

UNIVERZA NA PRIMORSKEM
FAKULTETA ZA MATEMATIKO, NARAVOSLOVJE IN
INFORMACIJSKE TEHNOLOGIJE

Zaključna naloga

Razvoj pametne stekleničke za vodo

(The development of a smart bottle)

Ime in priimek: Tadej Dragutinovič

Študijski program: Računalništvo in informatika

Mentor: doc. dr. Peter Rogelj

Koper, maj 2016

Ključna dokumentacijska informacija

Ime in PRIIMEK: Tadej DRAGUTINOVIĆ

Naslov zaključne naloge: Razvoj pametne stekleničke za vodo

Kraj: Koper

Leto: 2016

Število listov: 45

Število slik: 13

Število tabel: 3

Število prilog: 1

Število strani prilog: 3

Število referenc: 21

Mentor: doc. dr. Peter Rogelj

Ključne besede: Pametna steklenička, zdravo življenje, Voda, Arduino, senzor pretoka vode.

Izvleček:

Vsak med nami, ne glede na tehnološko znanje in potrebe je vsakodnevno obkrožen s tehnologijo. Zaključna naloga opisuje razvoj pametne stekleničke za vodo, katere namen je osveščanje ljudi glede pravilne hidracije. Cilj naloge je zajemal izdelavo prototipa pametne stekleničke za vodo, ki bi bil lep na pogled in vsestransko uporaben. V interesu mi je bilo izdelati uporabno rešitev, ki bi bila tehnološko podprta a hkrati dovolj enostavna, da jo lahko uporabljajo vsi, od otrok do starejših. V prvem delu naloge je opisana problematika ter dosedanje rešitve. V nadaljevanju sem opisal vso uporabljeno strojno opremo in primerjal dve možni rešitvi problema. Zaključil sem s primerjavo rezultatov in možnimi izboljšavami prototipa. Rezultat zaključne naloge je izdelan prototip stekleničke, ki spremlja hidracijo tekom dneva in nas po potrebi opominja na pitje tekočine.

Key words documentation

Name and SURNAME: Tadej DRAGUTINOVIĆ

Title of final project paper: The development of a smart water bootle

Place: Koper

Year: 2016

Number of pages: 45

Number of figures: 13

Number of tables: 3

Number of appendices: 1

Number of appendix pages: 3

Number of references:

21

Mentor: Assist. Prof. Peter Rogelj, PhD

Keywords: Smart bootle, water, Arduino, flow sensor.

Abstract: Each of us, regardless our technological knowledge and needs is every day surrounded with technology. My thesis describes the development of a smart water bootle, which aims to raise people's awareness regarding proper hydration. My goal was creation of a prototype, which would look good and be useful in everyday life. I wanted to create a useful solution that would be technologically supported yet simple enough that it can be used by everyone, from children to the elderly. In the first part of the thesis i described the problem and existing solutions. The second part describes the hardware i used and shows a comparison of the two best solutions. I concluded with the final solution and possible improvements of the prototype. The result of the thesis is a prototype water bottle, which monitors hydration during the day and reminds us to drink more water everyday.

Zahvala

Iskreno se zahvaljujem vsem, ki so mi v kakoršni koli meri pomagali pri izdelavi diplomske naloge.

Posebna zahvala gre mentorju prof. dr. Petru Roglju, za vse nasvete, strokovno pomoč in prijazno podporo.

Zahvaljujem se svoji družini, ki mi je omogočila študij in puncu, ki me je tekom študija in pisanja zaključne naloge podpirala.

Kazalo vsebine

1	Uvod	1
2	Uporabljena oprema	3
2.0.1	Mikrokrmilnik Arduino(podrobneje)	3
2.0.2	OLED prikazovalnik	4
2.0.3	Senzor pretoka	5
2.0.4	Senzor razdalje	6
2.0.5	Pospeškometer	7
3	Programska oprema	9
3.1	Razvojno okolje za Arduina	9
3.2	Programski jezik	9
3.3	Program	10
4	Prototipska rešitev	11
4.1	PROTOTIP 1	13
4.1.1	Strojni del - PROTOTIP 1	14
4.1.2	Programski del - PROTOTIP 1	15
4.1.3	Izpisovanje na zaslon	15
4.1.4	Merjenje pospeška in orientacije	18
4.1.5	Delovanje senzorja razdalje	19
4.2	PROTOTIP 2	20
4.2.1	Strojni del - PROTOTIP 2	20
4.2.2	Programski del - PROTOTIP 2	22
4.2.3	Merjenje pretoka	22
4.2.4	Stanje mirovanja	23
5	Uporaba in delovanje	25
5.1	Rezultati	26
5.1.1	Odločitev	28
5.1.2	Možne izboljšave	29

6 Zaključek	31
--------------------	-----------

7 Literatura in viri	33
-----------------------------	-----------

Kazalo tabel

1	Pričakovanja	12
2	Meritve	27
3	Kriteriji za izbor	30

Kazalo slik

1	OLED prikazovalnik	4
2	Senzor pretoka	5
3	Senzor razdalje	6
4	Pospeškometer	7
5	Izhodni graf vrednosti (pospeškometer)	8
6	Predprototip	11
7	Diagram senzorja razdalje	19
8	Izdelava stekleničke	21
9	Prikaz delovanja	25
10	Rezultati meritev	27
11	Steklenička	28

Kazalo prilog

Priloga A: Programska koda

Seznam kratic

<i>OLED</i>	Organic light-emitting diode
<i>LED</i>	Light-emitting diode
<i>AC</i>	Izmenični tok (Alternating current)
<i>DC</i>	Enosmerni tok (Direct current)
<i>USB</i>	Universal Serial Bus
<i>kHz</i>	Kilohertz
<i>ipd.</i>	in podobno
<i>GND</i>	masa/ozemljitev (Ground)
<i>SDA</i>	Serial data line
<i>SCL</i>	Serial clock line
<i>MEMS</i>	Micro Electro Mechanical System

1 Uvod

Ko nanese beseda na zdravo življenje, takoj pomislimo na šport in zdravo prehrano. Seveda pri tem ni nič spornega, vendar pogosto pozabljamo na glavno človeško potrebo, ki je potreba po tekočini. Človek lahko brez hrane zdrži nekaj dni, brez vode pa le nekaj ur. Vsakodnevne težave in stres nas pogosto odvrnejo od razmišljanja, koliko vode smo popili. Živimo v dobi tehnologije, vse okoli nas so računalniki, senzorji in celotno potrebno znanje za razvoj tehnološko dovršenih rešitev.

Moja ideja temelji na združitvi vsakodnevno uporabljene stekleničke za vodo in tehnologije, ki nam je na voljo za izdelavo tehnološko podprte stekleničke, ki bi podkrepila našo zavest po pitju tekočine. Koncept je oblikovan tako, da klasično stekleničko dopolnjuje z naprednimi tehnologijami s področja mikroprocesorjev. Pametna steklenička nam ponuja številne funkcije: od merjenja popite tekočine do obvestil, ko je popita količina prenizka. Pred samim razvojem pametne stekleničke sem raziskal že obstoječe prototipe in rešitve tega problema. Opazil sem, da se je že nekaj posameznikov lotilo izdelave prototipov, vendar se nobeden od že predstavljenih ni uspešno prebil na trg. Prav ta korak je bil odločilen, da se resnično lotim izdelave svojega prototipa. Prepričan sem, da je rešitev v preprostosti in enostavni uporabi. Stekleničko želim približati vsem uporabnikom, tako tistim mlajšim, ki so tehnično bolj podkovani, kot tistim malo zrelejšim, ki vedo o tehnologiji manj. Steklenička za svoje delovanje ne bo potrebovala ničesar drugega kot polno baterijo in nekaj vode.

V prvem delu diplomske naloge bom definiral pojem voda kot potreba našega telesa. Marsikdo je seznanjen z informacijo, koliko vode naj bi posameznik popil, vendar se ne zaveda, kakšne posledice lahko prinese premalo popite tekočine za naše telo. Nadaljeval bom s programskimi in strojnimi deli, kateri so bili nujni za samo realizacijo prototipa ter končal s predstavitvijo izdelane stekleničke in možnimi izboljšavami v prihodnosti. Odrasel človek izgubi v povprečju tri do štiri litre vode dnevno. Večino vode izločimo s prebavljanjem, znojenjem in uriniranjem. Ob večjih naporih ali povišanih temperaturah okolja je potreba po vodi še večja, zaradi česar moramo izločeno tekočino nadomestiti s pitjem.

Pomanjkanje vode v človeškem telesu imenujemo dehidracija. Ta se lahko pojavi kot posledica premajhnega vnosa tekočin ali bolezenskega stanja. V večini primerov je dehidracija posledica povišane telesne temperature, driske, zastrupitve s hrano ali

povečanega potenja pri višjih temperaturah. Vodo moremo telesu zagotoviti pravočasno, preden imamo občutek žeje, saj so takrat naše celice že dehidrirane. Občutek žeje se pojavi, ko se zaradi pomanjkanja vode v krvi poveča koncentracija natrija in mora telo s hipofizo v kri izločati antidiuretinski hormon, ki ohranja vodo v ledvicah. V primerih, ko je raven vode v telesu tako nizka, da je ne moremo več nadomestiti s pitjem, lahko ogrozimo svoje življenje. Odrasel človek naj bi za normalno delovanje organizma popil od 2 do 2,5 litra vode dnevno, kar je približno 35 ml vode na kilogram telesne teže. Vsi vemo, da brez vode ne moremo živeti, saj le-ta sestavlja kar 66 procentov človeškega telesa, vendar je malokdo seznanjen, da je predoziranje z vodo lahko usodno za človeški organizem. Študija revije *New England Journal of Medicine* [1] je razkrila podatke, da se šestina vzdržljivostnih športnikov sooča s problematiko hiponatremije oziroma redčenja krvi. Pri hiponatremiji se koncentracija natrija v krvi zmanjša pod minimalno mejo, kar se občuti kot glavobol, utrujenost, slabost, v hujših primerih pa pride do zastrupitve z vodo, kar lahko sproži psihično zmedenost in bruhanje. Problem pri merjenju optimalne količine tekočine pa je, da optimalna količina seveda ni enaka pri vsakem človeku. Zdrave ledvice v mirovanju izločijo od 800-1000 ml na uro. Če ista oseba teče na maratonu, se zaradi stresa dvigne raven vazopresina (antidiuretičnega hormona) zaradi česar se sposobnost izločene količine zmanjša na 100 ml na uro, pri čemer hitro pride do hiponatremije. V taki situaciji je 1000 ml popite tekočine veliko preveč, primerna pa bi bilo za istega človeka optimalno v mirovanju.

Moja ideja se mi je prvič porodila zaradi osebne izkušnje pred nekaj leti. Premajhna količina popite vode je bila povod za moje težave z ledvičnimi kamni. Ker običajno zaradi vsakodnevnih aktivnosti pozabljamo piti tekočino, so pred nekaj leti na trg prišle tako imenovane stekleničke za večkratno uporabo. Osnovna naloga stekleničk je zagotovljena, v kolikor imamo dovolj samokontrole, da jo uporabljamo in dovolj pogosto polnimo, vendar včasih tudi to ne zaleže. Sprva sem si dnevno zadajal cilj, koliko stekleničk popiti ter si nanje z vodoodpornim flomastrom risal črte, dokler seveda ni bila vsa flaška polna črt, po katerih se nisem več znašel. Povsod nas obdaja veliko dovršenih naprav in senzorjev za lažjanje ponavljajočih opravil in izboljšanje življenja. Pomislil sem, da imamo doma razvlaževalce zraka, termostate, razne senzorje gibanja ipd. Zakaj ne bi izkoristili preprosto tehnološko rešitev za opominjanje ljudi na pitje tekočine? Ideja mi je bila zelo všeč, želel sem rešitev zame, vendar so se kmalu po tem oglasili tudi sorodniki in prijatelji, ki so želeli nekaj podobnega tudi zase. Moja ideja je dobila smisel in tako je nastala pametna steklenička za vodo.

Cilj moje diplomske naloge je razviti preprost a učinkovit izdelek. Od osnovnega prototipa do končnega izdelka je kar nekaj razvijanja, zato se bom osredotočil na delovanje. Idealna steklenička bi bila steklenička iz ekoloških materialov, majhno porabo energije in privlačna na pogled.

2 Uporabljena oprema

Osrednji del pametne stekleničke je razvojna ploščica Arduino. V testni fazi sem uporabil Arduino uno verzijo, saj je zaradi svoje priročnosti in širokega nabora vhodov ter izhodov primerna za poskušanje delovanja. Za končni izdelek sem izbral Arduino nano verzijo, katera poleg svoje kompaktnosti prinaša tudi mini USB vhod, ki je v današnjem svetu zelo razširjen in predstavlja standard za polnjenje prenosnih naprav. Za potrebe zaznavanja količine in izpisovanja podatkov uporabniku sem uporabil vrsto senzorjev in modulov, ki dopolnjujejo samo razvojno ploščico. V testni fazi se bodo vse elektronske komponente napajale preko zunanje baterije (angl. *battery bank*), idealna steklenička bi napajala komponente preko integrirane polnilne baterije.

2.0.1 Mikrokrmilnik Arduino(podrobneje)

Mikrokrmilnik so razvili na šoli oblikovanja v italijanskem mestu Ivrea in predstavlja velik mejnik odprtokodne strojne opreme. Ploščica je bila osnovana in mišljena kot pripomoček za učenje programiranja v šolah, vendar je zelo hitro pridobila na svoji priljubljenosti tudi v ostalih tehnoloških institucijah. Danes lahko na spletu najdemo veliko različic Arduina, kar je bil tudi cilj ustanoviteljev. Ideja o odprtokodni ploščici z mikroprocesorjem je bila dodeliti pripomoček tistim, ki bi radi programirali, vendar ne vedo, kje začeti. Vsakdo lahko izdelava svoje vezje po Arduinovi shemi in ga poljubno dodela tako, da ustreza njegovim potrebam. Ploščica je zasnovana okrog mikroprocesorja ATMega328 s prijaznim razvojnim okoljem. Frekvenca delovanja je 16Mhz, za programe pa ima namenjenih 32 Kb flash spomina. Ploščica ima 14 digitalnih vhodov oz. izhodov ter 6 analognih vhodov. Na ploščici najdemo poleg USB vhoda ter reset tipke tudi 2 LED diodi. Napajanje se vrši preko USB vhoda, kar pride prav v fazi same izdelave in programiranja. Ko izdelamo kočno verzijo, lahko za napajanje uporabljamo namenski vhod z uporabo AC-DC električnega adapterja ali baterije. Ploščica deluje pri napetosti 6-20V oz. 6-12V pri baterijskem napajanju.

2.0.2 OLED prikazovalnik

Za razliko od ostalih že razvitih prototipov pametnih stekleničk sem želel svojo izdelati tako, da za svoje delovanje ne bi potrebovala druge naprave (mobilnega telefona ali pametne ure). Želel sem enostavno rešitev z majhnim prikazovalnikom, ki služi za odčitavanje popite tekočine.

Izbral sem OLED prikazovalnik, neznanega kitajskega proizvajalca, ki podpira I2C vodilo.

OLED je tehnologija, ki sloni na samoosvetljenem zaslonu, sestavljenem iz tankega večplastnega sistema organskih folij med anodo in katodo.

V primerjavi z LCD zasloni porabi OLED bistveno manj električne energije (poraba je največ 50 mA, povprečno pa približno 30 mA), saj za delovanje ne potrebuje osvetljenega ozadja (angl. *backlight*). Poleg velike svetilnosti ga odlikuje tudi širok vidni kot (več kot 160 stopinj) in hitra odzivnost. Zaradi svojih lastnosti, kot so: majhna poraba energije, tankost in vzdržljivost tudi pri ekstremnih temperaturah je zelo primeren za izdelavo pametne stekleničke za vodo.



Vir: (<http://img02.taobaocdn.com/imgextra/i2/773299280/T2W8WPXEXXXXXXXXXX-!!773299280.jpg>)
(24.03.2016)

Slika 1: OLED prikazovalnik

2.0.3 Senzor pretoka

Merjenje količine pretoka opravlja senzor kitajskega proizvajalca Sea industries, model ZF-S201. Senzor je sestavljen iz plastičnega ohišja ter turbine v notranjosti. Turbina je pritrjena na integriran magnetni senzor, ki oddaja električni impulz z vsako rotacijo turbine. Povezava poteka preko treh povezav: dve sta za napajanje in ena za merjenje izhodnega impulza. Z merjenjem impulzov enostavno izračunamo pretok vode. Impulzni signal je preprost kvadratni val, zato ga lahko enostavno pretovirimo v litre na minuto.

Algoritem 1: Lcd prikazovalnik

$$1 \text{ Frekvenca (Hz) } / 7,5 = \text{pretok v L / min.}$$

Vsak impulz je ekvivalenten 2,25 ml tekočine. Po specifikacijah proizvajalca lahko senzorjeva natančnost odstopa za 5 procentov. Delovanje senzorja je podrobneje opisano v poglavju 4.2.3.



(a) Prečni prerez senzorja pretoka



(b) Senzor pretoka

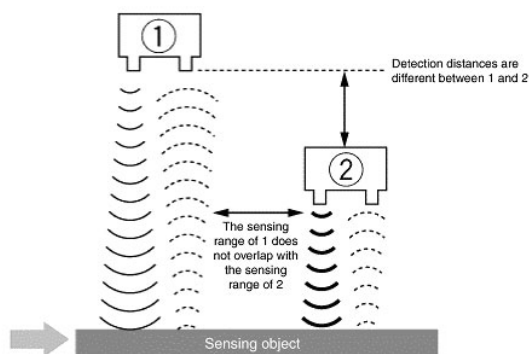
Vir: (<https://www.ia.omron.com/support/faq/answer/include/faq01408/img/faq01408-1.jpg>) (07.04.2016)

Slika 2: Senzor pretoka

2.0.4 Senzor razdalje

Za merjenje razdalje sem uporabil ultrazvočni senzor HC-SR04. Tovrstni senzori so zaradi ugodne cene in zanesljivosti pogosto uporabljeni v projektih robotike. Spekter uporabe ultrazvočnih senzorjev je zelo širok. Novejši fotoaparati uporabljajo poenostavljeno različico senzorja razdalje za samodejno ostritev slike, pametni sesalci in brezpilotni helikopterji (angl. *drone*) pa za odmikanje in zaznavanje objektov na njihovi poti. Senzor HC-SR04 ni vodoodporen, kar je predstavljalo velik problem pri samem razvoju prototipa.

Senzor je z Arduinom povezan preko štirih povezav. Dve sta namenjeni napajanju, dve pa prenosu podatkov. Za prenašanje podatkov imamo vhodni signal (angl. *Triger pulse input*) ter izhodni signal (angl. *Echo pulse output*). Senzor lahko meri oddaljenost v rangu od 2 cm do 400 cm pri 15-procentnem kotu.



(a) Delovanje senzorja



(b) Senzor razdalje

Vir: (<https://www.kjdelectronics.com/assets/images/HCSR04.jpg>) (07.04.2016)

Slika 3: Senzor razdalje

2.0.5 Pospeskometer

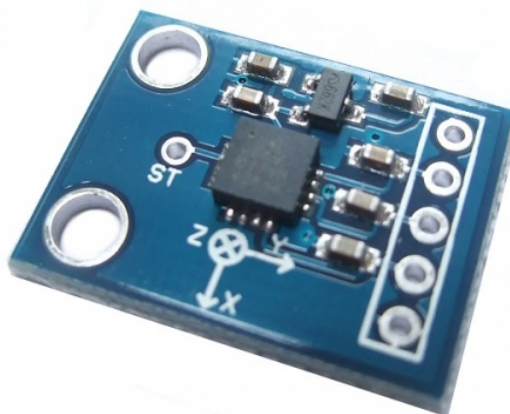
Za izvedbo stekleničke za vodo je bilo potrebno merjenje tekočine omejiti na tiste primere, kadar stekleničke ne prenašamo oz. ni podvržena prekomernemu premikanju, slednje namreč moti pravilno odčitavanje nivoja vode. Rešitev sem našel s pomočjo senzorja pospeška in gravitacije (model ADXL335).

Pospeskometre delimo v različne skupine po načinu delovanja.

Najpogostejši tipi so :

- uporovni,
- kapacitivni,
- optični,
- vibrirajoči kvarc.

Senzor, katerega sem uporabil v zaključni nalogi, spada med pospeškometre, ki uporabljajo princip MEMS. Taka vrsta senzorjev je zelo priljubljena zaradi svoje kompaktnosti in majhne porabe električne energije. Poleg veliko dobrih lastnosti, ima pa slabo lastnost, ki je zelo kompleksna izdelava. Izdelani so s pomočjo tehnologije mikrofabrikacija. Izdelujejo jih z nanašanjem tankih filmov in ionov na naelektreno podlago oz. v nekaterih primerih z jedkanjem. Materiali, iz katerih so senzorji sestavljeni, so silicijeve rezine, polisilikoni, silicijev nitrid, steklo, zlato, aluminij ...

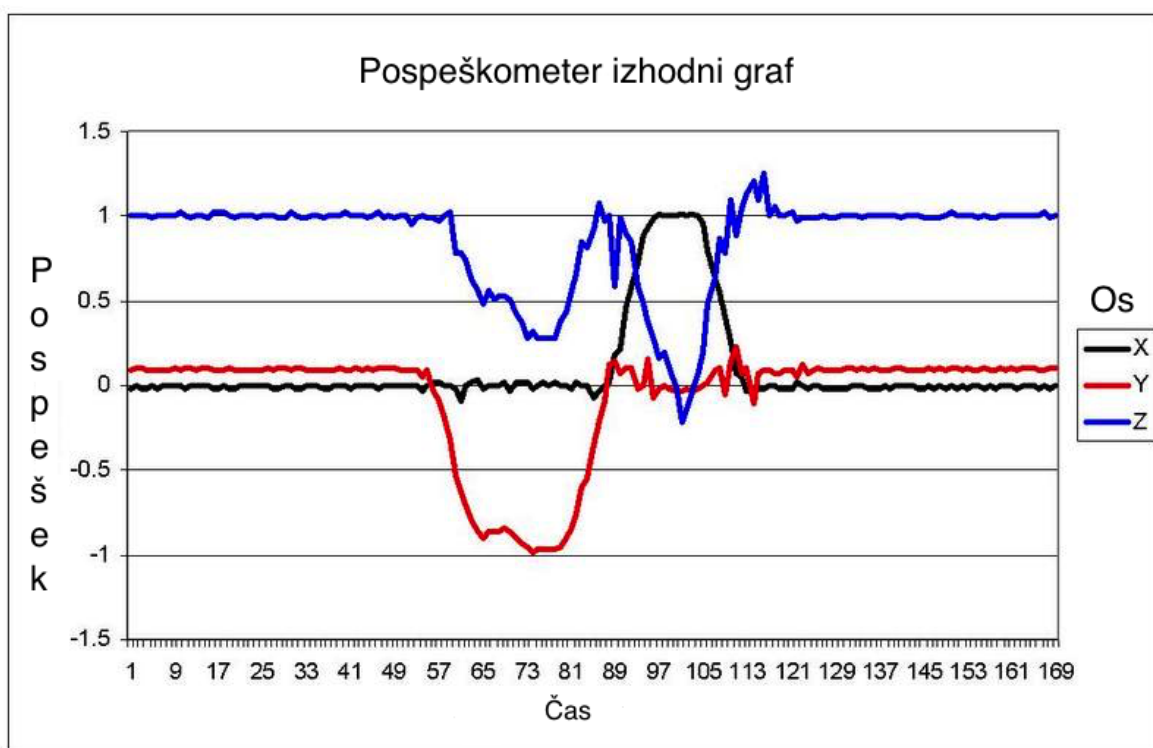


Vir: (<http://www.hobbytronics.co.uk/image/cache/data/ht/sensors/adxl335-breakout-500x500.jpg>), Avtor: Winifred E. Torres, (07.04.2016)

Slika 4: Pospeskometer

Danes je uporaba pospeškometrov zelo razširjena. Najdemo jih v večini mobilnih telefonov, fotoaparatom, igralnih konzolah ipd. Pospeškometri so uporabljeni za najrazličnejše naloge, od preprostih zaznavanj smeri zaradi orientacije zaslona do zaznavanja trkov in sprožitve varnostnih blazin v avtomobilih.

V osnovi poznamo tri vrste senzorjev, ki se delijo na enoosne, dvoosne in triosne. Razlika med njimi je v številu zmožnosti zaznavanja ortogonalnih pospeškov. Senzor, katerega sem uporabil, lahko meri pospeške v treh prostorskih smereh, zaradi česar ga imenujemo triosni senzor. Senzor ima tri analogne izhode (X, Y, Z) in vsak predstavlja eno os. Deluje na 3,3 V napetost in meri pospešek z minimalnim obsegom ± 3 g.



Vir: (<http://forums.pelicanparts.com/uploads16/3+axis+output+graph1265224544.jpg>), Avtor: Winifred E. Torres, (09.04.2016)

Slika 5: Izhodni graf vrednosti (pospeškometer)

3 Programska oprema

3.1 Razvojno okolje za Arduina

Veliko mero končnega izdelka zajema programska koda, ki se izvaja na razvojni ploščici. Razvojno okolje sestavlja paket Arduino programov in nam omogoča enostavno pisanje kode ter nalaganje programov na ploščico Arduino.

Razvojno okolje vsebuje grafični vmesnik, prostor za opozorila, konzolo in serijski monitor (terminal).

Grafični vmesnik je napisan v Javi, ki deluje na različnih operacijskih sistemih.

3.2 Programski jezik

Programski jezik temelji na jeziku Wiring in je soroden jezikoma C ter C++ v precej poenostavljeni različici. Poleg osnovnih podatkovnih struktur in krmilnih stavkov potrebujemo za programiranje tudi določene specifične ukaze za delo z vhodno/ izhodno komunikacijo.

Primer specifičnih ukazov:

- `analogRead()` - uporablja se, kadar želimo vhod brati v analognem načinu. (vrednosti so od 0-1023)
- `digitalWrite()` / `digitalRead()` - pisanje/branje vhoda kot logična 1 oz. 0
- `pinMode()`- v ukazu določimo ali bo neka nogica (angl. *pin*) vhod ali izhod.

Razvojno okolje vključuje knjižnice, ki predstavljajo najpogostejše ukaze in poenostavijo kodiranje. Glavna knjižnica, vsebovana v vsakem programu je arhitekturna Atmel AVR knjižnica.

Paket vsebuje tudi serijsko komunikacijo med ploščico in programom, v kateri lahko preizkušamo delovanje programa. Nalaganje kode na ploščico poteka preko Arduino "bootloader-ja".

3.3 Program

Vsak krmilni program je na začetku imenovan skica (angl. *sketch*).

Program sestavlja:

- opis programa,
- setup,
- zanka loop.

Tri glavne dele lahko dopolnjujejo dodatne funkcije. Prvi glavni del je opis programa, kjer v komentarju zapišemo kaj program počne in druge lastnosti, ki bi lahko nadaljnjemu uporabniku prišle prav pri razumevanju kode.

Druga se izvede inicializacijska metoda `SETUP()`. Ob vsakem zagonu se funkcija `SETUP()` zažene samo enkrat, na začetku izvajanja programa. V funkciji `SETUP` najdemo inicializacijo vseh spremenljivk, tip serijske komunikacije ter določimo tipe priključkov in knjižnice.

Funkcija `LOOP()` se, kot že iz imena razberemo, kliče v zanki in je osrednji del našega programa. Preden se napisani program prenese na ploščico, ga v konzoli preizkusimo in odpravimo morebitne napake.

4 Prototipska rešitev

Temeljni problem, okrog katerega se je samo iskanje rešitve začelo, je bil način merjenja tekočine. Za meritve lahko uporabimo najrazličnejše načine: od merjenja teže vode, merjenje pritiska, merjenja pretoka do merjenja razdalje. Po obširnem testiranju sem na podlagi kriterijev izbral dve po mojem mnenju najboljši metodi (v nadaljevanju imenovani PROTOTIP1 in PROTOTIP 2). Kriteriji za izbor so bili točnost meritev, težavnost implementacije ter učinkovitost in praktičnost metode (slednja predstavlja čiščenje, različne načine uporabe, zunanji izgled itd.). Prva rešitev - PROTOTIP1 je implementacija stekleničke s senzorjem razdalje (na spodnji sliki). Ultrasonični senzor oddaja visokofrekvenčni zvok in na podlagi odboja visokofrekvenčne motnje izračuna razdaljo. Druga rešitev - PROTOTIP2 predstavlja pametno stekleničko, ki deluje s pomočjo senzorja pretoka. (Slika 6.) Omenjeni senzor je sestavljen iz elektromagnetnega tipala ter vodne turbine, ki meri obrate in količino pretoka. Svojo raziskavo sem dopolnjeval sproti na bazi rezultatov obeh prototipov. Prototipiranje sem razdelil v tri faze: zamisel, izdelava in testiranje. Po omenjenih treh fazah, sem se na podlagi rezultatov odločil, kateri prototip izdelati do končne faze in katerega opustiti. Obe metodi sem realiziral na preprost način tako, da sem lahko testiral senzorja. Za izdelavo pred prototipa sem uporabil kozarec za vlaganje, v katerega sem izvrtal odprtini za senzorja. Nivo vode sem s pomočjo merice izmeril ter tako testiral oba senzorja. Začetne meritve so bile na merilcu razdalje zgolj v centimetrih, v merilcu pretoka pa v ml. Slednja rešitev je v začetku kazala veliko odstopanje od dejanske količine. Potrebna je bila kalibracija vrednosti pretoka v ml na sekundo. Po korekciji napak, sta bila senzorja natančna na ± 3 procenete. (Za rezultate meritev glej poglavje 4.3 v Tabeli 10).



Slika 6: Predprototip

Tabela 1: Pričakovanja

Pričakovanja		
Kriterij	PROTOTIP1	PROTOTIP 2
TOČNOST MERITEV	Senzor razdalje bo zaradi slabega odboja vodne gladine odstopal za več kot 5 procentov in ne bo konstanten pri svojih meritvah. Predvidevam da bo težava tudi premikanje vode in stik sensorja s tekočino.	Senzor za merjenje tekočine bo bolj natančen od sensorja razdalje in rezultati meritev bodo konstantni.
TEŽAVNOST IMPLEMENTACIJE	Senzor razdalje bo kompleksen za implementacijo zaradi svoje sestave, vendar predvidevam da bo mogoče sensor implementirati v pokrovček stekleničke in izolirati s tankim pleksi steklom da ne bo v stiku s tekočino.	Zaradi svoje velikosti bo senzor kompleksno implementirati v majhen pokrovček stekleničke. Vodoodpornost bo enostavno doseči z navojem in gumjasto podložko.
UČINKOVITOST	Senzor razdalje ne bo učinkovita metoda za merjenje tekočine. Senzor bo veliko enostavneje implementirati vendar bo njegova nekonstantnost razlog za zavrnitev prototipa	Senzor bo učinkovit pri meritvah vendar težaven za implementacijo.

KOMENTAR: Vsi zapisi znotraj tabele pričakovanja predstavljajo osebne predpostavke in hipoteze pred samo izdelavo in testiranjem prototipov.

4.1 PROTOTIP 1

Razvoj prototipa je od vsega začetka predstavljal določene težave. Pri izdelavi sem bil zelo skeptičen glede delovanja izvedbe, saj omenjeni senzor ponavadi ni uporabljen v rešitvah, ki predvidevajo tekočino.

Senzor sem sprva testiral s predmeti, da sem lahko potrdil njegovo natančnost, nakar sem poskusil z merjenjem razdalje do vode. Ultrasonični senzor, kot že ime pove uporablja visoko frekvenčne signale ter meri razliko med oddanim in prejetim časom signala. Zvok deluje v teoriji zelo podobno kot svetloba, le da je veliko bolj občutljiv na spremembe podlage, kar je tudi potrjevalo mojo skeptičnost do odboja zvoka. Zvok se na optično različnih sredstvih različno odbije. Delež odboja je tem večji, čim večja je razlika hitrosti širjenja valovanja. Največji del zvoka se odbije, če je material, od katerega se odbija, poponoma trden in pot, po kateri zvok potuje, točno definirana (čim ožja). V steklenički je prostor dobro zaprt, zato mu ta lastnost ni predstavljala težav. Problem je nastal pri valovanju vodne gladine.

Razdalja se je iz trenutka v trenutek spreminjala, zato so se tudi odčitki iz sekunde v sekundo razlikovali. Pri fazi testiranja se je to lepo opazilo, saj so centimetri skakali tudi za $+/- 2$ cm. Pri meritvi s pretvorbo centimetrov v mililitre in odštevanjem ml od skupne količine vode se je vedno upoštevala prva sprememba, kar pomeni, če je senzor zaznal veliko razdaljo, se je količina popite vode kar naenkrat zmanjšala za nekaj mililitrov, čeprav vode sploh nihče ni popil. Za omenjeno težavo sem našel dve potencialni rešitvi: zaradi sinusnega valovanja, lahko vsakič vzamemo srednjo vrednost med prvo in drugo meritvijo, kar ponavljamo toliko časa, dokler ne dobimo primernega približka tekočine. Problem take rešitve je, če zgolj odpremo pokrovček in ne pijemo ampak le pretresemo stekleničko, bo senzor zaznal srednjo vrednost valovanja in odštel količino vode.

Druga rešitev je bila dodatni senzor, ki bi zaznal premikanje in takrat ne bi izvajal merjenja. S tako rešitvijo se izognemo nepričakovanim padcem količine vode in z enostavno implementacijo pridobimo najboljše rezultate. Tudi ta rešitev je zgolj do neke mere pravilna, saj se ob primerih, ko pijemo in hodimo, nič ne odbije od skupne vrednosti tekočine, kar lahko prinese prazno stekleničko in veliko količine še ne popite tekočine na prikazovalniku. Pri izvedbi sem ocenil, da je primernejša druga rešitev in tako senzorju razdalje dodal senzor pospeška.

4.1.1 Strojni del - PROTOTIP 1

Steklenička omogoča preprosto brezdotično merjenje nivoja vode, zaradi česar je omejena rešitev zanimiva za uporabnika.

Steklenička je sestavljena iz pokrova (vrhnji del) ter zalogovnika za vodo.

Vrhnji del sestavljajo:

- mikroprocesor
Zajema in obdeluje vse vhodne podatke in na njih izvaja program. Na podlagi dobljenih podatkov, opozarja uporabnika na pitje preko zaslona in LED lučke.
- OLED zaslon
Prikazuje podatke o dnevnem cilju zaužite tekočine ter trenutno stanje.
- ultrasonični senzor razdalje
Na podlagi razdalje med senzorjem in vodno gladino, računa količino vode v steklenički.
- pospeškometer
Zaradi premikov na vodni gladini, senzor razdalje ne meri optimalno, iz tega razloga pospeškometer preverja lego stekleničke ter hitrost premikanja.
- LED lučka
Služi kot opozorilo uporabniku o kritično nizki količini zavžite tekočine.
- baterija
Napaja celotno elektroniko v steklenički-

Vrhnji del je zaradi lažje izvedbe narejen iz lesa. V spodnjem delu je navoj iz plastike, ki povezuje zalogovnik vode iz stekla ter ustnik.

4.1.2 Programski del - PROTOTIP 1

Programska koda je temelj našega prototipa. V prvi rešitvi je bil največji izziv predvideti merjenje le takrat, kadar steklenička miruje. V glavni zanki v intervalih preverjamo, ali se steklenička premika, ko prejmemo odziv TRUE ali FALSE, se odločimo, kako naprej.

Merjenje vršimo samo v primeru, da je steklenička na mestu, takrat izvedemo meritev in preračunamo milimetre v mililitre. Po računanju opravimo še spremembo spremenljivke, ki predstavlja popito tekočino na zadnjo vrednost.

Mikroprocesor se po vnaprej določenem intervalu neaktivnosti ugasne oz. preide v stanje mirovanja. Tako imenovano mirovanje imenujemo Standby, saj se v primeru prekinitve (angl. *interrupt*) mikroprocesor zbudi in ponovno vrne v zanko.

Stanje mirovanja je podrobneje opisano v nadaljevanju (poglavje 4.2.4) Popito tekočino in dnevni cilj izpisujemo na prikazovalniku. Prikazovalnik je popolnoma prilagodljiv, kar omogoča, da lahko zadovoljimo potrebo vsakega posameznika. Po potrebi lahko spreminjamo velikost pisave in stil črk. V kolikor je steklenička namenjena mlajši osebi, lahko uporabimo živahne barve z več grafične podobe, za slabovidne pa enostaven prikaz z velikimi črkami.

4.1.3 Izpisovanje na zaslon

Na področju OLED zaslonov najdemo veliko različnih modulov, ki se razlikujejo po vodilu in po vrsti krmilnika. Nekateri uporabljajo SPI, drugi pa I2C vodilo za priklop modula na mikroprocesor.

V primerjavi enega z drugim lahko opazimo, da je krmiljenje preko SPI vodila veliko hitrejšo, hkrati pa tudi bolj kompleksno, saj za svoje delovanje potrebuje več povezav. V našem primeru smo z izhodi na mikrokrmilniku zelo omejeni, saj uporabljamo Arduino nano verzijo. To je bila odločilna lastnost za izbiro oled zaslona s krmilnikom SH1106, povezanega preko I2C vodila. Modul deluje od 2 V do 3,6 V, zato ga vezemo na 3,3 V vhod. Za uporabo OLED zaslona ga je potrebno najprej povezati in definirati. Modul povežemo na ploščico s 4 priključki. Dva sta za napajanje ter SDA in SCL, ki prenašata podatke. Modul potrebuje za svoje delovanje celo kopico registrov, vsak opravlja določeno funkcionalnost - npr. prižig in ugašanje prikazovalnika, spreminjanje barv ipd. Vsak register ima svoj naslov v heksadecimalnem številu. Za vsako spremembo na prikazovalniku potrebujemo 3 parametre:

- Naslov naprave (npr 0x80),
- naslov registra (npr 0xAF),
- ukaz (npr `wire.endTransmission` - ugasne zaslon).

Najpreprostejša oblika zapisovanja na zaslon je definiranje pikslov.

Zaslon je razdeljen na 8 strani in 128 stolpcev. Vsaka stran vsebuje 8 pikslov, vsako stran določamo z enim heksadecimalnim številom - (število z osmimi številkami, vsak izmed osmih bitov predstavlja en piksel).

Primer: V kolikor bi želeli izpisati piko v zgornjem levem kotu prikazovalnika, bi napisali heksadecimalno število 0x01 (B0000 001) v podatkovni register.

V primeru, da bi bilo zapisanega zelo malo besedila/znakov in se le-to ne bi spreminjalo, je taka rešitev primerna, drugače ne.

Drugi način delovanja omogočata knjižnici:

-AdafruitGFX.h,
-gfxfont.h.

S pomočjo zgoraj omenjenih knjižic, ponudnika Adafruit lahko enostavno zapišemo kar želimo izpisati na zaslonu in to je tudi prikazano.

Algoritem 2: Lcd prikazovalnik

```
1 display.clearDisplay(); display.setTextSize(2); //določimo velikost pisave
  display.setTextColor(WHITE);
  display.setCursor(0, 0); //tukaj določimo začetno pozicijo prikazovalnika
  display.println("Popil si:"); //enostavno vpišemo željeno besedilo
  display.setTextColor(BLACK, WHITE); 'inverted' text
  display.setTextSize(2);
  display.setTextColor(WHITE);
  unsigned int frac; display.print("ml "); display.print(vsotaMililitrov); //tukaj
  izpisujemo tekočino, ki se je pretakala skozi senzor od vklopa display.print();
  display.display(); delay(1000); display.clearDisplay();
```

Vodilo I2C

Vodilo I2C (angl. *Inter Integrated Circuit*) je razvil Philips leta 1982. Vodilo je bilo sprva mišljeno kot povezava televizijskega procesorja z ostalo elektroniko. Kasneje se je zaradi uspešnega delovanja vodilo pričelo uporabljati tudi v drugih primerih pošiljanja podatkov, kjer imamo omejitve izhodnih/vhodnih priključkov. Protokol I2C je namenjen povezavi več naprav z uporabo samo dveh povezav. Prva se imenuje SDA in prenaša podatke, druga pa skrbi za urin takt (sinhronizacijo in hitrost komunikacije) imenujemo jo SCL.

Vsaka naprava ima svoj naslov v heksadecimalnem zapisu (med 0x00 in 0xFF), preko katerega jo Arduino nagovori, naprava pa odgovarja. Komunikacijo med napravama vedno prične nadrejeni. Takemu delovanju rečemo startni pogoj. SDA nivo potegne na nizko stanje, ko je SCL nivo visoko. Pošiljanje se vrši z 8 biti, prvih sedem je naslov naprave, 8 bit pa pove, ali želimo pisati ali brati. Če je na liniji več naprav, bodo vse, razen naslovljene, ignorirale sporočilo.

I2C vodilo deluje preko odprtega kolektorja. To pomeni da naprava, ki v nekem trenutku uporablja linijo, vklaplja ali izklaplja stikalo oz. tranzistor.

4.1.4 Merjenje pospeška in orientacije

Pospešek

Kot že omenjeno v opisu naprave, uporabljamo pospeškometer, ki zaznava pospeške na treh oseh - X, Y, Z. Princip delovanja pospeškometra ADXL335 je kapacitivni, kar pomeni, da pospešek meri s pomočjo kondenzatorjev.

S pomočjo frekvenčne modulacije lahko dobimo pospešek. Vhodna napetost vpliva na občutljivost pospeškov. Problem kapacitivnih senzorjev je ta, da delujejo nelinearno. Pred pričetkom testiranja sem pospeškometer testiral v trivialnem programu. Po uspešni inicializaciji imamo 3 spremenljivke X, Y in Z os. Vsaka spremenljivka odčitava vrednosti iz analognih vhodov Arduina. V primeru, ko je steklenička postavljena vodoravno, se vrednost spremenljivke Z bliža enici, ostali dve vrednosti sta v bližini ničle. Tako lahko ob odčitavanju vrednosti vedno vemo, v kateri poziciji je steklenička. Ob tresljajih in nenadnemu spreminjanju smeri, so odčitane vrednosti zelo visoke.

Po končanem testiranju sem potreboval še funkcijo merjenja orientacije stekleničke, s katero bi lahko zaznaval kot in ga pred vsako meritvijo preveril v *if* stavku.

Orientacija

Zaradi omejenosti senzorja razdalje, kateri bi ob primerih, ko je vsa voda v levi ali desni stranici stekleničke, podajal napačno vrednost, moramo preverjati, ali je steklenička pokonci pred pričetkom izvajanja meritev. Pred ugotavljanjem orientacije potrebujemo začetno vrednost. To je vrednost, pri kateri je steklenička vodoravno v mirovanju.

Pospešek izračunamo s pomočjo izhodne analogne napetosti. Ko je pospeškometer v mirovanju, je statična izhodna napetost v mojem primeru 1,72 V.

Kot, v katerem je steklenička, izračunamo tako, da pospešek pomnožimo s funkcijo arkus sinus. Kot, ki ga dobimo, je v radianih, zato ga je potrebno pretvoriti v stopinje. Pretvarjanje se izvaja z enostavno predpostavko, da je 1 radian enak 57,2957795 stopinj. Za preračunavanje uporabljamo spodnjo metodo. Kadar je steklenička znotraj kota 85 - 95 °, se merjenje izvaja, drugače ne.

Algoritem 3: Formula pretvarjanja

$$1 \text{ deg} = \text{rad} * 5729578 / 1000$$

4.1.5 Delovanje senzorja razdalje

Senzor razdalje, imenovan tudi ultrasonični senzor, je na Arduina povezan s štirimi povezavami. Dve sta napajanje (5V ter GND), ostali dve uporabljamo kot izhod in vhod podatkov. Podatkovni povezavi sta priključeni vsaka na svoj digitalni vhod na Arduinu.

Delovanje senzorja je podobno radarju, ki ga najdemo na plovilih (v zelo poenostavljeni obliki). Senzor odda kratek zvočni signal (pulz), ki v našem primeru traja signal 10 mikrosekund. Vsak pulz je sestavljen iz osmih ciklov po 40 khz. Senzor določa razdaljo na podlagi odmeva zvočnega signala. Ko Arduino preko Trigger priključka odda 5 V, se sproži zvočni signal in hkrati se prične štetje časa. Oddani signal se nato od najbližje površine odbije nazaj v senzor, ta ga zazna in preko Echo priključka opozori Arduina. Po prejemu signala izračunamo čas, kateri je pretekel od oddanega do prejetega signala. Oddajanje zvočnega signala je prikazano na Sliki 7.



Vir: (<https://mcuoneclipse.files.wordpress.com/2012/12/hc-sr04-timing-diagram.png>), Avtor: Erich Styger, (09.04.2016)

Slika 7: Diagram senzorja razdalje

Pretvorba časa v razdaljo

Kot že ime pove, uporabljamo senzor razdalje za merjenje razdalje in ne časa. Pred dokončnim rezultatom je potrebno pretečen čas z enostavno enačbo spremeniti v razdaljo. Zvok potuje s hitrostjo 340,29 metrov na sekundo. Razdalja je hitrost, pomnožena s pretečenim časom, deljena z dva, saj je dobljeni čas rezultat poti do objekta in nazaj, zanima nas pa samo pot do objekta.

Formula katero sem uporabil za merjenje razdalje:

Algoritem 4: Formula razdalja

1 razdalja = čas * hitrost (340) / 2;

2 Centimetri = uS / 58

4.2 PROTOTIP 2

Izvedba drugega prototipa, je bila manj kompleksna, zaradi uporabe bolj primere "periferije", namenjene merjenju tekočine. Za izdelavo mi je bil v veliko pomoč prvi prototip, saj sem lahko velik del prototipa samo prilagodil novemu senzorju. Naj omenim da deluje prebujanje stanja mirovanja popolnoma enako kot pri prototipu 1. Prvi problem s katerim sem se soočil je bil kalibriranje naprave. Senzor je na prvih testiranjih odčitaval vrednost, ki ni bila niti približno točna. Druga in največja težava je bila sama izvedba stekleničke. Za razliko od prototipa 1 kjer lahko na vsako stekleničko namestimo senzor razdalje, je bil pri prototipu 2 zaradi oblike senzorja problem najti stekleničko z ustnikom dovolj velikim za senzor.

Voda potuje preko turbine zaradi česar moramo tok usmeriti po kanalu. Najpreprostejša izvedba je bila s slamico, ki povezuje vodo in senzor.

4.2.1 Strojni del - PROTOTIP 2

Po obsežnem testiranju plovčkov, merilcev razdalj, teže in podobno, sem naletel na senzor pretoka. Podobne a večje senzore se uporablja v akvarijih in vodovodih. Senzor pretoka vsebuje elektromagnetno tipalo s katerim zaznava rotacijo turbine v senzorju. Na podlagi znanega volumna senzorja lahko izmerimo pretok v eni rotaciji turbine. Senzor pretoka je nameščen v zgornjem delu stekleničke in tako omogoča preprosto čiščenje spodnjega dela (del kjer se zadržuje voda).

Kompaktnost stekleničke nam ne omogoča veliko prostora, zato moramo pametno izbrati vsako komponento. Napajanje se vrši preko integrirane baterije, za čimmanjšo porabo uporabljamo funkcijo mirovanja. Steklenička miruje, z izjemo takrat kadar pijemo ali preverjamo količino popite tekočine. Meritev se prične, kadar odpremo varovalo na zgornjem delu stekleničke. Pod varovalom je gumb, kateri sproži zagon senzorja. V središču stekleničke najdemo slamico, pritrjeno na ustnik, po kateri pijemo tekočino v steklenički.

Poleg senzorja pretoka najdemo v steklenički tudi led diodo, ki oznanja prenizko raven popite tekočine ter led prikazovalnik, na katerem lahko odčitamo tako trenutno popito tekočino kot tudi dnevni cilj tekočine. Vsa "periferija" je priklopljena na osrednji del naprave, torej na mikroprocesor Arduino NANO.

Izdelava stekleničke

Po končanem testiranju senzorja je bilo potrebno vso elektroniko namestiti na delujočo stekleničko. Za zmanjšanje stroškov sem prototip izdelal iz obstoječe stekleničke za vodo. Moj načrt je bil prirediti zgornji del stekleničke (od navoja do ustnika), tako, da bi ustrezal standardnim zalogovnikom vode.

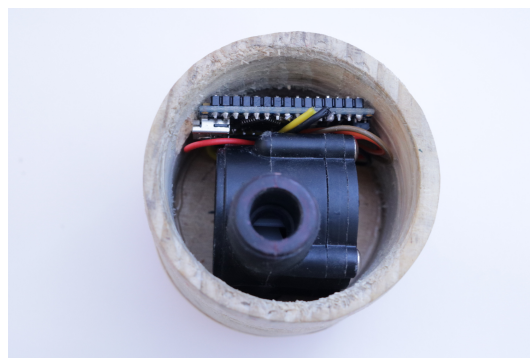
Za izdelavo sem uporabil les, ki predstavlja najlažjo surovino za ročno izdelavo in testiranje. Iz lesene grede sem s pomočjo vrtalnih kron izvrtal dve različno veliki luknji. Premer krone sem prilagodil širini plastičnega navoja iz stekleničke. Po vrtanju sem leseni del še obrusil in kasneje tudi pobarval, tako sem ga zaščitil pred vlago in poenostavil čiščenje.

Na zgornji del sem prilepil okrogel kos lesa v debelini 0,5 mm in vanj izvrtal luknjo premera velikosti senzorja. Na vrhnji del sem nato izrezal tudi odprtino za prikazovalnik ter USB vhod.

Največ težav mi je povzročala voda, zelo trivialno izgleda narediti stekleničko neprepustno vendar je stvar v praksi zelo kompleksna. Steklenička tesni tako, da jo navoj potiska v vrh pokrovčka kateri ima ob robovih gumo. V originalni steklenički je bila guma del ustnika vendar sem v svoji izvedbi ustnik namestil na senzor zaradi česar je bila potrebna ponovna izdelava gumijaste zaščite. Voda vstopa v senzor preko po slamici, preko vodne zaščite v zgornjem - elektronskem delu stekleničke in izhaja iz senzorja v ustnik. Druga težava, ki se mi je porodila ob izdelavi prototipa je bila iskanje pravičnega navoja. Senzor je namenjen kitajskemu trgu, kjer uporabljajo za vrtno cevi drugačen premer navoja kakor v Sloveniji. Težava je bila naredi ustrezen navoj, ki bi preprečeval vstop tekočine v del stekleničke kjer se nahaja elektronika. Ker navoja ni bilo mogoče najti sem ga bil primoran izdelati sam.



(a) Leseni del z elektroniko



(b) Elektronika stekleničke

Slika 8: Izdelava stekleničke

4.2.2 Programski del - PROTOTIP 2

Programska koda je v veliki večini podobna kodi iz 1. rešitve. Spremembe so v definiciji senzorja ter v for loop zanki. Steklenička je večino časa v mirovanju STANDBY, iz katerega pride ob prekinitvi.

Prekinitve se prožijo na tri načine (delovanje in opis vseh treh prekinitvev je v poglavju Stanje mirovanja).

Pred vsakim pitjem se najprej izvede *if* stavek, kateri preverja popito tekočino in potrebo po opozorilu, nakar se izvede meritev pretoka in popravek stare vrednosti popite tekočine z novo.

4.2.3 Merjenje pretoka

Poglavitni del stekleničke za vodo je seveda naprava, ki meri količino popite vode. V drugem prototipu sem poskusil za to namenjeni senzor, imenovan senzor pretoka. Senzor je z Arduinoom povezan preko treh povezav, dve sta napajanje (5 V - VCC vhodna napetost ter GND), tretja povezava pa prenaša signal/impulz iz senzorja. Signal znamo na digitalnem vhodu 2 kot zunanjo prekinitvev. Vsaka prekinitvev se vrši preko funkcije `stevec()`, ki sešteva pulze.

V povprečju se za vsak pretečeni liter tekočine, vsako minuto vrši med 4 in 5 signalov/pulzov. V spodnji formuli je prikazano merjenje pretoka v programu.

Algoritem 5: Merjenje pretoka

```
1 pretok = ((5700 / (ms() - starCas)) * stevec) / faktorKalibracije;
   starCas = ms();
   pretoceniMl = (pretok / 60) * 1000;
   vsotaMl += pretoceniMl;
```

4.2.4 Stanje mirovanja

Program se izvaja v mikroprocesorju v zanki, kar pomeni, da imamo konstantno porabo električne energije, čeprav je včasih ne potrebujemo. Steklenička ima zelo omejeno velikost, kar prinaša majhno baterijo in s tem zelo omejen čas delovanja. Za podaljšanje avtonomije sem uporabil možnost mirovanja.

Za izhod iz stanja mirovanja poznamo več različnih načinov, lahko uporabimo zunanjo prekinitvev, prekinitvev s časovnikom (angl. *timer*), prekinitvev s psom čuvajem (angl. *watch dog timer*) ali spremembo na vhodih/izhodih naprave.

Prekinitve se v našem primeru prožijo na tri načine. Prva se zgodi, kadar stekleničko prižgemo na glavnem stikalu - v prototipu je to, kadar steklenička prejme zunanje napajanje. Druga možnost prekinitve je sprememba na vhodu/izhodu naprava, lahko se zgodi ob sprožitvi stikala na pokrovčku stekleničke ali ko je pokrovček ostal odprt in pričnemo s pitjem tekočine ter takrat sprožimo prekinitvev zaradi spremembe v senzorju pretoka.

Tretji in tudi najpogostejši način prekinitve je časovnik. Na interval se steklenička prižge ter preveri popito tekočino. V kolikor je popita tekočina manjša od pričakovane, se prižge opozorilna lučka.

Algoritem 6: Pseudokoda zanka mirovanje

```
1 int zbudisePin = 2; // izhod na arduinu kjer prožimo prekinitvev preko stikala na
   pokrovčku
   int tatusPin= 0; // tukaj shranjujemo zahtevo za spanje
   int stevec = 0; // števec
```

```
   PreveriStanje();
   delay(30000) // 30 sekund po meritvi počakamo
   statusSpanje=digitalRead(statusPin); // preverimo stanje na vhodu
   if(statusSpanje ==LOW) // napravo pošljemo v mirovanje
   pojdiSpat();
```

Poznamo 5 različnih vrst mirovanja, ločijo se po svoji varčnosti in pa po načinu prekinitiv, katere stekleničko zbudijo.

- SLEEP-MODE-IDLE(najmanj varčen)
- SLEEP-MODE-ADC (zmanjšanje zvoka)
- SLEEP-MODE-PWR-SAVE
- SLEEP-MODE-STANDBY
- SLEEP-MODE-PWR-DOWN (steklenička preneha z vsemi procesi- najbolj varčen način)

Algoritem 7: Funkcija mirovanja

```
1 void pojdiSpat() // funkcija pošlje stekleničko v stanje mirovanja
  set-sleep-mode(SLEEPMODESTANDBY); // tukaj določimo željeni način
  mirovanja, v našem primeru je standby
  sleep-enable(); // Tukaj dostopamo do registra MCUCR kjer omogočimo bit
  spanja, brez tega koraka spanje ne bi bilo mogoče, bit spanja je neke vrste
  varovalo
```

```
prekinitiv(0,wakeUpNow, LOW); // tukaj določimo izhod na katerem bomo
prožili prekinitiv, to pomeni da moramo fizično priključiti stikalo ali senzor na
določen pin (npr. pin 2) v našem primeru se steklenička zbudi ko je vrednost na
vhodu 2 enaka LOW to velja za vse oblike spanja razen IDLE
```

```
sleep-mode(); // tukaj gre program v mirovanje, po prekinitvi se program začne
od tukaj naprej
sleep-disable(); // ko zbudimo stekleničko iz spanja onemogočimo spanje
detachInterrupt(0); // prekinitiv na pin 2 lahko sedaj odstranimo, da se ne bo
klicala metoda Zbudise tekom normalnega izvajanja programa
```

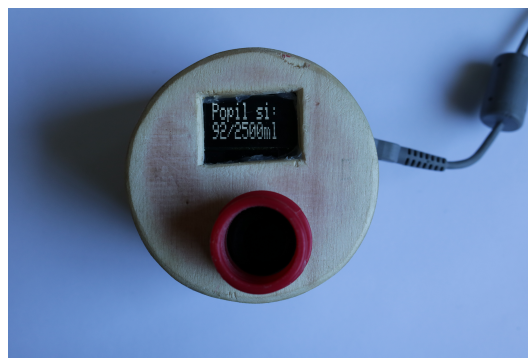
5 Uporaba in delovanje

Pametna steklenička za vodo je bila razvita kot nadgradnja vsakdanjih stekleničk za vodo. Moj prvotni cilj je bil poenostaviti način hidracije in opominjanja na zdravo življenje. Steklenička, ki sem si jo zamislil, je enostavna za uporabo in tako primerna tudi za osebe z manj tehničnega znanja.

Ob zagonu aplikacije se na zaslonu izpiše naš dnevni cilj tekočine. Tekočino je potrebno najprej nastaviti znotraj samega programa, saj aplikacija zaenkrat nima menija za nastavljanje količine. (izboljšava bi bila ob zagonu določiti dnevni cilj tekočine). Po vsaki uporabi stekleničke se na zaslonu prikaže dnevni cilj ter trenutno popita tekočina. Steklenička po minuti neuporabe žaspi in tako varčuje z energijo. V kolikor stekleničke ne uporabimo 2 uri, steklenička preveri, ali je bilo dovolj popite tekočine (3 dnevni cilji). V kolikor ne, nas s svetlobo opozori na pitje tekočine. Steklenička v testni izvedbi deluje zgolj z zunanjim napajanjem, v prihodnje pa bo baterija integrirana.



(a) Steklenička - delovanje



(b) Prikazovanik

Slika 9: Prikaz delovanja

5.1 Rezultati

Včasih so najpreprostejše rešitve, najboljše rešitve. Po testiranjih sem naletel na obilico problemov, s katerimi sem se moral soočiti med raziskavo. Pri prvi rešitvi je bil glavni problem nihanje vodne gladine pri premikanju. Moj cilj je bil izdelati stekleničko, ki jo je možno uporabiti ob vsakem trenutku. Slednje ni možno, če senzorju vsakič med premikanjem stekleničke vrednost popite tekočine eksponentno naraste. Prva rešitev, katere sem se domislil, je bila programskega tipa. Zamislil sem si, da bi lahko valovanje predvidel in zaradi sinusnega premikanja vedno vzel neko idealno srednjo vrednost. Teoretično bi taka rešitev pripomogla k zmanjšanju ekstremnih vrednosti meritev, vendar je slaba stran to, da meritve ne bi bile popolnoma natančne, temveč nek približek rezultata. Uporabnik bi tako po nekajminutni športni aktivnosti s stekleničko v roki, odčital popito tekočino, ne da bi sploh odprl pokrovček stekleničke. Takim primerov sem se želel izogniti. Po temeljitem razmisleku sem prišel do druge potencialne rešitve problema s senzorjem premika (angl. *gravity sensor*). Težavi sem se izognil tako, da sem pred merjenjem razdalje preveril, ali se steklenička premika, v primeru, da je G-senzor odgovoril pravilno, sem merjenje preskočil, v primeru, da je odgovoril napačno, pa sprožil meritev razdalje in popravek zadnje spremenljivke.

Meritve so tako potekale nemoteno, vendar so senzorji po kratkem času prenehali delovati. Ugotovil sem, da je v senzorjih, zaradi podvrženosti vlage prišlo do kratkega stika. Po večkratni menjavi senzorjev sem poskušal zaščititi senzor z vodoodbojnim materialom vendar neuspešno. Zaradi zaščite senzor ni več dobival povratnega signala in natančnost je bila zelo slaba.

Možna rešitev bi bila uporaba vodoodpornega senzorja, vendar so slednji cenovno v višjem rangu.

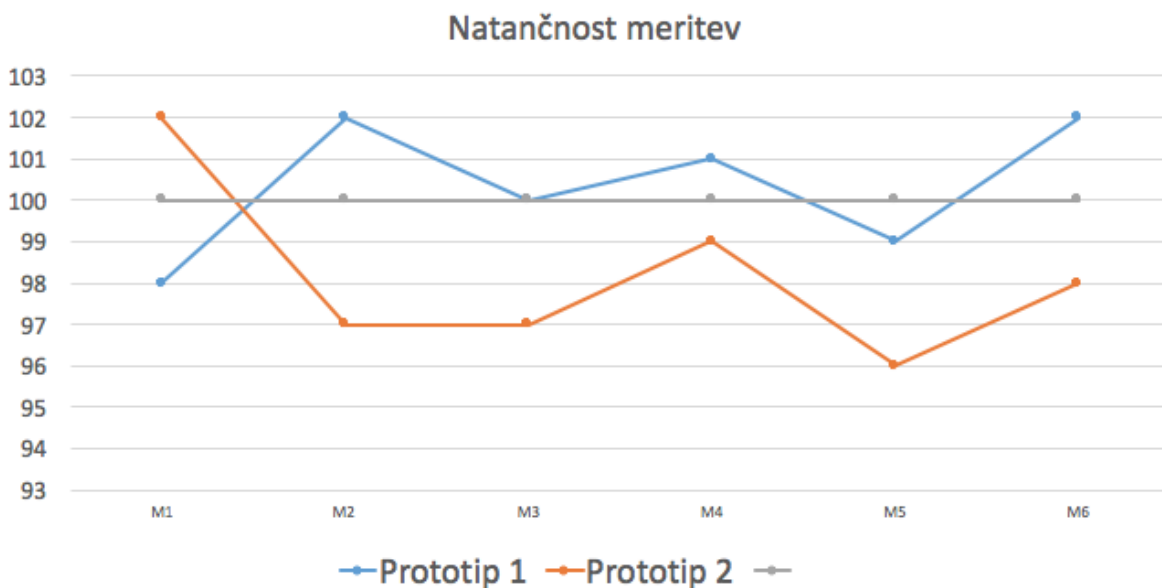
Veliko manj težav sem zaznal pri drugi rešitvi merjenja količine pretoka. Senzor je po specifikacijah proizvajalca natančen na ± 5 ml. To je pri meritvah velikih količin sprejemljiva natančnost za cenovno ugoden senzor. V mojem primeru je 5 ml kar velika razlika, zaradi česar sem senzor nekoliko prilagodil in tako izboljšal natančnost. V notranjosti senzorja je turbina na centralni osi, katera z rotacijo meri količino. Slabost takega načina je, da se ob udarcih turbina zarotira in tako izmeri pretok. Preprosta rešitev problema je bila podlaganje turbine z vsake strani, tako da ima manj manevrskega prostora pri rotaciji.

Natančnost meritev sem testiral z enostavnim poskusom. V merico sem natočil natančno 100 ml in tekočino izmeril s pomočjo prototipov. Meritev sem opravil šestkrat in se tako prepričal v odstopanje meritev.

S Tabele 2 lahko odčitamo, da je prototip1 vračal natančnejše in tudi bolj konsistentne meritve. Prototip 2 je bil na trenutke zelo natančen, v drugih trenutkih pa je natančnost odstopala celo bolj, kot je navedeno v specifikacijah proizvajalca. Prototip 1 je odstopal od 1 do 3 procente, prototip 2 pa od 1 do 4 procente.

Tabela 2: Meritve

Točnost meritev	M1	M2	M3	M4	M5	M6
Tekočina	100	100	100	100	100	100
Prototip1	98	102	100	101	99	102
Prototip2	102	97	97	99	96	98
Odstopanje (ml)P1	2	-2	0	-1	1	-2
Odstopanje (ml)P2	-2	3	3	1	4	2



Slika 10: Rezultati meritev

5.1.1 Odločitev

Pri sami raziskavi sem iskal najboljšo rešitev, ki bi pokrila več kriterijev izbire. Kot že napisano, sem iskal cenovno ugodno rešitev, ki bi bila enostavna za implementacijo in dovolj natančna pri meritvah, da bi lahko uporabnik spoznal dejansko stanje popite tekočine. Cenovno ugodne rešitve v večini primerov niso najbolj idealne in tudi moja rešitev je daleč od idealnega. Slabosti najdemo tako rekoč pri vsaki od rešitev.

Če se osredotočim na moji dve končni rešitvi, ima prva potencial v tem, da nima premičnih delov. Vse meritve so opravljene, ne da bi se dotikali tekočine, kar prinaša manj težav za čiščenje in enostavno uporabo. Slaba stran take rešitve so problemi v natančnosti in cena.

Druga rešitev je bila veliko lažja za implementacijo, cenovno ugodna in hkrati natančna. Problem rešitve so premični deli v notranjosti senzorja, v katerih se lahko po določenem času nabira nesnaga. Slaba stran je tudi to, da mora voda potovati preko turbine, zaradi česar moramo vodo piti po slamici. Zavedam se, da ni nobena rešitev idealna, vsakič pa lahko najdemo dobre in slabe strani. V mojem primeru sem prototip izvedel z drugo rešitvijo, saj sem s senzorjem pretoka imel manj težav pri implementaciji in dosegal boljše rezultate pri meritvah.



Slika 11: Steklenička

5.1.2 Možne izboljšave

Idealna rešitev je nekaj, k čemur stermi vsak izumitelj ali inovator. Ne glede na to, v kateri fazi je izdelek, so vedno možne dodelave in izboljšave produkta. V mojem primeru je steklenička šele v fazi prototipa. Steklenička meri pretok in opozarja glede premalo popite tekočine. OLED ekran prikazuje osnovne vrednosti v številkah. Prva možna izboljšava bi bila nadgradnja vmesnika tako, da bi bil uporabniku prijaznejši in bolj nazoren. Namesto številke bi lahko upodobili grafično vrednost. Steklenički bi lahko poleg merjenja popite količine lahko dodali še dodatne funkcionalnosti (na primer merjenje temperature tekočine z vgrajenim senzorjem temperature, opozarjanje na kontaminirano vodo in drugo).

Napajanje, ki se vrši preko zunanje baterije, bi lahko zamenjali z integrirano polnilno baterijo, ki bi se polnila brezžično, preko namenskega stojala za stekleničko. Stojalo bi bilo tudi povezava v oblak za vsakodnevno shranjevanje podatkov o hidraciji ter vmesnik za spreminjanje nastavitve vrednosti za vsakega posameznika.

Tabela 3: Kriteriji za izbor

Kriteriji za izbor		
Kriterij	PROTOTIP1	PROTOTIP 2
TOČNOST MERITEV	Točnost meritev je zelo natančna v primeru, da je vodna gladina nepremična.	Senzor je narejen za večje količine tekočine, zato je natančnost manj točna na majhne količine. Odstopanje je 3 procentno.
TEŽAVE PRI IZDELAVI	Največja ovira je bil stik s tekočino in meritev tekočine v gibanju.	Senzor je velik, kar otežuje implementacijo. Težava nastane pri čiščenju senzorja, saj gre za mehanično merjenje, kjer se lahko turbina zamaže.
POZITIVNE LASTNOSTI	Senzor nima stika s tekočino, čiščenje je enostavno.	Senzor je vodoodporen, enostavnoprogramska rešitev za natančne rezultate.
TEŽAVNOST IMPLEMENTACIJE	Implementacija je zelo težavna zaradi stika senzorja s tekočino. V kolikor ga zaščitimo, omejimo točnost podatkov.	Implementacija je težavna zaradi tekočine, vendar veliko bolj učinkovita kot prva rešitev.
UČINKOVITOST	Senzor deluje natančno, težavo glede premikanja tekočine sem premostil, vendar senzor ni namenjen uporabi z vodo, zaradi česar je imeplementacija težavna in posledično draga izvedba.	Prototip deluje po zadanih kriterijih, natančnost je zadovoljiva.

6 Zaključek

Ideja o pametni steklenički za vodo se mi je porodila prav na fakulteti, kjer smo v poletnih dneh pisali še zadnje izpite. Začelo se je samo z idejo, nato programiranjem in na koncu izdelanim delujočim prototipom.

Sama izvedba je bila zame izziv, saj je bilo potrebno izdelati rešitev od nadzora senzorjev do opozarjanja in meritev znotraj krmilnika. Programiranje v jeziku C, v katerem deluje Arduino, je bilo zame neznanka, poznal sem enostavne ukaze, kako prižgati in ugasniti lučko, kar je bilo tudi vse. Z orodjem sem spreten, zato mi je bila poleg programiranja zanimiva tudi sama izvedba stekleničke. Moram priznati, da sem imel več težav s fizično izdelavo stekleničke kot programiranjem. Veliko vlogo tukaj igra tudi potrebno orodje, katerega nisem imel na dosegu roke. Najverjetneje bi bila steklenička z uporabo 3D tiskalnika in stružnice za les, veliko bolj očem prijazen izdelek in nena zadnje tudi bolje delujoč.

Z obzirom na to, da je bila že od začetka načrtovana zgolj izdelava prototipa, sem zelo zadovoljen s končnim izdelkom.

V zaključni nalogi sem se osredotočil na idejo in izvedbo rešitve. Programski del sem poskušal razložiti tako, da predstavlja besedilo enostavno branje tudi tistim, ki nimajo programerskega znanja. V prvem delu naloge sem opisal problem in ozadje Arduina ter specifikacije senzorjev, katere sem uporabil v steklenički. Rešitev sem razdelil na dve rešitvi, kateri sem med raziskavo testiral in ugotavljal, katera bi bila lažja za izvedbo. Po lastni presoji na podlagi rezultatov in mnenj o težavnosti izvedbe sem se v drugem delu naloge osredotočil na rešitev Prototip 2. Izbrano rešitev sem sprogramiral do konca, optimiziral, izdelal in ne nazadnje tudi uporabljal.

Skozi nalogo sem se srečeval z navidezno banalnimi problemi, na katere verjetno marsikdo niti ne pomisli, ko uporablja stekleničko za vodo. Kot sem že omenil, je bilo težav več pri fizični izvedbi stekleničke kakor pri programerskem delu. Sproti sem z reševanjem težav spoznaval tudi samo delovanje Arduina in njegovih modulov.

Pri reševanju problemov sem prejel veliko uporabnih nasvetov s strani mentorja in tudi prijateljev, ki se ne ukvarjajo s programiranjem. Včasih logično razmišljanje popelje do ideje, ki se nato v programerskem svetu enostavno izvede v rešitev. Ko se ozrem nazaj in pogledam narejeno delo, vidim, da je možnosti za izboljšave in optimizacijo precej.

Kot sem že omenil, sem Arduina šele dobro spoznal pri izdelavi stekleničke, zato bi verjetno programer z več izkušnjami doprinesel k boljšemu hitrejšemu delovanju.

Menim, da se bodo prave pomanjkljivosti pokazale šele pri daljši uporabi prototipa in testiranju različnih ljudi z različnimi potrebami in navadami.

Glede same izvedbe sem pozitivno presenečen nad funkcionalnostjo in samim delovanjem s cenovno ugodno opremo. Če bi imel za projekt več sredstev, bi se lahko tudi funkcionalnost stekleničke precej povečala, vendar to ni bil moj namen. Cilj je bil izdelati stekleničko, ki deluje na preprost, vendar učinkovit način s tehnologijo, ki nam je blizu. Prototip stekleničke za vodo uspešno meri popito tekočino in te ob premalo popiti vodi tudi opozarja, kar je bil moj namen, ki sem ga tudi dosegel.

7 Literatura in viri

- [1] ALMOND CHRISTOPHER S.D., M.D., M.P.H., ANDREW Y. SHIN, *Hyponatremia among Runners in the Boston Marathon*, <http://www.nejm.org/doi/full/10.1056/NEJMoa043901t=articleTop>. (Datum ogleda: 11. 2. 2016.) (*Ni citirano.*)
- [2] ANDREJAŠIĆ MATEJ, *Seminar - MEMS ACCELEROMETERS*, <http://mafija.fmf.uni-lj.si/seminar/files/2007-2008/MEMS-accelerometers-končna>. (Datum ogleda: 1. 3. 2016.) (*Ni citirano.*)
- [3] KREKOČ PETER, Arduino : programirajmo Arduino z lahkoto. [*zbrani članki iz revije Svet elektronike*] 2014 () (*Ni citirano.*)
Feige
- [4] MLAKAR S. in GOLOB N., *Zdrav način življenja in voda : diplomsko delo*. Ljubljana, 2010. (*Ni citirano.*)
- [5] MIKELN J. , *Programirajmo mikrokontrolerje: priročnik za programiranje PIC mikrokontrolerjev v zbirniku in programskem jeziku C*, Ljubljana AX elektronika, 2004. (*Ni citirano.*)
- [6] MORGADO J., CHARAS A., ALCÁCER L., HENRIQUES R., MATOS M., BERNARDO G., *Organic Light Emitting Diodes (OLEDs) - Instituto de telecomunicacoes*, <http://www.lx.it.pt/alcacer/TM-Group/OLEDs.pdf>. (Datum ogleda: 24. 3. 2016.) (*Ni citirano.*)
- [7] NUSSEY JOHN in ARDUINO FOR DUMMIES, Založništvo John Wiley and Sons. 3 2013 (105–107, 143–204, 248– 260, 395–345) (*Ni citirano.*)
Nussey John
- [8] *Programiranje-I2C vodilo*, <http://www.svet-el.si/o-reviji/programiranje/2716-216-43>. (Datum ogleda: 24. 3. 2016.) (*Ni citirano.*)
ROTOVNIK-KOZJEK, Kaj naj pijem med maratonom? [članek -Polet : magazin Dela in Slovenskih novic]. *Rotovnik-Kozjek, Nada* - str. 24-25 (2005) .

- [9] *Sparkfun*, Tutorials - I2C.
<https://learn.sparkfun.com/tutorials/i2c>. (Datum ogleda: 25. 4. 2016.)
(Ni citirano.)
- [10] *Specifikacije Arduino Nano*, <https://www.arduino.cc/en/Main/ArduinoBoardNano>.
(Datum ogleda: 23. 3. 2016.) (Ni citirano.)
- [11] *Specifikacije pospeškometra ADXL335 - (datasheet)*,
<https://www.sparkfun.com/datasheets/Components/SMD/adxl335.pdf>.
(Datum ogleda: 20. 3. 2016.) (Ni citirano.)
- [12] *Specifikacije ultrasoničnega senzorja HC-SR04*,
<http://www.micropik.com/PDF/HCSR04.pdf>. (Datum ogleda: 24. 3. 2016.)
(Ni citirano.)
- [13] GUY W. in LECKY-THOMPSON, Just enough C/C++ programming . *Boston*.
Ma 10 (2008) 215–222; 250–300. (Ni citirano.)
- [14] [WILCHER DON] BERKELEY, CA NEW YORK , *Learn electronics with Arduino*
[Elektronski vir], 2012.124–162 (Ni citirano.)
- [15] *Slika 1: OLED prikazovalnik (Avtor neznan)*,
<http://img02.taobaocdn.com/imgextra/i2/773299280/T2W8WPXEdXXXXXXXXXX-!!773299280>
(Datum ogleda: 24. 3. 2016.) (Ni citirano.)
- [16] *Slika 2: Senzor pretoka (Avtor neznan)*, <https://www.ia.omron.com/support/faq/answer/in>
(Datum ogleda: 07. 4. 2016.) (Ni citirano.)
- [17] *Slika 3: Senzor razdalje (Avtor neznan)*,
<https://www.kjdelectronics.com/assets/images/HCSR04.jpg>. (Datum
ogleda: 07. 4. 2016.) (Ni citirano.)
- [18] *Slika 4: Pospeškometer (Avtor neznan)*,
<http://www.hobbytronics.co.uk/image/cache/data/ht/sensors/adxl335-breakout-500x>
(Datum ogleda: 07. 4. 2016.) (Ni citirano.)
- [19] *Slika 5: Diagram pospeškometer (Avtor: Winifred E. Torres)*,
<http://forums.pelicanparts.com/uploads16/3+axis+output+graph1265224544.jpg>.
(Datum ogleda: 07. 4. 2016.) (Ni citirano.)
- [20] *Slika 7: Diagram senzorja razdalje (Avtor: Erich Styger)*,
<https://www.ia.omron.com/support/faq/answer/include/faq01408/img>.
(Datum ogleda: 09. 4. 2016.) (Ni citirano.)

- [21] *Slika 6; 8; 9; 10; 11*, Avtor Tadej Dragutinović. (Datum ogleda: 01. 4. 2016.)
(*Ni citirano.*)

Priloge

A Programska koda - zgoščanka

- Izvorna programska koda Pametne stekleničke za vodo,
- knjižnica Adafruit-GFX-Library-master,
- knjižnica AdafruitSSD1306-master.