REVIJA ZA ELEKTRONIKO, AVTOMATIKO, RAČUNALNIŠTVO IN TELEKOMUNIKACIJE



Int P

SE

304







Raziskovalci uresničujejo organske dvojne tranzistorje



Merjenje zračne vlage in zračnega tlaka z ESP32

G Search

on the light

thing ON the lights

OK Google tu

.



Krmiljenje LED-ice z govorom

Brezplačna dostava pri naročilih nad 50 € ali 60 \$*

Digi-Key je prava izbira!





KO BOSTE PRIPRAVLJENI NA INOVACIJO OBIŠČITE DIGIKEY.SI.

*Pri vseh naročilih pod 50,00 € bodo zaračunani stroški pošiljanja v vrednosti 20,00 €. Pri vseh naročilih pod 60,00 USD bodo zaračunani stroški pošiljanja v vrednosti 30,00 USD. Vsa naročila so poslana prek UPS, Federal Express ali DHL in dostavljena v roku 2 do 4 dni (odvisno od končnega cilja). Brez stroškov obdelave. Vse cene so v evrih ali ameriških dolarjih. Digi-Key je pooblaščeni distributer za vse partnerske dobavitelje. Dnevno dodajamo nove izdelke. © 2022 Digi-Key Electronics, 701 Brooks Ave. South, Thief River Falls, MN 56701, ZDA

Supporting The Authorized Channel



KAZALO

UVODNIK 4 Življenje se ne bo ustavilo! Jure NOVICE 5 Polnjenje v prihodnost: Softbank vlaga v raziskave za razvoj litij-zrak baterij https://mainichi.jp 5 Raziskovalci uresničujejo vertikalne organsko prepustne dvojne tranzistorje za logična vezja https://techxplore.com 7 Nov emulator v vezju (ICE) https://www.microchip.com PREDSTAVLJAMO 8 Študentski kotiček-ADALM2000 Naredimo operacijski ojačevalnik iz diskretnih elementov https://svet-el.si 12 Hitra in učinkovita uvedba prilagodljivih polnilnih sistemov za EV Avtor: Rolf Horn https://www.digikey.com 17 Glasovno upravljanje, kako pametne naprave bolje slišijo in zvenijo Avtor: Anne Santhakumar, Jeff Hsieh in Sam Cheng https://www.rutronik.com 19 Razvijajte ločeno in integrirajte brezhibno z dvojedrnim krmilnikom Avtor: Harsha Jagadish https://www.microchip.com Mladi navdušenci nad robotiko so se zbrali 24 v Liubliani Avtor fotografij: Domen Ulbl https://www.um.si PROGRAMIRANJE 26 Merjenje zračne vlage in zračnega tlaka z ESP32 AVR, ESP, PIC in drugi, ko gre zares (7) Avtor: dr. Simon Vavpotič https://svet-el.si

- 33 Grafika v okenskih programih Avtor: Boštjan Šuhel https://svet-el.si
- 36 Krmiljenje LED-ice z govorom https://hackster.io
- 38 Arduino digitalni termometer z uporabo MAX30205, senzorja človeške telesne temperature https://circuitdigest.com
- 42 VoiceTurn glasovno vodene smerne luči za varnejšo vožnjo Avtor: Alvaro Gonzalez-Vila https://hackster.io
- 51 ATtiny I2C slave brez težav in odvečnih stroškov (2) Avtorja: Vladimir Mitrović in Robert Sedak https://svet-el.si

	<u>STIK</u>
58	Info in naročanje
	https://svet-el.si

OGLAŠEVALCI

ALKATRON	27	MICROCHIP	23
DIGI-KEY	02	STROMBOLI D.O.O.	41
ICM-IFAM	60	TZS	32

Raziskovalci uresničujejo vertikalne organsko prepustne dvojne tranzistorje za logična vezja

Integrirana vezja (IC), ki temeljijo na organskih tranzistorjih, imajo veliko dragocenih aplikacij, na primer pri izdelavi papirju podobnih

zaslonov ali drugih elektronskih komponent, ki se nahajajo na velikih površinah. V zadnjih nekaj desetletjih...

Stran: 5

Merjenje zračne vlage in zračnega tlaka z ESP32

V preteklem nadaljevanju nas je zanimalo, kako priključiti in sprogramirati cenen črni-beli SHARPov LCD grafični prikazovalnik brez tiskanine. Lotili smo se tudi preizkušanja med graditelji Arduino projektov priljub-



ljenega barvnega 1,25-inčnega Adafruitovega SSD1351 prikazovalnika, za priklop katerega ne potrebujemo kakih posebnih spretnosti...

Stran: 26

Krmiljenje LED-ice z govorom

Ali ni čudovito nadzorovati luči samo s svojim govorom? Uporabite Google pomočnika, IFTTT in platformo Bolt IoT za ustvarjanje glasovnega nadzora. Google pomočnik me je navduševal že v času, ko je bil



na voljo le na najnovejših pametnih telefonih. Ko sem odkril, da je mogoče IoT projekt ustvariti z Google pomočnikom, sem se odločil, da ga...

Stran: 36

VoiceTurn - glasovno vodene smerne luči za varnejšo vožnjo

Uporabnike opozorite na smer, v katero boste ubrali, samo tako, da rečete »levo« ali »desno«. Dobra praksa, ki jo morate upoštevati, ko ste v prometu, je, da druge uporabnike



na cesti opozorite na smer, katero boste ubrali, preden zavijete ali spremenite pas. Ta navada prispeva k bolj nemotenemu prometu in zmanjšuje nenadne premike... Stran: 42



Naslovnica: www.microchip.com





REVIJA ZA ELEKTRONIKO, AVTOMATIKO, RAČUNALNIŠTVO IN TELEKOMUNIKACIJE

Ustanovljena leta 1994, izhaja mesečno, 11 številk letno, julij/avgust ena številka.

Glavni in odgovorni urednik: JURIJ MIKELN, dipl.inž. Tel.: 01 549 14 00 E-pošta: stik@svet-el.si

Tehnični urednik: DTP studio AX d.o.o. E-pošta: dtp@svet-el.si

Prodajni servis, naročnine: E-pošta: prodaja04@svet-el.si

Marketing: Tel: 01 549 14 00 E-pošta: stik@svet-el.si

Prototipna tiskana vezja: Luznar d.o.o., Kranj Antivirusni program: PANDA security

Založnik in računalniški prelom: AX ELEKTRONIKA d.o.o. Depala vas 39, 1230 Domžale

Direktor: JURIJ MIKELN, dipl.inž.

Tisk: EVROGRAFIS d.o.o. Naklada do: 1.500 izvodov ISSN 1318 4679

Spletna revija: https://svet-el.si/category/revija/pretekle-stevilke

Cena za posamezni izvod je 4,50 EUR, za letno naročnino priznavamo 25% popust za dijake in študente s potrdilom o šolanju, 20% popust ostalim fizičnim osebam ter 10% popust za podjetja. V skladu s 25. členom 7. odstavka Zakona o davku na dodano vrednost se za revijo Svet elektronike plačuje in obračunava 9,5% DDV.

Izid publikacije finančno podprla Javna agencija za raziskovalno dejavnost RS iz sredstev državnega proračuna iz naslova razpisa za sofinanciranje domačih poljudnoznanstvenih periodičnih publikacij.

Uredništvo ne odgovarja za škodo, ki bi nastala zaradi nestrokovnega sestavljanja in uporabe naprav, ki so opisane v reviji, zaradi napak avtorjev ali napak v tisku. Uredništvo si pridržuje vse pravice do projektov, opisanih v reviji. Dovoljuje se izdelava naprav za lastno uporabo, prepoveduje pa se kakršnakoli reprodukcija projektov ali posameznih delov revije brez pisnega soglasja uredništva.

Jurij Mikeln Življenje se ne bo ustavilo!

Verjamem, da se strinjate z mano, drage bralke in dragi bralci. Življenje se zaradi Covid-19 ne bo ustavilo. Gotovo se bo upočasnilo ali pa spremenilo – saj veste, zdaj v noben zaprt prostor ne vstopimo brez zaščitne maske, kar bi pred 3 leti bilo nepredstavljivo. Pa vendar imajo maske in drugi zaščitni ukrepi svoj smisel. Nedolgo tega sem se pogovarjal z zdravnikom za pljučne bolezni. Povedal mi je, da je v letu 2021 imel zelo malo pacientov z »navadno« pljučnico, ker se zaradi zaščitnih ukrepov ne more prenašati iz človeka na človeka.

Poleg drugačnega življenja so se spremenili tudi načini dobave elektronike, o čemer smo pisali v pretekli številki. Po tem članku sem na spletu zasledil, da bo samo Intel vložil 20 milijard USD v novo tovarno polprevodnikov. Ja, veliki igralci vedo, kaj je strateška naložba. Podobno bi se morala vprašati tudi Evropa za področje polprevodnikov.

Mi v uredništvu pa smo povprašali vas bralce za odgovor na ne preveč zahtevno vprašanje. Nekatere odgovore boste lahko prebrali v tokratni številki. Izmed pravilnih odgovorov smo izžrebali naslednje dobitnike nagrade:



Vsem prejemnikom čestitamo! Nagrado – ADALM2000, digitalni osciloskop in logični analizator, bodo prejeli po pošti, dohodnina za nagrado je že plačana in s tem ne bodo imeli dodatnih stroškov.

V reviji, ki je pred vami, pa vam prinašamo obilico zanimivih in poučnih člankov. Tako boste lahko prebrali članek o organskih tranzistorjih, ki se bodo (kot kaže) uveljavili na papirju podobnih zaslonih ali komponentah, ki zavzemajo veliko površino.

No, bomo videli, ali se bodo organski tranzistorji »prijeli«. Hkrati pa si želimo, da bi se videli na sejmu IFAM, ki se bo odvijal v Ljubljani od 15. do 17. februarja na Gospodarskem razstavišču. Upamo, da nam Covid-19 ne bo preprečil prijetnega druženja s kolegi iz stroke.

Na sejmu bomo tudi mi – vabljeni, da nas poiščete v Hali A2.

Se vidimo v Ljubljani – če ne bo Covid-19 situacija tudi tokrat »odnesla« sejma IFAM.

Mainichi

Softbank Corp. bo vložil več kot milijardo jenov v raziskave za razvoj litij-zrak baterij, ki bi lahko imele petkrat več moči kot tiste, ki se trenutno uporabljajo za mobilne telefone, prenosne računalnike in električne avtomobile.

Polnjenje v prihodnost:

Softbank vlaga v raziskave

za razvoj litij-zrak baterij

Softbank se bo za projekt povezal z Nacionalnim inštitutom za znanost o materialih (NIMS), ki se nahaja v mestu Tsukuba v prefekturi Ibaraki. NIMS bo skupaj z zaposlenimi v Softbanku vzpostavil raziskovalno bazo s približno 50 člani in namera-

Junichi Miyakawa, vir: Softbank

vajo baterijo uvesti v praktično uporabo do približno leta 2025. Baterije delujejo tako, da združujejo shranjeni litij s kisikom v zraku, da ustvarijo litijev peroksid, ki se med polnjenjem ponovno razgradi na litij in kisik. Pričakuje se, da bo razvoj baterij močno povečal uporabo električnih naprav.

Softbank je od NIMS zaprosil za sodelovanje pri projektu, saj je znanstvena skupina pospeševala temeljne raziskave baterije. Trenutno vlaganje podjetij na Japonskem v razvoj litij-zrak baterij s strani univerz in raziskovalnih institucij zaostaja za Evropo in ZDA.

Na tiskovni konferenci za oznanitev partnerstva za projekt je glavni direktor NIMS Kazuhito Hashimoto dejal: "Ker smo globalni vodja, želimo razviti izdelke svetovnega razreda." Podpredsednik Softbank Junichi Miyakawa je dodal: "Če lahko dosežemo ta razvoj, bo postal velik nacionalni vir.

Povzeto po:

 https://mainichi.jp/english/articles/20180412/p2a/00m/0na/011000c

https://mainichi.jp



Science X

Integrirana vezja (IC), ki temeljijo na organskih tranzistorjih, imajo veliko dragocenih aplikacij, na primer pri izdelavi papirju podobnih zaslonov ali drugih elektronskih komponent, ki se nahajajo na velikih površinah.

V zadnjih nekaj desetletjih so inženirji elektronike po vsem svetu razvili različne vrste teh tranzistorjev.

Obetavna alternativa tem tranzistorjem so organski tan-

koplastni tranzistorji z navpičnimi kanali z dvojnimi vrati. Ti tranzistorji imajo več prednosti, kot so kratke dolžine kanalov in nastavljive mejne napetosti (VTH). Kljub tem prednostim se je zaradi pomanjkanja ustreznih p- in n-tipa naprav razvoj











)毎日新聞社 💽

NOVICE

komplementarnih inverterskih vezij za te tranzistorje doslej izkazal za izziv.

Raziskovalci na Technische Universitat Dresden, Helmholtz-Zentrum Dresden Rossendorf (HZDR) in Northwestern Polytechnical University so nedavno razvili vertikalne organsko prepustne tranzistorje z dvema bazama, ki bi jih bilo mogoče integrirati v logična vezja. V nedavnem članku [1], objavljenem v Nature Electronics, so ocenili potencialno uporabo teh tranzistorjev v kompleksnih integriranih vezjih.

"Tranzistor z dvojnimi vrati, ki smo ga razvili kot del naše prejšnje raziskave v Nature Communications [2], je sestavljen iz enega samega navpičnega kanala tankoplastnega tranzistorja z dodatnimi drugimi vrati in drugim dielektrikom, ki ga je mogoče uporabiti za nastavitev njegove mejne napetosti,", je povedala Erjuan Guo, ena od raziskovalcev, ki so izvedli študijo. "V naši novi študiji smo nadalje preučevali funkcijo in koristi vertikalnih kanalnih tranzistorjev z dvojnimi vrati v bolj zapletenih integriranih vezjih, na primer organskih komplementarnih inverterjih in obročnih oscilatorjih.

Guo in njeni sodelavci so ustvarili integrirane komplementarne pretvornike s

povezovanjem navpičnih n-kanalnih organsko prepustnih tranzistorjev z dvema bazama (OPDBT) in navpičnih p-kanalnih tranzistorjev z organsko prepustno bazo (OPBT). Predvsem druga vrata v OPDBT lahko nadzorujejo vklopljena in izklopljena stanja tranzistorjev in tako vplivajo na stanja pretvornikov. "Na podlagi meritev, ki smo jih zbrali, ugotavljamo, da tranzistorji z dvema bazama omogočajo širok razpon krmiljenja stikalne napetosti komplementarnega pretvornika nad 0,8 V pri vhodni napetosti <2,0 V na determinističen način," je dejala Guo. "Zato smo realizirali preklopno napetostno nastavljivo invertersko vezje z uporabo VTH-nastavljivega n-tipa OPDBT in p-tipa OPBT." Na podlagi dinamičnih odzivnih lastnosti lahko pretvorniki, ki so jih razvili Guo in njeni sodelavci, vzdržujejo visoke/nizke izhodne signale tudi pri 10 MHz vhodnega signala. Poleg tega lahko dosežejo tudi zelo kratke časovne konstante vzpona in padca 5 ns in 6 ns.

Poleg realizacije novega organskega komplementarnega pretvornika so Guo in njeni sodelavci izdelali sedemstopenjske komplementarne organske obročne oscilatorje, ki vključujejo 7 pretvornikov. Ti pretvorniki so jim omogočili demonstracijo prednosti vertikalnih organskih tranzistorjev za dinamično delovanje.

"Pri napajalni napetosti 4,0 V je izmerjena zakasnitev pri širjenju signala 11 ns na stopnjo za obročni oscilator v podobnem območju kot čas vzpona in padca signala posameznega



Fotografija: a,b, Shema vezja (a), statične prenosne karakteristike (b) organskega komplementarnega pretvornika, izmerjene z VCC = 3,5 V in VControl = 0,5, 1,0, 1,5 in 2,0 V. c, Fotografije izdelanega sedemstopenjskega obročnega oscilatorja z integracijo sedmih komplementarnih pretvornikov. d, Frekvence nihanja in zakasnitev pri širjenju signala na stopnjo sedemstopenjskega komplementarnega obročnega oscilatorja, izrisane kot funkcija napajalne napetosti VCC. Vezje lahko deluje z napajalno napetostjo do 2,0 V in z zakasnitvami pri širjenju signala do 11 ns na stopnjo. Vir: Guo et al.

> pretvornika," je dejala Guo. "Te zakasnitve signala so kratke v primerjavi s tistimi, o katerih so poročali do zdaj za organske obročne oscilatorje na katerem koli substratu pri napajalnih napetostih, manjših od 10 V."

> Guo in njeni kolegi so bili prvi, ki so za izdelavo integriranih komplementarnih pretvornikov uporabili vertikalne kanalne organske tankoplastne tranzistorje z dvema bazama. Njihovo delo bi lahko navdihnilo druge ekipe za ustvarjanje podobnih pretvornikov, s čimer bi utrlo pot k ustvarjanju novih elektronskih komponent.

> Študija potrjuje potencial uporabe vertikalnih organskih tranzistorjev za izdelavo visokofrekvenčnih logičnih vezij. Organski tranzistorji trenutno niso dovolj hitri, da bi jih lahko uporabljali v velikem obsegu, zato ekipa načrtuje nadaljnje študije, katerih cilj je povečati hitrost in zmanjšati velikost organskih tranzistorjev z dvema bazama in vertikalnimi kanali.

Viri:

- https://go.nature.com/3thORjq
- https://go.nature.com/3K4HEcs

Povzeto po:

https://bit.ly/3FIN5Qs



https://techxplore.com



Nov emulator v vezju (ICE)

Microchip Technology Inc.

MPLAB® ICE 4 In-Circuit Emulator podjetja Microchip je popoln sistem za emulacijo, programiranje in odpravljanje napak, ki vključuje brezžično povezljivost, močno odpravljanje napak in profiliranje kode v realnem času z uporabo sledenja.

Emulator povečuje produktivnost z bogatimi funkcijami pri programiranju in odpravljanju napak.

CHANDLER, Arizona, 4. januar 2022 – inženirji za ugnezdene rešitve, ki iščejo popolno analizo projektov za hiter razvoj, potrebujejo zmogljivo strojno opremo za emulacijo, ki je enostavna za uporabo. Microchip Technology Inc. (NASDAQ: MCHP) je predstavil MPLAB® ICE 4 naslednjo generacijo popolnega emulatorja v vezju, razvojnega orodja za odpravljanje napak in programiranja za mikrokontrolerje PIC® in AVR® (MCU), dsPIC®, krmilnike digitalnih signalov (DSC) ter SAM MCU in mikroprocesorje (MPU).

MPLAB ICE 4 In-Circuit Emulator je Microchipovo najhitrejše in najbolj zmogljivo orodje za emulacijo in programiranje svojih MCU-jev in MPU-jev, odpravljanje napak in programiranje z zmogljivim grafičnim uporabniškim vmesnikom MPLAB X Integrated Development Environment (IDE), ki je enostaven za uporabo. Emulator v vezju MPLAB ICE 4 zagotavlja prilagodljivo razvojno izkušnjo, vključno z naprednimi zmogljivostmi odpravljanja napak za pisanje energijsko učinkovite kode, skupaj z vsemi funkcijami, potrebnimi za skrajšanje časa odpravljanja napak. "Inženirji lahko razširijo svoje razvojne zmogljivosti s tem močnim "vse v enem " sistemom, ki ponuja nove možnosti in aplikacije z izboljšano strojno opremo in možnostmi brezžične povezljivosti," je povedal Rodger Richey, višji direktor poslovne enote za razvojne sisteme v podjetju Microchip. "Skupaj z naprednimi zmožnostmi nadzora porabe lahko inženirji ugnezdenega načrtovanja hkrati optimizirajo strojno in programsko opremo za celovit in energetsko učinkovit dizajn." MPLAB ICE 4 sistem emulatorja v vezju se povezuje z uporabo SuperSpeed USB 3.0 ali High-Speed USB 2.0 z možnostjo brezžične povezave z Ethernet ali Wi-Fi® povezljivostjo za boljšo fleksibilnost in enostavno uporabo. Sistem zagotavlja Ethernet ali Wi-Fi povezljivost za brezhibno brezžično programiranje in odpravljanje napak. Ethernet povezljivost omogoča oddaljeno odpravljanje napak za aplikacije, ki se spremljajo na dolge razdalje. Poleg tega je Wi-Fi povezljivost odlična za zagotavljanje izolacije od okoljskih pogojev, kot so visokonapetostne aplikacije za krmiljenje motorjev ali plavajoči sistemi brez ozemljitvene zanke. Zmogljiva strojna oprema MPLAB ICE 4 se integrira z nastavitvijo MPLAB X CI/CD prek Etherneta in ustvarja učinkovito kombinacijo za strojno opremo v zanki. Razvijalci lahko uporabijo CI/CD čarovnika za Jenkins in Docker nastavitev na najnovejši različici MPLAB X IDE v6.00.

Opremljen je z močnostnim odpravljanja napak za spremljanje, kako poraba energije ustreza kodi z uporabo MPLAB Data Visualizer. Poleg tega lahko inženirji za ugnezdene aplikacije iz kode izvlečejo več z uporabo dveh neodvisnih kanalov za zaznavanje toka z različnimi ločljivostmi za merjenje in optimizacijo porabe energije naprave. Inženirji lahko skrajšajo čas



razvoja z uporabo naprednih funkcij emulatorja v vezju MPLAB ICE 4, ki podpirajo številne široko uporabljene instrumente in zmožnosti sledenja ukazom. Več vmesnikov za razhroščevanje napak in programiranja ICE 4 ter možnosti za povezave v ciljni sistem z uporabo adapterskih plošč dodatno pomagajo skrajšati čas razvoja. MPLAB ICE 4 emulator v vezju razhroščuje in programira vse Microchip MCU, DSC in MPU, s čimer poenostavi proces načrtovanja, ko razvijalci migrirajo z enega Microchip MCU ali MPU na drugega. Z najnovejšo strojno opremo, podporo za vse naprave, več funkcijami, zanesljivostjo in brezhibno integracijo z najnovejšo različico MPLAB X IDE v6.00 zagotavlja popoln razvojni sistem. Poleg Microchipove lastniške brezplačne programske opreme, MPLAB X IDE za ugnezdene aplikacije, podjetje za razvijalce ponuja celotno paleto brezplačnih optimizirajočih prevajalnikov, licence za prevajalnik profesionalnega razreda, licence za funkcionalno varnost in licence za pokritost kode.

Za več informacij o vgrajenem emulatorju MPLAB ICE 4 obiščite Microchipovo spletno mesto: https://www.microchip.com/ en-us/development-tool/DV244140.

Cene in dobavljivost

MPLAB ICE 4 In-Circuit Emulator DV244140 je danes na voljo za 1.799 \$ in vključuje en komplet dodatne opreme. Dodatni kompleti pripomočkov so na voljo ločeno. Microchipov komplet dodatne opreme MPLAB ICE 4 AC244140 je na voljo za 350 \$. Za nakup izdelkov, omenjenih tukaj, obiščite naš nakupovalni portal ali se obrnite na pooblaščenega distributerja podjetja Microchip. Za dodatne informacije se obrnite na katerega koli prodajnega zastopnika podjetja Microchip ali pooblaščenega distributerja po vsem svetu.

Opomba: Ime in logotip Microchip sta registrirani blagovni znam-

ki podjetja Microchip Technology Incorporated v ZDA in drugih državah. Vse druge blagovne znamke, ki so morda tu omenjene, so last njihovih podjetij.



https://www.microchip.com

ELEKTRONIKE

Študentski kotiček—ADALM2000 Naredimo operacijski ojačevalnik iz diskretnih elementov

AX elektronika d.o.o.



AnalogDialogue

Analog Devices, Inc. V reviji Svet elektronike št. 302 smo v članku Študentski kotiček objavili nagradni vprašanji.

Za vezje na sliki 1, kakšno je ojačenje vhodnega signala W1, na izhod, ki ga vidimo na R,? Katera komponenta nastavlja to ojačenje?

Spremenite vrednost kondenzatorja C3. Kako večja ali manjša vrednost kondenzatorja vpliva na frekvenčni potek?

Prejeli smo več odgovorov, večina je bila pravilnih, objavili pa bomo dva odgovora, kjer sta se avtorja odgovorov malce bolj potrudila in vezje iz slike 1 tudi analizirala v LTSpice programu.

Po tehtnem premisleku in žrebu smo se v uredništvu odločili, da nagrade prejme-

jo: Aleš Simonič, prof. Jože Polak, Boštjan Tovšak

Avtor: Aleš Simonič

E-pošta: as3248@student.uni-lj.si

Pozdravljeni,

sem študent na Fakulteti za elektrotehniko in sem zasledil nagradno igro v vaši reviji v prispevku z naslovom Študentski kotiček-ADALM2000 Naredimo operacijski ojačevalnik iz diskretnih elementov.

Najprej bi rad opozoril na napako v shemi zaradi katere vezje ne deluje kot bi moralo. Označeno vozlišče ne sme biti vozlišče, slika 2.





8



Odgovor na 1. vprašanje (kaj določa ojačenje)

Ojačanje določata upora R2 in R1 in je enako -R2/R1, dejanska vrednost ojačenja je 4,7.

Odgovor na 2. vprašanje (vpliv kondenzatorja C3 na frekvenčno karakteristiko vezja):

- Manjša kapacitivnost: vezje začne oscilirati, ni več stabilno
- Večja kapacitivnost: kondenzator deluje kot nizkoprepustno sito in posledica je, da na izhodu ne dobimo več želenih frekvenčnih komponent, če je ta frekvenca že izven delovanja sita. Večja kot je kapacitivnost, manjša je zgornja frekvenčna meja vezja (večja kot je kapacitivnost, manjša *je impedanca* C3)

Ker diskretnih komponent nisem imel na voljo, sem vezje simuliral v programu LTSpice.

Vprašanje 1, slika 3. Vprašanje 2: (simuliranje več različnih vrednosti kondenzatorja C3), slika 4.







izris časovnega poteka izhodnega signala (zelena barva) in vhodnega signala (rdeča barva). Dodan je še izračun za ojačanje, ki se le malenkostno razlikuje od teoretične vrednosti zaradi neidealnosti komponent.



Izris AC analize (amplituda izhodnega signala) za drugo vprašanje pri spreminjanju kapacitivnosti C3. Opazimo zmanjšanje ojačanje pri višjih frekvencah.



Izris časovne analize za druge vprašanje pri spreminjanju kapacitivnosti C3. Vidimo zmanjšanje amplitud pri večjih vrednostih kapacitivnosti in nestabilnosti pri manjših.



PREDSTAVLJAMO

Avtor: prof. Jože Polak E-pošta: jfpolak040@gmail.com

Uporabljen simulator: LTspice

Vprašanje 1: Kolikšno je ojačenje vezja, če na vhod W1 priključimo sinusno napetost VW1=400mV frekvence 1kHz?

Odgovor 1: Ojačenje pri nastavljenih pogojih:

$$A = -\frac{V_{2+}}{V_{W1}} = \frac{-1,876V}{0,4V} = -4,69; \ A[dB] =$$

$$20logA = 20log|-4,69| = 13.4dB$$

Analiza 1: S povečanjem frekvence na f=4kHz izhodni signal nima več sinusne oblike.

Če bi zmanjšali amplitudo vhodnega signala na VW1=100mV, bi pri frekvenci 8 kHz na izhodnem priključku 2+ imeli še zadovoljivo sinusno obliko.

Zaključek Analiza 1:

Pri konstantnem ojačenju vezja z manjšo amplitudo vhodnega signala povečamo zgornjo frekvenčno mejo ojačenega signala. Zato je očitno, da vezje predstavlja invertirajoči operacijski ojačevalnik z dvema z dvema zunanjima uporoma R1 in R2, s katerima nastavljamo ojačenje.

Vprašanje 2: S katero komponento lahko spreminjamo ojačenje?

Odgovor 2: Ojačenje je odvisno od upornosti R1 in R2:

$$A = \frac{-R2}{R1} = \frac{-470k}{100k} = -4,7$$

Običajno spreminjamo ojačenje s spreminjanjem vrednosti upornosti v povratni vezavi, torej R2.

Vprašanje 3: Kakšen vpliv ima kondenzator C3?

Odgovor 3: V primeru, da ne bi uporabili kondenzatorja C3, bi ojačevalni sistem zanihal. Če bi povečevali kapacitivnost kondenzatorja C3, bi zmanjševali zgornjo frekvenčno mejo ojačenja.

fazno analizo ojačevalnika. Slika 6 prikazuje odziv, kjer je



Slika 1:Oiačevalno vezie s Spice modelom SSM2212



Slika 2:Oscilogram ojačanja V_{w1}=400mV, f=1kHz



Slika 3: Izhodni signal pri f=4kHz

Oscilograma na Sliki 5 in Sliki 6 prikazujeta frekvenčno in kondenzator C3=47nF, slika 7 pa odziv pri C3=470nF. Frekvenčno območje je od 10Hz do 10MHz.



Slika 4: Oscilogram VW1=100mV, f=8kHz



Slika 5: Invertirajoči operacijski ojačevalnik

Zaključek

Prav zanimivo nalogo so nam dali v podjetju Analog devices. In tudi zanimivo nagrado. ADALM2000 je multifunkcijska naprava, ki se priključi na USB in nam lahko služi kot generator signalov oziroma kot merilno mesto, saj med drugim vsebuje 16 kanalni logični analizator in dvokanalni osciloskop.



Slika 6: Bode analiza C3=47nF



Slika 7: Bode analiza C3=470nF

Vsi nagrajenci bodo po pošti prejeli ADALM2000, ki ga je sponzorirali podjetje Analog Devices. Čestitamo! In hkrati vabimo bralce, da sodelujejo tudi pri naslednjih nagradnih

https://svet-el.si

igrah, ki jih bomo objavili v reviji Svet elektronike.







Hitra in učinkovita uvedba prilagodljivih polnilnih sistemov za EV

Digi-Key Electronics Avtor: Rolf Horn

Trend v smeri e-mobilnosti je odvisen od pričakovane razpoložljivosti polnilne infrastrukture za električna vozila (EV) z javnimi postajami, ki jo dopolnjujejo ustrezni polnilni sistemi v domovih in na delovnih mestih uporabnikov.

Čeprav temeljne zahteve glede zasnove sistemov ostajajo pretežno usklajene, vsaka vrsta sistema prinaša posebne zahteve, kar predstavlja težavo, ki jo še poslabšujejo področne razlike pri dejavnikih, ki obsegajo vse od komunikacijskih platform do skladnosti z zahtevami.

Izziv za snovalce polnilne infrastrukture je torej v tem, da izpolnijo temeljne zahteve, njihova zasnova ponuja dovolj prilagodljivosti, da je dosežen najširši možen razpon zahtev glede končne uporabe in področnih zahtev, pri tem pa uravnotežijo stroške in čas do vstopa na trg.

V tem članku je opisana raznolikost zahtev glede zasnove javnih polnilnih postaj. Nato je predstavljena platforma prilagodljivih rešitev podjetja NXP Semiconductors, s katero je mogoče lansirati zasnove, ki so pripravljene tako, da izpolnjujejo te zahteve.

Soočanje z raznolikimi izzivi glede zasnove

Prizadevanja za pospešitev prehoda na električna vozila zahtevajo zadostno razpoložljivost učinkovitih polnilnih postaj (EVSE), bolj znanih pod imenom polnilni sistemi za EV. Potrebe glede lokalne vožnje je mogoče zadovoljiti s polnilniki, ki pretvarjajo izmenično napetost v enosmerno in so vgrajeni v vozila ter so namenjeni polnjenju doma ali na delovnem mestu, vendar ti polnilni sistemi ne zmorejo pomiriti zaskrbljenosti glede dosega EV, ki še naprej omejuje sprejetje EV. E-mobilnost velikega dosega je odvisna od razpoložljivosti javnih enosmernih polnilnih sistemov za EV, ki lahko EV napolnijo mnogo hitreje

kot vgrajeni polnilniki, ki pretvarjajo izmenično napetost v enosmerno. Hkrati morajo ti različni polnilni sistemi za EV ustrezati številnim standardom in predpisom glede varnosti, zaščite in zasebnosti.

Za razvijalce, ki ustvarjajo rešitve za polnilne sisteme za EV, potreba po zagotavljanju učinkovitih rešitev za vsak posamezen primer uporabe predstavlja tako izjemne priložnosti kot tudi precejšnje tehnične izzive. Razvijalci morajo poleg drugih izzivov zagotoviti širok nabor zmožnosti v izboru zasnov, ki lahko zagotovijo zahtevano delovanje in učinkovitost, hkrati pa izpolnijo posebne zahteve za posamezen način uporabe. Izpolnjevanje teh potreb zahteva prilagoditev temeljne arhitekture, ki je osnova vseh zasnov polnilnih sistemov za EV.

Prilagajanje osnovne arhitekture polnih sistemov za EV

Ne glede na posamezen ciljni način uporabe polnilne sisteme za EV sestavljata dva glavna podsistema – napajanje v ospredju in krmilnik za upravljanje napajanja v ozadju, ki ju ločuje izolacijska meja (Slika 1).

V ospredju, ki gleda proti vozilu in viru energije, podsistem z vmesnikom z vtičnico upravlja napajanje vozila. Na drugi strani izolacijske pregrade podsistem s krmilnikom skrbi za varnost, komunikacijo in druge funkcije na visoki ravni. Uvedba teh podsistemov je običajno odvisna od nekaj temeljnih gradnikov, ki izpolnjujejo posebne zahteve glede meroslovja, krmiljenja, funkcionalne varnosti, zaščite in komunikacije, ki so povezane s posameznim načinom uporabe.

Vsak gradnik prispeva ključno funkcionalnost k celotni zasnovi polnilnega sistema za EV. Meroslovna enota mora zagotoviti tako varen prenos energije kot tudi natančno merjenje energije za namene obračunavanja, v katerega ni mogoče nedovoljeno posegati. Krmilna enota zagotavlja zanesljivo izvajanje raznih protokolov, ki so zahtevani za prenos energije v eno smer in



Slika 1: Osnovna arhitektura za polnilne sisteme za EV vključuje ločena podsistema z vmesnikom z vtičnico in krmilnikom, ki ju ločuje izolacijska meja. (Vir slike: NXP Semiconductors) prenos podatkov v drugo smer, pri čemer gradi na zmožnostih za funkcionalno varnost in zaščito, hkrati pa podpira lokalne in področne zahteve glede protokolov za varno plačevanje in komunikacijo, ki se uporabljajo za komunikacijo z viri v oblaku.

V preteklosti so morali razvijalci prilagoditi zasnovo osnovne polnilne arhitekture za EV svojim zahtevam tako, da so uvedli vsak posamezen potreben gradnik, običajno s prilagojenimi zasnovami, ki so vključevale širok izbor splošnih naprav. Družina rešitev podjetja NXP za polnjenje EV ponuja učinkovito alternativo, ki razvijalcem omogoča združevanje vnaprej pripravljenih gradnikov za hitro ustvarjanje zasnov polnilnih sistemov za EV, ki ustrezajo širokemu izboru ciljnih načinov uporabe.



Slika 2: Serija Kinetis KM3x združuje celovit nabor funkcionalnih gradnikov, ki so potrebni za uvedbo preverljivega natančnega merjenja napajanja. (Vir slike: NXP Semiconductors)

Uvedba ospredja polnilnega sistema za EV

Rešitve podjetja NXP za polnjenje EV temeljijo na številnih družinah procesorjev, ki so zasnovani posebej za zagotavljanje delovanja in funkcionalnosti, ki sta zahtevana v zahtevnih načinih uporabe, kot so zasnove polnilnih sistemov za EV. Med temi družinami procesorjev so mikrokontrolerji serije Kinetis KM3x podjetja NXP zasnovani posebej za zagotavljanje preverljivega natančnega merjenja napajanja. Mikrokontrolerji Kinetis KM3x, ki temeljijo na 32-bitnem jedru ARM Cortex M0+, združujejo obsežen nabor funkcionalnih gradnikov za merjenje, zaščito, komunikacijo in sistemsko podporo ter Flash pomnilnik na čipu in statični pomnilnik (SRAM) (Slika 2).

Ospredje za merjenje z mikrokontrolerjem KM35xza poenostavitev uvedbe meroslovja združuje zelo natančen analogno-digitalni sigma delta (ADC) pretvornik, več analogno-digitalnih pretvornikov z registrom postopnega približevanja (SAR), do štiri ojačevalnike s programirljivim ojačanjem (PGA), hitri analogni primerjalnik (HSCMP), blok z logiko za kompenzacijo faze in zelo natančno notranjo napetostno referenco (VREF) z nizkim temperaturnim odzivom. Za zaščito celovitosti meroslovne enote zaščitna funkcionalnost na čipu podpira tako aktivno kot pasivno zaznavanje nedovoljenega poseganja s časovnimi žigi. Ti bloki na čipu v kombinaciji z zunanjimi tipali, releji in drugimi zunanjimi elementi zagotavljajo vso funkcionalnost, ki je potrebna za hitro uvedbo naprednega meroslovnega podsistema za ospredje z vtičnico polnilnega sistema za EV (Slika 3).

Uvedba krmilnika polnilnega sistema za EV

Kot že omenjeno, krmilnik polnilnih sistemov za EV organizira raznolike funkcionalne zmožnosti, potrebne v vsakem sistemu. Zahteve tega podsistema narekujejo uporabo procesorja, ki lahko zagotovi tako delovanje v realnem času, potrebno za zagotavljanje natančnega krmiljenja polnilnega sistema, kot tudi procesno zmogljivost, potrebno za podpiranje raznolikih protokolov, pri tem pa kar najbolj zmanjša okoljski odtis zasnove in stroške.



Serija i.MX RT procesorjev mešanih signalov podjetja NXP,

Slika 3: Z mikrokontrolerjem Kinetis KM razvijalci za uvedbo podsistema z vtičnico za EV potrebujejo le nekaj dodatnih zunanjih komponent. (Vir slike: NXP Semiconductors)

PREDSTAVLJAMO

ki temeljijo na jedru ARM Cortex-M7, zagotavlja zmožnosti v realnem času za ugnezdene mikrokontrolerje z delovanjem aplikacij na ravni procesorja. Procesorji i.MX RT, kot je i.MX RT1064, z delovno frekvenco 600 megahercov (MHz) in popolnim kompletom zunanjih elementov, lahko izpolnijo zahteve glede odziva v realnem času z nizko zakasnitvijo. Obenem funkcije, kot so velik pomnilnik na čipu, zunanji pomnilniški krmilnik, grafični podsistem in več vmesnikov za povezljivost, izpolnjujejo zahteve aplikacij. (Slika 4).

Zasnove polnilnih sistemov za EV morajo poleg izpolnjevanja ključnih zahtev glede realnega časa in delovanja zagotavljati zaščito na več področjih, vključno z zaznavanjem nedovoljenega poseganja ter preverjanjem pristnosti napajalnih povezav in plačilnih metod. Razvijalci lahko za zaščito podatkov, varen zagon in varno odpravljanje napak izkoristijo integrirane varnostne funkcije procesorja i.MX RT, kot so zagon z visoko zanesljivostjo, kriptografija strojne opreme, šifriranje vodil, varno obstojno shranjevanje in varen krmilnik JTAG (Joint Test Action Group).

Za dodatno utrditev zaščite krmilnika polnilnega sistema za EV se varnostne zmožnosti procesorja i.MX RT z zasnovo običajno dopolnijo tako, da se vklju-

či varnostni element EdgeLock SE050 podjetja NXP. Varnostni element SE050, ki je zasnovan za zagotavljanje celovite zaščite življenjskega cikla, zagotavlja pospeševalnike zaščite na osnovi strojne opreme za izbor priljubljenih kriptografskih algoritmov, funkcionalnost modula zaupanja vredne platforme (TPM), varne transakcije vodila in varno shranjevanje. Če razvijalci to napravo uporabljajo za zagotavljanje korena zaupanja (RoT) za izvajalno okolje, lahko zaščitijo ključne operacije, vključno s



Slika 4: Mešani procesor i.MX RT1064 združuje zunanje elemente in pomnilnik s podsistemom s procesorjem Arm Cortex-M7, ki je zasnovan za zagotavljanje tako izvajanja v realnem času kot tudi delovanje aplikacij na ravni procesorja. (Vir slike: NXP Semiconductors)

> preverjanjem pristnosti, varnim sprejemanjem, zaščito celovitosti in potrjevanjem.

> Če razvijalci uporabljajo procesor i.MX RT in napravo Edge-Lock SE05x, za uvedbo krmilnega podsistema, ki je zasnovan za izvajanje visokozmogljivega operacijskega sistema v realnem času (RTOS), potrebujejo le nekaj dodatnih komponent (Slika 5).





Slika 5: Mikrokontrolerji i.MX RT s svojimi integriranimi zmožnostmi funkcionalnosti in delovanja poenostavljajo zasnovo krmilnih podsistemov za polnilne sisteme za EV. (Vir slike: NXP Semiconductors)



Slika 6: Mikrokontroler KM3 in procesor mešaniih signalov i.MX RT v kombinaciji zagotavljata učinkovito temeljno strojno opremo za polnilne sisteme za EV. (Vir slike: NXP Semiconductors)

Prilagodljive rešitve za raznolike načine uporabe polnilnih sistemov za EV

Razvijalci lahko z združevanjem prej omenjenega napajalnega podsistema in krmilnih podsistemov z izbirnimi gradniki za možnosti plačevanja in komunikacije hitro uvedejo enofazni polnilni sistem za EV, ki lahko zagotavlja do 7 kilovatov (kW) (Slika 6).

Z relativno preprostimi prilagoditvami analognega ospredja je to isto zasnovo mogoče razširiti tako, da zagotavlja trofazni

polnilni sistem za EV, ki lahko zagotavlja do 22 kW (Slika 7).

Čeprav ta kombinacija naprav KM3x in i.MX RT ustreza mnogim primerom uporabe, drugi načini uporabe polnilnih sistemov za EV morda zahtevajo, da razvijalci optimizirajo druge vidike svojih zasnov. Rezidenčni polnilniki, ki so namenjeni zagotavljanju hitrejšega polnjenja od tistega, ki je možno z vgrajenimi polnilniki, na primer zahtevajo rešitve, ki optimizirajo stroške in okoljski odtis. Razvijalci lahko za te načine uporabe uvedejo cenejši krmilnik na osnovi ravni, ki uporablja stroškovno učinkovit mikrokontroler, kot je mikrokontroler LPC55S69 podjetja NXP.

Nasprotno imajo polnilniki komercialnih polnilnih postaj, ki so namenjeni javnim postajam, strožje zahteve glede hitre obdelave aplikacij in delovanja v realnem času. Te so potrebne za varno krmiljenje sistemov za shranjevanje baterij, ki delujejo na ravneh od 400 do 1000 voltov in zagotavljajo ravni polnjenja pri 350 kW ali več. V tem primeru je zmožnost izvajanja programske opreme na ravni aplikacije in programske opreme v realnem času ključna za delovanje in funkcionalnost. Pri teh sistemih uporaba procesorja, kot je procesor i.MX 8M podjetja NXP, razvijalcem omogoča lažjo uvedbo rešitev za polnjenje, ki lahko zagotavljajo obdelavo aplikacij na osnovi sistema Linux in delovanje v realnem času z omogočenim operacijskim sistemom v realnem času, ki sta potrebna v teh zapletenih zasnovah (Slika 8).

Hitra uvedba polnilnih sistemov za EV, povezanih z oblakom

Procesorji podjetja NXP, ki vključujejo Kinetis KM3x, i.MX RT, LPC55S69 in i.MX 8M, zagotavljajo prilagodljivo platformo za izpolnjevanje posebnih zahtev različnih načinov uporabe polnilnih sistemov za EV. Pri zapletenejših načinih uporabe pa lahko zamude pri postavitvi temeljne strojne opreme povzročijo precejšnje zamude pri razvoju celovitega načina uporabe polnilnega sistema za EV.

Podjetje NXP v izogib takšnim zamudam ponuja hitro pot do razvoja z naborom plošč in razvojnih kompletov na osnovi prej predstavljenih naprav. Modul TWR-KM34Z75M podjetja NXP na primer zagotavlja celotno meroslovno platformo, ki združuje meroslovni mikrokontroler Kinetis MKM34Z256VLQ7

PREDSTAVLJAMO

s celotnim kompletom podpornih komponent. Podobno razvojni komplet i.MX RT1064 podjetja NXP združuje procesor MIMXRT1064DVL6 z 256 Mb (megabit) pomnilnika SDRAM, 512 Mb bliskovnega pomnilnika in 64 Mb Flash pomnilnika QSPI, vse to na štirislojni plošči, skupaj z obsežnim naborom zunanjih priključkov, vključno z vmesnikom Arduino. Poleg tega plošča OM-SE050ARD podjetja NXP zagotavlja takojšen dostop do naprave EdgeLock SE050, razvojna plošča PNEV-5180BM podjetja NXP pa zagotavlja natično razvojno ploščo NFC za ospredje.

Razvijalci lahko z združevanjem plošče TWR-KM34Z75M podjetja NXP za meroslovje, procesorja i.MX RT1064 za krmilne funkcije ter plošč OM-SE050ARD in PNEV5180B hitro uvedejo platformo strojne opreme z vsemi funkcijami za sestavljanje načinov uporabe polnilnih sistemov za EV (Slika 9).

Rešitve na ravni plošč podjetja NXP v kombinaciji s storitvami v oblaku Microsoft Azure razvijalcem omogočajo hitro pripravo prototipa celovite rešitve za polnilni sistem za EV ter uporabo platforme kot osnovo za snovanje bolj specializiranih načinov uporabe.

Zaključek

Zadostna razpoložljivost polnilnih sistemov za EV je ključna za omogočanje emobilnosti, vendar stroškovno učinkovita uvedba različnih rešitev, ki so potrebne doma, v službi in na javnih postajah, ostaja ovira. Razvijalci lahko s platformo specializiranih naprav in rešitev s ploščami podjetja NXP Semiconductors hitro uvedejo zasnove z delovanjem, ki je potrebno za izpolnjevanje celotnega izbora načinov uporabe polnjenja EV, in prilagodljivostjo za odzivanje na

nastajajoče zahteve.

https://www.digikey.com

Slika 9: Razvijalci lahko s ploščami in razvojnimi kompleti podjetja NXP z razpoložljivimi storitvami v oblaku, kot je Microsoft Azure, hitro uvedejo celovite rešitve za polnjenje EV. (Vir slike: NXP Semiconductors)



Slika 7: Razvijalci lahko zasnovo, ki temelji na mikrokontrolerju KM3 in procesorju mešanih signalov i.MX RT, hitro prilagodijo tako, da podpira različne načine uporabe. (Vir slike: NXP Semiconductors)



Slika 8: Pri zapletenejših načinih uporabe, kot je izjemno hitro polnjenje EV, lahko razvijalci razširijo osnovno polnilno arhitekturo za EV z visokozmogljivimi procesorji, kot so procesorji i.MX 8M, ki podpirajo zapletenejše zahteve glede krmilnikov. (Vir slike: NXP Semiconductors)



Glasovno upravljanje, kako pametne naprave bolje slišijo in zvenijo

Rutronik GmbH

Avtor: Anne Santhakumar, Jeff Hsieh in Sam Cheng

Svetovna prodaja pametnih zvočnikov se je med letoma 2018 in 2019 povečala že z 99,8 na 134,8 milijona enot, analitiki pa napovedujejo do leta 2025 rast na 205,9 milijona enot. Bolj ko bodo prepričali s prvovrstnim zvokom in zanesljivim govornim upravljanjem, močnejša bo ta rast. Pri tem so lahko v pomoč določene smernice za načrtovanje.

Predvajanje glasbe, prebiranje vremenskih poročil in upravljanje pametnih naprav – vse to in še več zmorejo pametni zvočniki. Običajno so sestavljeni iz valjastega ohišja z navzdol obrnjenim zvočnikom. To mora enakomerno oddajati zvoke v vseh 360 stopinj prostora, tako da jih je mogoče optimalno slišati povsod. Za spre-



jem govornega upravljanja s strani uporabnikov so praviloma vgrajena polja mikrofonov z ustreznim algoritmom.

Difuzor čim bolj enakomerno porazdeli zvok po okolju. Širjenje zvočnih valov določata predvsem njegova geometrija in razdalja med difuzorjem ter membrano zvočnika. Podjetje Kingstate, dobavitelj akustičnih komponent in zvočnih rešitev, uporablja za optimizacijo kakovosti zvoka in zmogljivosti izdelkov analizo s končnimi elementi. S tem ustvari simulacijski model, ki napoveduje polarni diagram širitve zvoka (slika 1a/1b).

Napredni mikrokontrolerji za boljše razumevanje

Dolgo časa je bila ovira na poti do preboja naprav z govornim upravljanjem in sistemov za upravljanje doma pomanjkljivo prepoznavanje govora – uporabniki so prepogosto dobili odgovor »Ne razumem. Prosim, ponovite.« Eden od pristopov k reševanju te težave so naprednejši sodobni mikrofoni: Z njimi optimiziramo snemanje glasov in hkrati zmanjšamo neželene šume iz ozadja. To zagotavlja boljše prepoznavanje govora in več udobja za uporabnika.

Mikrofoni so zelo občutljive akustične komponente, ki pa ne sprejemajo le minimalnih zvočnih signalov v zraku, ampak tudi prevajane resonance in harmonska popačenja, ki jih povzročata nihanje zvočnika ter mehanika izdelka. Zaradi teh nelinearnih signalov DSP krmilnik (digitalni signalni procesor) ne more učinkovito obdelati signala s tehnologijo AEC (izničenje odmeva) in uporabnik sliši odmev. Kakovost zvoka se poslabša in nastane šum. V aplikacijah pa imamo možnosti, da mikrofone oblikujemo skupaj z zvočniki in tako dosežemo boljšo kakovost zvoka:

- Polje mikrofonov: Niz mikrofonov, sestavljen iz dveh do osmih mikrofonov, je mogoče kombinirati z DSP in algoritmi za oblikovanje snopa za izboljšanje kakovosti govora. Posamezni mikrofoni so postavljeni pod različnimi koti (razdalja do ustreznih mikrofonov je pomembna, da se izognemo težavam s fazo), da lahko zaznavajo in vrednotijo signale iz različnih točk.
- Občutljivost in frekvenca mikrofona: Mikrofoni polja morajo imeti čim bolj enake frekvenčne karakteristike, ki



Sliki 1a in 1b: Polarni diagram kaže širjenje zvoka. (Slika: Kingstate)

PREDSTAVLJAMO

se idealno ne razlikujejo za več kot 1 dB. Tako je mogoče zmanjšati odstopanje pri izračunih v krmilniku DSP. Skupna dolžina zvočnega predora, to je razdalja, ki jo zvok prepotuje med mikrofonom in zgornjim robom zaslona, ne sme biti daljša od 5 mm in ne sme pasti pod resonančno frekvenco mikrofonskega kanala, ki znaša 12 kHz (slika 5).

- Faza mikrofona: Faze mikrofonov morajo biti čim bolj enake, idealno do ± 5°. To skrajša čas odstopanja pri izračunih v DSP krmilniku in zagotavlja veliko točnost izračunanega zvoka ter smeri.
- Tesnjenje gumijastega držala mikrofona: Dodaten vidik, ki lahko popači kakovost zvoka, je tesnjenje gumijastega držala mikrofona. Zaradi preprečevanja prenosa zvoka notranjega zvočnika naprave v zvočni predor mikrofona skozi reže morajo biti gumijasto držalo in zaslon mikrofona brezhibno zatesnjena. Poleg tega mora zvočna izolacija znašati vsaj 20 dB SPL (decibelov zvočnega tlaka).
- Položaj in odpornost na udarce: Zaradi ojačenja zvočnika in AEC-obdelave signala mikrofona je zvočnik običajno postavljen na sredino naprave. Mikrofon naj bo čim bolj oddaljen od zvočnika. Da je mikrofon odporen proti udarcem in zrakotesen, mora biti prekrit z gumo. Ohišje zvočnika in luknje za vijake je treba zavarovati s penastim ali gumijastim držalom, da preprečite tresljaje. Sicer mikrofon sprejme signal

majo več ovir.

popačenja zvočnika (THD – Total Harmonic Distortion), kar vpliva na kakovost obdelave AEC.

S temi smernicami je mogoče izvesti zanesljive zasnove

akustičnih naprav – pametni zvočniki torej na svoji poti ni-



Slika 2: Z združevanjem informacij iz levega in desnega ušesa lahko ljudje določamo smer zvoka. Enako načelo omogočata dva ali več mikrofonov. (Slika: Kingstate)



Slika 3: Zasnova gumijastega držala mikrofona in zaslona (slika: Kingstate)

Rutronik GmbH,Podružnica v Ljubljani Motnica 5, 1236 Trzin, Slovenia E-pošta: rutronik_si@rutronik.com Tel. +386 1 561 09-80 https://www.rutronik.com





18 SVET ELEKTRONIKE



Razvijajte ločeno in integrirajte brezhibno z dvojedrnim krmilnikom

Microchip Technology Inc. Avtor: Harsha Jagadish

Ugnezdene aplikacije postajajo zapletene in izpopolnjene za doseganje večjega števila ciljev. Prvič, aplikacije morajo izboljšati učinkovitost, kar zahteva znatno zmogljivost krmilnika za izvajanje sofisticiranih algoritmov.

Nadalje, vseprisotna internetna dostopnost omogoča, da ugnezdene aplikacije postanejo "pametnejše" in bolj "povezane".

Tretji cilj je zmanjšati stroške z integracijo večjega števila funkcij, kot so vmesnik za senzorje, povezljivost, krmiljenje motorja, digitalna pretvorba moči, varnost in zaščita v enem samem krmilniku. Tako visoka stopnja integracije zahteva, da strokovnjaki za posamezne domene obravnavajo posebna funkcionalna področja ali module, nato pa je treba več funkcij integrirati v končno aplikacijo. Pogosto je zaradi multinacionalnih podjetij, ki imajo svoje ekipe razpršene po vsem svetu, še bolj pomembno, da je možno različne module razvijati ločeno in jih brezhibno integrirati z lahkoto, da se zmanjša razvojno tveganje in prizadevanja.

Izboljšanje učinkovitosti

Najprej poglejmo, kako cilj izboljšanja energetske učinkovitosti zahteva povečano zmogljivost krmilnika. Razmislite o primeru aplikacije za krmiljenje motorja. Industrija se odmika od enosmernih krtačnih motorjev, ki ponujajo 75-80-odstotno učinkovitost, in se usmerja k brezkrtačnim enosmernim



Slika 1: Današnje aplikacije so vse bolj "pametne" in "povezane" (Microchip)

(BLDC) motorjem ali novejšim sinhronim motorjem s trajnim magnetom (PMSM).

Ti motorji ponujajo do 85 - 90 odstotkov boljšo učinkovitost, zmanjšano glasnost in podaljšano življenjsko dobo izdelka. Tipično krmiljenje krtačnega enosmernega motorja zahteva zelo preproste tehnike nadzora smeri in hitrosti, ki jih je mogoče doseči z uporabo enostavnega in poceni 8-bitnega mikrokontrolerja. V primerjavi s tem je krmiljenje BLDC ali PMSM motorja brez senzoprjev s 'Field Oriented Control' (FOC) bolj izpopolnjeno in računsko bolj intenzivno.

Omogoča natančno kontrolo energije, ki jo porabi motor v širokem razponu obremenitve ali hitrosti, in pomaga znatno izboljšati učinkovitost. Dodatni krmilni algoritmi se lahko izvajajo tudi na podlagi zahtev aplikacije, kot so 'Zaznavanje in obnovitev zastoja rotorja', 'Prosto vrtenje po izklopu', 'Nasičenost PI zanke in preprečevanje navijanja', 'Slabitev toka' in 'Maksimalni navor na amper', ki pomagajo izboljšati zmogljivost, odzivnost na dinamično obremenitev in povečati splošno učinkovitost.

Vse te napredne tehnike nadzora so računsko intenzivne, vključujejo matematične operacije, kot so deljenje, množenje, kvadratno korenjenje in trigonometrične operacije, ki zahtevajo znatno pasovno širino centralne procesne enote (CPE). Ker je treba te nadzorne funkcije izvajati periodično z visoko frekvenco, je nujno, da CPE dodeli v določene časovne intervale. Takšno tesno izvajanje nadzorne zanke lahko zavzame večino pasovne širine CPE in lahko vpliva na druge časovno kritične funkcije v kompleksni aplikaciji.

Razvijalec ugnezdene programske opreme bo imel omejeno fleksibilnost pri dodajanju dodatnih funkcij, kot so komunikacija, varnostni nadzor, sistemske in "gospodinjske" funkcije, ki bi lahko motile časovno kritičen nadzor motorja. Izziv se povečuje v aplikacijah za digitalnihmočnostnih aplikacijah, kjer se morajo časovno kritične funkcije krmilne zanke izvajati na še višji frekvenci.

Kompleksna programska oprema

Zdaj pa razmislimo o naslednjem cilju, ki ga vodi povezljivost v internetu ali oblaku. Najnovejši trend v industriji je, da so aplikacije "pametne" in "povezane", ki ponujajo inteligenco in dostopnost od koder koli. Te pogoji zahtevajo, da ugnezdene aplikacije vključujejo več skladov programske opreme, kot so:

- Programska oprema za funkcijo glavne aplikacije. V našem primeru ta funkcija izvaja krmiljenje motorja, vzdrževanje in operacije uporabniškega vmesnika, ki so običajno potrebne v večini aplikacij.
- Komunikacijska programska oprema, ki poganja nujne omrežne protokole za povezljivost.
- Varnostna programska oprema za zaščito IP, zasebnost, integriteto podatkov, pristnost, nadzor dostopa in preprečitev vseh možnosti vdora.
- Če aplikacije vključujejo človeške operacije in lahko povzročijo telesne poškodbe zaradi okvare, potem mora biti celo programska oprema za funkcionalno varnost del takšnih

aplikacij, ki so kritične za varnost.

 Nekatere končne aplikacije imajo lahko tudi zahteve po prilagajanju, kjer bodo določene funkcije edinstvene za določene različice, namenjene različnim tržnim segmentom.

Vsi ti funkcijski pogoji zahtevajo, da se v razvoj ustreznih skladov programske opreme vključijo različne skupine strokovnjakov za področje in jih lahko optimalno in hitro integrirajo v končno aplikacijo. Strokovnjaki iz več področij se bodo morali zelo tesno uskladiti z arhitektom in implementirati končno aplikacijo. Ta scenarij se še bolj zaplete v multinacionalnih podjetjih, kjer bodo strokovne ekipe razporejene po vsem svetu.

Nižanje cene

Na koncu je optimizacija stroškov pomemben cilj, ki je skupen vsem končnim aplikacijam. Pogosto inženirji za ugnezdene aplikacije ne bodo imeli proračuna, da bi razmislili o zasnovi z več mikrokontrolerji, kjer se lahko posamezen sklad programske opreme izvaja na različnih mikrokontrolerjih z zelo malo koordinacije.

Najbolj optimalna rešitev bo izbira enega samega mikrokontrolerja z zelo visoko integracijo. To dodatno omogoča znižanje stroškov zaradi kompaktne zasnove PCB in zmanjšanega števila zunanjih komponent, kot so kristalni oscilatorji in pasivne komponente.

Kakšni so razvojni izzivi?

Za implementacijo sofisticiranih algoritmov in izvajanje več skladov programske opreme razvojniki ugnezdenih sistemov pogosto izberejo mikrokontroler z večjo zmogljivostjo. Vendar pa to morda ni najboljša izbira zaradi izzivov, povezanih s časovno kritično izvedbo, razvojem večjega števila skladov programske opreme, integracijo in testiranjem.

Preprost razporejevalnik ali operacijski sistem v realnem času (RTOS) lahko služi namenu načrtovanja in izvajanja večjega števila nalog iz različnih skladov na visoko zmogljivem CPE-ju na časovno spojen način. Vendar pa razporejevalnik ali RTOS dodaja dodatne stroške, ki porabijo pasovno širino v CPE, po-mnilnik in druge vire mikrokontrolerja.

Časovno spajanje prav tako poveča stroške preklapljanja in zmanjša učinkovito uporabo CPE. Scenarij se dodatno zaplete, ko je treba redno izvajati dve časovno kritični kompleksni krmilni zanki v natančnem in prekrivajočem se časovnem intervalu ali ko je treba v realnem času hkrati izvajati dve asinhroni varnostno kritični funkciji. V takih primerih uporaba mikrokontrolerja s še višjo zmogljivostjo ne bo vedno izpolnjevala sistemskih zahtev.

Tudi če ima visokozmogljiv enojedrni mikrokontroler dovolj pasovne širine CPU-ja, da sprejme več skladov programske opreme, morda skupaj z RTOS, je treba upoštevati še veliko drugih zapletov pri načrtovanju. Za razvoj, integracijo in preizkušanje večjega števila skladov programske opreme je potrebna precejšnja koordinacija med strokovnjaki za to področje. Zahteva razvoj združljive in modularne arhitekture



Slika 2: Tipičen blok diagram krmilnika z dvojnim neodvisnim jedrom (Microchip)

programske opreme, ki dinamično deli vire in informacije. Zapleti se še povečajo, če obstajajo stari skladi brez združljive arhitekture.

- Podedovani skladi imajo lahko različne arhitekture, ki temeljijo na načinu pozivanja ali načinu prekinitve
- Podedovani skladi morda uporabljajo iste vire mikrokontrolerja, ki jih je zdaj treba deliti brez kakršnih koli konfliktov, da bi se izognili nevarnostim, kot sta stanje dirke in zastoj
- Skladi lahko imajo več skupnih globalnih spremenljivk in funkcij z enakimi imeni
- Vsak sklad lahko deluje brezhibno, če se izvaja posamezno, vendar pa se lahko zgodi, da pri integraciji ne deluje pravilno. Odpravljanje napak tako integrirane rešitve bo nočna mora, ki podaljšuje čas razvoja.

Samostojni sklad, ki je že na voljo, morda ne pomaga vedno skrajšati razvojnega časa, če je implementiran na enojedrnem mikrokontrolerju. Vsi ti izzivi predstavljajo veliko razvojno tveganje in podaljšujejo čas trženja.

Dvojedrni krmilnik

Dvojedrni krmilnik pomaga izboljšati učinkovitost, poenostaviti razvojna prizadevanja in znižati stroške z naslednjimi ponudbami.

- Ponuja večjo zmogljivost kot podoben enojedrni krmilnik, ki deluje z dvakratno hitrostjo in je idealen za aplikacije, kjer sta dve ali več časovno kritičnih funkcij
- Poenostavlja razvoj programske opreme z dvojnimi neodvisnimi jedri, ki omogočata:
 - » Geografsko razpršen razvoj programske opreme
 - » Brezhibna integracija z zelo minimalno koordinacijo
 - » Enostavno prilagajanje funkcij v več različicah linije izdelkov

Dvojedrni krmilnik - boljša zmogljivost

Dvojedrni krmilnik omogoča večjo integracijo programske opreme, saj omogoča izvajanje različnih funkcij na dveh neodvisnih jedrih. To je še posebej koristno, če aplikacija zahteva občasno izvajanje dveh časovno kritičnih funkcij ob določenem času ali kot odziv na asinhrone dogodke. Med funkcijama ne bo spora z vsako časovno kritično funkcijo, ki se izvaja na dveh različnih neodvisnih jedrih.

To izboljša splošno izkoriščenost CPE-ja zaradi zmanjšanih stroškov preklapljanja konteksta med funkcijami ali pa celo brez stroškov. Številni dvojedrni krmilniki imajo namenske vire, kar dodatno zmanjša stroške preklapljanja in arbitraže.

Nekateri dvojedrni krmilniki imajo tudi namenski hitri programski RAM (PRAM), ki je povezan z enim od jeder, običajno s podrejenim jedrom, kar dodatno izboljša zmogljivost. Zato dvojedrni krmilnik ponuja večjo zmogljivost kot podoben enojedrni krmilnik, ki deluje z dvakratno hitrostjo.

Dvojedrni krmilnik – poenostavljen razvoj

Številni dvojedrni krmilniki ponujajo namenski pomnilnik, zunanje naprave in podporo za odpravljanje napak z vsakim jedrom. Prilagodljiva shema upravljanja virov nadalje omogoča skupno dodelitev virov enemu od jeder v skladu z zahtevami aplikacije.

Takšna arhitektura mikrokontrolerja omogoča neodvisen razvoj



Slika 3: Večfunkcijski kompleksni aplikacijski blok (Microchip)

programske opreme z zelo minimalno koordinacijo med domenskimi strokovnjaki in omogoča enostavno integracijo. Dvojedrni krmilnik še posebej poenostavi integracijo dveh programskih skladov, ki temeljita na različnih arhitekturah ali zahtevata podobna sredstva mikrokontrolerja, ki zdaj lahko delujeta na dveh neodvisnih jedrih.

To je podobno razvoju skladov za izvajanje na dveh različnih krmilnikih, vendar s prednostmi izboljšane zmogljivosti, optimalne izrabe virov in nižjih stroškov. To odpravlja vse zaplete, povezane z integracijo sklada, časovno združenimi viri in s tem povezanimi nevarnimi pogoji.

Dvojedrni krmilnik omogoča tudi enostavno odpravljanje napak po integraciji, saj ima vsako jedro lastne vmesnike za odpravljanje napak. Zaradi zmanjšane odvisnosti med skladi je odpravljanje napak izjemno poenostavljeno tako, da se izolirajo težave in se jih odpravi. Dvojedrni krmilnik, ki ponuja toliko prednosti, znatno zmanjša razvojno tveganje in čas za trženje.

Da dodamo še to na seznam prednosti: dvojedrni krmilnik omogoča enostavno prilagajanje brez spreminjanja glavne funkcionalnosti. Z arhitekturo glavne funkcionalnosti za delovanje na enem jedru je mogoče funkcije po meri implementirati na drugo jedro. Vse te ponudbe dvojedrnega krmilnika poenostavljajo načrtovanje programske opreme, tudi če je vključenih več ekip po vsem svetu, in omogočajo brezhibno integracijo z zelo minimalnimi prizadevanji za usklajevanje.

Dual-core Controller – Cost Reduction

Dvojedrni krmilnik z večjo zmogljivostjo omogoča razvijalcu ugnezdenih rešitev realizacijo kompleksnih aplikacij z enim samim mikrokontrolerjem. Dvojedrni krmilnik s poenostavitvijo razvoja znatno zmanjša čas načrtovanja in tveganje ter omogoča konkurenčne zasnove z nižjimi stroški in krajšim časom do trženja.

Da bi praktično uresničili vse zgornje prednosti dvojedrnega krmilnika, smo izvedli majhen poskus. V tej predstavitvi eno od jeder (običajno podrejeno jedro) izvaja krmiljenje motorja, ki izvaja FOC algoritem za nadzor BLDC motorja. Da bi ponudili grafični uporabniški vmesnik, drugo jedro (glavno jedro) izvaja grafični sklad za vmesnik OLED zaslona in izvaja sistemsko funkcijo za povezovanje potenciometra in tipk, ki nadzorujejo hitrost in stanje motorja.

Za prikaz preprostosti razvoja, ki jo ponuja dvojedrna naprava, sta grafični sklad in programsko opremo za krmiljenje motorja razvili dve različni skupini, ki sta bili geografsko ločeni. Zaradi prilagodljivosti za vzdrževanje neodvisne arhitekture programske opreme je bilo med obema ekipama potrebnega zelo malo usklajevanja.

Ekipa, ki ima strokovno znanje o krmiljenju motorja, bi lahko zelo hitro uvedla FOC algoritem za nadzor BLDC motorja. Ker ima druga ekipa strokovno znanje o razvoju grafičnega uporabniškega vmesnika, bi lahko obe ekipi izkoristili svoje izkušnje na posameznih področjih in hitro zaključili projekt.

Za vzpostavitev dogovora, kako bi prenašali status tipke in potenciometra med obema jedroma, je bilo potrebno zelo malo usklajevanja. Kot razširjen eksperiment sta obe ekipi uporabili že razpoložljive knjižnice programske opreme za izvajanje krmiljenja motorja in grafičnega vmesnika. To je privedlo do zaključka projekta v kratkem času z zelo malo truda, porabljenega za integracijo dveh različnih podedovanih skladov.

Zaradi visoke zmogljivosti jedra je bilo na obeh jedrih še vedno na voljo veliko pasovne širine CPE. Za premikanje meja je



Slika 4: Za prikaz demonstracije delovanja pojdite na povezavo [1]

bil na podrejeno jedro dodan tudi vmesnik za OLED zaslon, ki prikazuje dinamične parametre motorja brez vpliva na zmogljivost motorja. Tukaj je predstavitev aplikacije za krmiljenje motorja v živo z grafičnim uporabniškim vmesnikom, ki deluje na dvojedrnem digitalnem krmilniku signala.

Primer dvojedrnega krmilnika, ki ponuja vse te prednosti, je Microchipov najnovejši dvojedrni digitalni krmilnik signala (DSC) dsPIC33CH128MP508 [2]. Dvojedrni dsPIC33CH ponuja visoko zmogljivost z namenskim pomnilnikom in perifernimi napravami, specifičnimi za aplikacijo, zaradi česar je idealen za visoko zmogljive ugnezdene aplikacije, krmiljenje motorjev in digitalno pretvorbo energije. Dvojno jedro v tej družini razvijalcem omogoča, da ločeno razvijejo ugnezdeno programsko opremo za različne sistemske funkcije in jih nato nemoteno združijo, ne da bi se bloki kode med seboj motili.

Vir:

- 1: https://bit.ly/3pqjWz0
- 2: https://bit.ly/3po0vqw

Harsha Jagadish je vodja trženja izdelkov v poslovni enoti Microchip Technology za 16-bitne mikrokontrolerje. Ima osem let izkušenj v industriji polprevodnikov in se trenutno osredotoča na visoko zmogljive ugnezdene rešitve.

Dosegljiv je na:

- Twitter: @MicrochipTech
- Facebook: https://www.facebook.com/microchiptechnology/

Opomba: Ime in logotip Microchip sta registrirani blagovni znamki podjetja Microchip Technology Incorporated v ZDA in drugih državah. Vse druge blagovne znamke, ki so

morda tu omenjene, so last njihovih podjetij.



PREDSTAVLJAMO

https://www.microchip.com



Odprite vrata v IoT Wi-Fi[®] se sreča z Open-Source Linux[®]

Omogočite Wi-Fi na katerem koli MPU-ju, ki temelji na Linuxu z zaključenimi in pre-certificiranimi Wi-Fi moduli podjetja Microchip s podporo ta Mainline Linux gonilnik. Nudimo Linux gonilnike, ki so pripravljeni za uporabo za Wi-Fi Link krmilnike, ATWILC1000 (samo Wi-Fi) in ATWILC3000 (Wi-Fi + BLE 5). Skrajšajte čas razvoja z našo bogato izbiro virov na Github (https://github.com/linux4sam) vključujoč gonilnike, kernel, demo programje (z vključenim gonilnikom) in Getting Started uporabniškim navodilom. Nudimo programje za gostiteljske MPU-je ATSAMA5D2 in ATSAMA5D4 ali pa uporabite prednost njegove fleksibilnosti in uporabite MPU-je ostalih dobaviteljev preko Linux mainline kernel-a.

Začnite danes z ATWILC1000-SD razvojnim kitom.

- Wi-Fi IoT modul za povezavo in krmiljenje se poveže na AVR® MCU-je, MPU-je podjetja Microchip in druge procesorje
- · SPI in SDIO vmesnika gostiteljev optimizirana za Linux
- Podpira varnost IEEE 802.11 WEP, WPA, WPA2 tako osebno kot tudi v podjetju

Za več informacij obiščite Getting Started spletno stran na www.microchip.com/wifi-for-linux.



Mladi navdušenci nad robotiko so se zbrali v Ljubljani



UM FERI Avtor fotografij: Domen Ulbl

Osnovnošolci in srednješolci iz vse Slovenije so se pomerili v treh tekmovalnih kategorijah.

Maribor, 20. december – V soboto, 18. decembra 2021, je v Ljubljani v Digitalnem središču Slovenije potekal dogodek »Mladi in robotika: Revijalno robotsko tekmovanje RoboCupJunior«. Dogodek je organizirala Fakulteta za elektrotehniko, računalništvo in informatiko Univerze v Mariboru (UM FERI).

RoboCupJunior (krajše RCJ) je robotsko tekmovanje, organizirano na osnovi mednarodno veljavnih pravil z namenom promocije in popularizacije robotike, inovatorstva in znanosti med mladimi na temeljih projektno zasnovanega izobraževanja. RCJ ponuja več tekmovalnih disciplin oz. izzivov, pri čemer je pri vseh osnovni poudarek na sodelovanju, reševanju problemov in doseganju zastavljenih ciljev.

Udeležence tekmovanja so z uvodnim nagovorom v Digitalnem središču Slovenije v ljubljanskem BTC-ju pozdravili: doc. dr. Igor Kovač, direktor SRIP ToP; doc. dr. Boštjan Vlaovič, prodekan za izobraževalno dejavnost UM FERI; izr. prof. dr. Aleš Hace, predstojnik Inštituta za robotiko UM FERI; Erih Arko, predstavnik podjetja Yaskawa Slovenija.

Digitalno središče Slovenije, ki deluje pod sloganom »Tehnologije za ljudi«, je osrednji gospodarski dogodek slovenskega predsedovanja Svetu Evropske unije. Mesec december je bil posvečen vsebinam namenjenim Industriji 4.0 in robotiki, ki je tematski sklop še posebnega strateškega pomena za prihodnji razvoj Slovenije. Namen programa in izvedenih dogodkov je bilo izpostavljanje in podarjanje glavnih dosežkov slovenske industrije, ki so pomembni za nadaljnji razvoj Industrije 4.0 in robotike, promocija slovenskega znanja in dosežkov ter ustvarjanje priložnosti za internacionalizacijo, prav tako pa tudi združevanje vseh pomembnejših deležnikov in znanja z namenom plodnega sodelovanja. Še posebno pozornost so ob vsem naštetem posvetili človeškim virom in mladim talentom, kar je izrednega pomena za dolgoročni industrijski razvoj.

»Naša skupnost prepoznava izjemni pomen vzpodbujanja zanimanja mladih generacij za znanost in tehnologije prihodnosti. Robotika je prav gotovo tista tehnologija, ki kaže izjemen razvojni potencial in tehnologija, ki se bo v prihodnje razširjala izven industrijskega okolja ter nas vedno bolj in bolj obdajala tudi v našem vsakdanjem življenju,« je povedal izr. prof. dr. Aleš Hace. Od leta 2000 do leta 2019 smo na UM FERI organizirali in izvedli 20 državnih robotskih tekmovanj za osnovnošolce, srednješolce in študente. »V navedenem obdobju je sodelovalo okrog 3000 učencev osnovnih šol, okrog 600 dijakov srednjih šol in okrog 100 študentov iz celotne Slovenije in sosednjih držav. Z omenjenimi robotskimi tekmovanji smo opravili pionirsko delo pri popularizaciji robotike v Sloveniji, hkrati pa smo bili med prvimi v srednjeevropskem prostoru,« je še



pojasnil. »Zadnji dve leti državno tekmovanje zaradi ukrepov in koronavirusa žal ni potekalo, a upamo in veselimo se maja 2022, ko ponovno načrtujemo izvedbo in organizacijo državnega tekmovanja na naši fakulteti.«

Tekmovanje RoboCupJunuior (RCJ) ima sicer sledeče discipline: RCJ Reševanje Črta, RCJ Reševanje Labirint, RCJ Nastop, RCJ Nogomet in RCJ Reševanje Simulacija. V Ljubljani so potekale tri kategorijel.



RCJ Reševanje Črta

Tekmovanje RCJ Reševanje Črta predstavlja reševanje žrtev po naravni nesreči, npr. potresu. Naloga robota je, da se uspešno prebije do območja nesreče, tam poišče ponesrečence (žrtve) in jih nato prenese v varno območje. Tekmovalna arena je sestavljena iz dveh delov. Prvi del predstavlja pot do področja nesreče, drugi pa evakuacijsko področje. Pot je precej zahtevna, saj je polna ruševin, ovir in preprek. Proga je v tem delu arene označena s črno črto na beli podlagi, na progi pa so različne fizične ovire. Drugi del tekmovalne arene predstavlja evakuacijsko področje, kjer so žrtve (žogice), ki so naključno razporejene. Naloga robota je, da žrtve poišče in jih prenese do varnega območja (evakuacijske točke).



V kategoriji Reševanje črta se je na sobotnem dogodku pomerilo 18 ekip, 5 iz osnovnošolskih šol in 13 srednješolskih.

RCJ Reševanje Simluacije

Reševanje Simulacije je tekmovanje robotov v dveh svetovih simulacijskega okolja. Roboti imajo tri senzorje in pobirajo ter odlagajo predmete. Pobrane predmete morajo odložiti na določeno mesto, da dobijo zanje točke. Zmaga robot z najučinkovitejšim programom in strategijo vožnje, torej tisti, ki zbere v omejenem času največ točk za pobrane predmete in se med pobiranjem uspešno izogne vsem oviram in nevarnostim (močvirje, ipd.). Za vsako neuspešno izogibanje nevarnosti je robot kaznovan s časovno omejeno prepovedjo vožnje.

V kategoriji Reševanje Simulacije sta se v soboto pomerili dve srednješolski ekipi.

RCJ Reševanje črta SŠ

- mesto: ekipa SloTeam iz ŠC Celje, Gimnazija Lava
- mesto: ekipa GIMKP iz Gimnazije Koper
 - mesto: ekipa Programerji iz ŠC Nova Gorica

RCJ Reševanje Simulacija

- mesto: ekipa SERŠ TEAM iz Srednje elektro-računalniške šole Maribor
- mesto: ekipa SERŠ LER iz Srednje elektro-računalniške šole Maribor

RCJ Nogomet

- mesto: ekipa DNTŽ iz ŠC Ptuj, Elektro in računalniške šole Ptuj
- mesto: ekipa TŠC FootBot iz Tehničnega šolskega centra Maribor



https://feri.um.si https://www.um.si

RCJ Nogomet

Disciplina RCJ Nogomet je za obiskovalce skoraj najbolj zanimiva disciplina. Kot pove že samo ime discipline, se roboti med seboj pomerijo na nogometnem igrišču. Vsako ekipo sestavljata dva robota, tekma pa traja pet minut.

V nogometu sta se v soboto pomerili dve srednješolski ekipi.

Rezultati

RCJ Reševanje črta OŠ

- mesto: ekipa OŠKP iz OŠ Koper
- mesto: ekipa RoboMaks iz OŠ narodnega heroja Maksa Pečarja
- mesto: ekipa Garfield iz Prve OŠ Slovenj Gradec





Merjenje zračne vlage in zračnega tlaka z ESP32 AVR, ESP, PIC in drugi, ko gre zares (7)

Avtor: dr. Simon Vavpotič E-pošta: 15579247@users.siol.net Avtor fotografij: Simon Vavpotič

V preteklem nadaljevanju nas je zanimalo, kako priključiti in sprogramirati cenen črni-beli SHARPov LCD grafični prikazovalnik brez tiskanine. Lotili smo se tudi preizkušanja med graditelji Arduino projektov priljubljenega barvnega 1,25-inčnega Adafruitovega SSD1351 prikazovalnika, za priklop katerega ne potrebujemo kakih posebnih spretnosti spajkanja, saj je konektor z natičnimi kontakti priložen, na koncu pa še začeli izdelovati mini vremensko postajo z barvnim prikazovalnikom in merilniki temperature, zračnega tlaka in relativne zračne vlažnosti.

Tokrat moramo na vremensko postajo 3.0 priključiti še tipali za merjenje relativne zračne vlage in zračnega tlaka, poleg teh pa izdelati še zunanje merilne enote, ki jih bodo poganjali ESP32 moduli. Zdaj, ko imamo 262.143 barv, moramo na zaslon naložiti vsaj eno barvno fotografijo. Preverilo bomo tudi možnosti za priklop precej zmogljivejšega 4-inčnega Waveshareovega prikazovalnika z zaslonom na dotik, ki sem dočakal v prvih dneh januarja. Izdelan je kot Arduino ščit, a kljub temu ga lahko uporabljamo v drugih mikrokontrolerskih rešitvah in seveda z Raspberry Pi.

Z zaslonom na dotik odpade potreba po tipkovnici in enkoderju, zato je dizajn strojne opreme bistveno enostavnejši. Po drugi strani, lahko kot Arduini ščit zasnovan prikazovalnik z zaslonom na dotik povežemo tudi z Raspberry Pi. Dobro je tudi to, da njegova cena ni pretirana, saj sem zanj odštel okoli 35 EUR. Izdelali bomo tudi Ethernet prehod za ožičeno povezavo v Internet...

Komunikacija po vodilih SPI in I2C

Večina senzorjev za merjenje vremenskih parametrov komunicira preko vodil SPI in I2C. Slednje je v zadnjem času vse bolj popularno, saj lahko prek istega vodila z dvosmernim podatkovnim in taktnim signalom komunicira veliko naprav. Vremenska postaja 3.0 lahko zajema podatke iz ADT7410 (temperatura), BMP/BME 280 (zračni pritisk in temperatura) in HTU21DF modula (relativna vlažnost zraka). Vendar senzorjev ne moremo dodajati v nedogled, saj ima večina že vgrajena 10 k Ohmska upora za veliko naslovnih in podatkovnih linij navzgor, tako se upornost z vzporedno priključitvijo še enega senzorja na vodilo zmanjša na 5 k Ohm, s priključitvijo tretjega senzorja pa na 3,3 k Ohm.

Za preverjanje prisotnosti naprav na vodilu I2C lahko uporabimo naslednji Arduino C/C++ program, ki je še posebej uporaben, ko nismo prepričani o tem, na katerih naslovih so naprave, ki so priključene na vodilo:

```
nDevices = 0;
for(address = 1; address < 127; address++ ){
Wire.beginTransmission(address);
error = Wire.endTransmission();
if (error == 0){
Serial.print("I2C device found at address
0x");
if (address<16)Serial.print("0");
Serial.print(address,HEX);
nDevices++;
}
}
```

Tako ni potrebno ugibati, ali smo morebiti zgrešili naslov na I2C vodilo na novo povezane naprave, ampak se izpišejo naslovi vseh najdenih naprav. Podoben program lahko v programskem jeziku Python ali C++ lahko uporabimo tudi na Raspberry Pi.



Slika 1: Vremenska postaja 3.0

Vremenska postaja ter BMP280 in BME280 senzorja

Merjenje zračnega tlaka in relativne zračne vlage je povezano z merjenjem temperature, saj moramo pri obeh za natančnost meritev upoštevati tudi temperaturo prostora, v kateri sta izmerjena. Prav zato imajo (skoraj) vsa tovrstna digitalna tipala vgrajen tudi termometer. Dodaten digitalni termometer potrebujemo le tam, kjer morebiti potrebujemo večno natančnost merjenja temperature. Kakorkoli, za izgradnjo enostavne vremenske postaje zadošča že BME280 senzor, ki meri tlak, vlažnost in temperaturo, medtem ko starejši BMP280 meri samo tlak in temperaturo. Zato potrebujemo še dodatni senzor za merjenje relativna vlažnosti zraka.

Na Internetu najdemo različne cenene miniaturne razvojne ploščice s BMP280 ali BME280 senzorjem, ki se medsebojno ne razlikujejo po tem. Nekateri proizvajalci na razvojnih ploščicah celo navajajo oznako BME/BMP280, s čemer povejo, da gre za enako razvojno vezje, vendar je nanj lahko nameščen BMP280 ali zmogljivejši BME280. Slednji je precej dražji, zato je upanje, da bomo za okoli 8 EUR plus poštnino dobili še merilnik vlažnosti zraka zaman.

K sreči je Adadfruit pripravil programske knjižnici (github.com/adafruit/Adafruit_ BME280_Library in github.com/adafruit/Adafruit_BMP280_Library) za razvojno okolje Arduino za oba čipa. In kaj pa če ne vemo, kateri čip je vgrajen? V tem primeru v Adafruit BMP280 Library poiščemo primer bmp280_sensortest.ino ga prevedemo in naložimo v svojo z Arduinom združljivo razvojno ploščo. Če čip BMP280 ali združljiv, dobimo meritvi temperature in zračnega tlaka, v nasprotnem primeru, pa ID kodo čipa, po kateri lahko preverimo, kateri senzor uporabljamo. No, vsekakor je tu opcija tudi starejši BMP180.

Kakorkoli, moj čip nosi oznako 0x58, zato ne omogoča meritve vlažnosti zraka, sem pa zanj vseeno uporabil nekoliko prirejeno sodobnejšo programsko knjižnico, ki so jo pri Adafruitu razvili za BME280 senzor.

Merjenje zračnega tlaka

Miniaturni BMP in BME senzorji omogočajo vse vrste komunikacij SPI in I2C, odvisno od tega, kako vežemo njihove priključke. Sam sem nabavil cenejšo GYBMEP razvojno ploščico z le štirimi priključki, ki podpira samo I2C komunikacijo. Je pa pohvalno, da stebrasti kontakti niso prispajkani, saj je ploščico tiskanega vezja pri nekaterih aplikacijah smotrneje pritrditi neposredno na kabel in privarčevati prostor. Po drugi strani, imajo nekoliko dražje Adafruitove razvojne ploščice izvedene vse kontakte senzorja in tako omogočajo poljubno izbiro načina komunikacije.

GYBMEP razvojna ploščica ima za komunikacijo prek vodila I2C prednastavljen naslov 0x76, kar moramo upoštevati tudi pri uporabi Adafruitovih primerov, v katerih je predizbran naslov 0x77, razen če se potrudimo in na miniaturnem razvojnem vezju prespajkamo kontakt za izbiro naslova.

Kako neprijetno zna biti vzpostavljanje prvega stika z BMP280 senzorjev s pomočjo z Adafruitovih programskih knjižnic z ne-Adafruitovo razvojno ploščico sem izkusil tudi sam. S prvo od treh ploščic v kompletu, ki sem ga za okoli 8 EUR naročil prek nemškega Amazona, sem izgubil veliko časa, saj ni in ni hotela delovati, ker sem po nesreči zamenjal priključka SDA in SCL. Kot kaže, je bilo to tudi usodno, saj ni delovala niti, ko sem opravil vse težave, je pa nato takoj začela delovati druga ploščica iz kompleta.



Slika 2: Modul za merjenje temperature na osnovi ADT7410 temperaturnega senzorja



PROGRAMIRANJE

Kako prirediti novejšo programsko knjižnico za BME280 tudi za starejši BMP280?

Če programske knjižnice v Arduino razvojno okolje nameščamo ročno, ali pa jih enostavno skopiramo v projekt, je njihovo popravljanje in dopolnjevanje sorazmerno enostavno. Adafruit BME280 programska knjižnica ne podpira drugih senzorjev, saj v inicializacijskem podprogramu preveri, ali je ID koda senzorja 0x60. Kljub temu se nabora registrov BMP280 in BME280 za programerja skoraj ne razlikujeta. Tudi izmerjeni in kompenzirani vrednosti temperature in zračnega tlaka se ne razlikujeta glede na uporabljeno programsko knjižnico, zato pa ima programska knjižnica za BME280 sodobnejšo zasnovo. Z BMP280 jo lahko uporabimo, če v funkciji Adafruit_BME280::init() spremenimo pogoj:

Ker pa se BMP280 in BME280 razlikujeta le po enem registru, iz katerega programska knjižnica prebere vlažnost zraka, bomo pri BMP280 namesto dejanske relativne vlažnosti zraka vselej prebrali vrednost 0.

Kako umeriti merilnik tlaka glede na podatke ARSO?

Ostane še zadnja naloga, umerjanje glede podatke o zračnem tlaku, ki jih vsakodnevno sporoča Agencija Republike Slovenije za okolja (ARSO). ARSO zaradi lažje primerjave podatkov med različnimi kraji v Sloveniji vse meritve zračnega tlaka preračuna na nadmorsko višino 0 metrov, oziroma gladino morja. Za preračun vrednosti izmerjenega zračnega tlaka na ekvivalentno vrednost zračnega moramo poznati temperaturo prostora in nadmorsko višino, na kateri smo zračni tlak izmerili. Od zahtevanih podatkov s BMP280 ali BME280 senzorjem ne moremo izmeriti le nadmorske višine, vendar to navadno poznamo (ali preberemo iz GPS aplikacije na pametnem telefonu). Za Ljubljano je okoli 300 m. Potem je ekvivalentna vrednost zračnega tlaka na morski gladini (v skladu z ARSO meteorološkim priročnikom Uvod v meteorologijo, ki ga lahko prenesemo iz njihovih spletnih strani) naslednja:

$P(0)=P(z)/e^{(-g/(RT)*z)}$

Pri tem pomeni P(z) izmerjeni zračni na nadmorski višini z (metrov), P(0) pa tlak na gladini morja. Temperaturo T prav tako izmerimo, vendar jo moramo izraziti v Kelvinih, kar pa ni težko, če vemo, da je teoretično najnižja mogoča temperatura v vesolju -273,15 st. C oziroma 0 K. Torej moramo temperaturi v st. Celzija to vrednost prišteti, da dobimo vrednost v Kelvinih. Pri tem je vrednost konstante R enaka 287 K / (kg K), e je naravno število (2,718), g pa težnostni pospešek na Zemlji, ki znaša približno 9,81 m/s2. Če merimo zračni tlak v Ljubljani, ki je na



Slika 3: Priključek na modula za merjenje temperature

nadmorski višini približno 300 m pri 25 st. C in izmerimo tlak 991 hPa, tako dobimo:

P(0)=P(z)/e^{(-g/R/T*z)=P(z)*e^(g*z/R/T) P(0)=P(300) *e^(9,81*300/287/298,15) = 991,00 hPa * e^{(9,81*300/287/} (273,15+25)) = 1025,68 hPa.}

Vse, kar moramo zdaj še narediti, ko merilnik tlaka namestimo v prostor, je vnesti pravo nadmorsko višino. Poglejmo še, kaj moramo dodati v funkcijo readPressure v programski knjižnici Adafruit_BME280.cpp pred zadnji stavek return P:

float P= Pz*exp(9.81*elevation/287/ (273.15+Adafruit_BME280:: readTemperature()));

Pri tem v spremenljivki elevation podamo nadmorsko višino kraja v metrih, kjer smo izmeriti zračni tlak. Vrednosti elevation lahko spremenimo tudi med delovanjem vgrajene programske opreme s pomočjo terminalskega dostopa z ukazom E(<nova nadmorska višina v metrih>). Če nastavimo nadmorsko višino 0 metrov, so izmerjene vrednosti zračnega tlaka nekompenzirane, saj je e^0 enako 1. Vsekakor pa je ta nastavitev čisto v redu, če vremensko postajo namestimo v obmorskem kraju.

Naj omenim še to, da lahko enako kompenzacijo vgradimo tudi v programsko kodo vremenske postaje V2.0, katere osnova je PIC32MX mikrokontroler. Za enkrat je pri tej uporabljen enostavnejši način kompenzacije z referenčno vrednostjo tlaka, kar ima za posledico nekoliko večja odstopanja.

Kako BMP280 dodati merilnik relativne vlage zraka?

Če ste pomotoma kupili GYBMEP razvojno ploščico z BMP280

merilnikom, nam ne preostane drugega kot nabava HTU21DF razvojne ploščice za merjenje relativne vlažnosti zraka. K sreči imata GYBMEP in HTU21DF razvojni ploščici enako razporejene kontakte, zato ju lahko z uporabo dolgih stebričastih kontaktov povežemo v en senzor, ki ga nato vtaknemo v isti priključek, kot če bi imeli BME280. Vseeno pa je potrebno za uporabo HTU21DF dodati tudi ustrezno programsko knjižnico (Adafruit_HTU21DF). HTU21DF ima na I2C vodilu fiksen naslov 0x40, zato lahko na istem vodilu uporabljamo le en merilnik, lahko pa kot vhodni parameter funkcije begin podamo tudi izbrano I2C vodilo, če uporabljamo več I2C vodil.

Shranjevanje vremenskih podatkov

Ker je del vremenske postaje tudi VS1503 modul z ležiščem za micro SD kartico, bi bilo škoda, da ne bi izmerjenih vremenskih parametrov tudi shranjevali. Shranjevanje v datoteko praviloma poteka intervalno, denimo enkrat na minuto ali enkrat na uro, saj pogosteje ni smiselno, ker se vremenski parametri spreminjajo sorazmerno počasi.

Ker o shranjevanje podatkov v datoteke na micro SD kartice v Arduino okolju ni ravno zapleteno, poglejmo le, kako podatke dodajamo na rep obstoječe datoteka, saj večina primerov s spleta datoteko preprosto prepiše z novimi podatki:

```
if((fscn=SD.open("/log.txt",FILE_APPEN-
D))!=NULL){
fscn.write((uint8_t *)fprintout.c_
str(),fprintout.length());fscn.close();}
else {Serial.println("Cannot write to log.
txt error!");result=-1;}
```

Kot vidimo, je potrebno v ukazu open kot drugi parameter dodati atribut FILE_APPEND. Če ga ne podamo, se ukaz izvede s prednastavljeno vrednostjo FILE_WRITE, kar pomeni, da se najprej obstoječa vsebina datoteke izbriše, nato pa se vanjo zapiše nova, zato bi imeli v tem primeru vselej shranjeno le zadnjo meritev.

Poceni vremenska postaja z Raspberry Pi

Vodili I2C in SPI imajo tudi Raspberry Pi tipa B. Najstarejše različice imajo dvovrstični razširitveni konektor s 26-kontakti, ostali pa konektor s 40 kontakti. Povezava senzorjev BMP280, BME280 in HTU21DF z Raspberry Pi je zato prav tako enostavna kot s katerimkoli Arduino modulom. Paziti moramo le, da pravilno povežemo kontakte. Pri Raspberry Pi s konektorjem s 40 priključki vežemo priključek 1 na Vin, priključek 3 na SDA, priključek 5 na SCL in priključek 9 na GND (glej sliko). Za merjenje temperature, relativne vlažnosti zraka in zračnega tlaka zdaj potrebujemo le še programsko opremo. Postopek nameščanja, če je v Raspberry Pi že nameščen programski jezik Python, je naslednji:

sudo apt-get install python-setuptools

```
wget https://files.pythonhosted.org/packa-
ges/6a/06/
80a6928e5cbfd40c77c08e06ae9975c2a
50109586ce66435bd8166ce6bb3
/smbus2-0.3.0.tar.gz
```

Zdaj arhiv razširimo v imenik smbus2-0.3.0, nato pa programsko knjižnico s podporo za I2C vodilo za Pyhon še namestimo.

```
chdir smbus2-0.3.0
sudo python setup.py install
```

Pred ponovnim zagonom računalnika v nastavitvah omogočimo I2C vodilo in v datoteko /boot/config.txt dodamo naslednji vrstici:

```
dtparam=i2c_arm=on
dtparam=i2c arm baudrate=10000
```

S tem povemo, da želimo uporabljati I2C gonilnik pri hitrosti prenosa podatkov 10 kHz. Če BME280 senzor povežemo s kratkim kablom, lahko izberemo tudi precej večje hitrosti od 100 kHz do največ 3,4 MHz.

Zdaj potrebujemo še program za merjenje vremenskih parametrov, ki ga prenesemo na spodnji način in nato zaženemo:

wget -0 bme280.py http://bit.ly/bme280py

Opcijsko lahko pravilno delovanje povezave senzorja z računalnikom preverimo s programskim orodjem I2Cdetect, ki ga najprej namestimo z ukazom:

```
sudo apt-get install -y i2c-tools
Nato pa zaženemo:
sudo i2cdetect -y 1
```

S tem dobimo spisek vseh delujočih I2C naprav na vodilu.



Slika 4: Modul za merjenje temperature med vgradnjo v ohišje

Poglejmo še, kako izgleda program za branej z BME280 izmerjenih vrednosti temperature, relativna vlažnosti in zračnega tlaka v Pythonu:

```
import bme280
temperature,pressure,humidity = bme280.re-
adBME280All()
print "Temperature : ", temperature, "C"
print "Pressure : ", pressure, "hPa"
print "Humidity : ", humidity, "%"
```

Most med Wi-Fi in ožičenim Internetom

V petem nadaljevanju v SE302 smo omenili LILYGO TTGO T-Ethernet-POE modul z vgrajenim Ethernet LAN8720A vmesnikom za povezavo v ožičena omrežja. Obenem modul še vedno omogoča tudi brezžično Wi-Fi povezljivost. Če nanj namestimo programsko opremo, ki jo najdemo na spletni strani PC USB Projects (WIFI2ETH.ZIP), ga lahko spremenimo v most med ožičenim Ethernet dostopom do Interneta in brezžičnim Wi-Fi omrežjem. Za prevajanje programske kode potrebujemo IDF razvojno okolje, obenem je osnovni primer ETH2AP je na voljo tudi na spletnih straneh Espressif Systems v okviru tega razvojnega okolja. Ker pa je namestitev IDF za mnoge nekoliko zapletena, je IDF na voljo tudi v obliki navideznega računalnika z zastonjskim operacijskim sistemom Ubuntu (denimo: ubuntu1804.ova), ki ga lahko s pomočjo ustrezne programske opreme namestimo v poljuben osebni računalnik ali gostiteljski strežnik. Sam raje IDF uporabljam

neposredno v operacijskem sistemu Windows, saj tako deluje hitreje, hkrati pa je prenos nove ugnezdene programske opreme enostavnejši, saj imajo navidezni računalniki pogosto zelo omejen dostop do sistemskih sredstev, kot so USB vrata, in je zato težje zagotoviti sinhronizacijo RS232 vmesnikov, ki je potrebna pred nalaganjem nove vgrajene programske opreme.

Če ima vremenska postaja dostop do Interneta, lahko točen čas prenese iz javnih časovnih spletnih strežnikov, zato nam ure in datuma ob zagonu ni potrebno prenašati iz osebnega računalnika ali pametnega telefona, ali celo ročno vnašati prek enkoderja.

Oddaljeni merilni moduli

Vremenska postaja ne meri vremenskih parametrov samo v prostoru, v katerem je nameščena, temveč zbira tudi podatke iz oddaljenih modulov, ki so nameščeni v drugih prostorih stanovanja pa tudi zunanjo temperaturo, če oddaljeni modul namestimo na zunanji del stavbe. Oddaljeni moduli ne potrebujejo grafičnega prikazovalnika, saj delujejo kot Wi-Fi dostopne točke, s katerimi se poveže osnovni modul vremenske postaje 3.0, nato pa iz njih periodično zajema podatke. Osnovni modul obenem zazna morebitno porušitev Wi-Fi povezave in jo poskuša samodejno vzpostaviti nazaj.

Osnova oddaljenega merilnega modula je ESP32 modul. Slučajno sem imel pri roki ESP32-ROVER modul s priključkom za zunanjo anteno, za uporabo PCB antene pa moramo prestaviti 0-Ohmski mostiček. Dodal sem mu še ADT7410 modul



Slika 5: Električna shema oddaljenega modula za merjenje temperature

za merjenje temperatur, ki sem ga prav tako imel pri roki, seveda pa bi mu lahko dodal katerikoli drugi modul. Glede na to, da se vlažnost v večini prostorov stanovanja ne razlikuje dosti, zračni pritisk pa sploh ne, je taka odločitev razumna, je smiselno po en oddaljeni merilni modul namestiti v vsak prostor stanovanja, saj je tako najlažje zagotoviti zadostno ogrevanje ali poleti hlajenje vseh prostorov.

Za napajanje oddaljenega modula je mogoče uporabiti tudi do 12 V napajalnik iztrošenega mobilnega telefona, saj ima modul vgrajen LM317T napetostni regulator, ki je z uporoma nastavljen tako, da napetost zniža je 3,3 V. Oddaljeni modul se od osnovnega modula razlikuje predvsem po programski opremi in po tem, da ne potrebuje lastne mikro SD kartice za shranjevanje vrednosti vremenskih parametrov.



Slika 6: Lepljenje ESP32-ROVER modula na dvostransko prototipno vezje

Sestavljanje oddaljenega modula

Za oddaljeni modul bi lahko uporabil katerokoli razvojno ploščico z dovolj stebrastimi kontakti in vgrajenim mostom iz USB na RS232, vendar sem namesto tega raje uporabil vezje za prototipiranje, ESP32-ROVER modul in žico za povezovanje kontaktov na protoboardih. Če ESP32-ROVER modul s kovinskim pokrovčkom prilepimo na tiskano vezje tako, da s tem ne preprečimo dostopa do antenskega priključka, je izpostavljena zadnja stran modula z oznakami posameznih priključkov. Zato je z uporabo običajnega spajkalnika, žico za povezovanje kontaktov na protoboardih in povečevalnega stekla mogoče sorazmerno enostavno neposredno povezati kontakte na ESP32 modul.

Zagotavljanje združljivosti senzorjev na strojnem nivoju

Vsa vremenska digitalna tipala lahko povežemo z vodilom I2C, pri čemer je smiselna enaka razporeditev priključkov na konektorjih vseh modulov vremenske postaje. Tako so vsi povezovalni kabli enaki, tipala pa lahko izmenjujemo. Tako je mogoče senzorje medsebojno izmenjevati, hkrati pa jih je mogoče na posamezen modul vremenske postaje vezati tudi po več. Smiselno je, da sta signala SCL in SDA čimbolj narazen, saj lahko tako podatke prenašamo pri višjih taktih. K sreči lahko mednju vstavimo signala za napajanje, maso in napajalno napetost. Za svoj projekt sem izbral 1-vrstični konektor s 6 priključki:

Čeprav moramo modula BMP280 in BME280 napajati s 3,3
 V, pa sta v razvojno ploščico GYBMEP vgrajena tudi napetostni regulator in ustrezna napetostna prilagoditev, kar omogoča delovanje pri razponu napajalne napetosti med 3,3 V in 5 V. Zato jih lahko uporabljamo tako s 3,3 V kot 5 V mikrokontrolerji.

4-inčni Waveshare TFT z zaslonom na dotik Arduino ščit

ILI9486L krmilnik prikazovalnika z Intelovo procesorsko arhitekturo 8080 ima 16-bitno podatkovno vodilo, vendar omogoča tudi 8, 9 in 18-bitna podatkovna vodila, prav tako pa tudi SPI vodili s tremi ali štirimi priključki. Vendar naj bi cenejši TFT z ILI9486 krmilnikom, uporabljen v Waveshare TFT z zaslonom na dotik Arduino ščitu omogočal zgolj paralelno komunikacijo. Zato so pri Waveshare iz štirih čipov sami izdelali pretvornik iz SPI vodila na vzporedno 16-bitno vodilo, ki ima: en 12-bitni števec (74HC4040), dva 8-bitna pomikalna registra s pomnilnikom (74HC4094) in en negator (74HC04D). Pretvornik SPID16-bitno vodilo obenem deluje tudi kot prilagodilnik napetostnih nivojev, vendar pa, podobno kot implementacija SPI v ILI9486, ne omogoča povratne komunikacije TFT glavni mikrokontroler. Sliko na prikazovalniku zato lahko urejamo le v medpomnilniškem RAMu glavnega mikrokontrolerja. S sreči imamo pri ESP32-ROVER modulih vgrajen tudi dodatni RAM čip s serijskim dostopom, v katerem lahko hranimo zaslonsko sliko, medtem ko ima Raspberry Pi pomnilnika v vsakem pogledu več kot dovolj, saj prek (mikro) HDMI vmesnika omogoča bistveno večje ločljivosti kot 320x480.

Po drugi strani, XPT2046 krmilnik zaslona na dotik že v osnovi komunicira prek 1-bitnega vodila SPI, prav tako pa tudi ležišče za kartico mikro SD. SCLK, MOSI in MISO priključki vseh čipov Arduino ščita so združeni in so dostopni prek dveh rumenih enovrstičnih konektorjev, na katerih so tudi chip select signali za ILI9486L krmilnik, XPT2046 krmilnik in SD kartico. Napaja-nje Arduino ščita (med 3,3 V in 5 V) je izvedeno prek rdečega konektorja, na katerem do dostopen tudi signal reset. Modri konektor ni povezan je omogoča samo boljšo pritrditev ščita

PROGRAMIRANJE

na Adruino klasične Arduno ali STM32 razvojne plošče. Omenimo še AMS1 napetostni regulator z nizkim padcem napetosti, ki napajalno napetost zniža na 3,3 V. Zato lahko modul napajamo tudi s 5 V in krati z njim komuniciramo s 5 V mikrokontrolerjem.

Kako se lotiti programiranja?

Waveshare je na svojih spletnih straneh (waveshare.com/4inch-TFT-Touch-Shield.htm) objavil sorazmerno podrobno dokumentacijo pa tudi programske knjižnice za Arduino združljive mikrokontrolerske module, kakor tudi za module

z STM32 mikrokontrolerji. Zato moramo v programski kodi nastaviti predvsem priključke, prek katerih je povezan ščit. Pri tem za dostop do prikazovalnika, zaslona in mikro SD kartice uporabljamo ločene gonilnike. Za mikro SD kartico lahko neposredno uporabimo tudi splošno podporo v Arduino razvojnem okolju.

Priloženi primeri zadoščajo za preizkus delovanja zaslona, na katerem lahko prikažemo tudi barvne fotografije. Slednje pojasni, zakaj so se snovalci zaslona odločili dodati še ležišče za mikro SD kartico.

Prihodnjič

Prihodnjič bomo vremenski postaji dodali še funkcionalnosti spletnega radija, predvajalnika glasbe in ure budilke.



https://svet-el.si





32 SVET ELEKTRONIKE



Grafika v okenskih programih

Avtor: Boštjan Šuhel

E-pošta: bostjan.suhel@gmail.com

Danes nas zanima bližnjica Python grafika. Gre za sklop vaj , ki pokažejo nekatere zmožnosti grafike v okenskih programih.

Trenutni izgled namizja slovenske izdaje(Vir:02). Grafični uporabniški vmesnik(Vir:01) je okolje za delovanje okenskih programov.

Python grafika

Bližnjica Python grafika odpre terminalno okno, kjer v naslovu dobimo navodilo za ukaz ./naredi 01 do 08. Kar pomeni, da imamo na voljo 8 vaj iz grafike. Naučili se bomo kako uporabljati grafične elemente v okenskih programih.

Ukaz '/naredi 01 požene okenski program črta. Vidimo tudi izpis programa 01, ampak samo del, ki ga lahko sami spreminjamo. Okenski programi so, kot smo si pogledali v prejšnjem nadaljevanju, sestavljeni iz programa in razreda, ki opisuje samo okno. Ta razred uporabimo v našem programu z uporabo ukaza import.



Slika 1: Namizje slovenske izdaje



 Python grafika '/naredi 01 do 08'
 Low voltage warning

 Datoteka Uredi Zavihki Pomoč
 Please check your power supply

 class okno(QtWidgets.QWidget, Ui_Form):
 def _____init__(self, 'args, **kwargs):
 super(', '_____init__'):

 super(')._____init__('args, **kwargs):
 self.setupUs(self)
 self.setupUs(self)

 self.setupUs(self)
 self.exent(self, event):
 Crta
 * * *

 (tkidgets.QWidget, paintEvent(sets stikar = Qtou.QPainte(self)
 sikar = qetuu.QPainter(self)
 Sikar = getuu.QPainter(self)

 stikar.setPen(Qtou.QPainter(self)
 sikar.drawLine(self.crta)
 if ______name__ == '___main__':
 import sys

 if ______name__ == '____main__':
 import sys
 app.exec()
 zvajanje programa crta.py
 libEGit warning: DRI2: failed to aur

 libEGit warning: DRI2: failed to aur
 gt.qpa.xbc; Qkobconnection: xCB er
 sinor code: 20
 sinor code: 20

V2 🕑

Slika 2: Bližnjica Python grafika

Slika 3: Program 01 – Črta

12 uporabnik12 (uporabnik12) - VNC Viewer

S Python gr... Črta

Ko zapremo okenski program se v terminalu pojavi vprašanje ali ponastavim omizje in dodam vse datoteke vaje 01 na namizje. Če pritisnemo samo enter se ne zgodi nič, če pritisnemo d in enter se ponastavi celotno namizje in skopira vse datoteke, ki so potrebne za izvedbo vaje 01 na namizje. Tak način se je v praksi pokazal kot najučinkovitejši.

Ko imamo vse datoteke na namizju se začne pravo delo. Program crta.py je glavni program, tega lahko spremenimo. Crta_qt5.py je razred, ki opisuje naše okno. Tega razreda ne spreminjamo mi, ampak za spremembo okna uporabimo orodje Qt 5 Designer.

Crta.py

Sedaj imamo na namizju samo datoteke, ki pripadajo projektu črta. Na namizju imamo bližnjico Terminal, ki nas prestavi v mapo Desktop (namizje). Tu je že pripravljena skriptna datoteka »naredi«. Le-ta je prilagojena tako, da moramo samo odtipkati ./naredi. Na novo se prevede crta_qt5.ui v crta_qt5.py.

11:47

PROGRAMIRANJE

Crta_qt5.py je razred ki opisuje izgled (torej lastnosti) našega okna. Crta_qt5. py razred se potem uvozi (import) v crta. py. Program crta.py izriše črto na prazno okno :). Torej najprej s pomočjo Python bližnjic osnove, grafika, okna, opencv in gpio izberemo projekt in ga prekopiramo na namizje. Le-tu se lahko v projekt poglobimo.

Ker v tej vaji ne bomo spreminjali lastnosti okna, lahko samo kliknemo na crta. py. Pred nastavljeno se odpre v urejevalniku za python programe Thonny (Vir:03). Thonny je IDE za python (kakor recimo mednarodno predstavljena slovenska izdaja, ki vse opravlja v spletnem okolju). Z ukazom Run poženemo program crta.py in prikaže se črta :). Podobna logika je pri vseh okenskih in navadnih python programih, ki jih učimo na slovenski izdaji.



Slika 4: Kopiranje vaje na namizje



Slika 5: Ponastavljeno namizje s prekopiranimi datotekami



Slika 6: crta.py



Slika 7: Thonny



Slika 8: Boljša črta

Boljša crta.py

Sedaj je nastopil trenutek, ko lahko napišem svoj prvi okenski program, ki uporablja grafiko. Ugotavljanje razlike med prvotnim programom in programom, ki izpiše 8 vodoravnih črt



Slika 9: crta2.pv

Narisi	~	^	×

Slika 10: Narisi

	Zbrisi	~	^	×
Nariši				
Zbriši				



eno pod drugo prepuščam bralcem. Vidimo, pa, da je pisanje okenskih programov z uporabo razredov lahko zabavno.

Crta2.pv

Sledimo že opisanemu postopku in na povezavi Python grafika izberemo ./naredi 04

Odpremo in poženemo v Thonny-ju crta2.py. Tako lahko naredimo, ker ne bomo spreminjali lastnosti okna.

Vidimo, da lahko s pomočjo gumba Nariši, črto narišemo in s pomočjo gumba Zbriši, le-to zbrišemo :). Upam, da ste si že namestili slovensko izdajo, ker je polna primerov v python-u, c-ju, php-ju, html-ju, css-u in javascript-u. Gre namreč za

masovno sklicevanje na pretekle izkušnje in pridobljeno znanje. Da programiranje postane res zabavno in se začnejo rojevati res uporabni projekti, je potrebno kar veliko treninga :) . Za tiste, ki ste sledili do sem, vam to ne bo težka naslednja naloga.

Spremeni program crta2.py tako, da se bo izpisalo in izbrisalo osem vodoravnih črt, podobno, kakor v spremenjenem programu crta.py ? Hmmmm :)

Naslednjič

Naslednjič nas čaka zopet kratek izlet v C okolje. Raziskali bomo same meje realno časovnega delovanja v operacijskem sistemu brez nedovoljenih poseganj v samo jedro operacijskega sistema.

Videli in dokazali bomo, da lahko pokrivamo dogodke do nekako 100µs. S C-jem bomo iskali same meje, python ne zaostaja veliko, pa vendarle zaostaja, zato so veliki projekti še vedno vsi pisani v C-ju, ki je naravni jezik linux-a. Ali če hočete, morda je python namenjen produkcijskemu , C (in strojni jezik) pa razvojnemu delu (ne jemljite tega preveč dobesedno prosim). Še morda nekaj o Viru 04. Gre za slovensko izdajo postavljeno pri meni doma. Za to pošastjo se skriva rpi4 z 2Gbyte rama in priključen preko USB3.0 na SSD 0.5Tbyte. Pri IP številkah je problem, ker eduroam omrežje dovoljuje samo domene, zato vir 04 če ste direktno ali indirektno priklopljeni na to omrežje ne bo deloval. Rešitev je domače, arnes ali mobilno omrežje npr. Si pa na viru 04 vedno lahko pogledate v vaje in delujoče projekte. Pri delujočih projektih govorimo o orkestraciji razvojnih tehnologij. Postavljena je prva vzorčna učilnica nove mehatronike, ki med drugim uporablja vaje iz slovenske izdaje. Ključno je delo z MS teamsi in z nekaterimi portali, ki uporabljajo google prijavo. Znanje učencev na pametnih telefonih, uporabi pisarne je odlično. Poizkusi poenotenja formatov zapisa na glavnih platformah (MS, Linux, Anroid, Apple) je stalen proces. Raspberry je med tem prodan v 40M primerkih!! Imamo nov OS z imenom raspberry OS. Je prva strojna oprema na področju kartičnih računalnikov in prva evropska strojna oprema po spektrumu po skoraj 40 letih!!!. Prodaja se po celem svetu, je vodilna in ima evropske korenine. Vzporedno se v temeljih spreminja ožičenje naprav na splošno. Ukinjajo se signalne povezave, naprave so stalno pod napetostjo, krmilni elementi se selijo v podsklope, poraba se zmanjšuje za nekaj 10x. Slabo se piše kontaktorjem, relejem in mehanskim stikalom. Imajo hudo konkurenco v polprevodniških stikalih. Moderen razvoj se začne z uporabo medsebojno povezanih modulih. Sledi združevanje modulov na eno tiskano vezje. V ozadju se godi stalen razvoj na siliciju :). Brez panike vse skupaj se poenostavlja, potreben je samo pravi pogled na vse skupaj.

Viri:

- https://en.wikipedia.org/wiki/Graphical_user_interface
- http://77.38.33.244/indexorodja.php
- https://thonny.org/
- http://77.38.33.244/ •

https://svet-el.si



PROGRAMIRANJE



Krmiljenje LED-ice z govorom

Hackster Inc.

Ali ni čudovito nadzorovati luči samo s svojim govorom? Uporabite Google pomočnika, IFTTT in platformo Bolt IoT za ustvarjanje glasovnega nadzora.

Google pomočnik me je navduševal že v času, ko je bil na voljo le na najnovejših pametnih telefonih. Ko sem odkril, da je mogoče IoT projekt ustvariti z Google pomočnikom, sem se odločil, da ga nadaljujem. Tudi dajanje glasovnih ukazov stroju je precej učinkovita tehnologija, ki prihrani čas.

Stvari uporabljene v tem projektu:

- Hardverske komponente
- Bolt IoT Bolt WiFi Modul
- LED (navadna)

Softverske aplikacije in spletne storitve:

- Bolt IoT Bolt Cloud
- IFTTT WebHooks
- IFTTT Google Assistant

Nastavitve hardvera

Povežite pozitivni priključek LED-ice (daljši priključek) na pin 0. Povežite negativni priključek LED-ice (krajši priključek) na maso.

Povežite Bolt WiFi modul na napajanje. Prepričajte se, da je modul povezan na internet.

Uporaba Bolt API Ključa in Bolt Device ID

Logirajte se na "cloud.boltiot.com".

Pojdite na »devices tab« v levem kotu in poiščite ID številko svojega modula. Številka bo izgledala približno takole BOLTXXXXXXXX.

Pojdite na API zavihek v levem kotu. Izberite »enable and copy this key for further use«.

Nastavitev GPIO krmilnih ukazov

GPIO je kratica za splošni vhodi oziroma izhod. Bolt ima 5 digitalnih GPIO, in sicer 0, 1, 2, 3 in 4. Uporabljajo se za povezovanje vhodnih in izhodnih naprav na



Slika 1

ploščo. Ti priključki delujejo pri napetosti 3,3 V na Boltu.

Za nadzor naše LED-ice moramo nastaviti URL, ki bo uporabljal naš API ključ, ID naprave in GPIO priključek za povezavo z Bolt oblakom.

V tem primeru uporabljamo funkcijo digitalnega pisanja, saj je LED izhodna naprava. Funkcija digitalWrite vsebuje dva parametra, ki sta pin in stanje. Uporablja se za zapis HIGH ali LOW vrednosti na digitalni priključek.

Privzeta struktura ukaza bo izgledala takole:

 https://cloud.boltiot.com/remote/API_KEY/digitalWrite?pin=PIN_NUMBER&state=HIGH/LOW&deviceName=DEVI-CE_ID

€ ⇒ C	doud.boltiot.com/api		\$) Q (B 🕸 🚺
-1171-	API Credentials		O Droches Paint
Denter Products Anti-	 Manager API Kay API Documentations 	Cenerate Key Cites here for enabling the key Disable Disconsectory Baseling the set	Citch here for coung Lift key
		API Request Builder	
Or	API Teb	With this API builder, you can easily build API requests and also test them directly via the Bolt Cloud. Select the device you easil to reask the API request to -	



Parametri, ki jih bomo zamenjali v našem dejanskem ukazu, so:

- API_KEY: Omogočili in kopirali smo ga v prejšnjem razdelkıı.
- PIN_NUMBER: GPIO priključek, na katerega smo povezali našo LED-ico, je '0'.
- HIGH/LOW: V ta ukaz moramo vnesti 'HIGH' ali 'LOW'. HIGH bo prižgal luč, LOW pa jo bo ugasnil.
- DEVICE_ID: ID našega modula, ki smo ga zapisali v prejšnjem razdelku, tj. BOLTXXXXXXX.

Za vklop luči bom v 'state' vtipkal HIGH. In obratno, za izklop luči bom v 'state' vtipkal LOW.

Nastavitev aplikacije

Uporablja se programska oprema, imenovana IFTTT. To pomeni 'Če to, potem to'. Ta programska oprema pomaga pri povezovanju različnih aplikacij in naprav.

- Pojdite na 'ifttt.com' in kliknite na ustvari. Prijavite se s svojim Gmail računom.
- Kliknite na "If this add", slika 4.
- Poiščite »Google Assistant« in ga izberite.
- Kliknite na 'Say a simple phrase', slika 5.
- Določite stavke, ki jih želite uporabiti pri sprožitvi ukaza. Zahteva tri stavke. Za prižig luči sem uporabil 'Turn on the lights', 'Please turn the lights on' in 'Lights on'.
- Vnesite pomočnikov odgovor. V tem primeru 'Switching ON the lights', slika 6.
- Kliknite na 'create trigger'.
- Kliknite na "Then That add", slika 7.
- Izberite 'Webhooks', slika 8 in 9.
- Vnesite nadzorni ukaz GPIO (URL).
- Izberite GET način. GET je običajna HTTP metoda, ki se uporablja za zahtevanje podatkov iz določenega vira.
- V vrsti vsebine izberite 'application/ • json', slika 10.
- Kliknite na 'create action' na dnu te strani.
- Nato kliknite na "continue" in "finish".
- Za ugašanje luči sem uporabil 'Switch off the lights', 'Lights off', and 'Please turn off the lights.
- Vnesite pomočnikov odgovor. V tem primeru 'Switching OFF the lights', slika 11.
- Kliknite na 'create trigger'.
- Kliknite na "Then That add", slika 12.
- Izberite 'Webhooks', slika 13, 14.
- Vnesite krmilni GPIO ukaz (URL).
- Izberite GET način. GET je običajna HTTP metoda, ki se uporablja za zahtevanje podatkov iz določenega vira.



Slika 3

- V vrsti vsebine izberite 'application/json , slika 15
- Kliknite na 'create action' na dnu te strani.
- Nato kliknite na "continue' in 'finish".
- lf This Google Assistant



Slike od 4-15

(Bek)

0

PROGRAMIRANJE



Slika 16: LED-ica je vklopljena.

Zaključek

V tem projektu sem uporabil glas za prižiganje luči. Uporaba glasu kot ukaza je lahko zelo učinkovita in priročna za posameznike. Funkcija glasovnega nadzora zmanjša tiste neprijetne trenutke, ki so potrebni za vklop in izklop naprave. Uporaba programske opreme in aplikacij, kot so IFTTT, Google pomočnik in webhooks, je zmanjšala kompleksnost



Slika 17: LED-ica je izklopljena.

kode. Dajanje glasovnih ukazov stroju prihrani čas v tem hitro delujočem svetu.

Povzeto po:

https://www.hackster.io/drishtiipant/voice
 -controlled-led-8ce4e6

https://hackster.io



CírcuitDigest

Arduino digitalni termometer z uporabo MAX30205, senzorja človeške telesne temperature

Circuit Digest

Za medicinsko ali klinično uporabo je merjenje človeške telesne temperature pomemben parameter za ugotavljanje zdravstvenega stanja vsakega posameznika.

Obstaja veliko načinov za merjenje temperature, vendar ni vse dovolj natančno, da bi ustrezalo specifikacijam klinične termometrije. Temperaturni senzor MAX30205 je posebej zasnovan za to aplikacijo. Upoštevajte, da ta senzor ni brezkontaktni temperaturni senzor. Če iščete brezkontaktno IR merjenje temperature, si oglejte termometer MLX90614.

V tem projektu bomo povezali senzor človeške telesne temperature MAX30205, ki ga je mogoče enostavno namestiti na fitnes trak ali uporabiti v medicinske namene. Uporabili bomo Arduino Nano kot glavno mikrokontrolersko enoto in uporabljali tudi 7-segmentne displeje za prikaz zaznane temperature. Ko enkrat veste, kako uporabljati senzor, ga lahko



uporabite v kateri koli aplikaciji, ki vam je všeč, lahko si ogledate tudi projekt pametne ure narejene z Arduino [1], ki se v kombinaciji z MAX30205 lahko uporablja za spremljanje temperature posameznikov.

Komponente, ki jih potrebujemo za izdelavo

- Arduino NANO
- 7-segmentni displeji s skupno katodo 3 kos
- 74HC595 3 kos
- 680R upor 24 kos
- MAX30205 modul
- 5V napajanje
- Razvojna plošča
- Veliko priključnih žic
- Arduino IDE
- Kabel mikro-USB



Povezava MAX30205 z Arduinom

Celotna shema vezja za povezavo Arduino s senzorjem telesne temperature MAX30205 je prikazana na sliki 1. Vezje je zelo preprosto, a ker smo uporabili 7-segmentne displeje, je videti nekoliko zapleteno. 7-segmentni displeji z Arduinom so odličen način, kako prikazati vrednosti neke spremenljivke na veliko in svetlo z zelo nizkimi stroški. Seveda pa lahko te vrednosti prikažete tudi na OLED ali LCD, če želite.

Arduino Nano je povezan s tremi 74HC595. Trije 74HC595 so kaskadno vezani zato, da se prihranijo dodatni izhodni priključki na Arduino Nano za povezavo treh 7-segmentnih displejev. 74HC595 z Arduinom sem uporabil še v številnih drugih projektih [2], kot so Arduino Clock, LED Board Display, Arduino kača igra, itd.

Tiskano vezje modula MAX30205 zahteva dodatne »pull-up« upore, saj komunicira preko I2C protokola. Vendar pa nekaj vrst modulov ne potrebuje dodatnih »pull-up« uporov, saj so že





Slika 1: Shema vezja termometra

vgrajeni na modulu. Zato se je treba prepričati, ali ima modul vgrajene »pull-up« upore ali pa jih je treba dodati posebej. Modul, ki je uporabljen v tem projektu, ima že vgrajene »pull-up« upore znotraj modula.

Povezava Arduino s senzorjem telesne temperature MAX30205 Senzor, ki se tukaj uporablja, je MAX30205 podjetja Maxim (zdaj Analog Devices). Temperaturni senzor MAX30205 meri temperaturo z natančnostjo 0,1°C (37°C do 39°C). Senzor se poveže z mikrokontrolerjem preko I2C vodila.

Plošča modula lahko deluje z napajalno napetostjo 5 ali 3,3 V. Vendar je plošča konfigurirana za uporabo z 5V delovno napetostjo. Vključuje tudi prilagodilnik logičnih nivojev, saj senzor sam podpira največ 3,3 V za napajanje ali podatkovno komunikacijo.

Na izhodu se nahajajo trije 74HC595, 8-bitni premični registri, ki se uporabljajo za povezovanje treh 7-segmentnih displejev z Arduino NANO. Diagram priključkov lahko vidite na sliki 2.

PIN DESCRIPTIO	N
----------------	---

PIN No	SYMBOL	NAME AND FUNCTION
1, 2, 3, 4, 5, 6, 7, 15	QA to QH	Data Outputs
9	QH'	Serial Data Outputs
10	SCLR	Shift Register Clear Input
11	SCK	Shift Register Clock Input
13	G	Output Enable Input
14	SI	Serial Data Input
12	RCK	Storage Register Clock
8	GND	Ground (0V)
16	Vcc	Positive Supply Voltage

Slika 2: Diagram priključkov 74HC595

QA do QH so priključki za izhod podatkov, ki so povezani s 7-segmentnimi displeji. Ker so trije 74HC595 kaskadno povezani skupaj, bo podatkovni vhodni pin (PIN14) prvega premičnega registra povezan z Arduino NANO, izhodni priključek serijskih podatkov pa bo zagotovil podatke v naslednji premični register. Ta serijska podatkovna povezava se bo nadaljevala do tretjega 74HC595.

Programiranje MAX30205 v Arduino okolju

Celoten program za to vadnico je na voljo na povezavi na dnu tega članka. Razlaga kode je naslednja. Najprej vključimo standardno naslovno datoteko Arduino I2C knjižnice.

#include <Wire.h>

Zgornja vrstica bo vključevala knjižnico, ki jo prispeva Arduino iz protocentrala. Ta knjižnica ima pomembne funkcije za ko-

munikacijo s senzorjem MAX30205. Knjižnica je vzeta z GitHub povezave [3].

Naslednji dve vrstici sta pomembni za nastavitev parametrov. Spodnja vrstica bo zagotovila temperaturo v Fahrenheitih, če je nastavljena na »True«. Za prikaz rezultata v stopinjah Celzija je treba vrstico nastaviti na »False«.

```
const bool fahrenheittemp = true;
    // I'm showing the temperature
    in Fahrenheit, If you want to
    show the temperature in Celsius
    the make this variable false.
```

Spodnjo vrstico je treba konfigurirati, če se v strojni opremi uporabljajo 7-segmentni zasloni s skupno katodo. Naj bo nastavitev »False«, če se uporablja skupna anoda.

const bool commonCathode = true:

<pre>// I'm using common Cathode 7segment if you use common Anode then change the value into false.</pre>			
<pre>const byte digit_pattern[17] =</pre>			
{ // 74HC595 Outpin Connection with 7segment display.			
// 00 01 02 03 04 05 06 07			
// a b c d e f q DP			
0b11111100, // 0			
0b01100000, // 1			
0b11011010, // 2			
0b11110010, // 3			
0b01100110, // 4			
0b10110110, // 5			
0b10111110, // 6			
0b11100000, // 7			
0b11111110, // 8			
0b11110110, // 9			
0b11101110, // A			
0b00111110, // b			
0b00011010, // C			
0b01111010, // d			
0b10011110, // E			
0b10001110, // F			
0b0000001 // .			
};			

Zgornji niz se uporablja za shranjevanje vzorca številk za 7-segmentne zaslone.

V funkciji nastavitve se po nastavitvi priključkov 74HC595 inicializira protokol I2C in odčitavanje temperaturnega senzorja.

```
void setup() {
   // put your setup code here, to run once:
   // set the serial port at 9600
   Serial.begin(9600);
   delay(1000);
```

```
// set the 74HC595 Control pin as output
pinMode(latchPin, OUTPUT); //ST_CP of
74HC595
pinMode(clkPin, OUTPUT); //SH_CP of
74HC595
pinMode(dtPin, OUTPUT); //DS of 74HC595
// initialize I2C Libs
Wire.begin();
// start MAX30205 temperature read in con-
tinuos mode, active mode
tempSensor.begin();
}
```

V glavni zanki temperaturo bere funkcija tempSensor.getTemperature() in jo shrani v float spremenljivko z imenom temp. Po tem, če je izbran Fahrenheit temperaturni način, se podatki pretvorijo iz stopinj Celzija v stopinje Fahrenheit. Nato se tri številke iz pretvorjenih zaznanih temperaturnih podatkov nadalje ločijo v tri posamezne številke oziroma digite. Za to se uporabljajo vrstice kode (// separate 3 digits...).

Sedaj ločene tri digite pošljemo na 7-segmentne displeje s pomočjo 74HC595 premičnih registrov. Ker se je LSB prvič prikazal v tretjem 7-segmentnem prikazovalniku prek tretjega 74HC595, se najprej prenese 3. številka. Da bi to naredili, se priključek Latch potegne nizko in podatki se posredujejo 74HC595 s funkcijo shiftOut();

Na enak način se pošljeta preostala dva digita na ustrezni 74HC595 in tako ostaneta dva 7-segmentna displeja. Po pošiljanju vseh podatkov se Latch priključek sprosti in potegne



visoko, da potrdi konec prenosa podatkov. Ustrezne kode si lahko ogledate v kodi (// display digits into 3,...).

```
// display digits into 3, 7segment display.
digitalWrite(latchPin, LOW);
if(commonCathode == true){
shiftOut(dtPin, clkPin, LSBFIRST, digit_
pattern[dispDigit3]);
shiftOut(dtPin, clkPin, LSBFIRST, digit
pattern[dispDigit2]|digit_pattern[16]);
                         // 1. (Digit+DP)
shiftOut(dtPin, clkPin, LSBFIRST, digit
pattern[dispDigit1]);
}else{
shiftOut(dtPin, clkPin, LSBFIRST, ~(digit
pattern[dispDigit3]));
shiftOut(dtPin, clkPin, LSBFIRST, ~(digit
pattern[dispDigit2] digit_pattern[16]));
                         // 1. (Digit+DP)
shiftOut(dtPin, clkPin, LSBFIRST, ~(digit
pattern[dispDigit1]));
}
digitalWrite(latchPin, HIGH);
```

Arduino merilnik telesne temperature – testiranje Vezje je sestavljeno iz dveh sklopov razvojnih plošč, kot lahko vidite na sliki. Ko položimo prst na senzor, senzor zazna temperaturo in vrednost temperature se prikaže na 7 segmentnem displeju, na sliki je prikazana vrednost 98,5*F.

Celotno delovanje projekta si lahko ogledate v videoposnetku na povezavi [4]. Upam, da boste uživali pri gradnji projekta in ste se naučili kaj koristnega. Če imate kakršna koli vprašanja, jih pustite v razdelku za komentarje ali uporabite naše forume.

Povzeto po:

 https://circuitdigest.com/microcontroller-projects/arduino -max30205-human-body-temperature-measurement

Viri:

- https://bit.ly/3HbApNw
- https://bit.ly/3qrKRKg
- https://bit.ly/3qWvHww
- https://bit.ly/3nciRJI

https://circuitdigest.com



VoiceTurn - glasovno vodene smerne luči za varnejšo vožnjo

Hackster Inc. Avtor: Alvaro Gonzalez-Vila

Uporabnike opozorite na smer, v katero boste ubrali, samo tako, da rečete »levo« ali »desno«

Dobra praksa, ki jo morate upoštevati, ko ste v prometu, je, da druge uporabnike na cesti opozorite na smer, katero boste ubrali, preden zavijete ali spremenite pas. Ta navada prispeva k bolj nemotenemu prometu in zmanjšuje nenadne premike neozaveščenih voznikov. Pravzaprav avtomobili, motorna kolesa, tovornjaki, avtobusi in večina vozil, ki si jih lahko zamislite, vključujejo nekakšne naprave za smernike.

Komponente, ki jih potrebujemo za izdelavo:

- Arduino Nano 33 BLE Sense
- WS2813 RGB LED trak vodoodporen
- Kateri koli RGB LED trak, ki se ga da naslavljati bo dober. Potrebna dolžina: 20 cm.
- Povezovalne žice, 22 AWG
- Plastično ravnilo, 30 cm
- Katero koli ravnilo ali plošča ustreznih dimenzij
- Audio / Video kabel s konektorjem 3,5mm 4 polna vtičnica, 3,5mm Stereo Phone Jack vtikač × 2
- Nosilec ure za na kolo
- Li Ion paket baterij 2.5Ah (USB)
- USB micro-B kabel 1,5m
- Kabelske vezice

Softverske aplikacije in spletne storitve:

- Edge Impulse Studio
- TensorFlow
- Arduino IDE

Ročna orodja in stroji:

- Spajkalnik
- Spajkalni Flux,
- Škarje
- Držalo za TIV

Ta projekt je bil med petimi svetovnimi izbranci in je bil predstavljen na spletnem mestu Eksperimenti z Googlom [1].

Čeprav so kolesa med najbolj ranljivimi vozili na cesti, običajno nimajo vgrajenih signalnih naprav. Kolesarji pravzaprav ne opozorijo na zavoj ali pa morajo spustiti roko z krmila, da dajo znak drugim voznikom. Ta poteza zmanjša stabilnost kolesarja in je morda vsi ne razumejo pravilno.

Na trgu je mogoče najti nekaj dodatnih smernikov za kolesa, vendar običajno zahtevajo, da voznik pritisne gumb, da jih aktivira, in ponovno pritisne, da jih izklopi, podobno kot pri motorjih. In res jih je enostavno pozabiti izklopiti. Če dvakrat premislite, je to nekaj, kar je vredno izboljšati.

Od tu se rodi VoiceTurn, koncept osvetlitve smernikov, ki je bil

sprva zasnovan za kolesa, ki bi ga lahko razširili tudi na druga vozila. Cilj tega projekta je uporabiti algoritem strojnega učenja, da bi majhen mikrokontroler naučil razumeti besedi »levo in »desno« in da bi ustrezno ukrepal tako, da prižge ustrezen smernik.

Plošča

Mikrokontrolerska plošča, ki bo uporabljena, je Arduino Nano 33 BLE Sense: cenovno ugodna plošča, ki vsebuje 32-bitni ARM[®] Cortex[™]-M4 CPU, ki deluje pri 64 MHz, kup vgrajenih senzorjev, vključno z digitalnim mikrofonom, in Bluetooth Low Energy (BLE) povezljivostjo.

Kar pa naredi to ploščo kot odličnega kandidata za ta projekt je možnost, da na njej izvajamo Edge Computing aplikacije z uporabo Tiny Machine Learning (TinyML). Skratka, potem ko ustvarite modele strojnega učenja s TensorFlow Lite, jih lahko preprosto naložite na ploščo z uporabo integriranega razvojnega okolja Arduino (IDE).

Za prepoznavanje govora na Edgeu ni treba poslati glasu v strežnik v oblaku za obdelavo, s čimer se zmanjša zakasnitev omrežja. Poleg tega deluje brez povezave, tako da ste lahko prepričani, da vaši smerniki ne bodo prenehali delovati, ko greste skozi predor. Nenazadnje ohranja zasebnost uporabnikov, saj vaš glas ni nikjer shranjen ali nikamor poslan.

Izučite model strojnega učenja

Model za prepoznavanje besed je bil ustvarjen z uporabo Edge Impulse [2] razvojne platforme za ugnezdeno strojno učenje, ki je osredotočena na zagotavljanje neverjetne uporabniške izkušnje (UX), odlične dokumentacije in odprtokodnih kompletov za razvoj programske opreme (SDK). Na njihovi spletni strani piše:

Edge Impulse je bil zasnovan za razvijalce programske opreme, inženirje in strokovnjake za področje za reševanje resničnih težav z uporabo strojnega učenja na robnih napravah brez doktorata iz strojnega učenja.

To pomeni, da lahko imate malo ali nič znanja o strojnem učenju in še vedno uspešno razvijate svoje aplikacije.

To vadnico [3] lahko uporabite kot izhodišče za klasifikacijo zvoka z Edge Impulse. Naslednji koraki bodo opisali, kako je bil ta projekt prilagojen posebnim potrebam VoiceTurn.

Prva stvar, ki jo morate storiti, je, da se prijavite na Edge Impulse, da ustvarite brezplačno naročnino za razvijalce. Po koraku potrditve računa se prijavite in ustvarite projekt. Pozval vas bo čarovnik, ki bo vprašal o vrsti projekta, ki ga želite ustvariti. Kliknite na Zvok.

Audio

Listen to what's happening around you to create voice interfaces, listen to keywords, detect audible events, or to hear what's happening around your device.
V naslednjem koraku lahko izbirate med tremi možnostmi. Prva je, da sami sestavite nabor zvočnih podatkov po meri tako, da povežete razvojno ploščo, ki podpira mikrofon. Ta postopek zahteva snemanje velike količine zvočnih podatkov, da bi dosegli sprejemljive rezultate, zato ga bomo za zdaj prezrli. Druga možnost je nalaganje obstoječega nabora zvočnih podatkov, tretja pa sledenje vadnici. Kliknite Pojdi na nalagalnik v okviru možnosti Uvozi obstoječe podatke, da nadaljujete z VoiceTurn.

Import existing data If you already have audio data in WAV file format, you can upload it to Edge Impulse through the web interface or using the Edge Impulse CLI. C. Go to the uploader

Zvočni nabor podatkov, ki ga bomo uporabili, je nabor podatkov Googlovih govornih ukazov [4], ki ga sestavlja 65.000 enosekundno dolgih izrekov s 30 kratkimi besedami, na tisoče različnih ljudi. S te povezave [5] lahko prenesete različico 2 tega nabora podatkov.

Lahko si predstavljamo, da bi bile za usposabljanje modela potrebne samo podmnožice zvočnih posnetkov, ki ustrezajo besedama levo in desno. Kot pa Pete Warden iz Google Brain navaja v članku [6], ki opisuje metode, uporabljene za zbiranje in vrednotenje nabora podatkov:

Ključna zahteva za opazovanje ključnih besed v resničnih izdelkih je razlikovanje med zvokom, ki vsebuje govor, in posnetki, ki ga ne vsebujejo.

Zato bomo uporabili tudi podmnožico zvočnih posnetkov, ki jih vsebuje mapa _background_noise_, da bi naš model obogatili z nekaj šuma v ozadju. Poleg tega bomo bazo šuma dopolnili z zvoki iz mape šuma iz vnaprej izdelanega nabora podatkov [7] za opazovanje ključnih besed, ki je na voljo kot del dokumentacije Edge Impulse.

Imeti nabor podatkov, ki vsebuje besede levo in desno ter nekaj šuma v ozadju, ni dovolj, saj moramo modelu zagotoviti tudi dodatne besede. Na ta način, če se sliši druga beseda, ne bo razvrščena kot levo ali desno, ampak bo šla v drugo kategorijo. Za to lahko izberemo naključno zbirko zvočnih posnetkov iz prvega nabora podatkov, ki smo ga prenesli. Upoštevajte le, da mora biti skupna količina dodatnih zvočnih posnetkov podobna skupni količini posnetkov vsake besede, ki vas zanima.

Ko so zvočni posnetki zbrani, jih naložite v svoj projekt Edge Impulse. Upoštevajte, da boste morali naložiti 4 različne nabore podatkov, od katerih vsak ustreza drugi kategoriji. Brskajte po datotekah, ročno vnesite oznake, kot so levo, desno, šum in druge, in se prepričajte, da je izbrano »Automatically split between training and testing«. To bo pustilo ob strani približno 20 % vzorcev, ki bodo uporabljeni za kasnejše testiranje modela. Kliknite Begin upload.



Kot lahko opazite, so vsi zvočni vzorci, ki ste jih naložili, na voljo za preverjanje in poslušanje. Poskrbite, da bodo vsi trajali eno sekundo. Če so daljši, kliknite na pike na desni strani vrstice z zvočnim vzorcem in kliknite na Split sample, pri čemer nastavite dolžino segmenta 1000 ms.

Posledično bi morali zdaj videti skupno trajanje vaših podatkov o usposabljanju in testiranju, kot tudi vaše podatke, razdeljene v štiri kategorije.



PROGRAMIRANJE

Prav tako lahko dvakrat preverite, ali je trajanje vaših testnih podatkov približno 20 % celotnega trajanja nabora podatkov.



Naslednji korak je oblikovanje impulza, ki je celoten niz operacij, ki se izvajajo na vhodnih glasovnih podatkih, dokler besede niso razvrščene. Kliknite Create Impulse v levem meniju. Naš impulz bo sestavljen iz vhodnega bloka za rezanje podatkov, bloka za obdelavo za njihovo predhodno obdelavo in učnega bloka, ki jih razvrsti v eno od štirih predhodno definiranih oznak.

Kliknite Add an input block in dodajte Time series data block, pri čemer nastavite velikost okna na 1000 ms. Nato kliknite Add a processing block in dodajte blok Audio Mel Frequency Cepstral Coefficients (MFCC), ki je primeren za podatke človeškega govora. Ta blok ustvari poenostavljeno obliko vhodnih podatkov, ki jih je lažje obdelati z naslednjim blokom. Na koncu kliknite Add a learning block in dodajte Classification (Keras) blok, ki je nevronska mreža (NN), ki izvaja klasifikacijo in zagotavlja izhod.



Možno je konfigurirati bloke za obdelavo in učenje. Kliknite na MFCC v levem meniju in videli boste vse parametre, povezane s stopnjo obdelave signala. V tem projektu bomo stvari poenostavili in zaupali privzetim parametrom tega bloka. Kliknite Generate features na vrhu in nato kliknite tipko Generate features, da ustvarite MFCC bloke, ki ustrezajo zvočnim oknom.

Ko je opravilo končano, boste pozvani k raziskovalcu funkcij, ki je 3D predstavitev vašega nabora podatkov. To orodje je uporabno za hitro preverjanje, ali so vaši vzorci lepo ločeni v kategorije, ki ste jih določili prej, tako da je vaš nabor podatkov primeren za strojno učenje. Na tej strani si lahko ogledate tudi oceno časa, ki ga bo faza obdelave signala podatkov (DSP) potrebovala za obdelavo vaših podatkov, kot tudi porabo RAM-a pri izvajanju v mikrokontrolerju.



Zdaj lahko kliknete na NN Classifier v levem meniju in začnete trenirati svojo nevronsko mrežo, nabor algoritmov, ki lahko prepoznajo vzorce v svojih učnih podatkih. Oglejte si ta videoposnetek [8] za hiter pregled o principu delovanja nevronskih mrež in nekaterih njihovih aplikacijah. Večino privzetih nastavitev nevronskega omrežja bomo pustili nespremenjeno, vendar bomo minimalno oceno zaupanja nekoliko povečali na 0,7.



To pomeni, da bodo med treningom veljavne samo napovedi z verjetnostjo zaupanja nad 70 %. Omogočite povečanje podatkov

in nastavite Add noise na High, da bo naše nevronsko omrežje bolj robustno v resničnih življenjskih scenarijih. Kliknite Start training na dnu strani. Ko je usposabljanje končano, boste videli natančnost modela, izračunano z uporabo podmnožice 20 % podatkov o usposabljanju, dodeljenih za preverjanje. Prav tako lahko preverite Confusion matrix, ki je tabela, ki prikazuje razmerje med pravilno in napačno razvrščenimi besedami in oceno učinkovitosti v napravi.

Zdaj lahko preizkusite model, ki ste ga pravkar izučili z novimi podatki. Arduino ploščo je mogoče povezati z Edge Impulse [9] za izvedbo razvrščanja podatkov v živo. Vendar bomo model preizkusili s testnimi podatki, ki smo jih pustili ob strani med korakom pridobivanja podatkov. Kliknite Model testing v levem meniju in nato Classify all test data. Prejeli boste povratne informacije o delovanju vašega modela. Poleg tega vam bo Raziskovalec funkcij omogočil, da preverite, kaj se je zgodilo z vzorci, ki niso bili pravilno razvrščeni, tako da jih lahko po potrebi ponovno označite ali jih premaknete nazaj na usposabljanje, da izboljšate svoj model.



Končno lahko zgradite knjižnico, ki vsebuje vaš model in je pripravljena za uporabo v mikrokontrolerju. Kliknite Deployment v levem meniju, izberite ustvarjanje knjižnice Arduino in pojdite na dno strani. Tukaj je mogoče omogočiti prevajalniku EON™, da poveča zmogljivost v napravi za ceno zmanjšanja natančnosti. Ker pa poraba pomnilnika ni previsoka za Arduino Nano 33 BLE Sense, lahko to možnost onemogočimo, da bo delovala z najvišjo možno natančnostjo. Na koncu pustite izbrano možnost Quantized (int8) in kliknite gumb Build, da prenesete .zip datoteko, ki vsebuje vašo knjižnico. Projekt VoiceTurn Edge Impulse [10] je javno dostopen, tako da ga lahko neposredno klonirate in delate na njem, če želite.

Preverite, kako so besede razvrščene

Arduino IDE lahko uporabite za namestitev knjižnice, zgrajene z Edge Impulse, na vašo ploščo. Če ga še niste namestili, prenesite najnovejšo različico s strani Arduino programske opreme [11]. Po tem boste morali dodati paket gonilnikov, ki podpira ploščo Arduino Nano 33 BLE Sense. Odprite Arduino IDE in kliknite Orodja > Plošča > Upravitelj plošč...



V iskalno polje vpišite nano 33 ble sense in namestite Arduino Mbed OS Nano Boards.

Arduino Mbed OS Nano Boards by Arduino Beards incluido in this package. Collins: Hala. Mora Info.	
	2.3.1 * Install

Zdaj je vaš IDE pripravljen za delo z vašo ploščo. Povežite Arduino Nano 33 BLE Sense z računalnikom in uporabite naslednje nastavitve:

- Kliknite Tools > Board > Arduino Mbed OS Nano Boards in izberite Arduino Nano 33 BLE kot svojo ploščo.
- Kliknite Tools > Port in izberite serijska vrata, na katera je povezana vaša plošča. To se bo razlikovalo glede na vaš OS in vaša določena vrata.

Če želite knjižnico VoiceTurn dodati v vaš Arduino IDE, kliknite Sketch > Include Library > Add.ZIP Library... in pojdite na pot, kjer ste shranili svojo .zip datoteko knjižnice. Po tem kliknite File > Examples > VoiceTurn_inferencing > nano_ble33_sense_microphone_continuous, da odprete program za glasovno sklepanje, ki ga ponuja Edge Impulse. Ta program lahko že prevedete in naložite na svojo ploščo s klikom na dve ikoni, ki se nahajata v zgornjem levem kotu Arduino IDE. Ko je naložen, kliknite ikono Serial Monitor v zgornjem desnem kotu in videli boste rezultat predhodno razvitega klasifikatorja strojnega učenja. Točnost programa lahko preizkusite tako, da izgovorite besede »levo« ali »desno« in preverite verjetnost, da ste znotraj ene ali druge skupine, ki jo izračuna program. Prav tako lahko poskusite izgovoriti druge besede in preverite, ali so pravilno razvrščene v drugo skupino.

Predictions (DSP: 189 ms., Classification: 8 ms., Anomaly: 0 ms.): Left: 0.03125 Right: 0.91016 noise: 0.01172 other: 0.04688

PROGRAMIRANJE

Če želite izvedeti več o neprekinjenem vzorčenju zvoka, si oglejte to vadnico [12] v dokumentaciji Edge Impulse.

TensorFlow Lite za mikrokontrolerje

Na tej točki se morda sprašujete: ali niste omenili, da boste uporabljali TensorFlow? No, kot je omenjeno v tem zanimivem članku [13] iz bloga TensorFlow, TensorFlow sam po sebi uporablja Edge Impulse:

Edge Impulse uporablja ekosistem TensorFlow za usposabljanje, optimizacijo in uvajanje modelov globokega učenja v ugnezdene naprave.

Če se ozremo nazaj na prejšnje korake tega projekta, smo usposobili model za klasifikacijo besed, nato izvedli kvantizirano (int8) optimizacijo in končno zgradili Arduino knjižnico za namestitev na našo ploščo. V članku je navedeno tudi:

Razvijalci se lahko odločijo za izvoz knjižnice, ki za zagon modela uporablja TensorFlow Lite for Microcontrollers tolmač.

Skratka, VoiceTurn_inferencing knjižnica, ki je bila prej razvita z Edge Impulse, uporablja TensorFlow Lite tolmač [14] za mikrokontrolerje za izvajanje modela strojnega učenja za razvrščanje besed.

Pravzaprav je zelo enostavno sami preverite, ali ta knjižnica za izvajanje klasifikatorja uporablja TensorFlow Lite za mikrokontrolerje. V vašem Arduino IDE kliknite File > Preferences in preverite lokacijo Sketchbook. V raziskovalcu datotek pojdite na to lokacijo in z urejevalnikom besedil odprite VoiceTurn_inferencing/src/VoiceTurn_inferencing.h. V vrstici 41 boste našli:

#include "edge-impulse-sdk/classifier/ei_
run_classifier.h"

Zdaj se vrnite v mapo src in znova odprite edge-impulse-sdk/ classifier/ei_run_classifier.h z uporabo urejevalnika besedil. V vrstici 45 boste našli:

#include "edge-impulse-sdk/tensorflow/lite/
micro/micro_interpreter.h"

Kar se nanaša na uporabo knjižnice TensorFlow Lite za mikrokontrolerje. Pravzaprav je Litelibrary TensorFlow del SDK Edge Impulse, vključno z njegovo podknjižnico Micro C++.

Zgradite hardver

Smerniki bodo sestavljeni iz dveh LED trakov: enega na levi in drugega na desni strani kolesa. Uporabil sem naslovljiv RGB LED trak, sestavljen iz LEDic WS2813. Uporabite lahko druge LED trakove, če ste pozorni na ožičenje in da naknadno prilagodite kodo. Prva stvar je, da LED trakove odrežete tako, da imajo želeno dolžino. V tem primeru sem uporabil 10 cm dolg kos za vsako stran, kar ustreza 6 LEDicam. Pazite, da LED trakove odrežite po črtkani črti, da ne poškodujete električnih kontaktov.



Kot izbirni korak zdaj izvrtajte dve luknji na ravnilu ali ravni površini po vaši izbiri. Če želite strogo upoštevati dimenzije projekta, uporabite 30 cm dolgo (približno 12 palcev) površino in izvrtajte luknje na položajih 11,5 cm oziroma 18,5 cm.



Da bi lahko LED trakove povezali s preostalimi komponentami, moramo na njihove priključke dodati žice za priklop. Najprej s škarjami odstranite majhen del gel materiala, ki pokriva električne kontakte. Nato uporabite spajkalnik skupaj s spajkalnim fluksom, da prispajkate tri žice na priključke vsakega LED traku. Priporočam, da sledite običajni barvni kodi, tako da bo rdeča žica spajkana na 5V pin, črna pa na GND pin.



Kar zadeva druge priključke, sem uporabil rumeno žico za desno stran in zeleno žico za levo stran. Čeprav ima LED trak WS2813 štiri priključke, lahko DI (podatkovni vhod) in BI (rezervni vhod) zaradi preprostosti spajkate skupaj. Ko je ta korak opravljen, odstranite trak z LED trakovi in jih prilepite na ravno površino na položajih 0-10 cm oziroma 20-30 cm. Če ste v prejšnjem koraku izvrtali luknje na površini, skoznje napeljite žice. Če ne, jih preprosto postavite okoli roba.



Celotna nastavitev bo razdeljena na dva dela, ki se bosta med seboj povezali s pomočjo 3,5 mm priključkov za vtikač/vtičnico. Vzemite ustrezne kable in jih razrežite na približno 20-80 % dolžine, tako da boste na koncu dobili 80 % dolžine kabla, ki vsebuje vtičnico iz enega kabla, in še 80 % odstotno dolžino kabla, ki vsebuje vtič iz drugega kabla. Odstranite majhen del zunanje prevleke in videli boste, da je vsak kabel sestavljen iz 4 priključkov: treh žic in opleta. Vse žice opleta zvijte skupaj, da jih boste lažje spajkali.



Primite dolg kos kabla, ki vsebuje 3,5 mm priključek, in spajkajte žice na naslednji način (če je barvna koda drugačna, jo prilagodite svojim potrebam):

- Rdeča žica kabla do obeh rdečih žic dveh LED trakov (5V).
- Zelena žica kabla do zelene žice levega LED traku (DI-BI).
- Rumena žica kabla do rumene žice desnega LED traku (DI-BI).
- Oplet kabla do obeh črnih žic dveh LED trakov (GND).



Posledično bi morali imeti končan prvi del nastavitve, ki je sestavljen iz ravne površine, ki vsebuje smerne luči in kos kabla, ki se konča na 3,5 mm vtičnici. Izogibajte se stika med električnimi kontakti, tako da vsakega posamezno pokrijete z izolirnim trakom ali termoskrčljivimi božirkami.



Drugi del namestitve je enostavnejši za izdelavo. Vzemite dolg kos kabla, ki vsebuje 3,5 mm vtičnico, in prispajkajte žice na Arduino ploščo po enaki barvni kodi kot v prejšnjem koraku in Arduino priključkih:

- Rdeča žica do 3V3 Arduino priključka.
- Zelena žica do D4 Arduino priključka.
- Rumena žica do D7 Arduino priključka.
- Črna žica do enega od GND Arduino priključkov.

Shemo ožičenja najdete v razdelku Shema.

Tako preprosto nastavitev lahko uspemo, ker uporabljamo kratke LED trakove, ki vsebujejo le nekaj LEDic. Če uporabljate daljše LED trakove, jih boste morda morali napajati z zunanjim virom, da ne presežete največjega toka, ki ga podpira plošča. Če to storite, ne pozabite prispajkati 1000 µF kondenzatorja med priključki 5V in GND vaših LED trakov, da zagotovite stabilnost napajalne napetosti.

Dodajte kodi funkcionalnost

Prej smo v Arduino serijskem monitorju preverili, kako so besede, ki jih izgovorimo, razvrščene v vsako od štirih vnaprej določenih skupin. Ko je strojna oprema izdelana, je zdaj čas, da dodate zahtevano funkcionalnost za vklop enega od smernikov, potem ko izgovorite besedo, ki ustreza tej strani.

Najprej boste morali namestiti knjižnico, ki jo zahteva Voice-Turn za nadzor naslovljivih LED trakov, ki jih bomo uporabljali za smernike. Kliknite Sketch > Include Library > Manage Libraries...

Sketch Tools Help			
Verify/Compile	Ctrl+R		
Upload	Ctrl+U		
Upload Using Programmer	Ctrl+Shift+U		
Export compiled Binary	Ctrl+Alt+S		
Show Sketch Folder	Ctrl+K		
Include Library	*	Δ	
Add File	1	Manage Libraries	Ctrl+Shift+I

V iskalno polje vpišite NeoPixel in namestite knjižnico Adafruit NeoPixel.

xdafruit NeoPixel by Adafruit Arduino library for controlling single-wire-based LED pixels and strip. Ardu Hore info	ino library for controlling single-wire-based LED pixels and strip.
	Version 1.8.3 🔻 Install

Delali bomo na vzorčnem programu nano_ble33_sense_microphone_continuous, ki ga zagotavlja knjižnica VoiceTurn_inferencing, zgrajena in uvožena v prejšnjih korakih.

Za vsako smer je potreben en LED trak, levo ali desno, vsak LED trak pa ima skupaj 6 LEDic. Poleg tega bo vgrajena Arduino RGB LED plošča uporabljena, da bo kolesar hitro vedel, ali je plošča pripravljena poslušati besedo ali še vedno uporablja smernike. Določite razpored priključkov na začetnem delu programa, pri čemer si zapomnite priključke, ki ste jih uporabili za zelene in rumene žice, ko ste zgradili del strojne opreme:

```
/* LED strip pinout */
#define LED_PIN_LEFT 4
#define LED_PIN_RIGHT 7
#define LED_COUNT 6
/* Built-in RGB LED pinout */
#define RGB_GREEN 22
#define RGB_RED 23
#define RGB BLUE 24
```

Vključite NeoPixel knjižnico poleg drugih že vključenih knjižnic:

```
#include <Adafruit NeoPixel.h>
```

Deklarirajte dva LED trakova tako, da določite njuno število LEDic, podatkovni priključek, na katerega sta povezana, in vrsto traku. Tukaj bomo za vrsto traku uporabili NEO_GRB + NEO_KHZ800, saj ta ustreza LED trakom WS2812 in podobnim. To lahko spremenite s pomočjo knjižnične dokumentacije, če uporabljate drugo vrsto LED traku.

*/
/* LED strips forleft and right signals
Adafruit_NeoPixel left(LED_COUNT, LED_PIN_
LEFT, NEO_GRB + NEO_KHZ800);

Adafruit_NeoPixel right(LED_COUNT, LED_PIN_ RIGHT, NEO_GRB + NEO_KHZ800);

Deklarirajte dve konstanti, ki bosta koristni za nastavitev trajanja svetlobne signalizacije. Obdobje je približno čas čakanja med vklopom LED in naslednjim vklopom: višji kot je nastavljen ta čas, počasnejša bo animacija. Spremenljivka ciklov se nanaša na število ponovitev animacije vsakič, ko se zazna beseda levo ali desno.

```
static int period = 100;
static int cycles = 3;
```

Zdaj v funkciji setup() dodajte naslednje vrstice kode za inicializacijo dveh LED trakov in nastavite njuno raven svetlosti na približno 20%. Slednji deluje kot programsko voden nadzor toka, tako da pazimo, da ne uničimo LEDic ali ne prekoračimo maksimalnega toka, ki ga podpira Arduino plošča predvsem takrat, ko na priključke trakov ni prispajkan noben zaščitni zaporedno vezan upor.

```
// Initialization of LED strips:
left.begin();
right.begin();
left.show();
right.show();
// Set BRIGHTNESS to about 2/5 (max = 255)
left.setBrightness(100);
right.setBrightness(100);
```

Inicializirajte priključke LED RGB kot IZHOD.

pinMode(RGB_RED, OUTPUT); pinMode(RGB_GREEN, OUTPUT); pinMode(RGB_BLUE, OUTPUT);

Kot morda veste, koda znotraj loop() funkcije vsebuje programska navodila, ki se bodo izvajala znova in znova. Dodajte naslednjo vrstico na začetek funkcije, da kolesarju sporočite, da plošča posluša. Nato bo ustvarjena funkcija rgb_green().

rgb_green(); // Shows the board is READY

Po tem se izvede klasifikacija besed in izhod programa se shrani v spremenljivko result.classification. Ta spremenljivka je indeksirana po vrstnem redu označevanja, ki smo ga uporabili pri usposabljanju modela strojnega učenja z Edge Impulse. Ne pozabite, da smo v tem vrstnem redu uporabili oznake levo, desno, šum in drugo. Zato result.classification[0] vsebuje informacije, ki ustrezajo levi skupini, result.classification[0].value pa je verjetnost, da izgovorjena beseda pripada levi skupini. Podobno result.classification[1] vsebuje informacije, ki ustrezajo desni skupini, result.classification[1].value pa je verjetnost, da izgovorjena beseda pripada desni skupini. Indeksa 2 in 3 ustrezata skupini šuma oziroma drugim skupinam, čeprav ju v tem projektu ne bomo uporabljali.

Ustrezen LED trak želimo aktivirati, če je beseda zaznana z

verjetnostjo nad določenim pragom. Za ta projekt je prag verjetnosti nastavljen na 80 % za besedo levo in na 85 % za besedo desno, da se čim bolj izognemo lažnim pozitivnim rezultatom. Pragovi so različni, saj so nastavljeni glede na izhod faze testiranja modela, opravljenega z Edge Impulse. Dodajte ta del kode na konec funkcije loop(), potem ko je vsa obdelava strojnega učenja končana.

```
// 0 ->LEFT; 1 -> RIGHT
if (result.classification[0].value>= 0.80)
{
  turn(left);
}
if (result.classification[1].value>= 0.85)
{
  turn(right);
}
```

Programu je treba dodati le funkcije, ki so zadolžene za aktiviranje LED trakov in nadzor nad vgrajeno RGB LED. Funkcija turn() sprejme LED trak, da se aktivira kot vhodni parameter, in vklopi LED eno za drugo v oranžni barvi po animaciji, podobni tistim v sodobnih avtomobilih. Ko je vnaprej določeno število ciklov končano, se LED trak izklopi. Poleg tega je kolesar opozorjen na stanje programa tako, da vgrajeno RGB LED nastavite v rdečo barvo. Ko se LED trakovi uporabljajo, vgrajeno RGB LED izklopi. Dodajte to funkcijo na konec programa, potem ko se funkcija loop() zaključi.

```
{
static void turn(Adafruit NeoPixel&strip)
rgb red(); // Shows the board is BUSY
for (int i = 0; i < cycles; i++) {</pre>
for (int j = 0; j <strip.numPixels(); j++) {</pre>
strip.setPixelColor(j,
                            strip.Color(255,
104, 0)); // Color: Orange
strip.show();
delay(period);
}
strip.clear();
strip.show();
delay(2 * period);
}
rgb off(); // The board has FINISHED ligh-
ting the LED strip
}
In končno dodajte tri preproste funkcije, ki
nadzorujejo vgrajeno RGB LED.
{
voidrgb red()
     digitalWrite(RGB RED, HIGH);
digitalWrite(RGB GREEN, LOW);
digitalWrite(RGB BLUE, LOW);
}
voidrgb green(){
digitalWrite(RGB RED, LOW);
digitalWrite(RGB GREEN, HIGH);
digitalWrite(RGB BLUE, LOW);
}
```

```
voidrgb_off(){
digitalWrite(RGB_RED, LOW);
digitalWrite(RGB_GREEN, LOW);
digitalWrite(RGB_BLUE, LOW);
}
```

Ko program prevedete, ga naložite na ploščo Arduino in povežite strojno opremo na svojo ploščo. Nato boste lahko sami preizkusili nastavitev VoiceTurn. Ko izgovorite besedo »levo«, naj se vklopi levi smernik in po besedi »desno« enako bi se moralo zgoditi z desnim smernikom.

Če vaši testi niso tako uspešni, kot ste pričakovali, boste morda morali prilagoditi občutljivost postopka klasifikacije besed vašim potrebam. To lahko storite tako, da nastavite dva parametra:

- Verjetnost zaznave besede, ki je sprva nastavljena na 85 %: z znižanjem te številke boste povečali število odzivov naprave na vaš glas, povečali pa boste tudi količino lažno pozitivnih rezultatov.
- Konstanta EI_CLASSIFIER_SLICES_PER_MODEL_WIN-DOW, definirana na začetku programa: to je število koščkov, na katere je okno modela razdeljeno. Več informacij o tem parametru najdete na tej povezavi [15].

Povezavo do GitHub, ki vsebuje kodo projekta, sem vključil v razdelek Code, zato jo preverite, če ste se kdaj izgubili ali če želite neposredno preizkusiti mojo kodo.

Nastavite VoiceTurn na kolesu

Preizkus koncepta VoiceTurn je bil izveden na kolesu. Za pritrditev vseh komponent nanj so bile uporabljene plastične vezice, tako da je nastavitev mogoče enostavno namestiti in odstraniti brez trajnih sprememb na kolesu. Če želite, da je nastavitev trajno nameščena, lahko uporabite druge načine.

Prvi del je sestavljen iz Arduino plošče in dveh kablov, ki sta nanjo pritrjena: tisti, ki vsebuje žice, spajkane na Arduino priključke in se konča na 3,5 mm vtičnici, in mikro-USB na USB kabel, ki ploščo priključi na napajalno baterijo. Postavljenih je nekaj vezic, s katerimi sta oba prej omenjena kabla pritrjena na zadnjo zavorno žico. Oba kabla sta dovolj trda, da Arduino plošča ostane v lebdečem in stabilnem položaju, pri čemer je mikrofon obrnjen proti kolesarju.



Oba kabla moramo pritrditi na kolo, tako da lahko 3,5 mm vtičnica doseže zadnji del kolesa, USB vtič pa do sedeža, kjer bo nameščena baterija. Kabli so pritrjeni vzdolž zgornje strani okvirja kolesa.



Sedaj namestimo nosilec kolesarske ure okoli palice za sedež, tako da dobimo ravno površino, kjer montiramo ravnilo, ki vsebuje smernike. Uporabite nekaj kabelskih vezic v križni razporeditvi, da pritrdite drugi del nastavitve na palico za sedež na mestu pritrditve nosilca ure. Preostale žice pritrdite tudi na palico za sedež.



Priključite 3,5 mm vtikač na ustrezno vtičnico in preostali kabel pritrdite okoli okvirja kolesa, da ju ne boste med vožnjo nenamerno razklenili.



Na koncu postavite power bank v režo pod sedežem, ga trdno pritrdite s kabelskimi vezicami in nanj priključite USB kabel in tako da je VoiceTurn pripravljen za uporabo.



Uživajte!

Zdaj je čas, da sestavite in preizkusite VoiceTurn sami. Lahko ga celo vzamete kot osnovni projekt in ga še naprej izboljšujete: zmanjšajte ožičenje z uporabo vgrajene Arduino BLE povezljivosti, dodajte besedo za bujenje, kot je Hey Bike!, razširite nabor besed in dodajte več funkcionalnosti ... veliko različnih možnosti obstaja.

Viri:

- https://experiments.withgoogle.com/voice-turn
- https://www.edgeimpulse.com/
- https://docs.edgeimpulse.com/docs/responding-to-your-voice
- https://ai.googleblog.com/2017/08/launching-speech
 -commands-dataset.html
- http://download.tensorflow.org/data/speech_commands_ v0.02.tar.gz
- https://arxiv.org/pdf/1804.03209.pdf
- https://docs.edgeimpulse.com/docs/keyword-spotting
- https://www.youtube.com/watch?v=bfmFfD2RIcg
- https://docs.edgeimpulse.com/docs/arduino-nano-33-blesense
- https://studio.edgeimpulse.com/public/39038/latest
- https://www.arduino.cc/en/software
- https://docs.edgeimpulse.com/docs/continuous-audio -sampling
- https://blog.tensorflow.org/2021/06/how-tensorflow-helps -edge-impulse-make-ml-accessible.html
- https://www.tensorflow.org/lite/microcontrollers
- https://docs.edgeimpulse.com/docs/continuous-audio -sampling

Povzeto po:

https://www.hackster.io/a_g_v/voiceturn
 -voice-controlled-turn-lights-for-a-safer-ride
 -3a8e78



https://hackster.io



ATtiny I2C slave brez težav in odvečnih stroškov (2)

Avtorja: Vladimir Mitrović in Robert Sedak E-pošta: vmitrovic12@gmail.com

V prvem nadaljevanju smo predstavili programske knjižnice za Bascom-AVR, s pomočjo katerih lahko na enostaven način nek ATtiny mikrokontroler pretvorimo v I2C slave.

Prav tako smo predstavili I2C slave modul z ATtiny85 mikrokontrolerjem in RGB diodo in za njega napisali ustrezen komunikacijski program. V tem nadaljevanju bomo opisali kako realizirati I2C slave modul z ATtiny mikrokontrolerjem iz Arduino IDE, in nato bomo predstavili ustrezne master programe za razvojni sistem Shield-A za obe platformi.

Arduino I2C slave

Arduino IDE ne podpira ATTiny takoj po instalaciji, ampak moramo instalirati dodatno podporo. Obstaja več projektov z odprto kodo. Projekt ATTinyCore avtorja Space Konde se je pokazal kot stabilen, ukazi so kompatibilni z originalnim Arduino IDE. Na strani[1] se nahajajo navodila za instalacijo. Možno je uporabiti več načinov programiranja ATTiny mikrokontrolerja: brez bootloadera ali z Optoboot ali Micronucleus bootloaderjema, ki zavzemata del spomina mikrokontrolerja. Predlagamo uporabo Arduino Uno kot ISP za programiranje ATTiny, da ne bi bilo potrebno programirati bootloader v ATTiny.

ATTinyCore ima knjižnico Wire ki uporablja USI in dela zelo zanesljivo.

Arduino rešitev programske naloge

Programska logika Arduino IDE rešitve je identična kot v Bascom-AVR. Ukaze, ki ne obstajajo v Arduino IDE, bomo napisali kot makro ukaze.

Program ATtiny85_RGB_slave_1.ino

Na začetku programa definiramo, da bomo uporabljali knjižnico Wire:

#include <Wire.h>

definiramo makro ukaze RED, GREEN in BLUE in v njih shranimo informacijo, katere priključke bomo uporabljali:

#define RED PB1 #define GREEN PB3 #define BLUE PB4

Za bolj pregleden program definiramo makro ukaz vključevanja in izključevanja zelene LEDice in Timerja0,

```
#define GREEN_OFF (PORTB &= ~(1 << GREEN))
#define GREEN_ON (PORTB |= (1 << GREEN))
#define STOP_TIMER0 (TCCR0B &=</pre>
```

~((1<<CS00)|(1<<CS01))) #define START_TIMER0 (TCCR0B |= (1<<CS00)|(1<<CS01))

definiramo I2C naslov "0b1111000" za ATTiny, polje v katerem bomo shranili vrednosti, ki nam jih je poslal master in katero verzijo RGB LED uporabljamo (s skupno katodo ali skupno anodo):

```
byte I2C_addr = 0b1111000;
volatile byte i2c_rcv_byte[3] = {0, 0, 0};
const char rgb common ac = 'a';
```

V funkciji setup() konfiguriramo Timer0 tako, da šteje impulze, ki jih dobi z deljenjem osnovne frekvence (16 MHz) s faktorjem 64, postavimo registre OCR0A in OCR0B v začetni strani in vključimo prekinitve:

```
void setup() {
  cli();
  TCCR0A = 0;
  TCCR0B = 0;
  START_TIMER0;
  TCCR0B |= (1<<CS00) | (1<<CS01);
  CLKPR |= (1 << CLKPCE);
  CLKPR = 0;
  TIFR |= (1 << OCF0A) | (1 << OCF0B);
  TIMSK |= (1 << OCIE0A) | (1 << OCIE0B);
  OCR0A = 0;
  OCR0B = 255;
  TCNT0 = 0;
  sei();</pre>
```

Namen vsakega od teh ukazov lahko izveste iz priloženega programa, kjer je večina ukazov izdatno komentirana.

Sedaj inicializiramo I2C protokol, dodelimo I2C slave naslov zapisan v spremenljivki I2C_addr in definiramo, da se bo izvršila funkcija receiveEvent(), ko knjižnica Wire ugotovi, da I2C master pošilja podatke:

```
Wire.begin(I2C_addr);
Wire.onReceive(receiveEvent);
```

Moramo še definirati priključke, ki so vezani na RGB LEDico in izvršimo začetni demo program:

pinMode(RED, OUTPUT);

```
pinMode(GREEN, OUTPUT);
pinMode(BLUE, OUTPUT);
rgb_demo();
} // end setup()
```

Ker uporabljamo funkcije prekinitve, v funkciji loop() nam ni treba imeti niti enega ukaza:

```
void loop() {
{
```

Opomnimo, da se to bistveno razlikuje od Bascom-AVR rešitve, v kateri se je I2C komunikacija spremljala v glavni programski zanki. Za krmiljenje z zeleno LEDico uporabljamo dve prekinitvi vezani na Timer0. Prva prekinitev vključuje LEDico ko timer doseže v OCR0B vpisano vrednost 255, druga prekinitev izključuje LEDico ko timer doseže vrednost vpisano v OCR0A. S spremembo vrednosti OCR0A vplivamo na trajanje stanja "vključeno" in s tem na intenziteto zelene barve.

```
ISR (TIMER0_COMPA_vect) {
 GREEN_OFF;
 } // end ISR (TIMER0_COMPA_vect)
ISR (TIMER0_COMPB_vect) {
 GREEN_ON;
 } // end ISR (TIMER0_COMPB_vect)
```

V funkciji receiveEvent() prevzemamo podatke z I2C vodila in jih shranimo v polje i2c_rcv_byte[], in nato zaženemo funkcijo execute_i2c_command():

```
void receiveEvent(int howMany) {
i2c_rcv_byte[0] = Wire.read();
i2c_rcv_byte[1] = Wire.read();
i2c_rcv_byte[2] = Wire.read();
execute_i2c_command();
}
// end receiveEvent(int howMany)
```

V funkciji execute_i2c_command() najprej preverjamo ali imamo RGB LED s skupno anodo ali katodo. Pri RGB LEDcah s skupno katodo shranjene vrednosti v polju i2c_rcv_byte[] lahko direktno uporabimo; za rdečo in modro barvo bomo uporabili ukaz analogWrite() in vrednosti iz i2c_rcv_byte[0] in i2c_rcv_byte[2], medtem ko bomo za zeleno LED vrednost iz i2c_rcv_ byte[1] vpisovali v OCR0A register Timerja0. Kot tudi v Bascom-AVR rešitvi, na ta način ni možno v popolnosti izključiti zelene barve in bomo, kadar je uporabljena vrednost 0, "ročno" zaustavili Timer0 in izključili zeleno LEDico. Če uporabljamo RGB LED s skupno anodo, moramo vrednosti, ki jih dobimo od masterja, pred uporabo komplementirati, tako da shranjeno vrednost v polju i2c_rcv_byte[] odštejemo od 255:

void execute_i2c_command(){
 if (rgb common ac == 'a') {

```
analogWrite(RED, 255-i2c rcv byte[0]);
if (i2c rcv byte[1] == 255) {
STOP TIMER0;
GREEN ON;
} else {
OCR0A = 255-i2c rcv byte[1];
START TIMER0;
}
analogWrite(BLUE, 255-i2c rcv byte[2]);
} else {
analogWrite(RED, i2c rcv byte[0]);
if (i2c_rcv_byte[1] == 0) {
STOP TIMER0;
GREEN OFF;
} else {
OCR0A = i2c rcv byte[1];
START TIMER0;
}
analogWrite(BLUE, i2c rcv byte[2]);
}
} // execute i2c command()
```

Funkcija rgb_demo() je identična kot v Bascom-AVR rešitvi: program menja intenzivnost vseh treh barv od najmanjše do največje in nato nazaj do najmanjše, kot signal, da je RGB slave modul pripravljen. Ko naredi to animacijo, funkcija nima nobenega vpliva na delo modula in je tukaj ne bomo analizirali, saj bo modul pravilno funkcioniral tudi, če jo popolnoma izpustimo. Tukaj moramo poudariti, da funkcija for v C jeziku ne bo delovala dobro, če uporabljamo 8-bitni števec in kot mejno vrednost postavimo 255 (ali 0, kadar zanka šteje navzdol). Takšno zanko moramo "ročno" dodelati, kot v funkciji rgb_demo() v priloženem programu.

Program Attiny85_RGB_slave s softversko podporo za I2C

V Adruino IDE okolju trenutno ne obstaja uradna knjižnica, ki bi podpirala softverski I2C slave način dela (knjižnica "SoftWire" podpira samo master). Zato nimamo svobodne izbire priključkov





za I2C komunikacijo, ampak smo omejeni na priključke, ki so povezani z USI vezjem. S komunikacijskega vidika to ne bo predstavljalo problema, ker je hardverska rešitev hitrejša in zanesljivejša od softverske emulacije. Vendar pa lahko nekaj izgubimo na nekem drugem nivoju: tako smo v našem primeru zaradi navedene omejitve izgubili možnost hardverskega generiranja enega od treh PWM signalov.

Arduino I2C master

Kot I2C master smo uporabili razvojni sistem Shield-A postavljen na Arduino

UNO. "Master" programi, ki jih bomo predstaviti, so pisani za takšno okolje, katerega shema je prikazana na sliki 5. Oznake posameznih komponent prihajajo iz Shielda-A, modro pobarvane oznake pa ustrezajo oznakam z Arduino Uno ploščice. Enako dobro bo služil kateri koli podobni razvojni sistem ali vezje, bazirano na nekem ATmega mikrokontrolerju, seveda z ustreznimi spremembami programa.

Slika 6 pokazuje kako povežemo Shield-A z RGB modulom. Na sliki so simbolično prikazane komponente Shielda-A, ki jih uporabljamo za interaktivno nastavljanje barv: alfanumerični displej, tipki SW1 in SW2 ter potenciometer RV1. S tipko SW1 izberemo barvo, s potenciometrom nastavimo njeno intenzivnost svetenja in s tipko SW2 pošljemo ustrezen ukaz RGB modulu.

Slika 7 vizualizira izbiro na alfanumeričnem displeju: oznake v gornji vrstici displeja označujejo rdečo, zeleno in modro barvo, triznakovne številke pod temi oznakami pa trenutno intenzivnost posamezne barve. Četrta številka v spodnji vrstici je odvisna od položaja potenciometra RV1; z vrtenjem njegove osi od enega do drugega skrajnega položaja, se prikazana vrednost menja v razponu 0-255, prav tako kot nam ustreza za intenzivnost ene od barv.

Barvo, katere intenzivnost nastavljamo, izberemo s pomočjo tipke SW1z naslednjim vrstnim redom:

niti ena -> rdeča -> zelena -> modra -> vse tri -> niti ena

To je na sliki 7 prikazano z rdečimi puščicami, oznaka trenutno izbrane barve je na displeju izpisana z velikimi črkami. Ko izberemo eno od barv in s potenciometrom nastavimo želeno intenzivnost, ga s pritiskom na tipko SW2 vpišemo pod oznako barve (to je na sliki 7 prikazano z modrimi puščicami) in preko I2C vodila pošljemo ustrezen ukaz RGB modulu. Strukturo teh ukazov smo opisali v predhodnem nadaljevanju. Z vrtenjem osi potenciometra RV1 lahko po volji nastavljamo intenzivnost izbrane barve, vendar se bo prikaz na LCDju spremenil in ukaz RGB modulu poslal šele po pritisku na tipko SW2.

Če držimo tipko SW2 pritisnjeno dalj od ene sekunde, se program prestavi v avtomatski način dela (spodnja vrstica na sliki 7): v gornjem desnem vogalu displeja se izpisuje oznaka "auto", program pa vsakih 50 ms prebere napetost na drsniku potenciometra RV1, ga preračuna v intenzivnost, ažurira







Slika 7: S pomočjo izbire na alfanumeričnem displeju formiramo ukaz, ki ga bo master program poslal RGB modulu

intenzivnost izbrane barve (ali vseh barv) in pošlje ustrezen ukaz RGB modulu. Obračanje osi potenciometra RV1 se odrazi v trenutni spremembi prikaza na LCDju in barve RGB diode na RGB modulu. V "ročni" način dela se vračamo s kratkim pritiskom na tipko SW2.

Bascom-AVR rešitev: program Shield-A_RGB_master.bas

Program uporablja naslednje spremembe:

- *Rd, Gn in Bl vsebujejo trenutno intenzivnost rdeče, zelene in modre barve.*
- Adc_value vsebuje trenutno branje A/D pretvornika
- Rgb_value služi za oblikovanje prikaza numeričnih vrednosti na LCDju
- Rgb_index določa izbrano barvo in prikaz na LCDju (0 = nobena barva, 1 = rdeča, 2 = zelena, 3 = modra, 4 = vse barve)
- Auto_flag določa način dela (0-100 = ročni, >100 = avtomatski)

Na začetku programa definiramo I2C naslov slave mikrokontrolerja, kateremu bo master pošiljal sporočila Const I2c_address = &B11110000

in nato tudi hitrost I2C komunikacije ter SCL in SDA priključke:

```
$lib "i2c_twi.lbx"
Config Twi = 100000 'SCL = 100kHz
Config Sda = Portc.4
Config Scl = Portc.5
I2cinit
```

S tem, ko smo vključili I2C_TWI.lbx knjižnico, smo Bascom -AVRju omogočili da namesto standardne softverske emulacije uporablja TWI vezje ATmega328P mikrokontrolerja. Njegova SDA in SCL priključka sta priključena na priključke PC4 in PC5, in smo to morali upoštevati v konfiguracijskih ukazih. Izbrali smo standardno hitrost komunikacije 100 kHz.

A/D pretvornik bomo konfigurirati tako, da kot referenčno napetost uporablja napetost napajanja:

```
Config Adc = Single , Prescaler = Auto ,
Reference = Avcc
```

Zato se bo polni razpon A/D pretvorbe (0-1023) pokril z vrtenjem osi potenciometra RV1 od enega do druge skrajnega položaja. Še bomo na običajni način konfigurirali LCD in vhodne priključke PC1 in PC2, nakar vstopimo v glavno Do-Loop zanko. V njej vsakih 50 ms preverjamo stanja tipka SW1 in SW2 ter izvršujemo podprogram Disp_adc in če je izbran avtomatski način dela, tudi podprogram Disp_send:

```
Do
Debounce Pinc.1 , 0 , Sw1_sub , Sub
Debounce Pinc.2 , 0 , Sw2_sub , Sub
Gosub Disp_adc
If Auto_flag > 100 Then
Gosub Disp_send
End If
Waitms 50
Loop
```

V podprogramu Disp_adc merimo napetost drsnika potenciometra RV1, delimo izmerjeno vrednost s 4 (tako smo dobili razpon vrednosti 0-255), in jo prikazujemo na LCDju:

```
Disp_adc:
Adc_value = Getadc(0)
Adc_value = Adc_value / 4
Locate 2 , 13
Rgb_value = Str(adc_value)
Rgb_value = Format(rgb_value , "000")
Lcd Rgb_value
Return
```

Odvisno od vsebine spremembe Rgb_index, v podprogramu Disp_send najprej ažuriramo intenziteto trenutno izbrane barve ali barv,

Disp_send: If Rgb_index = 1 Then

```
Rd = Adc_value
Elseif Rgb_index = 2 Then
Gn = Adc_value
Elseif Rgb_index = 3 Then
Bl = Adc_value
Elseif Rgb_index = 4 Then
Rd = Adc_value
Gn = Adc_value
Bl = Adc_value
End If
```

in nato naslavljamo RGB modul in mu pošljemo trenutno nastavljene intenzitete vseh treh barv:

```
I2cstart
I2cwbyte I2c_address
' Waitms 1
I2cwbyte Rd
I2cwbyte Gn
I2cwbyte Bl
' Waitms 1
I2cstop
```

Waitms ukazi so potrebni samo če slave potrebuje več od 2 µs za obdelavo naslova oziroma podatkov, ki mu jih je master poslal. Program našega RGB modula je dovolj hiter, zato so Waitms ukazi zakomentirani. Bascom-AVR postavlja sistemski Err bit, če je prišlo do napake v komunikaciji (npr., če slave ni odgovoril z ACK po vsakem prejetem bajtu). Preverili bomo Err bit in če je detektirana napaka, bomo izpisali ustrezno sporočilo:

```
If Err = 1 Then
Locate 2 , 1
Lcd "Err
Waitms 250
Err = 0
End If
```

Na koncu podprograma bomo na LCDju izpisali trenutne intenzitete barv in se vrnili v glavno zanko:

Gosub Disp_rgb Return

Isti podprogram kličemo tudi ko program detektira da je pritisnjena tipka SW2,

Sw2_sub: Gosub Disp_send

nato merimo, kako dolgo je tipka bila pritisnjena:

```
For Auto_flag = 0 To 100
If Pinc.2 = 1 Then Exit For
Waitms 10
Next
```

Če je tipka bila pritisnjena več kot 1 s, bo vrednost števca Auto_flag bila 101 in program gre v avtomatski način dela; če je pritisnjena krajši čas, se program vrača v ročni način dela. Sporočilo o trenutno veljavnem načinu dela se izpisuje na LCDju, nakar se program vrača v glavno zanko:

Locate 1 , 13
If Auto_flag > 100 Then
Lcd "auto"
Else
Lcd ""
End If
Return

Kot smo prej pokazali, se vrednost spremembe Auto_flag preverja tudi v glavni zanki in se, če je avtomatski način dela aktiviran, RGB modulu pošlje sporočilo vsakih 50 ms. V nasprotnem primeru se bo sporočilo poslalo samo kadar pritisnemo tipko SW2, tako, kot je v programski nalogi tudi predvideno.

Podprogram Sw1_sub se izvršuje vsakič, ko pritisnemo tipko SW1. V njemu menjamo vrednost spremembe Rgb_indeks v razponu od 0 do 4, in s tem izberemo novo barvo v skladu s programsko nalogo in prikazom na sliki 7. Ta podprogram in druge dele master programa, ki nimajo direktnega stika z osnovnim namenom, tukaj ne bomo analizirali.

Program Shield-A_RGB_master.ino

Uporabili bomo identične spremembe in programsko logiko kot v Bascom-AVR rešitvi.

Na začetku programa definiramo knjižnice, ki jih bomo uporabljali in deklariramo funkcijo debounce(), ki jo bomo kasneje tudi definirali.

#include <Wire.h>
#include <LiquidCrystal.h>
void debounce(byte, byte, void (*)());

Definiramo objekt s pomočjo katerega krmilimo LCD in potrebne spremembe:

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
byte rd, gn, bl, rgb_index, auto_flag;
const byte I2c_address = 0b1111000;
int adc_value;
```

V funkciji setup() bomo vključili osvetlitev ozadja s pomočjo tranzistorskega stikala T2, definirali priključke tipke, zagnali I2C komunikacijo, definirali kateri LCD displej uporabljamo, definirali začetno vrednost spremembe rgb_index in zagnali funkcijo sw1_function() da bi prikazali vrednosti spremenljivk na displeju.

```
void setup() {
pinMode(17, OUTPUT);
digitalWrite(17, HIGH);
pinMode(A1, INPUT_PULLUP);
pinMode(A2, INPUT_PULLUP);
Wire.begin();
lcd.begin(16, 2);
```

```
rgb_index = 4;
sw1_function();
```

} // end setup()

V funkciji loop() vsakih 50 ms preverjamo stanja tipk SW1 in SW2 in izvršujemo podprogram disp_adc(). Če pa je izbran avtomatski način dela, pa tudi podprogram disp_send():

```
void loop() {
  debounce(A1, 0, sw1_function);
  debounce(A2, 0, sw2_function);
  disp_adc();
  if (auto_flag > 100) disp_send();
  delay(50);
  } // end loop()
```

V funkciji disp_adc() merimo napetost drsnika potenciometra RV1, delimo izmerjeno vrednost s 4 (tako smo dobili razpon vrednosti 0-255), in jo prikažemo na LCDju:

```
void disp_adc(){
  adc_value = analogRead(A0)/4;
  lcd.setCursor(12,1);
  if (adc_value < 100) lcd.print("0");
  if (adc_value < 10) lcd.print("0");
  lcd.print(adc_value);
  } // end disp_adc()</pre>
```

V funkciji disp_send(), odvisno od vsebine spremembe rgb_index, najprej ažuriramo intenziteto trenutno izbrane barve ali barv,

void disp send(){ switch(rgb index){ case 1: rd = adc value; break; case 2: gn = adc value; break; case 3: bl = adc_value; break; case 4: rd = adc value; gn = adc value; bl = adc value; break; } disp rgb(); send rgb(); } // end disp_send()

nato prikažemo vsebino barv na displeju s pomočjo funkcije disp_rgb()

```
void disp_rgb(){
lcd.setCursor(0,1);
if (rd < 100) lcd.print("0");</pre>
```

```
if (rd < 10) lcd.print("0");
lcd.print(rd);
lcd.setCursor(4,1);
if (gn < 100) lcd.print("0");
if (gn < 10) lcd.print("0");
lcd.print(gn);
lcd.setCursor(8,1);
if (bl < 100) lcd.print("0");
if (bl < 10) lcd.print("0");
lcd.print(bl);
} // end disp rgb()
```

in pošljemo podatke preko I2C protokola s pomočjo funkcije send_rgb():

```
void send_rgb(){
Wire.beginTransmission(I2c_address);
Wire.write(rd);
Wire.write(gn);
Wire.write(bl);
Wire.endTransmission();
} // end send_rgb()
```

Ko program detektira, da je pritisnjena tipka SW2, najprej merimo, kako dolgo je tipka bila pritisnjena:

```
void sw2_function(){
for (auto_flag = 0; auto_flag <= 100; auto_
flag++){
if (digitalRead(A2) == 1) {
break;
}
delay(10);
}</pre>
```

Če je tipka bila pritisnjena več kot 1 s, bo vrednost števca auto_flag biti 101 in program preide v avtomatski način dela; če je pritisnjena krajši čas, se program vrača v ročni način dela. Sporočilo o trenutno veljavnem načinu dela se izpisuje na LCDju, nakar se program vrača v glavno zanko:

```
lcd.setCursor(12,0);
if (auto_flag > 100) {
lcd.print("auto");
} else {
lcd.print("");
}
disp_send();
} // end sw2_function()
```

Funkcija sw1_function() se izvršuje vsakič, ko pritisnemo tipko SW1, algoritemsko je identična rešitvi za Bascom-AVR in je tukaj ne bomo analizirali.

Za tiste, ki želi vedeti več

Arduino IDE okolje nima definirane funkcije debounce(), zato jo bomo sami napisali po opisu iz dokumentacije za Bascom

-AVR. Funkcija v razmaku 25 ms dvakrat preveri, ali je tipka pritisnjena, da bi odstranila "lažno" branje zaradi nepopolnosti mehanskih stikal. Funkcija prav tako spremlja, če je tipka spuščena in bo pridruženo funkcijo naslednjič izvršila šele, ko spustimo in ponovno pritisnemo tipko.

V našem programu spremljamo dve tipki, SW1 in SW2, ki sta povezani na Arduino priključke A1 (= 15) i A2 (= 16). Zaradi enostavnosti bomo napisali funkcijo, ki je uporabna samo za priključke A1-A5:

```
void debounce(byte pin, byte pin_state, void
(*function)()){
static uint8_t allpinsstate = 0;
uint8 t pinshift = pin-14;
if (digitalRead(pin) == pin state) {
delay(25);
if (digitalRead(pin) == pin state) {
if (!((allpinsstate & (1 << pinshift)) >>
pinshift)){
function();
allpinsstate |= (1<< pinshift);
}
}
} else {
delay(25);
if (digitalRead(pin) != pin state) {
allpinsstate &= ~(1 << pinshift);
}
}
}
```

Tako napisana debounce() funkcija v popolnosti ustreza postavljenim zahtevam vendar poudarjamo, da ne bo primerna za digitalne priključke 0-13! * * *



Slika 8: Preverjanje RGB modula

Slika 8 potrjuje, da opisani programi delajo tudi v realnosti :)! No, ko že vse tako dobro dela, moramo še preveriti, kaj se bo zgodilo, če s Shielda-A snamemo alfanumerični displej in namesto njega povežemo I2C LCD (slika 9)! Takrat v Bascom-AVR programu delamo konfiguracijo displeja na naslednji način:



Slika 9: RGB modul v kombinaciji z I2C LCD modulom

```
Config Lcd = 16 * 2
$include "I2CLCD.sub"
Lcd$init &B1111
```

Standardni konfiguracijski ukazi LCDa, Config Lcdbus in Config Lcdpin, nam niso več potrebni, z vključenjem knjižnice I2CLCD. sub smo vse Bascomove ukaze za delo z LCDjem preusmerili I2C LCDju. Konstanta &B1111 v Lcd\$init ukazu ustreza I2C LCD modulu s PCF8574A čipom in vsi trije naslovni priključki v stanju "1" (slika 9 zgoraj desno). Knjižnica I2CLCD.sub je funkcionalno identična knjižnici PCF8574_LCD\$SE.sub, o kateri smo podrobno pisali v Svetu elektronike 262, v okviru Arduino serije.

V Arduino IDE je možno za krmiljenje I2C LCDjem instalirati knjižnico "LiquidCrystal I2C". Ta knjižnica ima iste ukaze kot tudi knjižnica "LiquidCrystal" in zato bomo imeli samo nekaj sprememb v programu. Za instaliranje knjižnice je potrebno izbrati padajoči meni "Tools->Manage Libraries..." kot je prikazano na Sliki 10. Z izborom padajočega menija se prikaže dodatno okno "Library Manager" (slika 11). V gornjem desnem polju vpisujemo ključne besede "liquidcrystal i2c" za filtriranje spiska knjižnic. Iskano knjižnico bomo našli na sredini spiska; postavimo kazalnik miške na trak z opisom knjižnice in kliknemo na virtualno tipko "Install". Po instalaciji zapremo okno s klikom na virtualno tipko "Close".

😣 🗇 💿 Shield-A	A_18 Arduino 1.8.16	
<u>Eile Edit Sketch</u>	Tools Help	1.000
	Auto Format Archive Sketch Fix Encoding & Reload	Ctrl+T
void setup()	Manage Libraries	Ctrl+Shift+I
}	Serial Monitor Serial Plotter	Ctrl+Shift+M Ctrl+Shift+L
<pre>void loop() {</pre>	WiFi101 / WiFiNINA Firmware Updater	

Slika 10: Padajoči meni za krmiljenje s knjižnicami

Clbrary M	lanager	And the second se	
ype All	+ Topic All	* liquidcrystal i2c	
A library with displays (u8git More info	you can generate a menu' s ba b),	ed on the nested set model with multi layers Supports serial mont	tor, liquidorystal, iZe, graphic
LiquidCrystal	12C		
by Marce Schw A library for E MIGHT NOT BE More info	vartz 2C LCD displays. The library allo COMPATIBLE WITH EXISTING SK	es to control IZC displays with functions extremely similar to LiquidC FTCHES.	rystal library. THIS LIBRARY
			Version 1.1.2 * Install
LiquidCrystal	AIP31068		
by Andriy Gala A library for A LiquidCrystal lit More info	JYNYA JP31068 12C/SPI LCD display brary. THIS LIBRARY MIGHT NOT	. The library allows to control AIP31068 based I2C/SPI displays with E COMPATIBLE WITH EXISTING SKETCHES.	functions extremely similar to
theddenated	I2C_Hangul		
LiquidCryscal			

Slika 11: Okno za krmiljenje s knjižnicami

Program bomo začeli z navajanjem knjižnic, ki jih bomo uporabljali:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

Za razliko od Bascom knjižnice, knjižnica LiquidCrystal_I2C ustvari zasebne objekte in zato ni možno definirati I2C naslova LCDja v spremenljivki, da bi z njeno pomočjo krmilili z LCDji. Zato bomo definirali objekt za LCD in ga poimenovali "lcd":

```
LiquidCrystal_I2C lcd(0x27,16,2);
```

Sočasno smo navedli tudi kateri I2C naslov uporabljamo za ta objekt in vrsto displeja. Ostane nam še, da v funkciji setup() inicijaliziramo objekt lcd, in vključimo osvetlitev ozadja s pomočjo funkcije backlight():

```
lcd.init();
lcd.backlight();
```

S temi ukazi smo zamenjali "standardne" ukaze za konfiguracijo LCDja in vklop osvetlitve ozadja s pomočjo tranzistorskega stikala T2, in zato jih več ne potrebujemo:

```
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);
...
pinMode(17, OUTPUT);
digitalWrite(17, HIGH);
...
lcd.begin(16, 2);
```

Ostali ukazi so identični.

Opomba: Programe ATtiny85_RGB_slave_1.ino, Shield-A_ RGB_master.bas, Shield-A_RGB_master.ino, Shield-A_RGB_ master_2.ino in knjižnico I2CLCD.sub lahko brezplačno dobite od uredništva revije Svet elektronike!

Viri:

https://github.com/SpenceKonde/ATTinyCore



https://svet-el.si

KAKO, KJE in KAJ potrebujem za naročilo?

- Naročilo je možno poslati po pošti (AX ELEKTRONIKA d.o.o., Depala vas 39, 1230 Domžale), po telefonu (01 528 56 88 ali 01 549 14 00) ali e-pošti (prodaja04@svet-el.si). Naročeni material pošiljamo preko Pošte Slovenije.
- Garancija za gotove izdelke velja 12 mesecev (datum na računu), KIT kompleti nimajo garancije.
- Plačevanje je možno po povzetju (plačilo ob prevzemu), na obroke (2 obroka), po predračunu, kreditnimi karticami ali po vnaprej dogovorjenem plačilnem roku!
- Naročene izdelke pošljemo najkasneje v roku 48 urah od prejema naročila oziroma vam sporočimo predvideni rok dobave. Vračilo izdelkov je možno v osmih dneh po prevzemu. Kontaktna oseba za naročila in vprašanja je Nataša Stružnik.
- Katerikoli <u>brezplačni PDF letnik revije Svet elektronike</u> si lahko izbere vsak novi naročnik ali obstoječi naročnik, ki podaljša naročnino.
- Popust na vse stare letnike revije Svet elektronike v PDF in v pisni obliki imajo vsi trenutni naročniki na revijo Svet elektronike.
- Pri obeh naročninah (pisni + internet) dobite internet naročnino za 50% ceneje.
- Konec leta vsak naročnik prejme stenski planer.

Naročnine na revijo Svet elektronike

- **PRAVNE OSEBE (1 leto)**. Naročnina na revijo Svet elektronike, za pravne osebe. Naročnina velja eno leto (11 številk, julij/avgust dvojna). Cena naročnine z vštetim popustom je <u>44,95</u>
 EUR. Plačilo po predračunu, katerega pošljemo po pošti.
- FIZIČNE OSEBE (1 leto). Naročnina na revijo Svet elektronike, za fizične osebe. Naročnina velja eno leto (11 številk, julij/avgust dvojna). Cena naročnine z vštetim popustom je 39,95 EUR. Plačilo po položnici, ki jo pošljemo po pošti.
- ŠOLAJOČE SE OSEBE (1 leto, potrdilo o šolanju). Naročnina na revijo Svet elektronike, za šolajoče se osebe. Naročnina velja eno leto (11 številk, julij / avgust dvojna). Cena naročnine z vštetim popustom je <u>37,46 EUR</u>. Plačilo po položnici, ki jo pošljemo po pošti. Brez potrdila o šolanju se naročniku avtomatično pošlje naročnino z 20% popustom.
- INTERNET NAROČNIKI (1 leto, fizične ali pravne osebe). Naročnina na internet revijo Svet elektronike. Naročnina velja eno leto (vpogled revije v PDF datoteki na www.svet-el.si). Cena naročnine znaša 19,99 EUR. Nujna je prijava na spletni strani, kjer si lahko ogledate tudi svoj vse informacije glede naročnine.
- VSI NAROČNIKI (-50% popusta pri internetni naročnini 1 leto). Pri naročilu na pisno revijo Svet elektronike in internet naročnino vam za internetno naročnino priznamo <u>50%</u> popust. Izberite si želeno pisno naročnino in jo obkrožite skupaj z internet naročnino. Vsi pogoji ostanejo enaki, lahko si jih ogledate v zgornjih naročninah. Za vse ostale informacije smo vam na voljo na tel.: 01 549 14 00 ali e-naslov: prodaja04@svet-el.si.
- AVTORJI člankov imajo brezplačno pisno naročnino (svojo naročnino lahko tudi podarijo komur koli)

Vse cene naročnin in izdelkov »od podjetja AX, d.o.o. in revije Svet elektronike« vsebujejo DDV.

Več naročnin vam prihrani denar. Pravnim osebam, ki naročijo več izvodov

revije Svet elektronike, nudimo za 2. naročen izvod 50% popust, za

3. izvod in vse naslednje pa 70% popust do preklica. Velja tudi za

podaljšanje naročnine. Vsi izvodi revije morajo imeti istega plačnika.

Naročilnica za revijo Svet elektronike

podjetje / fizična oseba (ime in priimek)

ulica / hišna številka / pošta / kraj

davčna številka / zavezanec (da ali ne)

telefon

e-pošta

podpis / žig

Podarite naročnino ali darilni BON • Obdarovanje svojih najbližjih

je vsako leto težje. Imamo že toliko stvari, da ne vemo več kaj potrebujemo in kaj si v življenju res želimo, zato je obdarovanje včasih težko, ker ne vemo natančno kaj podariti. V uredništvu revije Svet elektronike smo za take primere pripravili nekaj novosti. Lahko podarite naročnino na revijo Svet elektronike ali vrednostni BON. Oboje vam

olajša odločitev kaj podariti.

7	
-	
	1 -2 -2 -
	ZA
	to/oi



CUEKTRONIK	Ime in Priimeks
	terip Son Addisona je smachk Resjo Son Addisona je smachk Randona at objete 11 terik Vadansezika, klabi loter na Rospidje PDP kask relj no nasj Rose nas prjeti kravje

Brezplačni PDF letnik za naročnika

 Svet elektronike nagradi vsakega naročnika z brezplačnim letnikom preteklih revij v PDF obliki od leta 2004 po svoji izbiri. Vsak naročnik se ob

podaljšanju naročnine odloči, kateri letnik bi želel prejeti. Svojo odločitev nam lahko sporočite po elektronski pošti, telefonu ali preko virtualne trgovine.



Brezplačno vsi letniki do 2006

Vsi letniki do 2006 so sedaj brezplačno na naši spletni strani!



Vsi naročniki

 50% popusta pri internetni naročnini 1 leto. Pri naročilu na pisno revijo Svet elektronike in internet

naročnino, vam za internetno naročnino priznamo **50% popust**. Izberite si želeno pisno naročnino, ter jo obkrožite skupaj z internet naročnino.



Brezplačni ogledni izvod

 Verjamemo, da se želite prepričati, zakaj je Svet elektronike najboljša revija za prave elektronike. Ker smo ponosni na to, kar delamo, vam bomo z veseljem poslali brezplačni ogledni izvod na vaš naslov - seveda brez zaračunanih stroškov poštnine!

STIK



Smo uradni zastopnik za ELNEC v Slovniji!



AX, d.o.o. • Depala vas 39 • 1230 Domžale • 01 528 56 88 • https://trgovina.svet-el.si • stik@svet-el.si



AX ELEKTRONIKA

110001100110101010 100011001101010 1100011001101010 01010101010111001010110011 1100110100110



TRÓNIKA

Rebetics

uluiluuluuluu

.....



powered by **i C** *M* www.icm.si