

PROGRAMIRANJE IN ZVOK

impe but

LABBOOK

TEENSYnth

NAREDI SAM (DIY)



LABBOOK

TEENSYnth

Programiranje in zvok.3

Avtorica: Tara Pattenden

V tem učnem gradivu oziroma labbooku so navodila, po katerih boste lahko sestavili TEENSYnth. S TEENSYnthom boste lahko za proizvajanje zvoka uporabili sadeže! Naprava je izdelana na osnovi Teensyja – mikrokrmilnika, ki ga je mogoče programirati za najrazličnejše stvari.

Stopnje težavnosti

- 1. RADOVEDNEŽ Primerno za vsakogar
- RAZISKOVALEC Ne poznam področja, ampak bom zmogel z malo razmišljanja
- 3. POZNAVALEC Imam dovolj znanja za samostojno delo
- MOJSTER Sem kar vešč, vstopam v polje poglobljenega razumevanja umetniških in znanstvenih trikov
- 5. RAZVIJALEC Znam že toliko, da lahko vodim druge iz kategorij 1–3
- 6. MENTOR Razumem vsebine, obvladam tehnologije, samostojno razvijam in posredujem znanja



Creative Commons: To delo je objavljeno pod licenco Creative Commons priznanje avtorstva 4.0 mednarodna.

KAZALO

Kaj je TEENSYnth?	4
Kaj potrebujemo?	4
Namestitev programske opreme za Arduino	6
Kaj je Teensy?	12
Priprava Teensyja za uporabo na testni plošči	15
Vaja 1 – zvoki sadežev	18
Vaja 2 – spreminjanje višine tona/ogled kode	22
Vaja 3 – branje vhodnih signalov	25
Programiranje Teensyja	27
Vaja 4 – dodajanje več sadežev	38
Vaja 5 – uravnavanje višine tona	42
Priloge	46
O avtorici	48
Kolofon	49

Kaj je TEENSYnth?

To so navodila, po katerih boste lahko sestavili TEENSYnth. Z njim boste lahko za proizvajanje zvoka uporabili sadeže! Naprava je izdelana na osnovi Teensyja – mikrokrmilnika, ki ga je mogoče programirati za najrazličnejše stvari. To učno gradivo vključuje tudi druge projekte in vire; tako boste lahko s Teensyjem po zaključku delavnice še naprej sestavljali različne naprave. Zdaj bomo sestavili eno tako napravo in z njo raziskali svet zvoka in senzorjev.

Kaj potrebujemo?

- 1. Teensy 3.2 z nogicami oz. pini
- 2. testna plošča
- 3. približno 10 skokic (jumper wires)
- 4. žica (približno 15–20 cm dolžine)
- 5. vtičnica za 3,5-milimetrski jack
- 6. slušalke ali zvočnik
- 7. podatkovni USB-kabel (micro-USB-USB-A)
- 8. računalnik z naloženo programsko opremo Arduino
- 9. sadje

OPCIJSKO:

- 10. 2 x 10K LIN potenciometer
- 11. LDR upor



Namestitev programske opreme za Arduino

Pred nami je namestitev programske opreme za Arduino v vaš računalnik in njen preizkus.

1. korak – namestitev vse potrebne programske opreme

S krmilnikom Teensy komuniciramo z uporabo programske opreme za Arduino. Brez nje Teensyja ne boste mogli programirati.

Prenesite si naslednja dva programa:

1. Arduino IDE

https://www.arduino.cc/en/Main/Software

POZOR: Če poganjate sistem Windows, pazite, da ne izberete možnosti Windows app, saj vam ta ne omogoča namestitve dodatne programske opreme Teensyduino, ki jo bomo tudi potrebovali.





2. Teensyduino

https://www.pjrc.com/teensy/td_download.html

Namestitev programske opreme za Arduino

Namestite in zaženite program Arduino. Zaprite program Arduino.



Namestitev programske opreme Teensyduino

Program Arduino nima vgrajene podpore za krmilnike Teensy, zato morate zagnati tudi namestitveni program Teensyduino in na ta način programski opremi za Arduino dodati module za podporo krmilnikom Teensy.

Namestitveni program vas bo vprašal po mestu, kjer je nameščena programska oprema za Arduino. Izberite mesto namestitve. Gumb Next (Naprej) se aktivira šele, ko izberete mapo, v kateri je programska oprema za Arduino.



Klikajte Next, dokler namestitev ni končana.

Za uporabnike sistema Windows

Če uporabljate operacijski sistem Windows, morate zagnati tudi program Windows Serial Installer, ki si ga prenesete z naslova <u>https://www.pjrc.com/</u> <u>teensy/serial install.exe</u>. Izberite možnost USB Serial, ki čarovniku Odkrita nova programska oprema (Windows Found New Hardware Wizard) sistema Windows omogoča najti primeren gonilnik za Teensy.



2. korak – preizkus programske opreme za Arduino

V Teensy bomo naložili program, ki sproži utripanje svetlobne diode na krmilniku, in tako preverili, ali programska oprema deluje pravilno.

To storite tako, da odprete datoteko: File > Examples > 01. Basics > Blink.

Zdaj je treba program poslati v krmilnik Teensy.

Najprej je potrebnega nekaj nastavljanja:

Izbira prave plošče (board) v programu Arduino

Prepričajte se, da je v programu nastavljena prava plošča. To storite v meniju z orodji (*Tools*).

Tools > Board: Teensy 3.2/3.1.

1	Tools	Help			
3:		Auto Format	Ctrl+T		
		Archive Sketch			
		Fix Encoding & Reload			
		Manage Libraries	Ctrl+Shift+I		
		Serial Monitor	Ctrl+Shift+M		
		Serial Plotter	Ctrl+Shift+L		
		WiFi101 / WiFiNINA Firmware Updater			
		Board: "Teensy 3.2 / 3.1"	;	-	Boards Manager
	USB Type: "Serial"		;		
		CPU Speed: "96 MHz (overclock)"	>		Teensyduino
		Optimize: "Faster"	3		Teensy 4.1
		Keyboard Layout: "US English"	2		Teensy 4.0
		Port			Teensy 3.6
		Get Board Info			Teensy 3.5
		Programmer: "AVRISP mkll"	;	•	Teensy 3.2 / 3.1
		Burn Bootloader			Teensy 3.0
					Teensy LC
					Teensy++ 2.0

Izbira vrat (port) v programu Arduino

V meniju z orodji (Tools > Port) izberite vrata, kjer zraven piše Teensy.



3. korak – Compile & Upload (nalaganje kode)

Zdaj bomo kodo sestavili in jo naložili na Teensy. Sestavljanje kode je postopek, ki ga izvaja programska oprema; v njem se koda pretvori v obliko, ki jo Teensy razume.

S pomočjo USB-kabla priključite Teensy na svoj računalnik.

V programu Arduino pojdite v meni *Tools > Boards* in tam izberite *Teensy* 3.2.

Nato pojdite v meni *Tools > Port* in izberite vrata, kjer zraven piše *Teensy*.

Zdaj kliknite gumb za nalaganje z ikono puščice, ki ga najdete zgoraj desno od kode programa.



Program bo preveril veljavnost kode, jo sestavil in nato naložil na krmilnik Teensy.

Gumb z ikono kljukice kodo sestavi in preveri njeno veljavnost, a je ne naloži na priključeni mikrokrmilnik.

Ko se koda za utripanje prenese na vaš Teensy, bi morala ledica začeti utripati.

Pogoste težave pri nameščanju programske opreme

Če se vaša koda noče naložiti, obstaja za to nekaj pogostih vzrokov. Če je težava povezana s krmilnikom, se na dnu zaslona prikaže sporočilo o napaki.

Tu, na primer, Teensy ni priključen na računalnik.



Če se vaš program ne naloži, preverite naslednje:

Ali je USB-kabel, ki ga uporabljate, primeren za prenos podatkov?

Ste izbrali prava vrata?

Ste izbrali pravo ploščo?

Poglejte, če vas v orodju Teensy Loader čaka kak napotek.

https://coolcomponents.co.uk/blogs/news/common-arduino-issues



Teensy Loader

Vaš Teensy s programom Arduino komunicira prek orodja Teensy Loader. To je del programske opreme Teensyduino, ki ste jo namestili maloprej. Izvaja se v ozadju in vaše pozornosti običajno ne potrebuje, a včasih javi napako in izpiše opozorilo, da je potrebno ročno upravljanje. V tem primeru je treba pritisniti tipko na Teensyju, da vanj program naložite v ročnem načinu.



Kaj je Teensy?

Najprej si poglejmo, kaj Teensy je. Teensy je vrsta mikrokrmilnika. Mikrokrmilnik je nekakšen majhen računalnik, ki lahko poganja različne programe. Ima 22 priključkov, ki jih lahko uporabljate bodisi kot vhodne bodisi kot izhodne priključke – to se določi v programu. Zaznavali bomo dotik jabolka in ob vsakem dotiku predvajali zvok.

Teensy je na voljo v številnih izvedbah. Mi bomo uporabili Teensy 3.2, saj ta zadostuje našim potrebam.

Pred začetkom morate v Teensy naložiti program, ki opravlja želeno nalogo. O samem programu bomo več povedali kasneje.

Na Teensyju je devet priključkov, ki omogočajo kapacitivno zaznavanje dotika. S pomočjo te funkcije bomo dosegli, da bo košček sadja proizvajal zvok.

Kaj je kapacitivno zaznavanje dotika?

Kapacitivno zaznavanje dotika je tehnologija, ki omogoča zaznavanje bližine. Kapacitivni senzorji delujejo tako, da ustvarijo električno polje in nato zaznavajo motnje v tem polju, ki jih povzročijo predmeti v bližini.

Kapacitivni senzorji so vse okrog nas. Isto tehnologijo za zaznavanje dotikov uporabljajo zasloni pametnih telefonov.

Za zaznavanje dotikov lahko uporabimo mnoge vsakdanje predmete – edini pogoj je, da prevajajo elektriko. Večina kovinskih predmetov dobro prevaja elektriko. Prav tako tudi sadje in druge organske snovi, kar pomeni, da jih lahko uporabimo kot senzorje. A ni to kul?

Oglejmo si Teensy 3.2



Vaš Teensy bo najverjetneje že opremljen s pini oziroma nogicami in bo videti takole.

Vsaka od teh nogic na Teensyju je povezana z vhodnim oz. izhodnim priključkom. Tem nogicam rečemo pini. Pine lahko programiramo za najrazličnejše naloge. Mi jih bomo uporabili za kapacitivno zaznavanje dotika.

Za kapacitivno zaznavanje dotika lahko uporabimo pine 0, 1, 15, 16, 17, 18, 19, 22 in 23.



Testna plošča oz. breadboard

To je testna plošča, ki omogoča uporabo brez spajkanja. Omogoča vam, da s Teensyjem brez spajkanja povežete razne naprave.



V notranjosti testne plošče so bakreni trakci, ki medsebojno povezujejo luknjice v isti vrsti.

Priprava Teensyja za uporabo na testni plošči

Odvisno od tega, kakšne komponente imate, vas morda čaka nekaj spajkanja. O spajkanju govori super strip z naslovom Soldering is easy (Spajkanje je preprosto), ki ga lahko zastonj prenesete na naslovu <u>https://mightyohm.com/files/soldercomic/FullSolderComic_EN.pdf</u>.

Teensy

Teensy je na voljo tako s pini kot brez njih. Če vaš nima pinov, jih boste morali vzdolž obeh dolgih stranic naspajkati sami, da ga boste lahko uporabljali s testno ploščo.



Teensy 3.2 s pini



Teensy 3.2 brez pinov

Če vaš Teensy nima pinov, lahko za njihovo izdelavo uporabite takšno letvico.



Namig: vstavite pine v testno ploščo in nataknite Teensy nanje, da se med spajkanjem ne bodo premikali. Spajkajte čim hitreje, da se pini preveč ne segrejejo, saj lahko sicer začnejo taliti testno ploščo. Za dostop do priključkov na spodnji, krajši stranici Teensyja bomo nanje prispajkali letvico z vhodnimi priključki; tako bomo lahko do njih dostopali, ne bodo pa povezani s testno ploščo. Če bi na priključke prispajkali pine, bi se ti znašli v medsebojno povezanih luknjicah testne plošče, in sicer v isti vrstici kot pin 13, to pa bi Teensyju povzročalo težave. Vaš Teensy bi moral imeti vhodne priključke na tej spodnji stranici prispajkane že vnaprej. Če jih nima, jih boste morali prispajkati sami.



Zvok

Če vaša vtičnica avdio konektorja (jack) ni opremljena z žičkami, jih boste morali prispajkati sami. Ena nogica avdio konektorja je namenjena ozemljitvi, druga pa zvočnemu izhodu.



Avdio konektor bomo uporabili za priklop zvočnika na DAC-pretvornik (več spodaj). To je stereo konektor. Teensy je opremljen z mono izhodom. Uporabite lahko stereo ali mono konektor.

Kako delujejo avdio konektorji?

Konektorji tipov TS in TRS se pogosto uporabljajo za avdio namene. Na voljo so v različnih velikostih, najpogostejša sta 3,5 mm (mini jack vtičnica) in 6,35 mm (kitarska jack vtičnica). TS je mono, TRS pa stereo konektor. Te oznake pomenijo: Tip Sleeve in Tip Ring Sleeve.



Ta slika prikazuje 6,5-milimetrska vtiča v izvedbi TRS in TS. Tip (konica), ring (obroček) in sleeve (rokav) so različne medsebojno ločene povezave. Rokav je vezan na ozemljitev, konica in obroček pa na mono oziroma stereo kanal.



Vtič se z vtičnico poveže takole:



DAC izhod na Teensyju

Avdio izhod (DAC oz. Digital Analogue Converter – digitalno-analogni pretvornik) je na Teensyju na pinu A14/DAC. Ta je na spodnji stranici Teensyja, torej tisti, ki leži nasproti USB-priključku.

Vaja 1 – zvoki sadežev

1. korak

Prenesite si projektne datoteke. Datoteke, ki jih potrebujete, so na voljo na naslovu: <u>https://gitlab.com/kons-platforma/teensynth</u>

T TEENSYnth (Proc. 10. 343000) In Michael III (Michael Lond Workshop Taxa Pattern	ම ි කි. ඒ ම කළා වු 1.88 මංක 1110ක හාරා	Million 🗟 La Millionay		(0 × (0 m) ≥ (¥ res
main - ban	ayah J + +		Haloy Fed D	Vest + d + Dee
they branch 'warder	f lines header (inc)			Lamborac (
1 Lipitani File 🔯 MitAZINE	(B AN LICENE ID SH NPCYCD	R An Develop	E AN CONTRACTOR	🗟 Instite Auto DevOps
Name		Last commit		Last up de
Es the Dackers Touch + Pa	a	initial spread		110/1
D Die Declarer teach		initial accord		1 hours
The Ostinter Touch in Physical		initial sprenit		3 hour a
The collision such		initial commit		1 hour a
HINCHLON		initial assessed		Theory
B READINE and				
TEENsynth	nti propiti			



Kliknite ikonco Download, in si prenesite vse datoteke, tako da izberete možnost Download ZIP. Shranite jih v mapo na svojem računalniku.

2. korak

Položite Teensy na testno ploščo tako, da bo premostil sredinski žleb na plošči. Povežite pina VCC in GND na Teensyju s testno ploščo. Nekatere komponente, ki jih bomo dodali, morajo biti priključene na vir napetosti in na ozemljitev. Kot vir napetosti bomo uporabili Teensy, zato ga bomo povezali s stranskima vodiloma, da ju bomo lahko uporabili kot pozitivni (+) in negativni (-) priključek.



Priključite barvno žico avdio konektorja na DAC-izhod, ki je na nasprotni strani Teensyjeve ploščice USB-vhoda (po potrebi preverite shemo na strani 7), črno žico avdio konektorja pa povežite z ozemljitvijo (GND) na testni plošči.



4. korak

Priključite jabolko na pin 23 (A9). Preprosto zapičite en košček žice v jabolko!



Priključite USB-kabel za napajanje Teensyja.



6. korak

S programom Arduino odprite datoteko one osc touch.ino. Sestavite kodo in jo tako kot prej naložite v svoj Teensy s klikom na ikono puščice.





7. korak

V avdio konektor vklopite slušalke.



Dotaknite se jabolka; ob dotiku bi morali zaslišati zvok. UAU!



Zdaj lahko eksperimentiramo z različnimi drugimi materiali namesto jabolka, da vidimo, kateri od njih so prevodni – nadomestite jabolko s kakim drugim predmetom in preverite, ali prevaja elektriko. Seznam materialov, ki jih lahko preizkusite, najdete v prilogi te knjižice.

Vaja 2 – spreminjanje višine tona/ogled kode

Zdaj se bomo malo poigrali s kodo, da spremenimo zvok. Kako točno koda deluje, bomo razložili kasneje; za zdaj bomo samo spremenili ustrezno vrednost.

1. korak – odprite program Arduino

Odprite dokument one_osc_touch.ino.

To je program, ki ga morate naložiti na Teensy, da ob dotiku sadja sproži predvajanje zvoka.

Vklopite oštevilčevanje vrstic

Številke vrstic bomo kasneje uporabili za navigacijo po kodi. Za vklop prikaza številk vrstic kode v programu Arduino odprite meni *File* (Datoteka) in izberite možnost *Preferences* (Nastavitve). Tu označite oz. potrdite okvirček *Display line numbers* (Prikaži številke vrstic).

Preferences		×		
Settings Network				
Sketchbook location:				
C: \Users\Tara\Documents\A	Arduino Bro	owse		
Editor language:	System Default v (requires restart of Arduino)			
Editor font size:	12			
Interface scale:	✓ Automatic 100 ÷ % (requires restart of Arduino)			
Theme:	Default theme \lor (requires restart of Arduino)			
Show verbose output during: Compilation Upload				
Compiler warnings:	None 🗸			
Display line numbers	Enable Code Folding			
Verify code after upload	Use external editor			
Check for updates on sta	tartup Save when verifying or uploading			
Use accessibility features				
Additional Boards Manager U	RLs:			
More preferences can be edited directly in the file				
C: Users (Tara \AppData \Local \Arduino 15) preferences.txt				
(edit only when Arduino is no	ot running)			
	OK C	ancel		

2. korak – spremenite kodo

Pojdite na 27. vrstico kode.

Ta je videti tako:

waveform1.begin(0, 100, WAVEFORM_TRIANGLE);



V tej vrstici nastavimo lastnosti zvoka, v tem primeru glasnost, višino in valovno obliko.

Spremeniti hočemo višino zvoka. To je druga številka v oklepaju, ki predstavlja frekvenco v hercih (Hz).

Razpon človeškega sluha je običajno med 20 in 20.000 Hz, a so med posamezniki bistvene razlike. 20 Hz je zelo globok, 20.000 Hz pa zelo visok zvok.

https://www.szynalski.com/tone-generator/

Spremenimo to drugo številko – izberite nekaj med 20 in 20.000.



3. korak – sestavljanje in nalaganje kode (Compile & Upload)

Zdaj bomo kodo sestavili in jo naložili na Teensy. Sestavljanje kode oz. kompilacija je postopek, ki ga izvaja programska oprema; v njem se koda pretvori v obliko, ki jo Teensy razume.

S pomočjo USB-kabla priključite Teensy na svoj računalnik.

V programu Arduino pojdite v meni *Tools > Boards* in tam izberite *Teensy* 3.2.

Nato pojdite v meni *Tools > Port* in izberite vrata, kjer zraven piše *Teensy*.

Zdaj kliknite gumb za nalaganje z ikono puščice, ki jo najdete zgoraj desno od kode programa

Program bo preveril veljavnost kode, jo sestavil in nato naložil na krmilnik Teensy.

Gumb z ikono kljukice kodo sestavi in preveri njeno veljavnost, a je ne naloži na priključeni mikrokrmilnik.

Dotaknite se jabolka in bodite pozorni na to, kako je zvok zdaj drugačen.

4. korak – prisluhnite različnim frekvencam

Ponavljajte koraka 2 in 3 in preizkusite nekaj različnih frekvenc. Tu je seznam s frekvencami, ki ustrezajo posameznim tonom glasbene lestvice.

https://pages.mtu.edu/~suits/notefreqs.html

5. korak – prisluhnite različnim valovnim oblikam

Zdaj pa poslušajmo, kako se v zvoku odražajo različne valovne oblike. Spremenite vrsto valovne oblike v vrstici 28 v eno od teh na spodnjem seznamu; valovna oblika je navedena takoj za frekvenco in je trenutno nastavljena na WAVEFORM_TRIANGLE. Po spremembi valovne oblike kodo sestavite in naložite ter prisluhnite spremembi zvoka.



Vaja 3 – branje vhodnih signalov

V tej vaji si bomo s pomočjo terminala ogledali, kakšni podatki se izmenjujejo, ko se dotaknemo jabolka.

1. korak – odprite terminal (Serial Monitor)

Preverite, ali je Teensy priključen, in v programu Arduino odprite terminal, tako da kliknete na ikono povečevalnega stekla v zgornjem desnem kotu. Odprlo se bo okno, v katerem od zgoraj dol tečejo številke. Terminal je zelo uporabno orodje, s katerim lahko spremljamo, kaj se dogaja s Teensyjem.

Serial Monitor 👂

2. korak – dotaknite se jabolka

Medtem ko spremljate terminal, se dotaknite jabolka; videli boste, da se ob tem številke spremenijo.



3. korak – spremenite občutljivost

Če bi radi povečali ali zmanjšali občutljivost na dotik vašega »jabolčnega upravljalnika«, lahko to dosežete s spreminjanjem vrednosti, ki jo uporabljamo kot prag zaznavanja dotika jabolka. Sprememba te številke je potrebna, na primer, če jabolko ne zazna, da smo se ga dotaknili, ali pa dotik zaznava ves čas, tudi ko se ga ne nihče ne dotika. Ta številka mora biti višja od številk, ki se v terminalu izpisujejo, ko se jabolka ne dotikamo, in nižja od tistih, ki se izpisujejo, ko se jabolka dotikamo.

Številka za prag zaznavanja je v vrstici 19 našega programa.

```
int thresh = 2300; // variable to store the touch threshold
```

Po naših izkušnjah je vrednost 2300 primerna za uporabo s celo vrsto različnih sadežev. Med drugim tudi z jabolki.

4. korak – nalaganje in preizkus kode

Sestavite in naložite spremenjeno kodo in se poigrajte z jabolkom, da vidite, na kakšen način se odraža sprememba.

Kaj je terminal (Serial Monitor)?

S kodo spodaj določimo, katere informacije se bodo izpisovale v terminalu.

V okviru začetne nastavitve se v vrstici 31 najprej povežemo s terminalom, in sicer s pomočjo te kode: če te kode ni, se Teensy ne bo povezal s terminalom.

Serial.begin(9600);

V tistem delu kode, ki je znotraj zanke, določimo, kateri podatek želimo spremljati s pomočjo terminala. To storimo s to kodo v vrstici 63: v tem primeru je vrednost current spremenljivka, v kateri je zabeležena trenutna vrednost na vhodu sistema za kapacitivno zaznavanje dotika.

Serial.println(current);

Programiranje Teensyja

V tem poglavju si bomo podrobneje ogledali kodo, ki nam omogoča, da s pomočjo kapacitivnega zaznavanja dotika »igramo na jabolko«.

Orodje za oblikovanje zvoka Teensy

Orodje za oblikovanje zvoka Teensy se imenuje The Audio System Design Tool. To orodje uporabljamo za oblikovanje zvoka, ki bi ga radi prožili z uporabo kode. Omogoča nam oblikovanje zvokov, zvočnih učinkov in mikserjev na vizualen način. Kodo, ki je potrebna za nastavljanje sintetizatorja zvoka, ustvari samodejno in vam na ta način prihrani veliko časa. Tako je videti shema zvočnega sistema za projekt zaznavanja dotika za en sadež.



Nekaj uporabnih informacij

Program za Teensy je napisan v jeziku C++ v programu Arduino. Ko napišete kodo, je treba preveriti njeno veljavnost in jo sestaviti, da jo lahko naložite na Teensyja. V postopku preverjanja veljavnosti se izvede pregled kode za morebitne napake – koda razlikuje med velikimi in malimi črkami, pomembna pa so tudi uporabljena ločila. Za začetek bomo uporabili kodo v dokumentu. Naj vas to, da o programiranju ne vemo še nič, ne skrbi – za zdaj bomo zgolj zaganjali in spreminjali obstoječo kodo. To je dober učni pristop.

// Dve poševnici pred vrstico pomenita, da se ta vrstica uporablja zgolj kot opomba

// in se pri sestavljanju oz. kompilaciji programa ne upošteva. Takšne opombe lahko uporabite,

// da razložite delovanje kode.

Če pogledate kodo v programu Arduino, boste videli, da je v njej veliko opomb, ki razlagajo delovanje kode.

Poglejmo si kodo

V programu Arduino odprite datoteko one_osc_touch.ino. To je koda za proizvajanje zvoka z dotikanjem sadežev, ki je trenutno naložena v Teensyju. Oglejmo si torej kodo, ki to omogoča. Koncepte in kodo bomo razložili sproti.

//One oscillator capacitive touch // uporabili bomo kodo, ki jo je spisal Sebastian Tomczak // https://little-scale.blogspot.com/2017/05/teensy-36-basicstouchread.html // spodnja koda naloži vse knjižnice, ki jih potrebuje Teensy #include <Audio.h> #include <Audio.h> #include <Wire.h> #include <SPI.h> #include <SPI.h> #include <SD.h> #include <SerialFlash.h>

Nalaganje knjižnic

To je uvodni del dokumenta, kjer navedemo informacije o tem, kaj je namen programa, in naložimo potrebne knjižnice (knjižnice prepoznamo po končnici .h). Kot boste videli kasneje, kodo v knjižnicah za nas ustvari orodje za oblikovanje zvoka.

// GUItool: begin automatically generated code AudioSynthWaveform waveform1; //xy=387,283 AudioOutputAnalog dac1; //xy=549,283 AudioConnection patchCord1(waveform1, dac1); // GUItool: end automatically generated code

Ta koda pripravi sintetizatorje zvoka, ki jih bomo uporabili, napisalo oz. ustvarilo pa jo je orodje za oblikovanje zvoka.

Kaj je spremenljivka (variable) v programiranju?

Tu pripravimo spremenljivke. Spremenljivke so nekakšne virtualne posode, v katere lahko pod poljubnim imenom shranjujemo podatke. Pri uporabi Arduina spremenljivko ustvarimo tako, da najprej deklariramo oz. najavimo vrsto spremenljivke in ji dodelimo ime. Po želji ji lahko takoj tudi že dodelimo vrednost, ni pa to nujno potrebno.

Pri programiranju se spremenljivke uporabljajo iz številnih razlogov. Nekateri med njimi so: **Učinkovitost** – spremenljivke uporabljamo, da si prihranimo čas in si olajšamo spreminjanje kode v prihodnje. Pripravili smo, recimo, spremenljivko z imenom *touchRead_pin*, v kateri je shranjena številka vhodnega pina, ki ga bomo uporabili za zaznavanje dotika; ta spremenljivka je kasneje v kodi večkrat uporabljena oz. se nanjo večkrat sklicujemo. Če želimo spremeniti številko, lahko to storimo tu v spremenljivki. Namesto spremenljivke bi lahko v kodi povsod neposredno uporabili številko pina, torej A9, a če bi kasneje želeli to številko spremeniti, bi morali pregledati celotno kodo, poiskati vsa mesta, kjer je uporabljena, in jo povsod spremeniti. V tem primeru je veliko preprosteje uporabiti spremenljivko.

Shranjevanje dinamičnih podatkov – še en razlog za uporabo spremenljivk je shranjevanje podatkov, ki se spreminjajo glede na neko vhodno vrednost.

Obstaja več različnih vrst oz. tipov spremenljivk, ki so namenjene shranjevanju različnih vrst oz. tipov podatkov. Mi uporabljamo tipa *int* in *float*.

Float – vrednost tipa float oz. vrednost s plavajočo vejico mora biti številska in ima lahko decimalke.

Int – integers oz. cela števila so primarni tip podatkov za shranjevanje številskih vrednosti. Te vrednosti so lahko le cele številke brez decimalk.

Kaj je funkcija?

//set up variables
int touchRead_pin = 0; //variable to store the input pin
int thresh = 2300; // variable to store the touch threshold
int play_flag = 0; // variable that flags if sound is being played or not 1
= playing; 0 = not playing
int current; // variable used to store the value from the touch input pin

Void Setup()

Ta funkcija je nujno potrebna za poganjanje kode. Koda, ki je med zavitima oklepajema {}, se izvede enkrat, in sicer na samem začetku izvajanja programa.

void setup() {
// put your setup code here, to run once:

AudioMemory

S to funkcijo določite, koliko spomina v Teensyju bo dodeljenega v uporabo izključno zvočni knjižnici. VNESITE KOLIČINO!

AudioMemory(50); // Dynamic memory is allocated to be used exclusively by the Audio library

Waveform.begin()

Začetna nastavitev zvoka sintetizatorja s parametri glasnosti, frekvence in valovne oblike.

waveform1.begin(0, 100, WAVEFORM_TRIANGLE);
// ta del kode določa parametre valovne oblike (glasnost, frekvenca,
tip valovne oblike)

Terminal (Serial Monitor)

Terminal omogoča spremljanje podatkov iz vhodnih priključkov. Ta koda pripravi teren za komunikacijo s terminalom; kasneje ji povemo še, katere podatke naj spremlja. 9600 predstavlja hitrost, s katero se podatki prenašajo v terminal. Hitrost prenosa podatkov merimo v bitih na sekundo (baudih).

// initialize serial communication with computer - sets up the serial // monitor

Serial.begin(9600);

Void loop()

Program izvede ukaze v zanki (loop), torej kodo v zavitih oklepajih – {}. Nato se vrne na začetek zanke in začne znova. To se dogaja blazno hitro.

// This is the loop, the program runs through these items then goes
back to the start of the loop and does it again.
// This happens very very fast.
void loop() {
}

TouchRead()

Funkcija TouchRead se uporablja za branje vhodne vrednosti kapacitivnega zaznavanja dotika. S funkcijo TouchRead odčitamo električno polje v bližnji okolici; ta vrednost se ob dotiku predmeta, ki se uporablja kot senzor, spremeni. Spremljajte vrednost in bodite pozorni na spreminjanje številke. Načeloma je prag nastavljen na 2300, če pa je treba spremeniti občutljivost predmeta, na katerem kapacitivno zaznavamo dotik, to vrednost spremenite v skladu s tem, kar razberete v terminalu.

Najprej pripravimo spremenljivko current, v kateri se bodo shranjevali podatki iz kapacitivnega senzorja v jabolku.

current = touchRead(touchRead_pin); // setting the 'current' variable
// Now the value from the touchRead function is compared to the
threshold value.

Spremenljivke za shranjevanje teh podatkov nastavimo na začetku našega programa. Spremenljivke so current, thresh in play_flag. Naša spremenljivka za vrednost praga (thresh) je nastavljena na 2300.

If()

S t. i. if-izjavo (*if-statement*) preverimo, ali se nekaj dogaja oz. ali je prisoten določeni pogoj. Če je pogoj prisoten, se izvede koda v zavitih oklepajih; če ni, se začne izvajanje naslednjega dela kode. Tu imamo dvoje if-izjav, s katerima preverjamo, ali jabolko zaznava dotik in ali se zvok trenutno predvaja.

Naš sintetizator je lahko v enem od štirih različnih stanj in glede na trenutno stanje so potrebna različna dejanja.

Stanje	Kaj ga določa	Rezultat
Dotaknili smo se jabolka.	Vrednost iz senzorja dotika v jabolku (current) je višja od vrednosti praga in zvok se ne predvaja.	Nastavitev play_flag na vrednost 1, ki pomeni, da se zvok predvaja. Zvišanje glasnosti zvoka.
Še naprej se dotikamo jabolka.	Vrednost iz senzorja dotika v jabolku (current) je višja od vrednosti praga in zvok se predvaja.	Nič se ne zgodi.
Nehali smo se dotikati jabolka.	Vrednost iz senzorja dotika v jabolku (current) je nižja od vrednosti praga in zvok se predvaja.	Nastavitev play_flag na vrednost 0, ki pomeni, da se zvok ne predvaja. Znižanje glasnosti zvoka.
Ne dotikamo se jabolka.	Jabolko ne zaznava dotika – vrednost iz senzorja dotika (current) je nižja od praga – in zvok se ne predvaja (da se zvok ne predvaja, vemo po tem, da je play_flag nastavljen na 0).	Nič se ne zgodi.

```
if(current > thresh && play_flag == 0) {
     play_flag = 1;
     waveform1.amplitude(0.2); // turn the volume up to 0.2 - 1 is
full volume.
}
```

Če se jabolka dotikamo, je vrednost iz senzorja dotika (*current*) višja od vrednosti praga. Če gre za nov dotik, torej če se tik pred tem jabolka nismo dotikali, bo vrednost spremenljivke *play_flag* znašala 0.

Če je vrednost iz senzorja dotika (*current*) višja od nastavljenega praga (thresh) in se ton oz. zvok ne predvaja (vrednost *play_flag* je 0), se sproži dogodek vklopa tona, vrednost spremenljivke *play_flag* pa se spremeni v 1, kar sporoča, da se ton trenutno predvaja.

Če je vrednost iz senzorja dotika (*current*) višja od nastavljenega praga (*thresh*) in se ton oz. zvok trenutno predvaja (vrednost *play_flag* je 1), se ne zgodi nič.

Jabolka se ne dotikamo in zvok se ne predvaja.

```
if(current < thresh && play_flag == 1) {
    play_flag = 0;
    waveform1.amplitude(0); // turn the volume to 0 - no sound
}</pre>
```

Če je vrednost iz senzorja dotika nižja od nastavljenega praga in se ton trenutno predvaja, potem se sproži dogodek izklopa tona in vrednost *play_flag* se spremeni v 0, kar sporoča, da se ton ne predvaja. Če je vrednost iz senzorja dotika pod nastavljenim pragom, ton pa se ne predvaja, se ne zgodi nič.

Serial.println(current); // this sends the data from the input pin which is stored in the variable - "current" - to the serial monitor

Delay()

Časovni zamik, podan v milisekundah. Kratek premor, preden se zanka začne ponovno izvajati od začetka.

delay(100);

AudioSynthWaveform

Valovna oblika za predvajanje s sintetizatorjem; nanjo se v kodi sklicujemo z uporabo spremenljivke, zato da imamo lahko več različnih.

Bodite pozorni na spremenljivki waveform in waveform2.

Lastnosti – valovna oblika, amplituda, frekvenca

Valovna oblika je vrsta zvočnega vala, ki bo uporabljena. Podprte so naslednje valovne oblike:

- WAVEFORM_SINE
- WAVEFORM_SAWTOOTH
- WAVEFORM_SAWTOOTH_REVERSE
- WAVEFORM_SQUARE
- WAVEFORM_TRIANGLE
- WAVEFORM_TRIANGLE_VARIABLE
- WAVEFORM_ARBITRARY
- WAVEFORM_PULSE
- WAVEFORM_SAMPLE_HOLD

Frekvenca

Z njo določamo višino tona oz. zvoka. Razpon človeškega sluha je običajno med 20 in 20.000 Hz, a so med posamezniki bistvene razlike. 20 Hz je zelo globok, 20.000 Hz pa zelo visok zvok.

Amplituda

Ta določa glasnost zvoka in ima lahko vrednosti med 0 in 1, pri čemer 0 pomeni izklopljen zvok, 1 pa največjo glasnost.

Funkcije

Vse lastnosti valovne oblike lahko določite z eno samo funkcijo, in sicer tako:

variableName.begin(level, frequency, waveform);

Tako je to videti v naši kodi – waveform1 je ime spremenljivke za valovno obliko, na katero se sklicujemo. S tem ukazom nastavimo glasnost zvoka na 0,5 in višino (frekvenco) na 300 Hz, kot vrsto valovne oblike pa določimo sinusni val.

waveform1.begin(0.5, 300, WAVEFORM_SINE);

Parametre valovne oblike lahko določite tudi posamič, in sicer tako:

begin(waveform);

Konfiguracija vrste valovne oblike, ki jo bomo ustvarili.

Na primer:

waveform1.begin(WAVEFORM_SINE);

frequency(freq);

Sprememba frekvence.

Na primer:

waveform1.frequency(300);

amplitude(level);

Sprememba amplitude. Nastavite na 0, da signal izklopite.

Na primer:

waveform1.amplitude(0.5);

Vaja 4 – dodajanje več sadežev

Zdaj s Teensyjem povežimo še eno jabolko, da bomo lahko imeli dva različna tona.

1. korak

Pin 22 (A8) na Teensyju povežite z drugim jabolkom.



2. korak

Naložite datoteko two_oscillators_touch.ino.

V programu Arduino naložite datoteko two_oscillators_touch.ino v Teensy in prisluhnite, kako zvenita dve jabolki.

Poglejmo si kodo

Za priključitev drugega jabolka na Teensy sledite naslednjim korakom. Preberite si te zapiske in nato iste korake uporabite za dodajanje tretjega jabolka.

1. korak – posodobitev valovnih oblik z orodjem za oblikovanje zvoka

Za dodajanje kode za drugo jabolko moramo najprej s pomočjo orodja za oblikovanje zvoka posodobiti valovne oblike. Ker imamo zdaj več zvokov, bomo morali uporabiti tudi mešalko oz. mikser.

Dodajmo mikser in novo valovno obliko. Valovni obliki povežite z mikserjem tako, kot kaže spodnja slika.



Izberite možnost Export code in dobljeno kodo prilepite na začetku dokumenta, do tja, kjer piše:

// GUItool: end automatically generated code

https://www.pjrc.com/teensy/gui/



Ustvarjena koda

<pre>#include <audio.h> #include <wire.h> #include <spi.h> #include <sd.h> #include <serialflash.h< pre=""></serialflash.h<></sd.h></spi.h></wire.h></audio.h></pre>	>		
// GUItool: begin automatically generated code			
AudioSynthWaveform	waveform1;	//xy=290,213	
AudioSynthWaveform	waveform2;	//xy=291,266	
AudioMixer4	mixer1;	//xy=487,237	
AudioOutputAnalog	dac1;	//xy=632,237	
AudioConnection	patchCord1(waveform1, 0, mixer1, 1);		
AudioConnection	patchCord2(waveform2, 0, mixer1, 2);		
AudioConnection	patchCord3(mixer1, dac1);		
// GUItool: end automatically generated code			

2. korak – dodajanje spremenljivk

Ustvarite še en komplet spremenljivk, ki jih uporabljamo za sklicevanje na jabolko in z njim povezane podatke.

Najprej moramo dodati novo spremenljivko, ki nam bo omogočila razlikovati med obema jabolkoma. Na koncu imena nove spremenljivke dodajte številko 2.

Dodajte spodnjo kodo v vrstico 22, takoj za razdelkom za pripravo spremenljivk. S tem pripravimo nove spremenljivke za uporabo z drugim sadežem, ki ga boste dodali.

//variables for the second piece of fruit

int touchRead_pin2 = A8; //variable to store the second input pin
int thresh2 = 2300; // variable to store the second touch threshold
int play_flag2 = 0; // variable that flags if sound from the second apple
is being played or not 1 = playing; 0 = not playing

int current2; // variable used to store the value from the second touch
input pin

V vrstici 51 (približno; v vaši kodi bo morda malenkost drugače), takoj pod vrstico, kjer je nastavljena spremenljivka current, prilepite to kodo:

```
current2 = touchRead(touchRead_pin2); // setting the 'current2'
variable
```

3. korak – nastavljanje parametrov zvoka

V vrstici 37, takoj pod vrstico, kjer so nastavljeni parametri za waveform1, dodajte to:

```
waveform1.begin(0, 180, WAVEFORM_TRIANGLE); // this sets up the
parameters of the second wave form
```

4. korak – preverjanje, ali se zvok predvaja

Preverite, ali se ob dotiku jabolka začne predvajati zvok. Pri tem se ustrezno spremeni glasnost zvoka jabolka.

Dodajte to kodo, s katero preverimo, ali se zvok drugega jabolka predvaja, v vrstico 60, torej takoj pod delom, kjer preverjamo predvajanje zvoka prvega jabolka.

```
//repeat for the second apple
if(current2 > thresh && play_flag2 == 0) {
    play_flag2 = 1;
    waveform2.amplitude(0.2); // turn the volume up to 0.2}
```

```
//repeat for the second apple
if(current2 < thresh && play_flag2 == 1) {
    play_flag2 = 0;
    waveform2.amplitude(0); // turn the volume to 0 - no sound}</pre>
```

5. korak – dodajanje več jabolk

S pomočjo tega postopka lahko zdaj dodate še več jabolk. Poskusite dodati tretje jabolko, pri čemer spremenljivke oštevilčite s številko 3 namesto 2.

Vaja 5 – uravnavanje višine tona

V tej vaji bomo jabolku dodali funkcijo spreminjanja višine tona.

1. korak

Izklopite drugo jabolko iz testne plošče.

2. korak

Na testno ploščo dodajte potenciometer:

- srednji pin potenciometra povežite s pinom 11 na Teensyju;
- levi pin potenciometra priključite na vir napetosti;
- desni pin potenciometra priključite na ozemljitev oz. minus.

Opomba: če se višina tona spreminja v napačni smeri, poskusite zamenjati, kateri pin na potenciometru je priključen na ozemljitev, kateri pa na vir napetosti.



Odprite datoteko one_osc_pitch.ino ter kodo v njej sestavite in naložite na Teensy. S potenciometrom boste zdaj lahko upravljali višino tona, ki se predvaja ob dotiku jabolka.

Poglejmo si kodo, ki to omogoča – zlasti nove elemente v kodi, s katerimi usposobimo potenciometer:

v vrstici 22 pripravimo naslednje spremenljivke:

int pitchPin = A0; // variable storing which pin the pitch potentiometer will connect to float pitchData; // variable used to store the data coming from the pitch potentiometer float scaledPitch; //variable used to store the scaled data from the pitch potentiometer

Opombe v kodi nam povedo naslednje:

pitchPin je spremenljivka, v kateri je shranjena številka pina, povezanega s srednjim pinom potenciometra.

pitchData je spremenljivka, v kateri se shranjuje trenutna vrednost iz potenciometra. Potenciometer pošilja vrednosti v območju od 1 do 1023.

scaledPitch je spremenljivka, v kateri bomo shranili »mapirano« (preslikano) [T1] vrednost; več o tem spodaj.

V zanki v vrstici 44 spremenljivko pitchData nastavimo tako, da prevzema vrednosti, ki jih pošilja potenciometer.

```
pitchData = analogRead(pitchPin);
// Read the data coming from the pitch potentiometer and save it in
the variable pitchData
```

Мар

Tu uporabimo funkcijo za mapiranje oz. preslikavanje, da vrednosti, ki jih pošilja potenciometer, pretvorimo v želen razpon višine tonov.

Ob obračanju potenciometra se vrednost na njegovem izhodu spreminja od 1 do 1023. Enako velja za večino standardnih potenciometrov. Te številke bi radi pretvorili v razpon zvočnih frekvenc. To dosežemo tako, da s funkcijo za mapiranje določimo razpon višine tona od 20 do 8000 Hz.

Funkcija za mapiranje je zelo uporabna. Z njo številko iz nekega razpona vrednosti preslikamo v drugi razpon. Natančneje, vrednost fromLow se bo preslikala v toLow, vrednost fromHigh v toHigh, vmesne vrednosti pa se bodo preslikale v ustrezne vrednosti v ciljnem območju.

Funkcijo zapišemo tako:

map(value, fromLow, fromHigh, toLow, toHigh)

Poglejte si našo kodo – v njej vrednosti v območju izhodnih vrednosti potenciometra (1–1023) mapiramo oz. preslikamo v vrednosti v izbranem območju zvočnih frekvenc (v našem primeru 20–8000 Hz).

scaledPitch = map(pitchData, 1, 1023, 20, 8000);
// Scale the value from the 'pitchData' variable and store it in the
'scaledPitch' variable

S tu uporabljenimi številkami lahko eksperimentirate in spremenite razpon višine tona, ki jo upravljamo s potenciometrom. Zapomnite si, da je razpon človeškega sluha sicer od 20 do 20.000 Hz, da pa je naš sluh najbolj občutljiv v frekvenčnem območju od 2000 do 5000 Hz.

Upravljanje višine tona

Zdaj bomo višino tona valovne oblike programirali tako, da jo bo mogoče upravljati s potenciometrom. Spremenljivka scaledPitch predstavlja mapirano oz. preslikano izhodno vrednost potenciometra.

waveform1.frequency(scaledPitch); // Set the frequency of waveform1 to the value of the 'scaledPitch' variable.

Uporaba LDR upora

Namesto potenciometra lahko uporabite t. i. LDR upor (Light Dependent Resistor oz. svetlobno odvisen upor), ki mu rečemo tudi fotoupor. Ker ima LDR-upor le dve nogici, za razliko od potenciometra s tremi, ga je treba povezati nekoliko drugače.

Ena nogica LDR upora mora biti priključena tako na ozemljitev (in sicer prek majhnega upora z vrednostjo 3,3–10K ohmov) kot na vhodni pin na Teensyju; drugo nogico priključimo na vir pozitivne napetosti (+v).

To bo videti nekako tako:



[vir: narejeno v programu Fritzing]

Priloge

Prevodnost materialov

Tu je nekaj primerov prevodnih in neprevodnih materialov.

Prevodni materiali	Neprevodni materiali
baker	papir
aluminij	teflon
platina	steklo
zlato	guma
srebro	olje
voda z vsebnostjo ionov	destilirana voda
rastline in sadeži	materiali iz steklenih vlaken
železo	porcelan
jeklo	keramika
medenina	zrak
bron	bombaž
grafit	les
	plastika

Testne plošče

Več informacij o testnih ploščah:

https://www.sciencebuddies.org/science-fairprojects/references/how-to-use-a-breadboard



Teensy pinout



Vrste zvočnih valov/kako je videti zvok?

Zvok so vibracije (zvočni valovi), ki jih lahko slišimo. Te zvočne valove povzročajo vibracije predmetov (tresljaji sem in tja). Velikost in oblika zvočnih valov določata, kakšen bo zvok, ki ga slišimo. Obliko zvočnih valov lahko narišemo.



Kode https://gitlab.com/kons-platforma/teensynth



O avtorici

Tara Pattenden je umetnica, organizatorka in pedagoginja, ki se ukvarja z različnimi novimi mediji, med drugim z elektroniko, zvokom, videom in kiparstvom. Pod imenom Phantom Chips uporablja krmilnike Teensy za izdelavo nosljivih elektronskih glasbil na osnovi tkanin, ki omogočajo nove ekspresivne načine igranja glasbe; na ta glasbila igra tudi sama. Njena glasbila občinstvo kar vabijo k sodelovanju, saj omogočajo ustvarjanje zvoka z gibanjem in kretnjami (raztezanje, božanje in stiskanje). Živi in dela v Brisbanu v Avstraliji, kjer vodi Elektrolab – prostor za učenje tehnologij za ustvarjanje. <u>https://www.phantomchips.com</u>

Kolofon

Naslov TEENSYnth, Naredi sam (DIY)

Avtorica labbooka Tara Pattenden

Urednica labbooka Tina Dolinšek

Tehnična pomoč in prelom Lovrenc Košenina

Fotografije Matjaž Rušt, Rea Vogrinčič, Fritzing

TEENSYnth Github <u>https://github.com/problemmaths/TEENsynth</u>

Prevod in lektura Zadruga Soglasnik (<u>https://www.soglasnik.si/</u>)

Produkcija Zavod Projekt Atol Platforma konS

Naslovna pasica https://pixabay.com/photos/coding-computer-hacker-hacking-1841550/

Kraj in založba Novo mesto, LokalPatriot

Leto izida 2022

Naslov knjižne zbirke, štetje v zbirki Labbook kons, 7. knjižica

Naklada 200

Brezplačna publikacija



Zapiski

CIP - Kataložni zapis o publikaciji Narodna in univerzitetna knjižnica, Ljubljana

004.4:681.5:534(035) 004.2/.3(035)

PATTENDEN, Tara

TEENSYnth : labbook : naredi sam (DIY) : programiranje in zvok 3 / [avtorica Tara Pattenden ; fotografije Matjaž Rušt ; prevod Zadruga Soglasnik]. - Novo mesto : LokalPatriot, 2022. - (Labbook kons ; knjižica 7)

ISBN 978-961-92137-6-6 COBISS.SI-ID 109368579



PROJEKT KONS.PLATFORMA ZA SODOBNO RAZISKOVALNO UMETNOST je namenjen spodbujanju prebojnih umetniških stvaritev in vzpostavitvi produkcijskega okolja, v katerem bo mogoče umetniške ideacije prevajati v priporočila za inovacije boljših, varnejših, bolj trajnostnih in etičnih produktov ter storitev. S spodbujanjem izjemnosti v umetniških delih želimo ustvariti navdihujoče okolje za ustvarjalce prihodnosti med otroki in mladimi ter za odločevalce in zainteresirane strokovnjake, ki sodelujejo pri nastajanju novih tehnoloških aplikacij in družbenih inovacij.



PARK

V vozliščih ustvarjamo prostor za mlade raziskovalne in ustvarjalne posameznike ter skupine. Posvečamo se raziskovalnim, nemirnim umom. Z navdihujočim programom spodbujamo uporabo visokih tehnologij in hkrati vzgajamo kritičnost, spodbujamo kreativnost in negujemo inovativnost. Skozi aktivno participacijo in razvoj zmogljivosti oblikujemo nove ustvarjalne skupnosti. Naše aktivnosti so namenjene otrokom, mladim in tudi odrasli zainteresirani javnosti.

kons-platforma.org



REPUBLIKA SLOVENIJA MINISTRSTVO ZA KULTURO

Projekt konS - Platforma za sodobno raziskovalno umetnost je bil izbran na javnem razpisu za izbor operacij "Mreža centrov raziskovalnih umetnosti in kulture. Naložbo sofinancirata Republika Slovenija in Evropska unija iz Evropskega sklada za regionalni razvoj.



indicator u .cart-icon sparent head i.current_p i.current-m a:hover > . otn a:hover li.current-m lient-cart,. olor:#ffffff op nav>ul>li ea-toggle a anspare