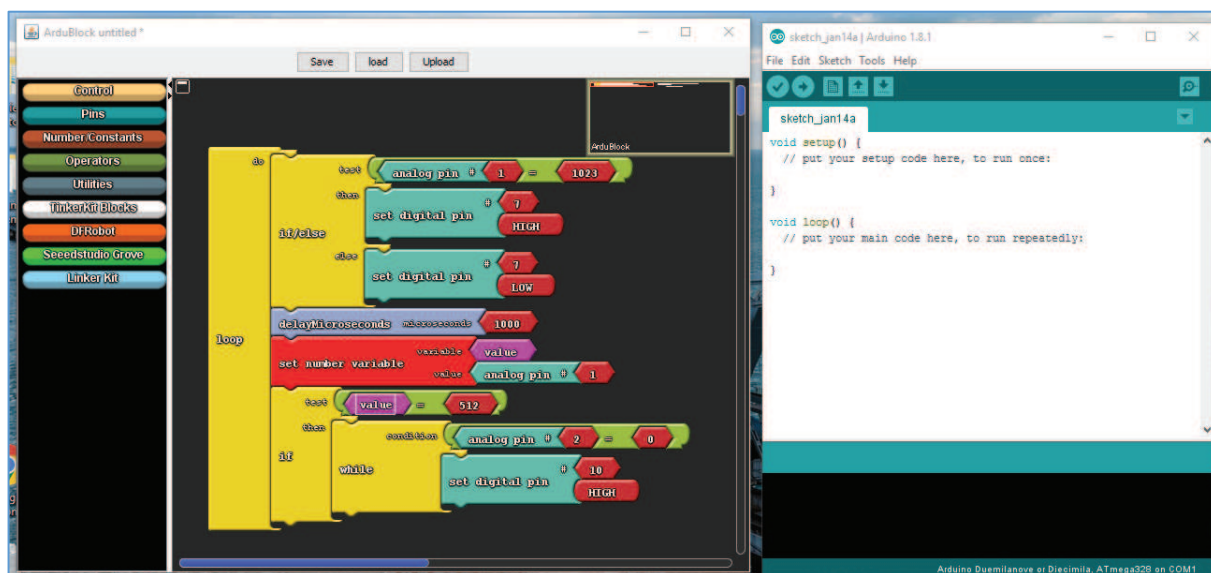
- V razvojnem okolju je mogoče programirati v grafičnem programskem jeziku in v vsaj enem drugem tekstovnem programskem jeziku (basic, python, c++ ...).
- Razvojno okolje je razumljivo in enostavno, tako da ga razumejo učenci 6. razreda OŠ, brez predhodnih izkušenj na področju računalništva ali robotike.
- Razvojno okolje je kompatibilno z izbranim mikrokontrolerjem (glej poglavje »Strojna oprema«).
- Razvojno okolje lahko deluje na Windows 7.
- Znotraj programa lahko definiramo svoje funkcije ali dodamo razširitve.
- Program za mikrokontroler se lahko prenese preko Bluetooth-a.
- Razvojno okolje je brezplačno.

## 3.2.2 Primerna programska oprema

Čeprav je izbira razvojnih okolij za Arduino, ki uporablja grafične programske jezike, zelo skromna, smo vendarle uspeli najti nekaj takšnih, ki bi bila lahko primerna za naš projekt. Ta razvojna okolja so ArduBlock, mBlock ter miniBloq.

### 3.2.2.1 ArduBlock

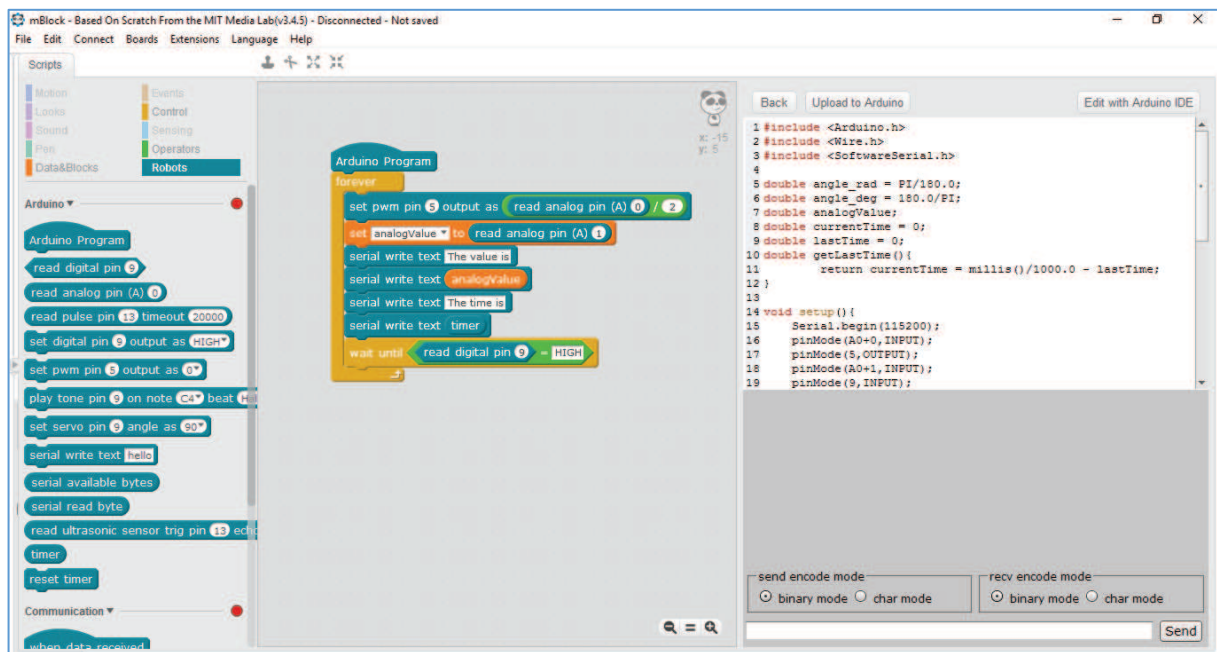
ArduBlock je programsko okolje razvito z namenom, da poenostavi programiranje na Arduino na drag-and-drop nivo. Namesto pisanja stavkov, se pri ArduBlock-u, kot vsakem drugem grafičnem programskem jeziku, uporabljajo »block«-i, ki se pritrdijo na druge »block«-e in tako tvorijo program. ArduBlock deluje kot razširitev programskega okolja Arduino IDE. (19)



Slika 6 Programsko okolje ArduBlock

### 3.2.2.2 mBlock

MBlock je programsko okolje, ki ga je razvil Makeblock za izdelovanje programov za Arduino ter mBot robote, ki jih proizvaja sam Makeblock. MBlock je razvit na podlagi odprtokodnega razvojnega okolja Scratch 2.0, ki ga razvija MIT Media Lab Lifelong Kindergarten. (20) Programski jezik bo verjetno zelo razumljiv, saj ima isto sintakso kot Scratch, ki se ga učenci učijo v okviru tečaja računalništva na šoli. Za programski jezik je možno razviti tudi razširitve, ki jih pišemo v JavaScript-u.

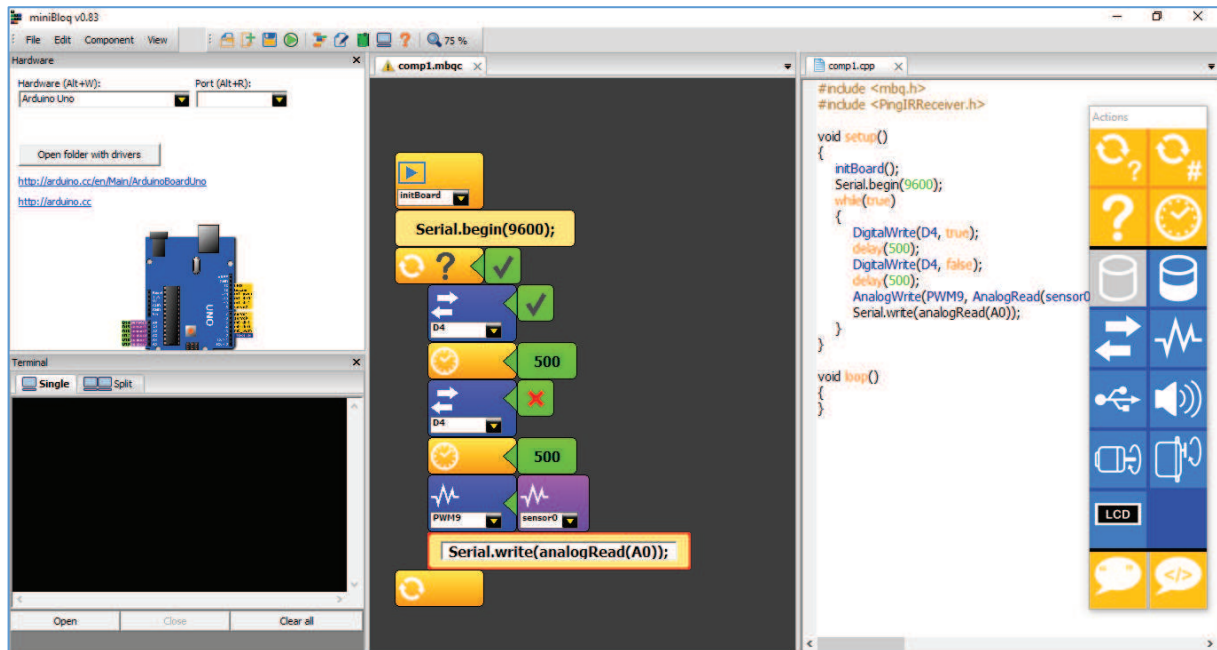


Slika 7 Programsko okolje mBlock



### 3.2.2.3 miniBLoq

MiniBLoq je odprtokodni grafični programski jezik za Arduino in druge njemu podobne mikrokontrolerje, kot je DuinoBot. Trenutno je v obdobju razvoja. Med izdelovanjem programa z grafičnim programskim jezikom se generira koda v C++. (21) Kot je vidno na spodnji sliki, je izbira funkcij znotraj programskega jezika zelo skromna in bi bilo potrebno dodati številne nove funkcije, da bi lahko to programsko opremo uporabili v našem projektu.



Slika 8 Programsko okolje miniBLoq

### 3.2.3 Primerjava

Spodnja tabela prikazuje ustreznost posameznih programskih okolij kriterijem.<sup>3</sup> Očitno je vidno, da sta programska okolja mBlock in miniBloq mnogo primernejša za naš projekt, kot Ardublock, pri katerem ne moramo dodati razširitev, jezik ni razumljiv ... Čeprav sta mBlock ter miniBloq na dokaj enakem nivoju, mBlock vendarle nekoliko bolj ustreza našim kriterijem, saj kot smo že omenili v poglavju »Primerna programska oprema«, program temelji na Scratchu, kar je odličen programski jezik za začetnike v osnovnih šolah, ki se prvič seznanjajo s programiranjem.

Tabela 3: Primerjava programske oprema

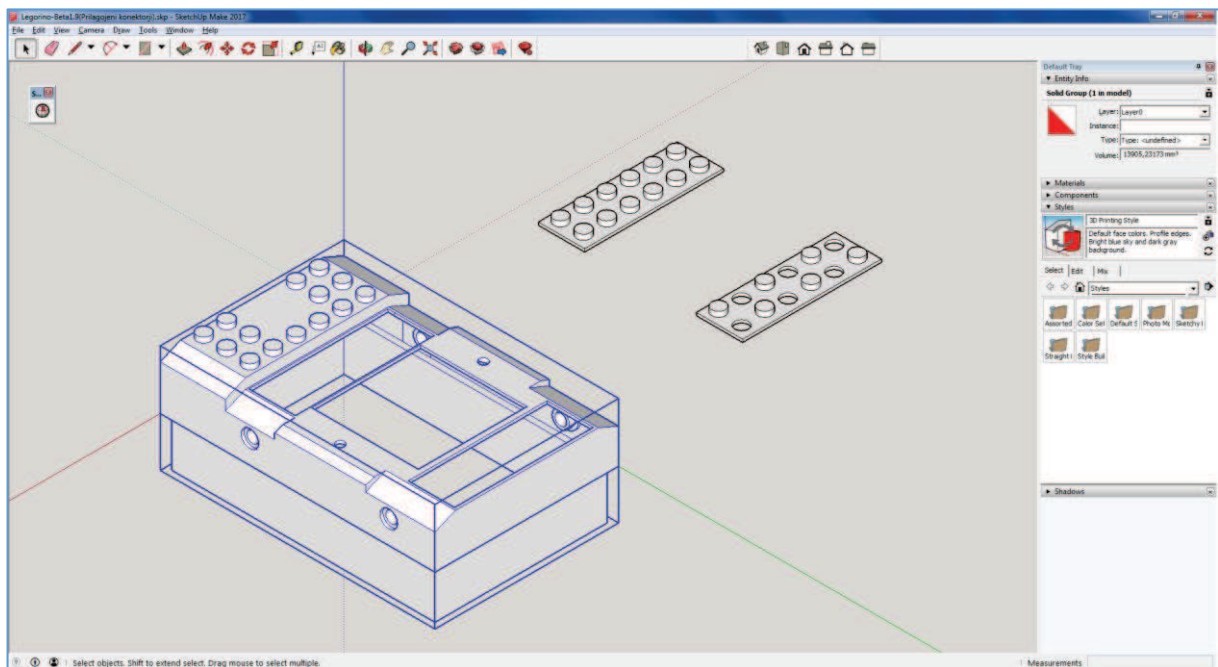
Kriterij	Adrublock	mBlock	miniBloq
Programiranje v graf. prog.jeziku	Da	Da	Da
Programiranje v tekst. prog. jeziku	Ne	C++	C++
Razumljivost in enostavnost	Dobra	Prav dobra	Zadostna
Kompatibilnost z Arduino Nano	Da	Da	Da
Deluje na Windows 7	Da	Da	Da
Dodajanje funkcij ali razširitev	Ne	Da	Da
Prenos podatkov preko Bluetooth-a	Da	Da	Da
Brezplačno razvojno okolje	Da	Da	Da

<sup>3</sup> Ocena razumljivosti in enostavnosti posamezne programske opreme temelji na osebni pogledu avtorjev inovacijskega projekta na posamezno razvojno okolje.

## 4 METODOLOGIJA DELA

### 4.1 Modeliranje ohišja

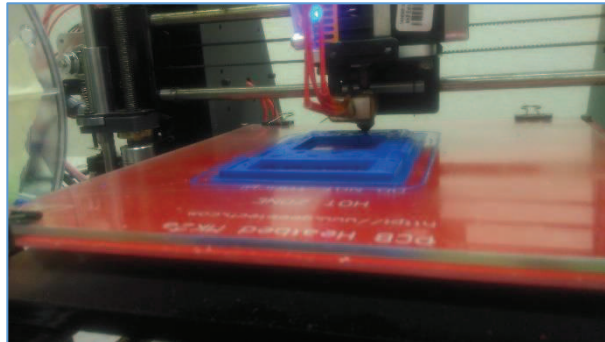
Izdelave 3D modela kocke smo se lotili znotraj programa namenjenega 3D modeliranju, SketchUp Make 2017. Izmerili smo obstoječo RCX »pametno kocko« in mere vnesli v program. Izdelali smo dva modela, model pokrova kocke ter ohišje kocke. Ob delu smo si pomagali z razširitvami Solid Inspector<sup>2</sup> in STL Import & Export. Po končanem delu smo 3D modele izvozili v formatu .stl. V prilogi lahko najdete tudi tehniške risbe ohišja ter pokrova, ki smo jih narisali s programom LayOut.



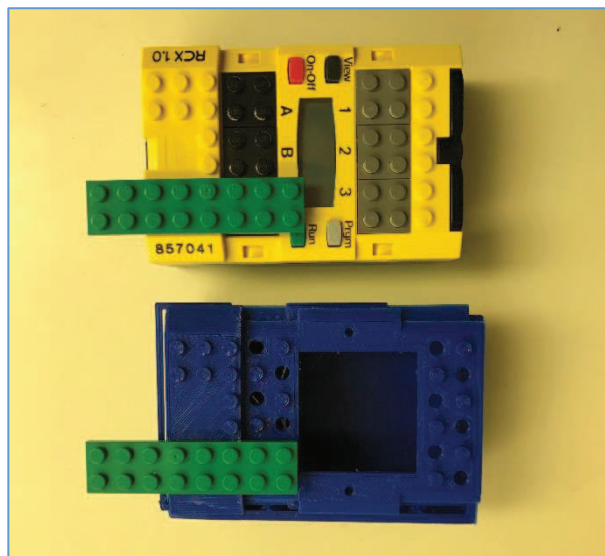
Slika 9 3D model ohišja "pametne kocke"

## 4.2 3D tisk ohišja

.stl datoteka ohišja ter pokrova smo uvozili v odprtokodni program za »slicing«, imenovan Sli3r, iz katerega smo potem izvozili g-kodo za 3D tiskalnik. Svoje modele smo nato natisnili v lokalni odprti delavnici.



Slika 10: 3D tisk ohišja



Slika 11: Testiranje modela, natisnjene s 3D tiskalnikom

## 4.3 Priprava programskega okolja

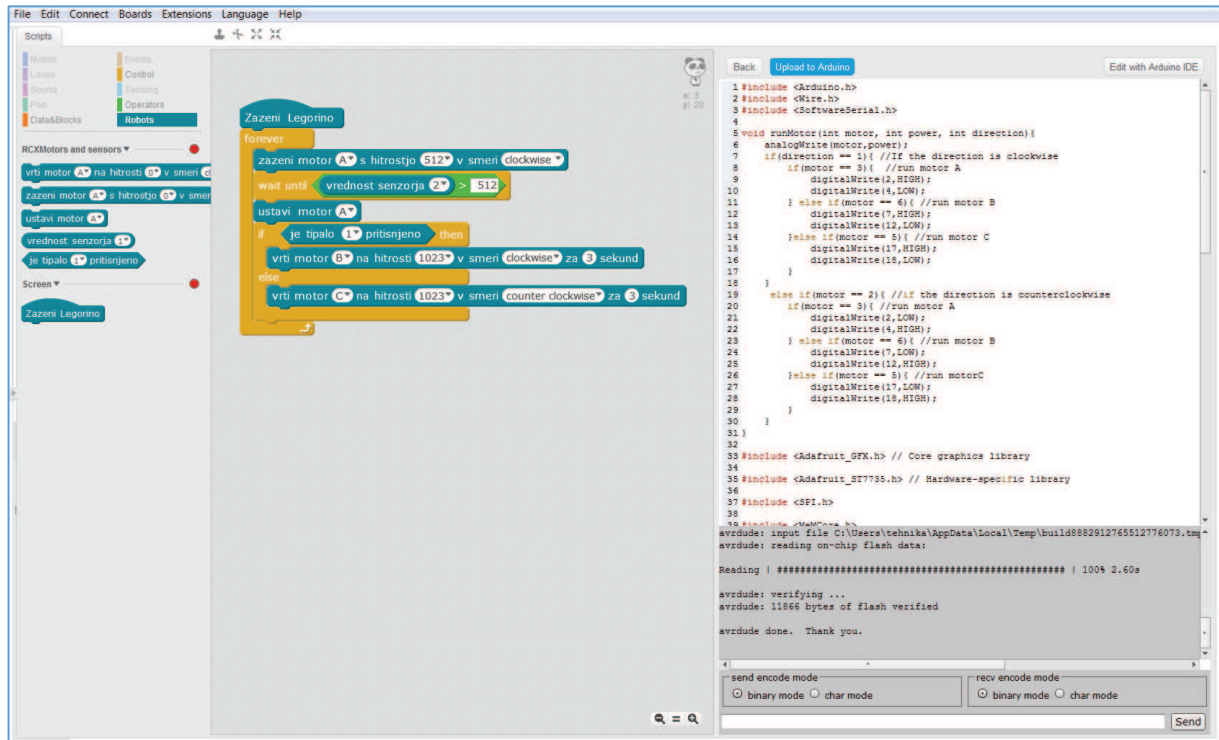
Po inštalaciji mBlock-a 3.4.5 za Windows sisteme smo se lotili razvijanja razširitev za razvojno okolje. V mBlock so bile že prej vključene druge razširitve kot so mBot, Arduino, Communication ...

Začeli smo s pisanjem razširitve, ki bo omogočala uporabo senzorjev ter motorjev. Razširitve za mBlock sestavljajo 3 enote (mapa »src«, ki vsebuje »knjižnice«, program v JavaScriptu ter .s2e datoteka). MBlock deluje na dva načina Scratch Mode (pri kateremu uporabnik krmili »Sprite« in se program izvaja na računalniku, tako kot pri programskem okolju Scratch, na katerem mBlock temelji) in Arduino Mode (kjer se tvori program v C++, ki se nato naloži na Arduino). Program v JavaScriptu je tisti, ki definira, kako bo razširitev delovala v Scratch Mode, .s2e datoteka pa tista, ki določa, kako se bo grafični programski jezik v Arduino Mode prevajal v C++. Ker v našem projektu ne nameravamo uporabljati Scratch Mode, vendar brez JavaScript programa mBlock ne dovoli uvažanja razširitev, smo uporabili JavaScript datoteko iz razširitve za Arduino, ki je bila priložena mBlocku.

Spodaj je navedena slika .s2e datoteke, ki se piše v JSON jeziku. Za lažje razumevanje programa so spodaj z dvema poševnicama zaznamovani komentarji, a ker JSON ne podpira komentarjev, smo jih pred uvozom v mBlock izbrisali. Izpis programske kode lahko najdete v prilogi.

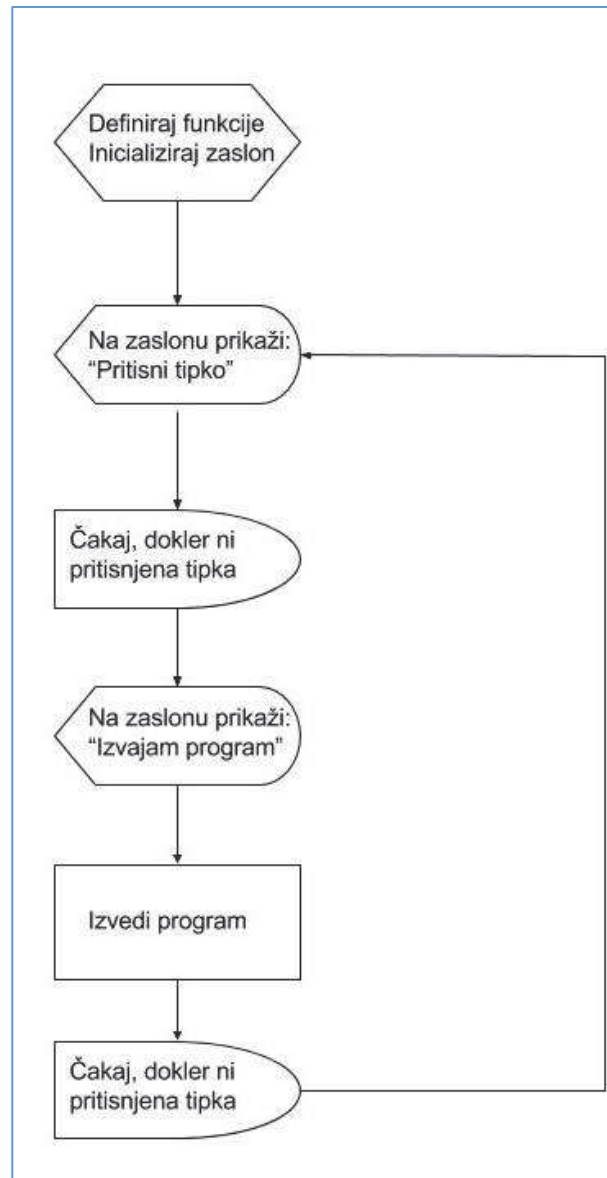
Po istem principu smo naredili razširitev za zaslon na »pametni kocki«. Ob zagonu »pametne kocke« se bo na ekranu izpisalo sporočilo »Pritisni tipko za nadaljevanje«. Ko bo ta pritisnjena, se bo na zaslonu izpisalo »Izvajam program«. Po končanem programu, ki ga je uporabnik ustvaril, se bo ob kliku na drugi gumb vrnila »pametna kocka« v prvotno stanje. Izpis kode lahko prav tako najdete v prilogi.

Po končanem razvijanju razširitev smo znotraj okolja imeli 6 »block«-ov. 1 »header block« za zagon programa, uvoz knjižnic, priprava zaslona ..., 3 »block«-e za krmiljenje motorjev ter 2 »block«-e za senzorje (en zaznamuje integer value, drugi pa boolean value).



Slika 12 Program znotraj mBlocka

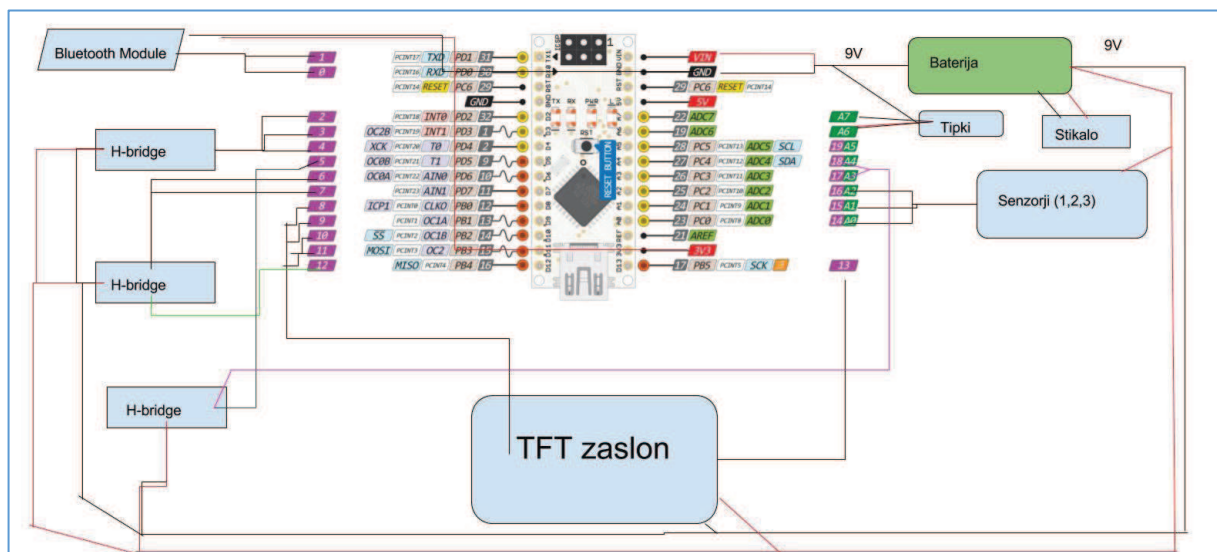
Spodnji diagram poteka prikazuje, kako se izvaja program na mikrokontrolerju. Vsi členi diagrama, razen procesa »Izvedi program«, ostajajo vedno isti in se izvedejo vedno, medtem ko se znotraj člena »Izvajam program«, izvaja tisti program, ki ga je izdelal uporabnik, tako kot pri Lego Mindstorms »pametnih kockah«.



Shema 3: Diagram poteka programa

## 4.4 Sestavljanje vezja

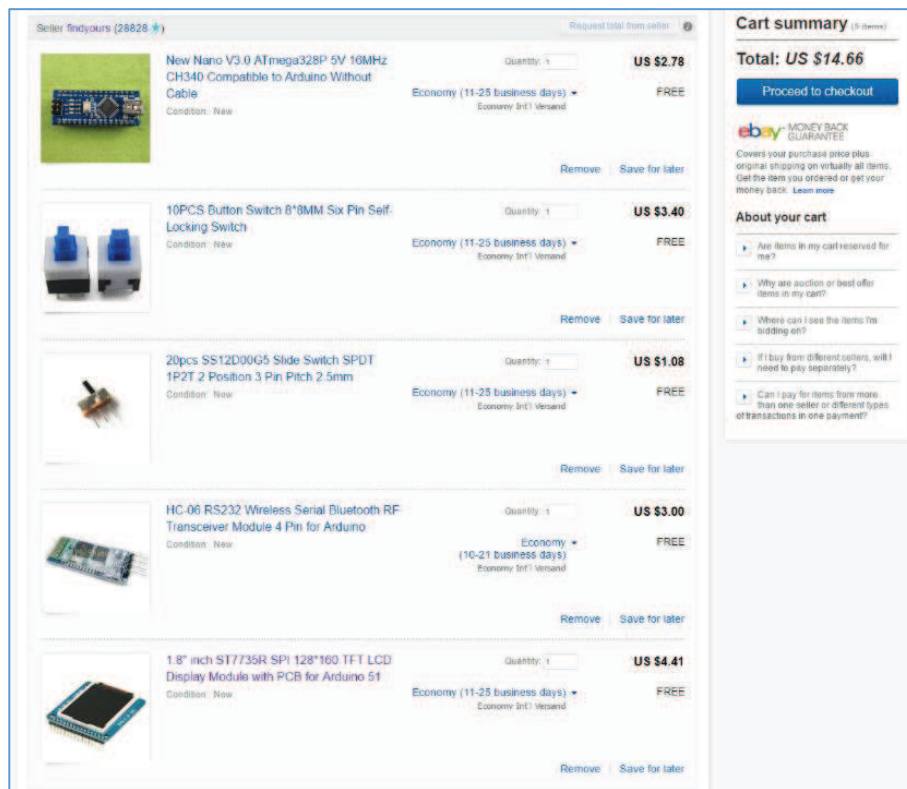
Potem ko smo za projekt izbrali primeren mikrokontroler, smo se lotili izdelave sheme nove »pametne kocke«. Ta bo vsebovala mikrokontroler Arduino Nano, Bluetooth modul, 1.8 palčni zaslon, 2 push-button stikala, ON/OFF stikalo, 3 H-mostove za krmiljenje motorjev ter baterijo. Spodnja slika prikazuje, na katere izhodne ali vhodne pine mikrokontrolerka bodo te komponente povezane.



Shema 4 Komponente povezane na Arduino Nano (22)



Komponente smo nato naročili preko spletne trgovine.



Slika 13 Naročene komponente v spletni trgovini

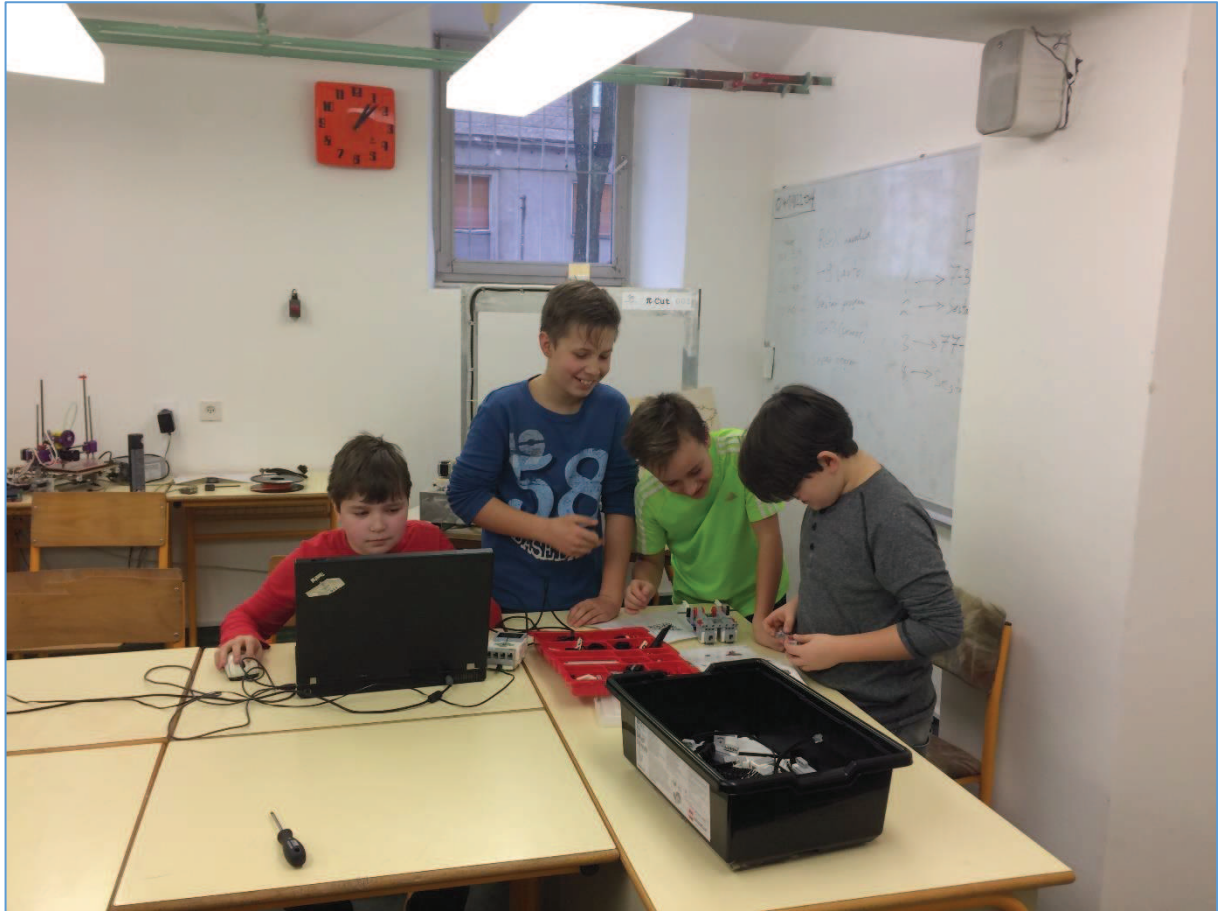
Kupljene komponente smo povezali z Arduino in jih vstavili v natisnjeno ohišje. Za napajanje smo uporabili, kot smo že v poglavju »Strojna oprema« predvidevali, 6 alkalnih baterij. Ko je naša »pametna kocka« v stanju pripravljenosti (prižgani so le mikrokrmilnik, Bluetooth modul ter zaslon) porabi okoli 87 mA toka (8 mA za modul, 19 mA za Arduino ter do 50 mA za zaslon). (13) (23) (24) S tolikšno porabo, bi morala naprava delovati 100 ur, preden se baterije izpraznijo. (17)

Na ohišje smo vgradili še priključke, na katere se lahko priključijo motorji ter senzorji iz originalnega kompleta.

## 4.5 Testiranje s skupino učencev

Učencem 6. razreda naše šole smo dali nalogo, naj sestavijo avto po načrtih ter naredijo program, ki avto vodi dokler ne zazna ovire. Navodilo za delo je priloga tej nalogi.

Ena skupina učencev je sestavljala avto s pametno kocko RCX, druga z EV3, tretja pa z našo pametno kocko, ki smo jo poimenovali »Legorino«.



*Slika 14: Testiranje uporabnosti s skupino šestošolcev*

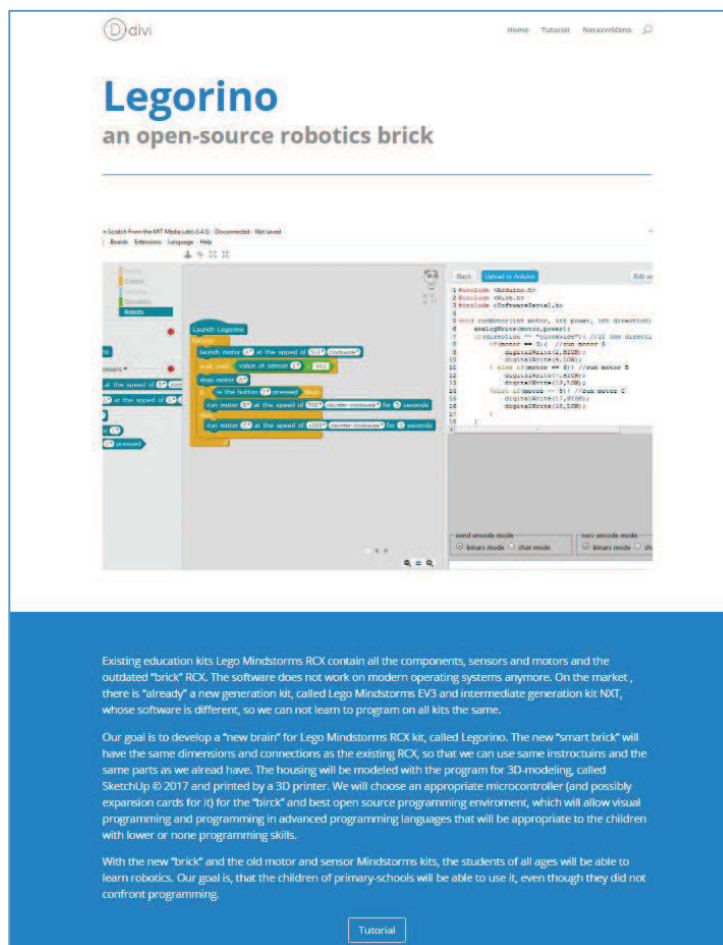
## 5 DRUŽBENA ODGOVORNOST

Ker želimo, da bi naš projekt uporabljalo čim več šol in posameznikov po celem svetu, smo programsko kodo in načrte javno objavili v angleščini pod imenom »Legorino«. S tem bomo omogočili, da bodo lahko šole in posamezniki znova uporabili Lego kocke, ki ležijo v omarah. S tem si bodo zmanjšali stroške za nabavo novih, stare pa ne bodo pristale na odpadu.

### 5.1 Spletna stran

Postopek izdelave ter navodila za uporabo smo na splet objavili v obliki lastnega spletišča. Na spletišču bo lahko kdorkoli dostopal do navodil za izdelavo »pametne kocke«. Znotraj strani bodo povezave do spletnega repozitorija GitHub in strani na družabnem omrežju Facebook, ki sta navedena v naslednjih poglavjih. Za izdelavo spletišča smo uporabili spletna orodja Arnes Splet in Wordpress ter temo spletni strani Divi. Spletišče je na voljo na spletni povezavi:

<http://legorino.splet.arnes.si/>

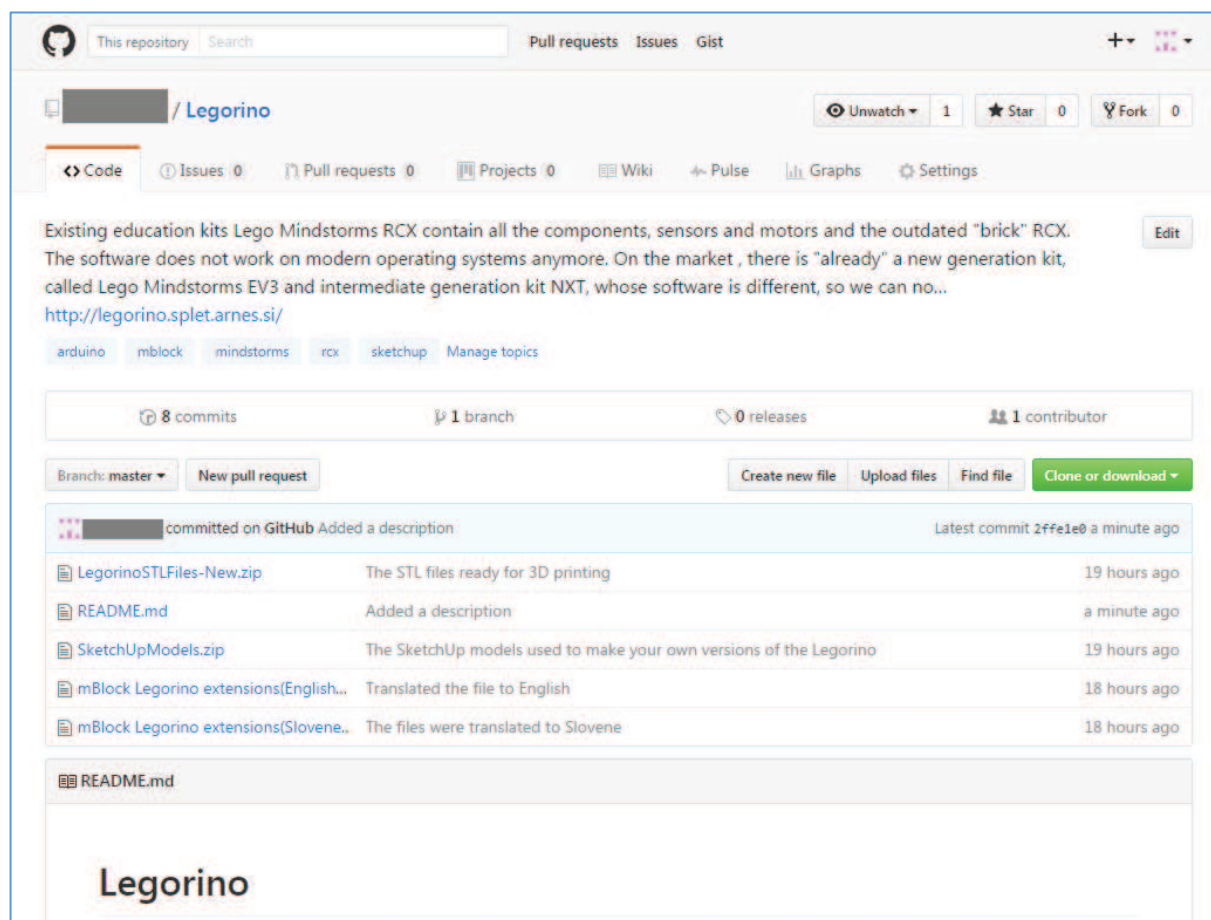


Slika 15 Naslovna stran Legorino spletišča

## 5.2 GitHub

Vse načrte, .skp in .stl modele ter razširitve, ki smo jih uporabili, smo objavili na spletnem repozitoriju GitHub. Vsak si lahko prenese najnovejšo programsko opremo za projekt. Uporabniki lahko obveščajo o morebitnih napakah v programski opremi in pomagajo pri njenem razvoju. Ker so datoteke odprtokodne, jih lahko kdorkoli spreminja, izboljšuje ali prenavlja. S tem bomo dosegli, da bodo lahko drugi nadaljevali projekt, ki smo ga zasnovali.

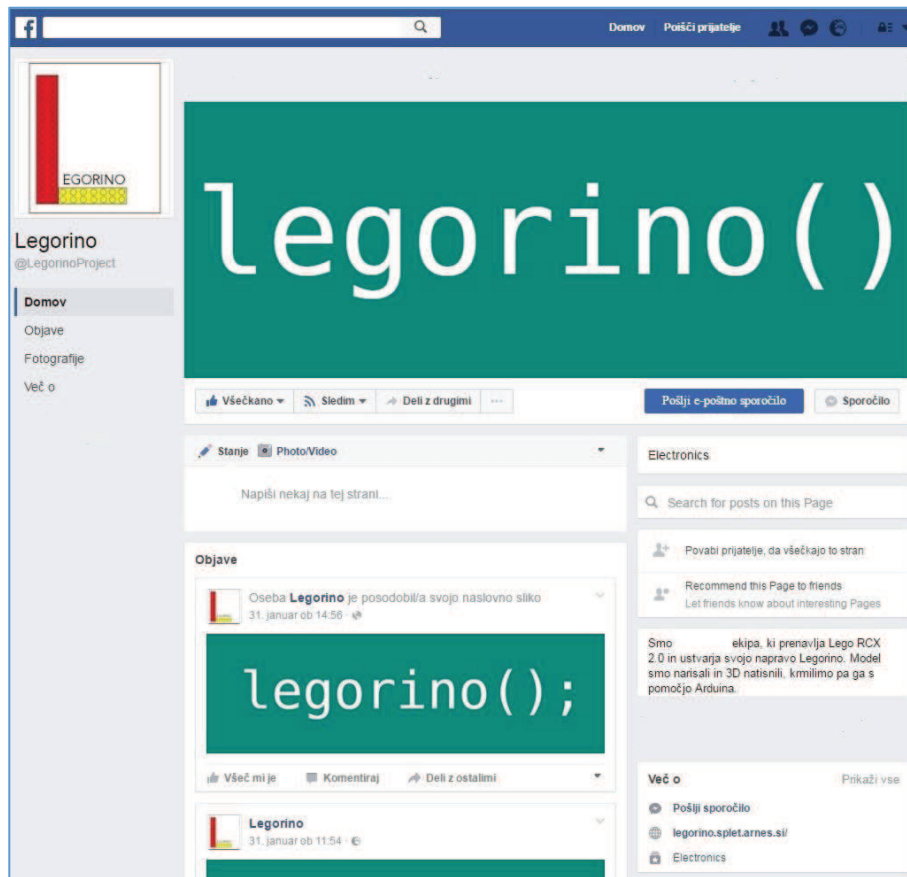
Zaradi anonimnosti avtorjev so na spodnjem posnetku zaslona, njihova imena cenzurirana.



Slika 16 GitHub stran za Legorino

## 5.3 Facebook stran

Da bi naš projekt dosegel čim večje število bodočih uporabnikov, smo ustvarili Facebook stran za naš projekt. Na stran bomo objavljali posodobitve ter novice našega projekta.



Slika 17 Facebook stran

## 6 VIRI

1. **Microcontroller.** *Wikipedia.* [Elektronski] Wikimedia Foundation, Inc. [Navedeno: 7. 12 2016.] <https://en.wikipedia.org/wiki/Microcontroller>.
2. **Analog vs. Digital.** *Sparkfun.* [Elektronski] SparkFun Electronics ®. [Navedeno: 13. 12 2016.] <https://learn.sparkfun.com/tutorials/analog-vs-digital>.
3. **Understanding Microcontroller Pin Input/Output Modes .** *StratifyLabs.* [Elektronski] Stratify Labs, Inc. [Navedeno: 7. 12 2016.] <https://stratifylabs.co/embedded%20design%20tips/2013/10/21/Tips-Understanding-Microcontroller-Pin-Input-Output-Modes/>.
4. **What is the Difference Between a Line Driver, Open Collector, and Push-Pull Quadrature Encoders and Which One Should I Use?** *National Instruments.* [Elektronski] © 2014 National Instruments Corporation, 22. 7 2015. [Navedeno: 5. 1 2017.] [http://digital.ni.com/public.nsf/websearch/0C5091E9099059BC86256FC1007947AA?opendocument&Submitted&&node=133020\\_US](http://digital.ni.com/public.nsf/websearch/0C5091E9099059BC86256FC1007947AA?opendocument&Submitted&&node=133020_US).
5. **PWM.** *Arduino.* [Elektronski] Arduino. [Navedeno: 13. 12 2016.] <https://www.arduino.cc/en/Tutorial/PWM>.
6. **Lego Mindstorms.** Wikipedia : Wikimedia Foundation, Inc.
7. **LEGO. History of LEGO Robotics.** [Elektronski] [Navedeno: 07. december 2016.] <https://www.lego.com/en-us/mindstorms/history>.
8. **LEGO 9V Technic Motors.** [Elektronski] [Navedeno: 18. Januar 2017.] <http://www.philohome.com/motors/motorcomp.htm>.
9. **3D modeling.** *Wikipedia.* [Elektronski] Wikimedia Foundation, Inc. [Navedeno: 7. 12 2016.] [https://en.wikipedia.org/wiki/3D\\_modeling](https://en.wikipedia.org/wiki/3D_modeling).
10. **3D tiskalniki.** *3D tiskalniki.* [Elektronski] [Navedeno: 1. februar 2017.] <http://www.3dtiskalnik.si/3Dtiskalniki.html>.
11. **RASPBERRY PI 3 MODEL B.** Rapsberry Pi : RASPBERRY PI FOUNDATION.
12. **Arduino UNO.** Arduino : ARDUINO S.R.L., 12. 12 2016.

13. **Arduino Nano.** Arduino : s.n.
14. **FAQS. Raspberry Pi.** [Elektronski] RASPBERRY PI FOUNDATION. [Navedeno: 14. 12 2016.] <https://www.raspberrypi.org/help/faqs/>.
15. **Power Consumption Arduino.** *Arduino.* [Elektronski] Arduino. [Navedeno: 14. 12 2016.] <http://forum.arduino.cc/index.php?topic=5536.0>.
16. **What is the Max Power Consumption of Arduino Nano 3.0.** *Stack Exchange.* [Elektronski] Stack Exchange Inc. [Navedeno: 14. 12 2016.] <http://arduino.stackexchange.com/questions/926/what-is-the-max-power-consumption-of-arduino-nano-3-0>.
17. **Product Technical Data Sheets. Duracell.** [Elektronski] [Navedeno: 30. 1 2017.] [https://d2ei442zrkqy2u.cloudfront.net/wp-content/uploads/2016/03/QU1500\\_US\\_UL1.pdf](https://d2ei442zrkqy2u.cloudfront.net/wp-content/uploads/2016/03/QU1500_US_UL1.pdf).
18. **PI 3 BENCHMARKS: THE MARKETING HYPE IS TRUE.** *Hackaday.* [Elektronski] [Navedeno: 5. 1 2017.] <http://hackaday.com/2016/03/01/pi-3-benchmarks-the-marketing-hype-is-true/>.
19. **Alternative Arduino Interfaces.** s.l. : SparkFun Electronics.
20. **Getting Started With mBlock.** *mBlock.* [Elektronski] [Navedeno: 14. 1 2017.] <http://download.makeblock.com/mblock/docs/getting-started-with-mblock.pdf>.
21. **What is miniBloq.** *miniBloq.* [Elektronski] 9. 3 2011. [Navedeno: 15. 1 2017.] <http://blog.minibloq.org/2011/03/what-is-minibloq.html>.
22. **Can't Get I2C to Work on an Arduino Nano? (Pinout Diagrams).** *Big Dan the Blogging Man.* [Elektronski] [Navedeno: 6. 2 2017.] Big Dan the Blogging Man.
23. **HC-06 Bluetooth Module Datasheet.** *Olimex.* [Elektronski] [Navedeno: 7. 2 2017.] <https://www.olimex.com/Products/Components/RF/BLUETOOTH-SERIAL-HC-06/resources/hc06.pdf>.
24. **1.8" Color TFT LED display with MicroSD Card Breakout- ST7735R.** *Adafruit.* [Elektronski] Adafruit ®. [Navedeno: 7. 2 2017.] <https://www.adafruit.com/product/358>.

**25. ENERGIZER NH15-2300. *Energizer*. [Elektronski] [Navedeno: 30. 12 2016.]**

**<http://data.energizer.com/PDFs/nh15-2300.pdf>.**

**26. LEGO 9V Technic Motors . [Elektronski] [Navedeno: 18. Januar 2017.]**

**<http://www.philohome.com/motors/motorcomp.htm>.**



## 7 KAZALO SLIK

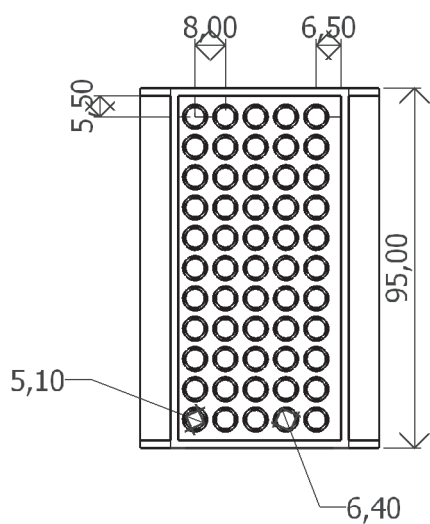
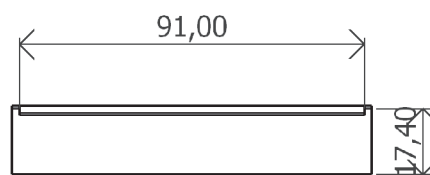
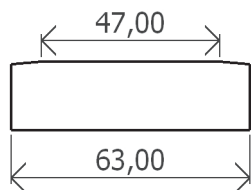
Slika 1: Elektromotorji in senzorji EV3 .....	10
Slika 2: Elektromotorji in senzorji Lego RCX .....	11
Slika 3 Raspberry Pi 3 Model B (11) .....	18
Slika 4 Arduino Uno (12) .....	19
Slika 5 Arduino Nano 3.0 (13) .....	19
Slika 6 Programsko okolje ArduBlock .....	23
Slika 7 Programsko okolje mBlock .....	24
Slika 8 Programsko okolje miniBloq .....	25
Slika 9 3D model ohišja "pametne kocke" .....	27
Slika 10: 3D tisk ohišja.....	28
Slika 11: Testiranje modela, natisnjene s 3D tiskalnikom .....	28
Slika 12 Program znotraj mBlocka .....	30
Slika 13 Naročene komponente v spletni trgovini .....	33
Slika 14: Testiranje uporabnosti s skupino šestošolcev .....	34
Slika 15 Naslovna stran Legorino spletišča .....	35
Slika 16 GitHub stran za Legorino .....	36
Slika 17 Facebook stran .....	37

## 8 KAZALO TABEL

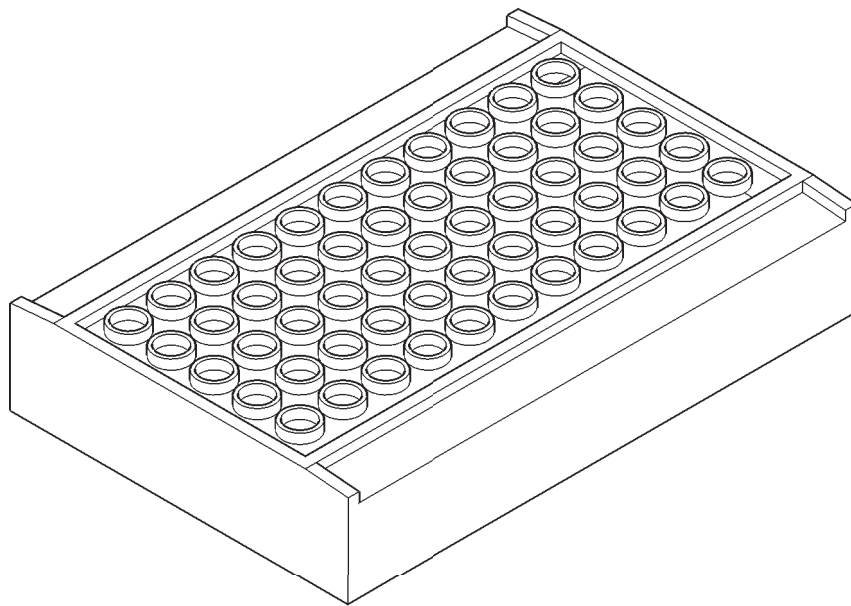
Tabela 1 Specifikacije motorjev v kompletih Mindstorms (8) .....	13
Tabela 2 Primerjava mikrokontrolerov .....	21
Tabela 3: Primerjava programske oprema .....	26

## 9 KAZALO SHEM

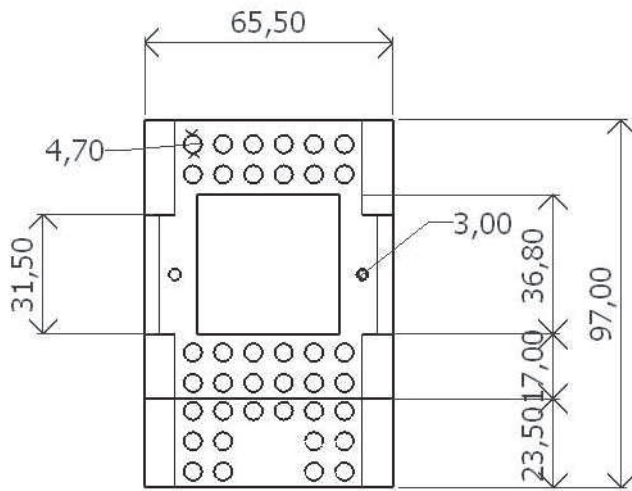
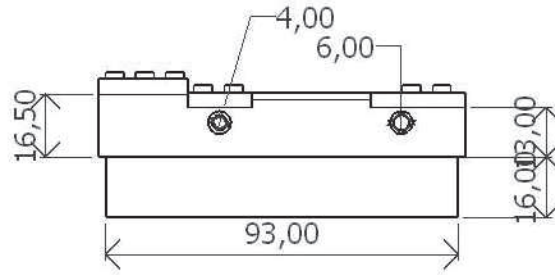
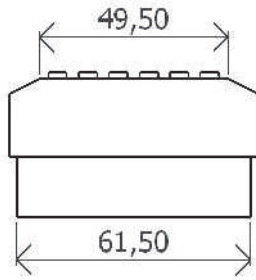
Schema 1: Razlika med digitalnimi vhodi s histerezo in brez nje (3).....	8
Schema 2 Tipi izhodnih enot mikrokontrolerov (4).....	9
Schema 3: Diagram poteka programa .....	31
Schema 4 Komponente povezane na Arduino Nano (22) .....	32



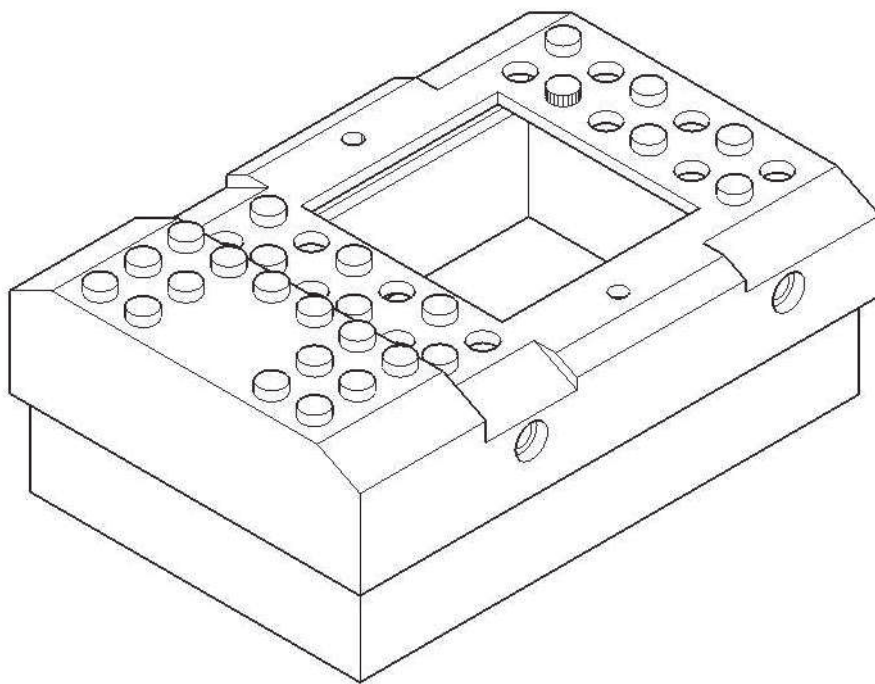
Pozicija: 1		Masa: 24g		
Material: ABS				
Merilo: 1:2		Datum	Ime	Naziv: Legorino pokrov
	Risal	4.2.17	****	
	Pregl.			
	Potrd.			
	Številka risbe: Legorino 1			List: 1
	Osnovna risba:			



Pozicija: 2		Masa: 24g	
Material: ABS			
Merilo: 1:1		Datum	Ime
	Risal	4.2.17	****
	Pregl.		
	Potrd.		
		Naziv: Legorino pokrov	
		Številka risbe: Legorino 1	
Osnovna risba			List: 2



Pozicija: 1		Masa: 32,5g		
Material: ABS				
Merilo: 1:2		Datum	Ime	Naziv: Legorino ohišje
	Risal	4.2.17	*****	
	Pregl.			
	Potrd.			
			Številka risbe: Legorino 1	List: 1
Osnovna risba				



Pozicija: 2		Masa: 32,5g			
Material: ABS					
Merilo: 1:1		Datum	Ime	Naziv: Legorino ohišje	
	Risal	4.2.17	*****		
	Pregl.				
	Potrd.				
				Številka risbe: Legorino 1	List: 2
	Osnovna risba				

```

{
  "extensionName": "RCXMotors and sensors", //Naslov razširitve
  "version": 1.0,
  "sort":1,
  "extensionPort": 0,
  "firmware":"1.0",
  "javascriptURL":"js/arduino.js", //Datoteka za Scratch Mode
  "blockSpecs": [

    // "Block", ki vrtil motor, ki vrtil motor na določeni hitrosti in smeri za omejen čas
    ["w", "vrtil motor %d.motors na hitrosti %d.power v smeri %d.direction za %n
    sekund", "runPwm", "A", "0", "clockwise", "1",

    {"setup":"pinMode({0},OUTPUT);\n", "inc":"","", "def":"","", "work":"runMotor({0},{1},{2});\n_d
    elay({3});\nanalogWrite({0},0);\n", "loop":""}],

    // "Block", ki vrtil motor, ki vrtil motor na določeni hitrosti in smeri za neomejen čas
    ["w", "zaženi motor %d.motors at the speed of %d.power v smeri
    %d.direction", "runDigital", "A", "0", "clockwise",

    {"setup":"pinMode({0},OUTPUT);\n", "inc":"","", "def":"","", "work":"runMotor({0},{1},{2});\n",
    "loop":""}],

    // "Block", ki ustavi določen motor
    ["w", "ustavi motor %d.motors", "runTone", "A",
    {"setup":"","", "inc":"","", "def":"","", "work":"digitalWrite({0},LOW);\n", "loop":""}],

    // "Block", ki vrne analogno vrednost senzorja (0-1023)
    ["R", "vvrednost senzorja %d.sensors", "getAnalog", "1",

    {"setup":"pinMode(A0+{0}-1,INPUT);\n", "inc":"","", "def":"","", "work":"analogRead(A0+{0}-1)",
    "loop":"delay(60);\n"}],

    // "Block", ki vrne boolean vrednost (True ali False)
    ["B", "ali je %d.sensors pritisnjen", "getDigital", "1",

    {"setup":"pinMode(A0+{0}-1,INPUT);\n", "inc":"","", "def":"","", "work":"analogRead(A0+{0}-1)==
    1023", "loop":"delay(60);\n"}],

  ],
  "menus": {
    //Definirani meniji
    "motors":["A","B","C"],
    "power":["0","256","512","768","1023"],
    "direction":["clockwise","counter clockwise"],
    "sensors":["1","2","3"],
  },
  "values":{
    //Definirane vrednosti
    "A":3,
    "B":6,
    "C":5,
    "clockwise": 1,
    "counter clockwise": 2,
  },
  "translators":{
  }
}

```

```

{  "extensionName": "Screen", //Naslov razširitve
   "sort":1,
   "extensionPort": 0,
   "firmware":"1.1.103",
   "javascriptURL":"js/arduino.js", //Datoteka za Scratch Mode
   "blockSpecs": [
     ["h", "Launch Legorino", "runArduino",
      {"encode":"","setup":""}
    ]
  ]
}

//Vspostavi serijsko povezavo (namenjeno bug-fixingu)
Serial.begin(9600);

//Inicializiraj zaslon
Adafruit_ST7735 tft = Adafruit_ST7735(TFT_CS,  TFT_DC, TFT_RST); //Inicializiraj zaslon

//Nasavi načine pinov
pinMode(A6, INPUT);
pinMode(A7, INPUT);
pinMode(2, OUTPUT);
pinMode(4, OUTPUT);
pinMode(7, OUTPUT);
pinMode(12, OUTPUT);
pinMode(17, OUTPUT);
pinMode(18, OUTPUT);

//Nastavitve zaslona
tft.initR(INITR_BLACKTAB);
tft.fillScreen(ST7735_BLACK);
tft.setCursor(0, 0);
tft.setTextColor(ST7735_WHITE);
tft.setTextWrap(true);

//Izpiši sporočilo
tft.print("\nPritisni tipko za nadaljevanje");
Serial.println("\nPritisni tipko");

//Počakaj, da je tipka pritisnjena
while(analogRead(A7) < 1023) {
}

//Izpiši sporočilo
tft.fillScreen(ST7735_BLACK);
tft.setCursor(0, 0);
Serial.println("\nIzvajam program");
tft.print("\nIzvajam program");

"inc":

//definiraj funkcijo runMotor
void runMotor(int motor, int power, int motorDirection){
analogWrite(motor, power);

//Če je smer urinega kazalca
if(motorDirection == 1){
if(motor == 3){ //vrti motor A
digitalWrite(2, HIGH);
digitalWrite(4, LOW);
} else if(motor == 6){ //vrti motor B
digitalWrite(7, HIGH);
digitalWrite(12, LOW);
} else if(motor == 5){ //vrti motor C
digitalWrite(17, HIGH);
digitalWrite(18, LOW);
}
}

//če je naspr. smer ur. kaz.
} else if(motorDirection == 2){
if(motor == 3){ //vrti motor A
digitalWrite(2, LOW);
digitalWrite(4, HIGH);
} else if(motor == 6){ //vrti motor B
digitalWrite(7, LOW);
}
}

```

```
digitalWrite(12,HIGH);\n}else if(motor == 5){ //vrti motor C\ndigitalWrite(17,LOW);\n digitalWrite(18,HIGH);\n}\n}\n}\n\n//Vstavi knjižnjice\n#include <Adafruit_GFX.h> // Core graphics library\n#include <Adafruit_ST7735.h> // Hardware-specific library\n#include <SPI.h>\n",\n"def": "\n\n//Definiraj pine, ki krmilijo zaslon\n#define TFT_CS      10\n#define TFT_RST     9\n#define TFT_DC      8\n#define TFT_SCLK   13\n#define TFT_MOSI   11\n",\n"work": "", "loop": "\n\n//Če je tipka pritisnjena, ustavi program in pojdi na začetek\nif(analogRead(A6)==1023){\nsetup();\n}\n}\n",\n  ],\n  "menus": {\n  },\n  "values":{\n  },\n  "translators":{\n  }\n}\n}
```



## Projekt Legorino

1. Po načrtih iz škatle sestavite avtomobil (stran 1). Izdelajte program, ki bo avto vodil, da bo vozil v ravni črti, dokler ne bo pritisnil gumba STOP.
2. Izdelajte program, ki bo avto vodil, da bo vozil v ravni črti 2 sekundi.
3. Izdelajte program, ki bo avto vodil, da bo vozil ravno do ovire.
4. Po končanem delu naj 1 član skupine slika QR kodo in skupaj izpolnite anketo.

