

IT Girls

tinkering_with_e-waste



VJEŽBE ZA
SREDNJE ŠKOLE

Muamer Halilović



podržava



za svako dijete



UJEDINJENE NACIJE
BOSNA I HERCEGOVINA

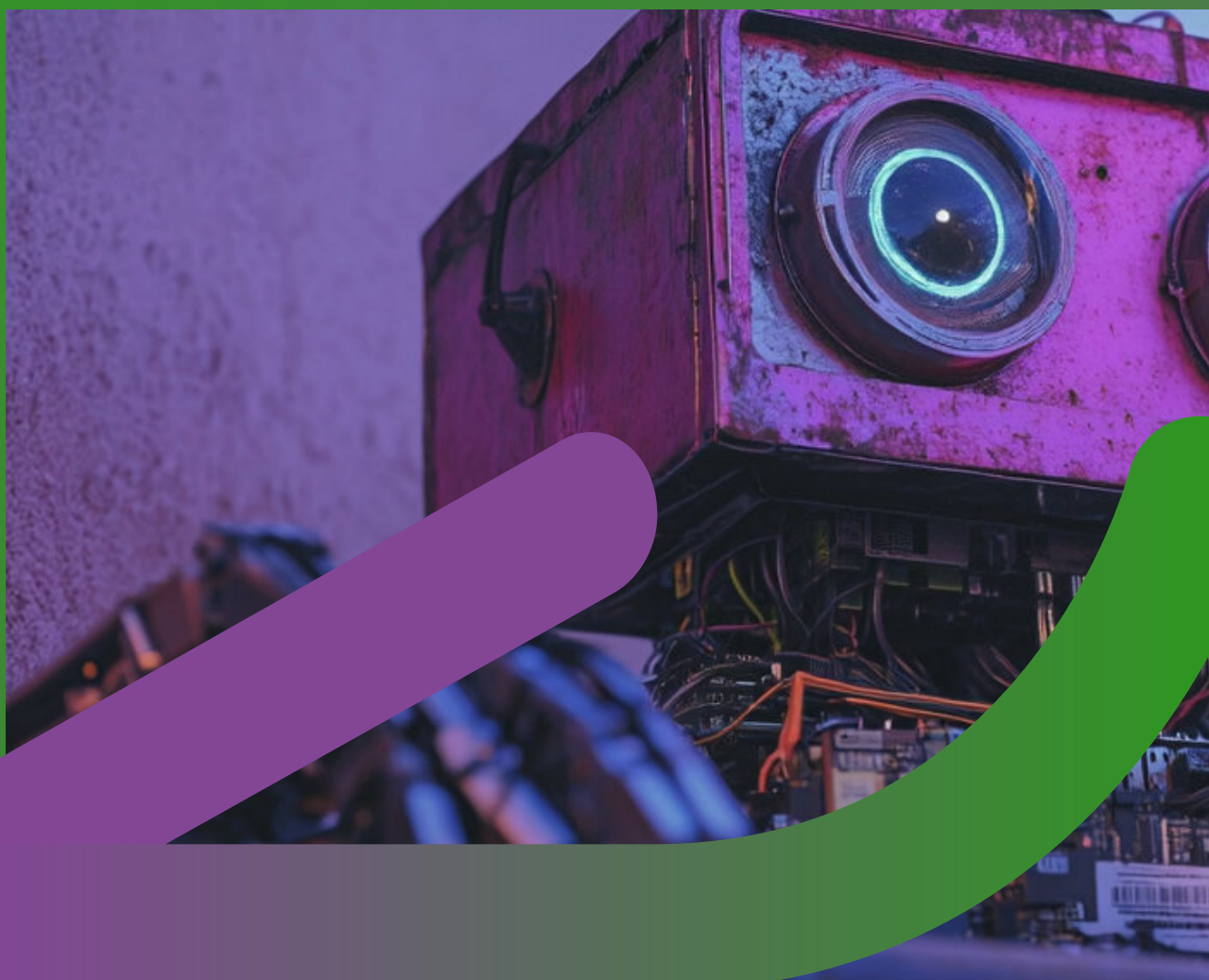
UNICEF ne podržava nijedan brend, proizvod ili uslugu.

SADRŽAJ

4_---	Uvod
6_---	Kako starom računararu dati novi život
16_--	1. VGA monitor + Arduino - Prikaz poruke
25_--	2. Tastatura (PS/2) + LCD 2X16
31_--	3. CD-Rom Portalni robot
37_--	4. CPU ventilator kao indikator temperature
45_--	5. Morzov kod iz zvučnika matične ploče
49_--	Zaključak

➤ UVOD

E-otpad je sav otpad nastao odbacivanjem elektroničkih uređaja (računara, mobitela, televizora, itd.). Kao takav u sebi sadrži štetne tvari (olovo, živa, kadmij) koje mogu zagađiti tlo i vodu ako se ne zbrinu pravilno.





Ali, sadrži i korisne materijale (bakar, plastiku komponente) koji se mogu **ponovno iskoristiti**.

Cilj nam je naučiti kako **LED-ice i tasteri** iz starih računara mogu dobiti **novi život** kroz edukaciju i kreativnost!



Kako starom računar dati novi život?

Glavna zvijezda ovog seta vježbi će biti stari računar (eng. Personal Computer, PC), koji je poslije vjerne službe svom korisniku, spreman da bude zamijenjen novim, bržim i jačim nasljednikom.

Iako ima malo tuge u ovom uvodu, sretna stvar je što će se kroz ovaj set vježbi naučiti da ovo ne znači kraj života za stari PC. Naime vidjet ćete da se neke od komponenti mogu iskorisiti u zanimljivim projektima, koji će pored toga što su korisni, smanjiti količinu elektronskog otpada.

Korak 1 - Priprema PC kućišta za proces demontaže

Na slici je prikazano standardno radno okruženje PC kućišta. Prvenstveno treba voditi računa da se prije bilo kakvog rada s PC-m isti otpoји s mrežnog napona.

Nakon što se taj dio posla završili potrebno je otpojiti sve kablove koji su spojeni na Vaš PC.

To obično podrazumjeva sledeće kablove i uređaje:

- napojni kabel,
- miš,
- tastatura,
- mrežno/LAN kabel,
- VGA/DVI/HDMI kabel....

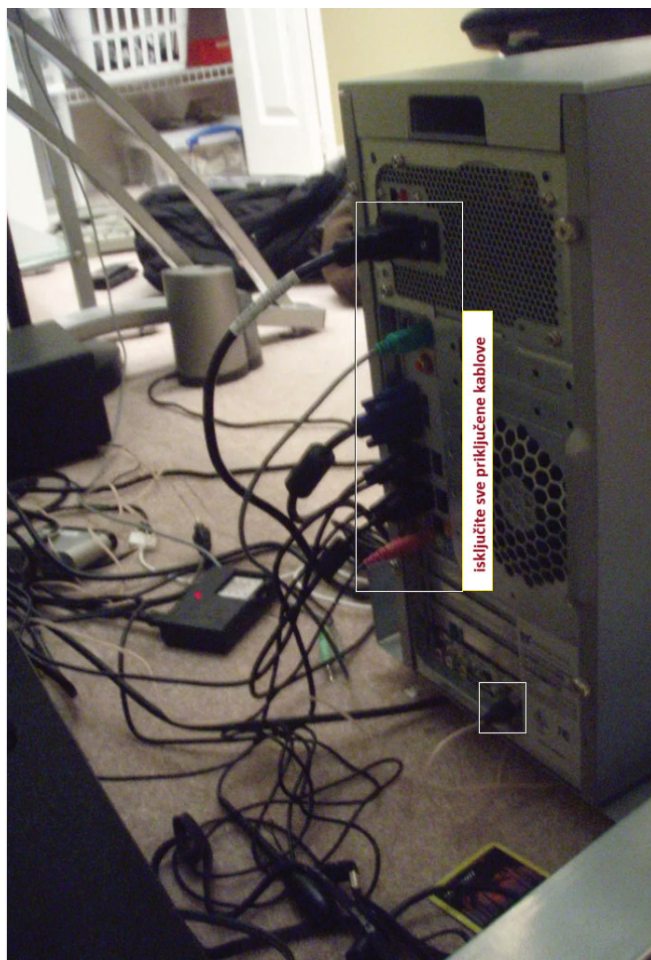


Slika 1. PC kućište u prirodnom okruženju



OPREZ!!!
OPASNOST OD STRUJNOG UDARA

ISKLUČITI SVE PRIKLJUČNE KABLOVE



Slika 2. Kablovi za odvajanje od PC-a

Kada ste sve kablove i uređaje odspojili od vašeg PC-a, potrebno je pristupiti procesu otvaranja kućišta.

Korak 2 - Demontaža, otvaranje Kućišta PC-a



ODVRNI VIJKE



OPREZ!!!

OPASNOST OD POSJEKOTINA

Nakon uklanjanja vijaka, potrebno je izvući bočne strane kućišta. Gledano sa zadnje strane izvlačenje desnog poklopca kućišta otvara put ka matičnoj ploči i ostalim komponentama PC-a.



POVUCI PANEL

Nakon skidanja desnog bočnog panela, potrebno je isto uraditi i s lijevim bočnim panelom.



Slika 3. Otvaranje kućišta



Slika 4. Skidanje desnog bočnog panela

Nakon skidanja bočnih panel i gonjeg poklopca kućišta može se pristupiti skidanju prednjeg panela. On je obično od plastike i pričvršćen je za metalnu konstrukciju kućišta tzv. "žabicama". Koje je potrebno pritisnuti da iskoče iz utora u metalnom kućištu.



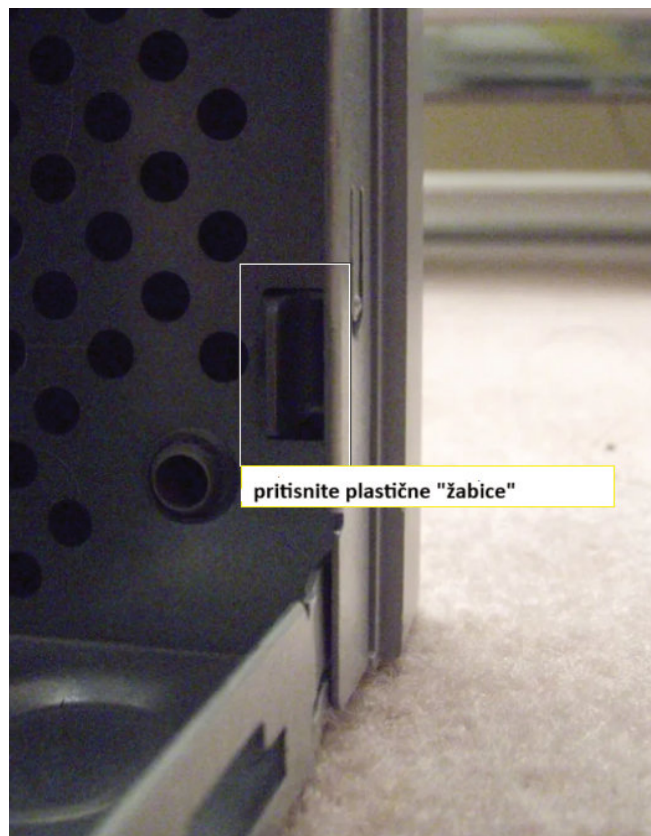
PRITISNITE PLASTIČNE "ŽABICE"



OPREZ!!!

OPASNOST OD UKLIJEŠTENJA

Proces pritiskanja plastičnih "žabica" treba ponoviti sa svih strana prednjeg panela, da bi se on mogao odvojiti od metalnog kućišta.



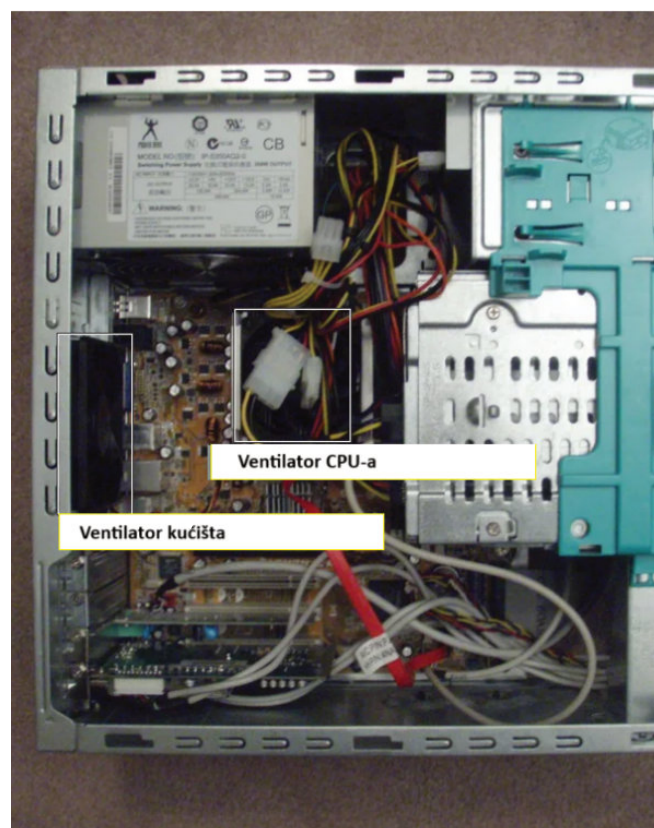
Slika 5. Priprema za skidanje prednjeg panela



Slika 6. Odvajanje prednjeg panela

Korak 3 - Vađenje komponenti

VENTILATOR CPU-a / KUĆIŠTA



Slika 7. Odvajanje prednjeg panela

Proces vađenja komponenti iz kućišta PC-a se ne može generalizirati, jer postoji veliki broj specifičnih i nestandardnih rješenja. U konkretnom primjeru će se pokazati procedura za izdvajanje nekoliko komponenti koje se koriste u vježbama.

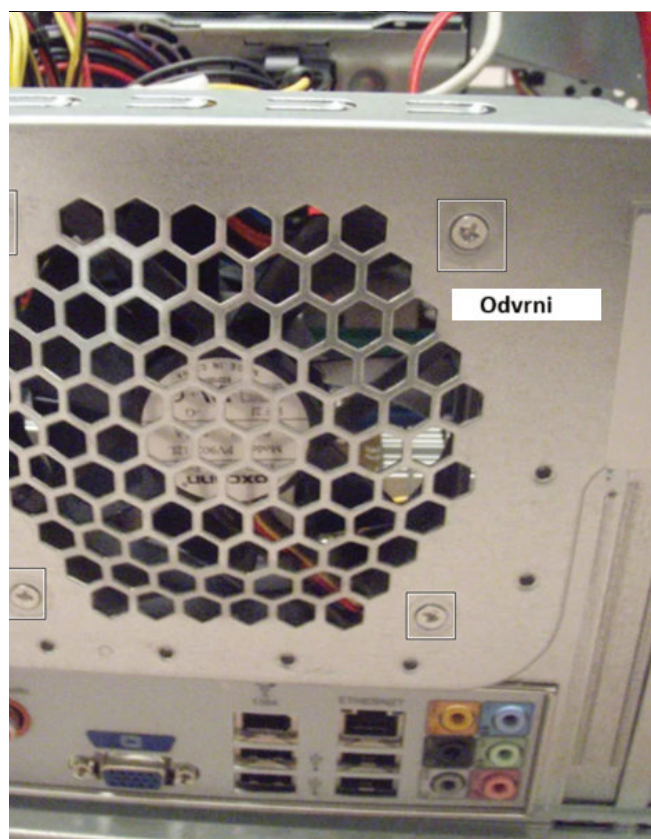
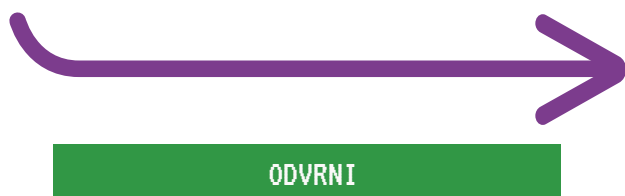
Prva od komponenti za demontažu je ventilator kućišta, on je obično priključen direkto na matičnu ploču PC-a, te je potrebno prvo otpojiti konektor od ploče.

KONEKTOR VENTILATORA KUĆIŠTA



Slika 8. OTPAJANJE KONEKTORA VENTILATORA

Sljedeći korak je odvrtnje vijaka na kućištu PC-a kojima je ventilator pričvršćen za kućište.



Slika 9. Odvrtnje vijaka ventilatora kućišta



Slika 10. Ventilator kućišta

Sličnu proceduru je potrebno ponoviti i za CPU ventilator s jednom razlikom, on je obično pričvršćena na tijelo hladanjka CPU-a.

Naredna komponenta je CD/DVD uređaj. Prije same demontaže potrebno je otpojiti kablove kojima je CD/DVD uređaj spojen na napojnu jedinicu i matičnu ploču.



OTPOJITE NAPOJNO I DATA KABLO



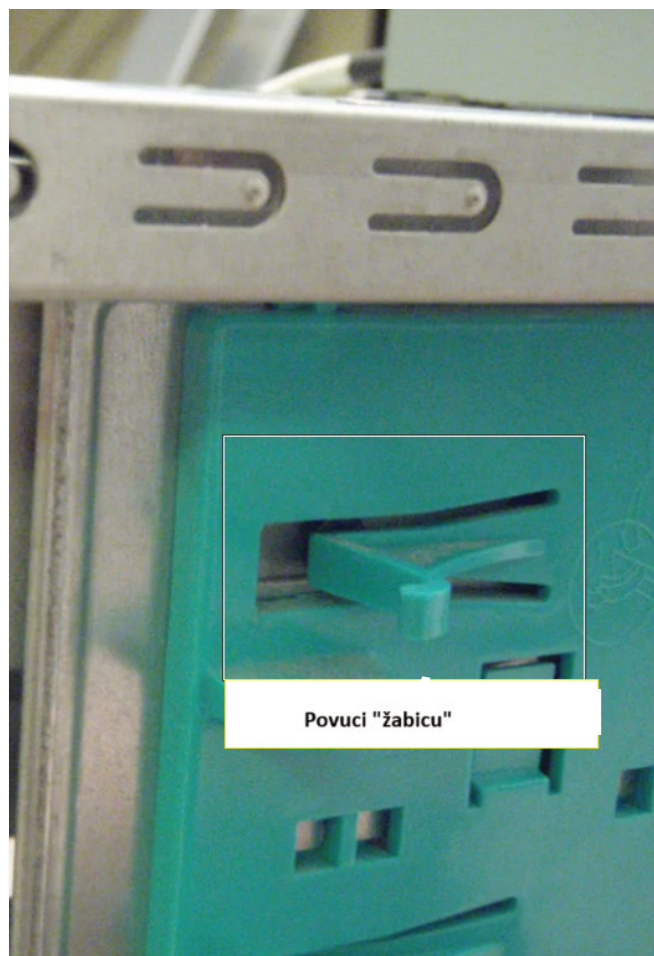
Slika 11. Mjesto spajanja podatkovnog(data) kabla CD/DVD-A

Više je idejnih rješenja kako je CD/DVD uređaj instaliran unutar kućišta PC-a, tako je moguće vidjeti realizaciju s vijcima koji su bočno zavrnuti u CD/DVD uređaj ili kao što je to u ovom slučaju, pomoću plastičnih "žabica".

Njih je potrebno povući ka vani dok se istovremeno CD/DVD gura ka vani. Ovo je jednostavnije rješenje i omogućava brzu instalaciju ili deinstalaciju uređaja.



POVUCI "ŽABICU"



Slika 12. Demontaža CD/DVD iz kućišta PC-a.



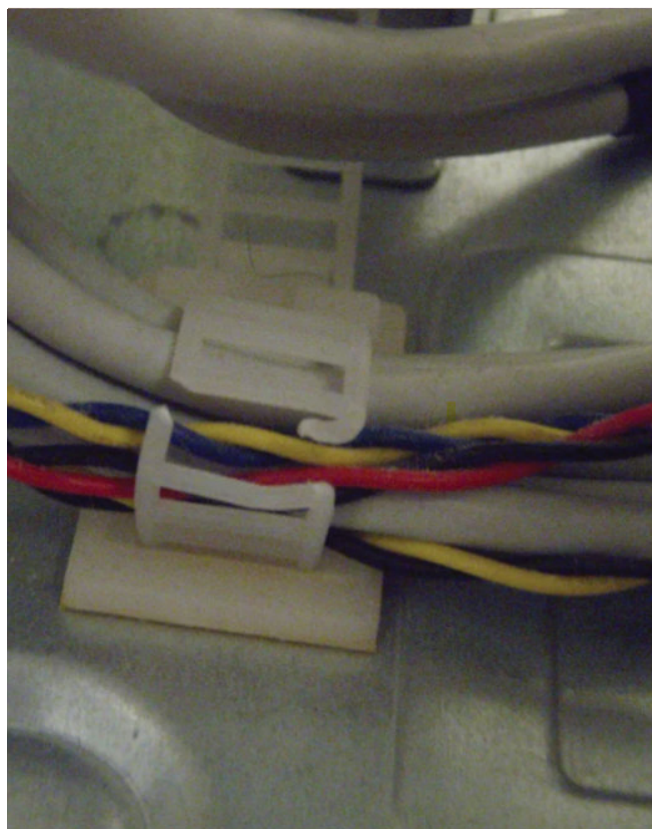
Slika 13. CD/DVD izvučen iz kućišta

Naredni element koji je potrebno demonirati je nosač POWER/HDD LED i START tastera na prednjem panelu PC-a. On je u većinom za kućište PC-a spojen s plastičnim "žabicama" koje je potrebno osloboditi da bi krenuli u proces demontaže.



Slika 14. Izgled nosača POWER/HDD LED i START tastera

Unutar samog kućišta kablovi koji od nosača do matične ploče su radi estetike povezani vezicama s drugim kablovima unutar kućišta PC-a. Vezice je potrebno osloboditi, te je nakon isključenja kablova s matične ploče nosač ledica i tastera slobodan za izvlačenje iz PC-a i za korištenje u novim projektima.



Slika 15. Vezice za organizaciju kablova unutar kućišta



Slika 16. Nosač spreman za upotrebu

Korak 4 - Oprezno nastavite dalje S demontažom ostalih Elemenata PC-a

Ukoliko ste oprezni i prije same demontaže elemenata unutar PC-a odradite analizu kako su oni montirani, te vodeći računa o obaveznoj upotrebi zaštitne opreme i potrebnih alata, element će se postepeno btiti demontirani iz kućišta PC-a. Pa bi potpuno demonitrano kućište PC-a izgledalo kao na slici ispod.



Slika 17. Demontirano kućište PC-a

Nakon uspješne demontaže elemenata PC, ostaje nam da neke od njih iskorisitmo u zabavnim projektima i nečemu što je planirano da postane otpad udahnemo novi život. Sretno!

1

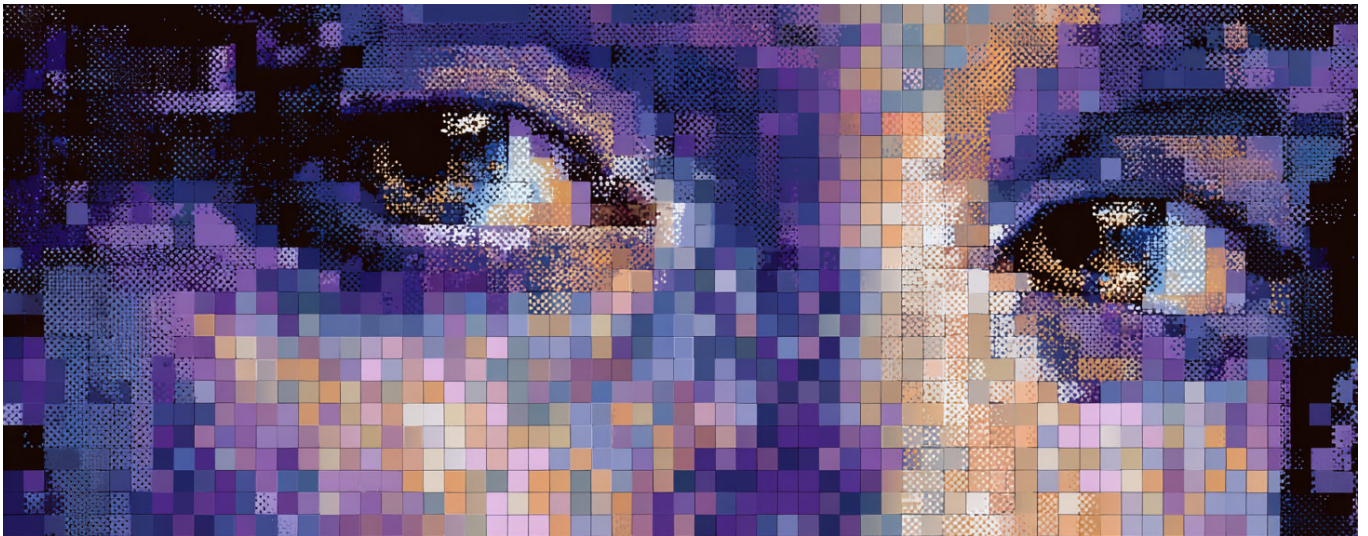


VGA monitor + Arduino - Prikaz poruke

CILJEVI VJEŽBE:

- Naučiti kako se Arduino može koristiti za generisanje VGA signala.
- Prikazati piksele različite boje na VGA monitoru pomoću biblioteke VGAX.
- Podesiti boje.

Pikselovano lice (AI Generisana)



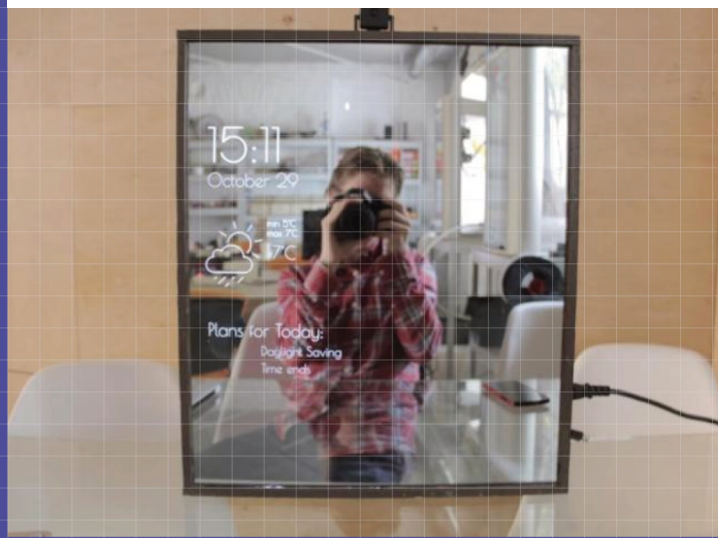
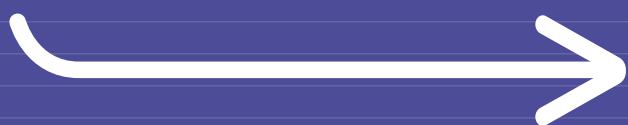
Jedan od najčešćih e-waste uređaja je stari monitor sa VGA ulazom. Tehnologija monitora napreduje tako brzo da danas standardni monitori imaju performanse gaming premium monitora. Stoga monitori, pogotovo kod profesionalnih korisnika vrlo brzo zastarijevaju.

Elektronika monitora je pouzdana i u velikoj većini slučajeva se monitori odlažu kao ispravni, naravno da se kao takvi mogu iskoristiti u obrazovnom procesu. Pri čemu im se kao starim uređajima može udahnuti novi život s neobičnim dizajnom.



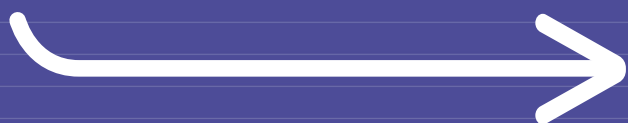
Slika 1.1. Monitor s VGA inputom

Jedna od ideja je tzv. “magic mirror” koji obično koristi daleko skuplje uređaje za upravljanje (Raspberry Pi), ali i Arduino se može iskorisiti kao jeftinina alternativa. Na način da se Arduino uparen sa VGA monitorom iskoristi za prikaz datuma, vremena ili temperature.



Slika 1.2. Magično ogledalo (Izvor: www.hackster.io)

Naravno neizostavan je i razvoj retro igrica, gdje mašta i vještina programera i programerki može napraviti čudo.



Slika 1.2. Retro igrica (Izvor: www.hackster.io)

Potrebni materijali:

- Arduino Uno (ili neki drugi 8-bitni mikrokontroler)
- VGA kabl i muški VGA konektor (DB15)
- 2x otpornici od 470Ω
- 2x otpornici 650Ω
- Stari CRT ili LCD VGA monitor
- Par žica, malo strpljenja i dobra volja

Šema spajanja

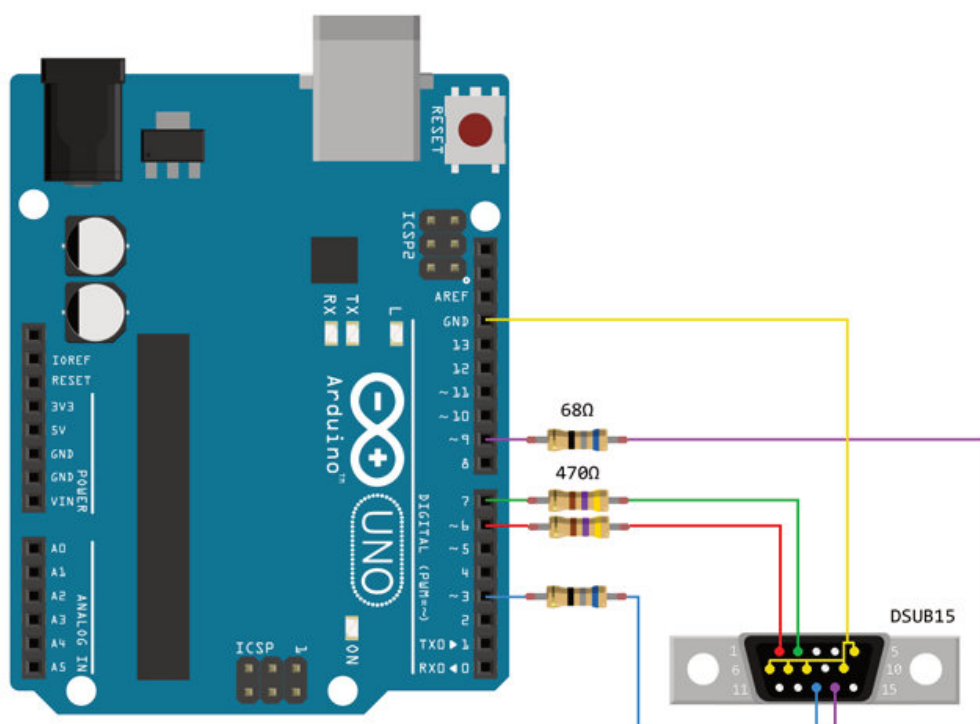
Na VGA konektoru sve pinove na šemi spajanja označene žutom bojom spojiti na niži naponski potencijal (GND). Jako je važno naglasiti da zbog hardverskih ograničenja mikrokontrolera nije preporučljivo, na VGA konektoru, koristiti sva tri pina za boje, ali je moguće odabirom pinova definisati moguće boje koje se mogu koristiti.

Četiri boje generirane iz biblioteke nisu unaprijed definirane, tako da na Arduino UNO trebate spojiti pin 6 i pin 7 sa dva VGA RGB pina.

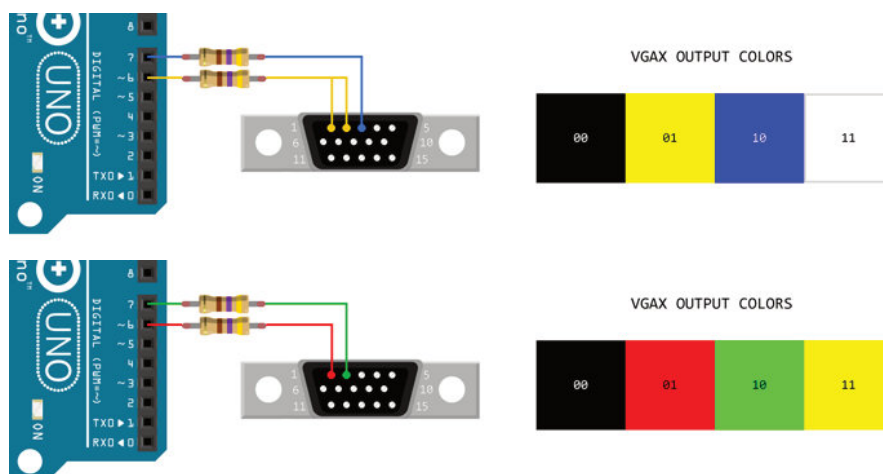
Odabirom kombinacije pinova dobijete različite boje.

Neke od mogućih kombinacija bez dodatnih hardverskih komponenti su prikazani na slici ispod.

Slika 1.3. Šema spajanja (Izvor: www.github.com)



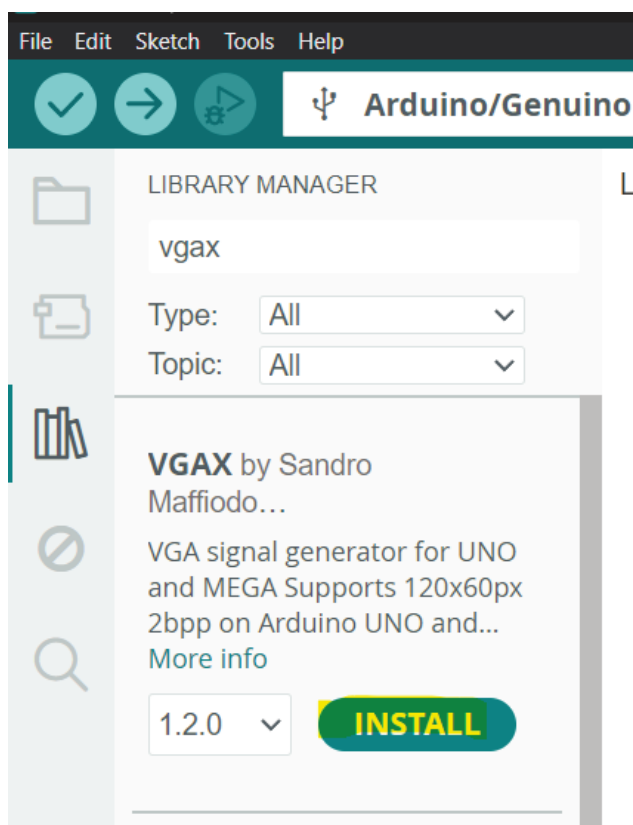
Slika 1.4. Neke od mogućih kombinacija za boju (Izvor: www.github.com)



Da bi bilo moguće realizovati ovu vježbu potrebno je koristiti optimiziranu biblioteku VGAX, ovo je VGA biblioteka i za Arduino UNO i za Arduino MEGA. Kako je već navedeno za korištenje biblioteke potrebna su vam samo 4 otpornika i jedan DSUB15 konektor ili VGA kabl presječen na jednom kraju. Ova biblioteka zahtijeva ATmega328 ili noviji mikrokontroler, a dodatno je bitno naglasiti da ista ne podržava rad s ATTINY ili ATmega168 porodicama mikrokontrolera

Instalacija biblioteke je jednostavna (Tools->Manage Libraries)i ista dolazi sa setom primjera.

Slika 15. VGAX instalacija biblioteke

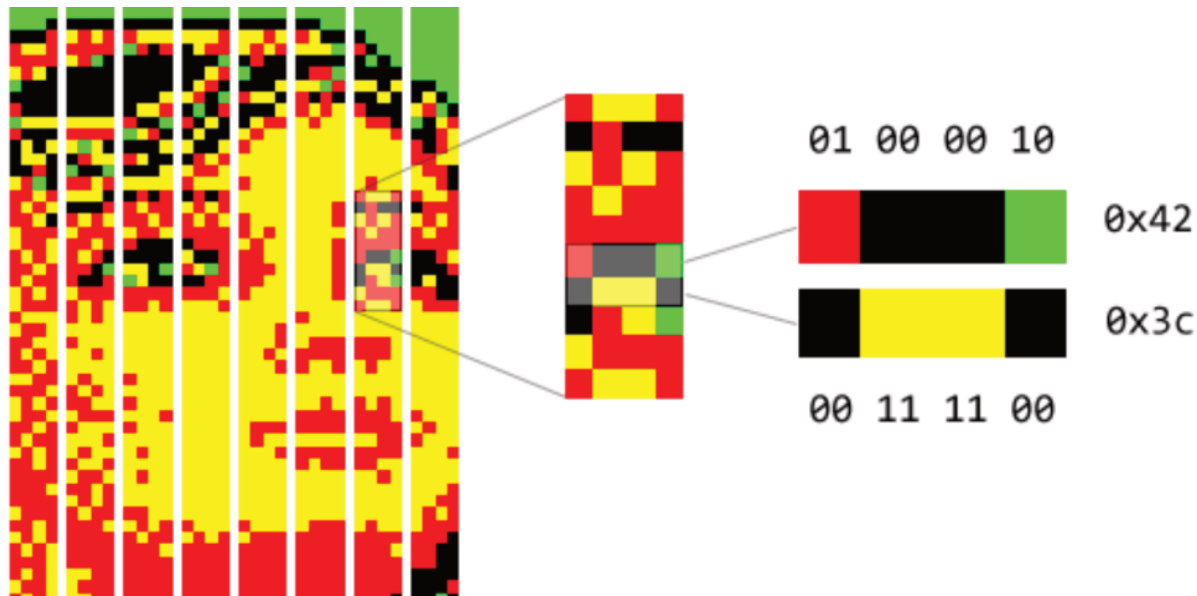


Biblioteka implementira 120x60px framebuffer (međuspremnik) gdje se svaki piksel pohranjuje kao 2 bita (4 boje). Framebuffer se pohranjuje unutar SRAM-a (StaticRAM) i zahtijeva najmanje 1800 bajtova.

To znači da na ATmega328 vaši programi ne mogu koristiti više od 200 bajtova, to je jako važno naglasiti, stoga budite oprezni kod planiranja aplikacije! Moguće rješenje ovog problema je upotreba dva Arduino UNO-a, jedan za upravljanje onim koji koristi VGAX biblioteku, a drugi za aplikaciju.

VGAX framebuffer koristi 2 bita za svaki piksel. Što znači da su unutar svakog bajta pohranjena 4 piksela spakovana ovim redoslijedom: krajnji lijevi piksel je u najviša 2 bita, krajnji desni piksel je u najniža 2 bita.

Slika 1.6. Prikaz 1Byte i 4 piksela (Izvor: www.github.com)



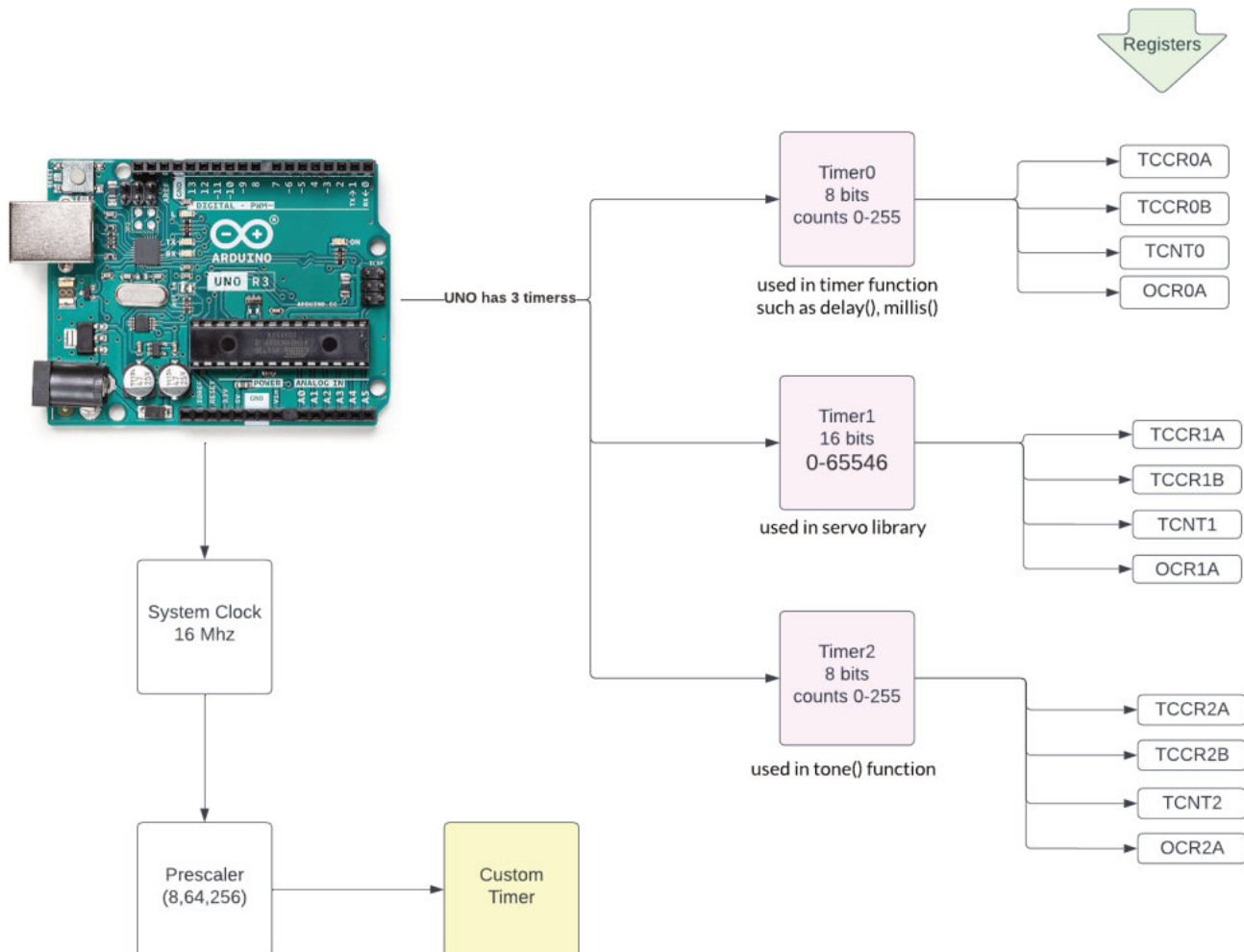
VGAX biblioteka spada u grupu visoko kompleksnih biblioteka jer optimizira hardver mikrokontrolera za generisanje video signala koristeći samo interrupt, tako da unutar funkcije `main()` možete raditi što god želite. Vaš kod će biti prekinut kada treba generirati VGA signal.

To naravno ima i ograničenja, jer resursi CPU su oslobođeni ali ne možete dodati druge interrupt signale ili će generiranje VGA signala biti prekinuto.

Biblioteka koristi sva 3 tajmera ATmega328 mikrokontrolera, a kako Arduino razvojno okruženje koristi `TIMER0` za implementaciju funkcija **`millis()`**, **`micros()`**, **`delay(ms)`**, **`delayMicroseconds(us)`**, njihovo korištenje uz ovu biblioteku nije moguće.

Kako smo spomenuli Timer 0 na Arduino UNO, ukratko ćemo ga pojasniti.

Slika 1.7. Tajmeri kod ATMEGA328p - UNO



Arduino UNO ima više tajmera, a Timer 0 je 8-bitni tajmer, radi automatski u pozadini – ne mora se dodatno "aktivirati". Pokreće se čim se uključi mikrokontroler. Radi na osnovu sistema taktova (clock) – broji impulse. Te služi gore navedenim funkcijama **millis()**, **micros()**, **delay(ms)**, **delayMicroseconds(us)**. Ove funkcije blokiraju pristup drugim funkcijama koje bi htjele koristiti isti tajmer.

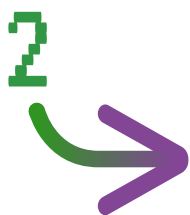
To predstavlja i izazov za korisnike i korisnice VGAX biblioteke jer se moraju koristiti alternativne verzije koje pruža ta biblioteka.

Za kraj vježbe je izdvojen jedan jednostavan primjer.

Kodiranje

```
#include <VGAX.h>
// kreiranje instance
VGAX vga;
void setup() {
    vga.begin();
    vga.clear(11);
    for (int y=0; y!=VGAX_HEIGHT; y++) {
        for (int x=0; x!=VGAX_BWIDTH; x++) {
            // pogledajte header fajl za opis metode putpixel4
            vga.putpixel4(x, y, y%2 ? 27 : 228);
        }
    }
}
void loop() {
}
```





Tastatura (PS/2) + LCD 2X16

CILJEVI VJEŽBE:

- Povezati tastaturu s Arduinoom.
- Pročitati pritisnutu tipku.
- Prikazati unos na LCD displeju.

Tipke tastature lete (AI Generisana)



Naredna komponenta koju je razvoj USB tehnologije polako gurnuo u zaborav je PS2 tastatura, vjerovatno u školi imate nebrojeno ovih starih tastatura koje iako ispravne su zamjenjenje onima sa USB interfejsom.

Naravno ove tastature ne moraju odmah postati e-otpad, moguće im je udahnuti novi život kroz upotrebu s Arduino razvojnom pločicom.



Slika 2.1. PS/2 tastatura

PS/2 tastatura

PS/2 tastatura komunicira s računarom (ili u našem slučaju s Arduinoom) preko serijskog protokola koristeći samo dva pina:

- **CLK (clock)** – signal koji označava “početak slanja podatka sa tastature”
- **DATA** – linija kojom se šalje informacija sa tastature, bit po bit

Važno znati da sama tastatura generiše clock signal. Raspored pinova muškog i ženskog PS/2 konektora je prikazan na sljedećoj slici.

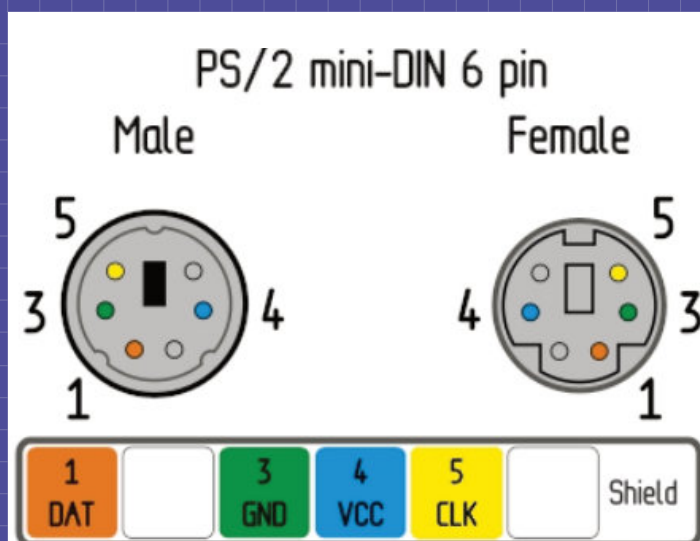


Prije samog upuštanja u izazove kodiranja bilo bi važno razumjeti kako izgleda jedan “pritisak tastera”?

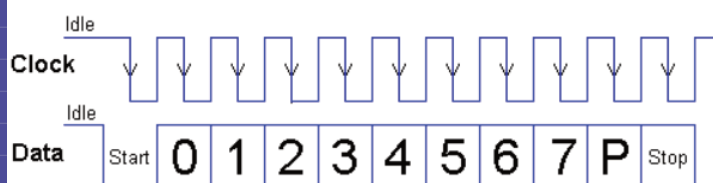
Svaki put kad pritisnemo tipku na tastaturi desi se sljedeće:

1. Tastatura generiše 11-bitni podatak.
2. Ti bitovi idu jedan po jedan, uz ritam koji tastatura diktira (clock).
3. Arduino ih “hvata” i dekodira pomoću biblioteke PS2Keyboard.

Format jednog slanja (11 bita) je dat na sljedećoj slici.



Slika 2.2. PS/2 pinout



Slika 2.3. PS/2 vremenski diagram

BIT	FUNKCIJA	VRIJEDNOST
1	Start bit	Uvijek 0
2-9	Data (8-bita)	ASCII kod ili scancode
10	Parity bit	Za provjeru greške
11	Stop bit	Uvijek 1

- Pritisnete A
- Tastatura šalje: 0 00011100 P 1
- Gdje je 00011100 = **0x1C** → scancode za A

Jako je važno da se zapamti da scancode nije isto što i ASCII kod, stoga bi trebali memorisati sebi ovu informaciju na sljedeći način:

Scancode ≠ ASCII

Prednost je što zu pomoć biblioteke koju ćemo koristiti u primjeru vježbe PS2Keyboard odrađuje pretvaranje **scancodea u karakter**, tako da pritisak tastera vraća identičnu vrijednost pritisnutog tastera: "A", "I", "Enter" itd. Za slučaj puštanja tipke na tastaturi se šalje dodatni kod:

- **0xF0**="Key released"

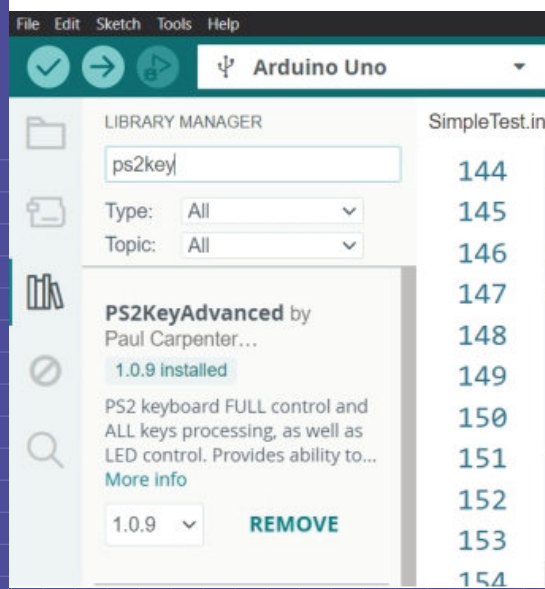
Instalacija biblioteke se radi na identičan način, kao i kod ostalih biblioteka, s tim da za PS/2 imate mnoštvo biblioteka. Bez želje da se preferira bilo koja od njih samo za primjer je prikazana odbrana biblioteka za testiranje aplikacije.

- Slijedi kod tastera koji si pustio

Pri čemu se u većini slučajeva otpuštanje tastera ("key release") zanemaruje, osim ako pravimo naprednije aplikacije (npr. igre). Takt slanja je oko 10-16 kHz, što je dovoljno sporo da mikrokontroler na Arduino Uno razvojnoj pločici može sve da prati i razumije bez problema.

Zašto je PS/2 super za Arduino e-Waste projekte:

- Radi na **5V** – savršeno za Uno
- Nema potrebe za naprednim protokolima kao USB
- Tastatura samo "priča" – Arduino samo "sluša"



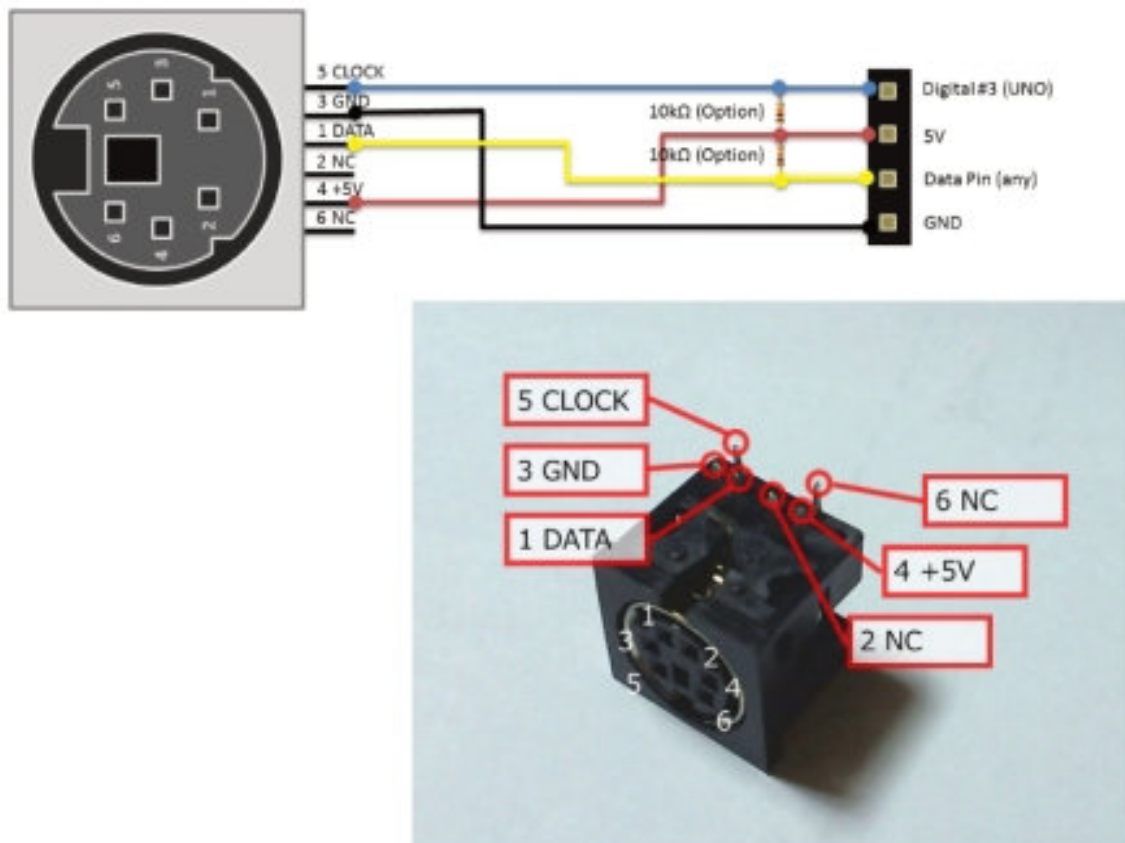
Slika 2.4. PS2KeyAdvanced biblioteka

Potrebni materijali:

Arduino Uno / Nano
 PS/2 tastatura (ili membrana PS/2 tastature)
 PS/2 konektor (ili isječeni kabl iz tastature)
 LCD 2x16 (klasičan HD44780)
 10kΩ potencijometar (za kontrast LCD-a)
 Vodiči, matador ploča

Šema spajanja

Slika 2.5. Šema spajanja



Kako su vježbe sa LCD displejom već rađene, a ovo treba da bude nadogradnja vašeg znanja ovdje će biti prikazana samo šema spajanja PS/2 konektora sa Arduino razvojnom platformom.

Razumijevanjem rada starih PS/2 tastatura otvara vam se čitav niz potencijalnih cool projekata, za početak, krenut ćemo s jednostavnim primjerom ispisivanja otkucanog teksta na Serial monitoru.

Kodiranje

```
#include <PS2KeyAdvanced.h>
#define DATAPIN 4
#define IRQPIN 3
uint16_t c;
PS2KeyAdvanced keyboard;
void setup( )
{
    // IRQ pin mora biti 3 radi postavki biblioteke
    keyboard.begin( DATAPIN, IRQPIN );
    Serial.begin( 115200 );
    Serial.println( "PS2 Advanced Key Simple Test:" );
}
void loop( )
{
    if( keyboard.available( ) )
    {
        // čitaj pritisnute tipke
        c = keyboard.read( );
        if( c > 0 )
        {
            Serial.print( "Value " );
            Serial.print( c, HEX );
            Serial.print( " - Status Bits " );
            Serial.print( c >> 8, HEX );
            Serial.print( " Code " );
            Serial.println( c & 0xFF, HEX );
        }
    }
}
```





CD-ROM

Portalni robot

CILJEVI VJEŽBE:

- Iskoristiti mehanizam za linearno pozicioniranje lasera (jednoosno, dvoosno ili troosno).
- Upravljanje motorom pomoću Arduina.
- Upoznati se s motorima, drajverima i osnovama logike pozicionera.

CD/DVDovi (AI Generisana)



Kada ste zadnji put vidjeli novi prijenosni računar (PC) s ovim nekada neizostavnim uređajem. Možda bi se pitanje moglo drugačije formulirati, da li ste ikada naišli na neispravan CD-ROM?

I ovdje se ponovo radi o izuzetno pouzdanom uređaju, koji ne treba postati e-otpad, već ga je moguće iskoristiti za izradu novih interesantnih uređaja.

Iako se u Arduino starter setu nalaze DC, servo i step motori ono što CD-ROM izdvaja je mehanizam koji se može iskoristiti za pozicioniranje kao jednoosni mehanizam, ili kao portalni robot s dva ili čak tri stepena slobode kretanja.

U ovom priručniku će biti pokazan osnovi koncept upravljanja, a vjerujem da vam ideja neće nedostajati.

CD-ROM

U svakom starom CD-ROM uređaju nalazi se:

- Mali DC motor ili stepper motor
- Zupčanici ili navojna šipka koja pomjera lasersku glavu

Sam process demontaže nepotrebnih elemenata je poprilično jednostavan i zahtijeva samo osnovni alat. Također, bilo bi dobro uraditi selekciju otpada prilikom demontaže jer se većina elemenata može reciklirati. Krajnji cilj je dobiti stanje kao na sljedećoj slici.



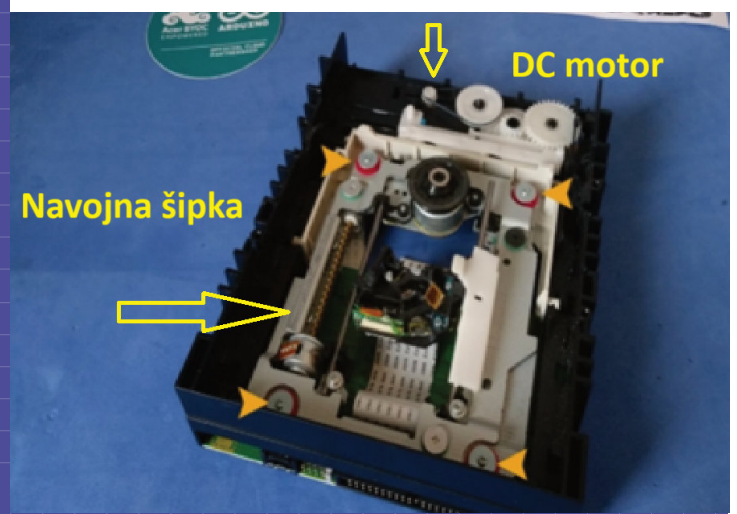
Često se kod projekata vezanih za arduino zaboravlja na važnost znanja vezanih za mehaniku, ali ovo je jedan primjer gdje učenice i učenici mogu vidjeti kako su iskusni mašinski inženjeri i inženjerke realizovali rješenje koje ne zahtijeva mnogo troškova a koje ima izuzetnu pouzdanost.

Ovaj mehanizam se zbog toga jednostavno i lako pretvara u:

- Linearnu rampu
- X-ili-Y osu pozicionera
- Mali plotter
- Portalni robot

Kao što smo ranije već naveli, zavisno od modela u CD-ROM se najčešće mogu naći sljedeći motori:

- Mali DC motor (dvije žice)
- Stepper motor (četiri ili više žica)



Slika 3.1. CD-ROM-a nakon skidanja zaštitnog lima

Bez obzira koji je motor unutar CD-ROM-a za upravljanje možemo iskorisiti dobro poznati L239D motor drajver jer je on svakako najjednostavniji za upravljanje, a kako je ovaj drajver kao i sam stepper motor detaljno opisan u prethodnim priručnicima ovdje ćemo se fokusirati na realizaciju projekta pozicioniranja i iskorištavanja same mehanike CD-ROM-a.



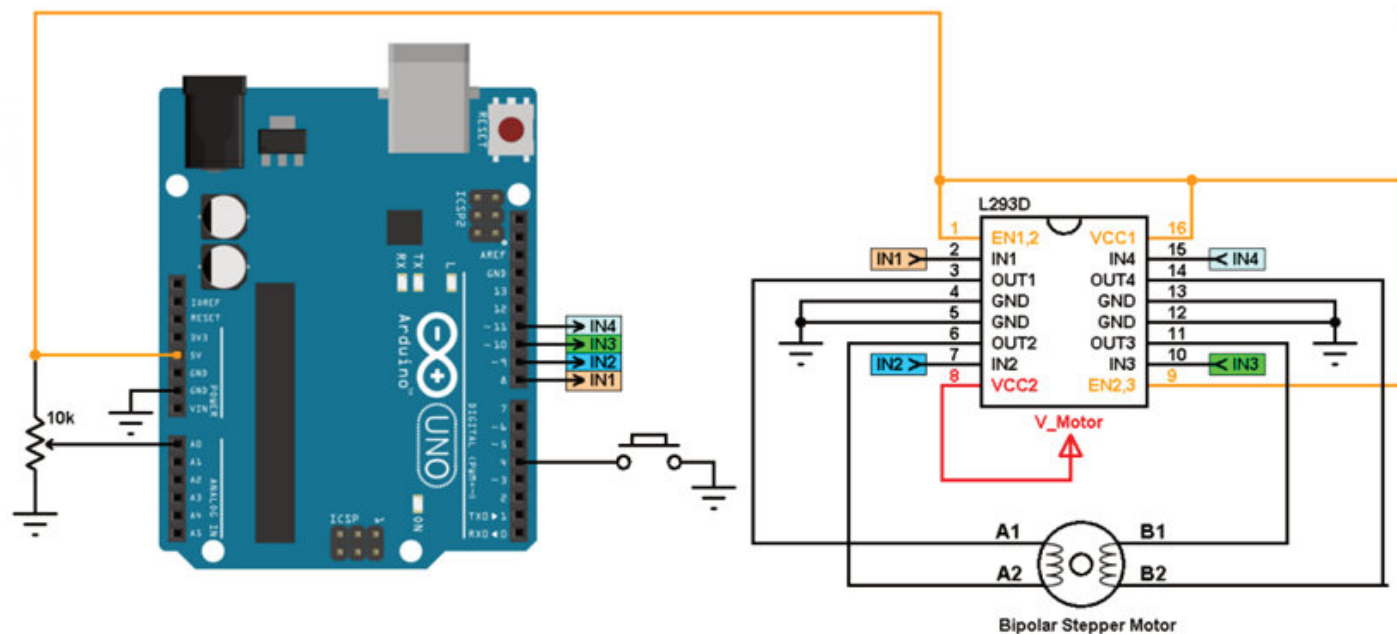
Slika 3.2. Mehanizam spreman za upotrebu

Potrebni materijali:

- Arduino Uno
- Stari CD-ROM (izvaditi ladicu i motor)
- L293D drajver
- Napajanje (12V, npr. AC/DC adaptera ili baterija)
- Vodiči, odvijači, matador ploča
- (Opciono) 2x CD-ROM za dvoosnu kontrolu

Šema spajanja

Slika 3.3. Šema spajanja



Kao što se vidi iz same šeme spajanja, ideja je da se realizira linearni pozicioner s kontrolom brzine pozicioniranja putem analognog ulaza na koji je spojen potencijometar i tasterom podešenim sa aktiviranim internim pull-up za kontrolu smjera pozicioniranja motora.

Kodiranje

```
#include <Stepper.h>
#define STEPS 20
// kreiranje instance i podešavanje korištenih pinova
Stepper stepper(STEPS, 8, 9, 10, 11);
const int button = 4;
const int pot     = A0;
void setup()
{
    pinMode(button, INPUT_PULLUP);
}
int direction_ = 1, speed_ = 0;
void loop()
{
    if ( digitalRead(button) == 0 )
        if ( debounce() )
        {
            direction_ *= -1;
            while ( debounce() ) ;
        }
    int val = analogRead(pot);

    // map value
    if ( speed_ != map(val, 0, 1023, 5, 100) )
    {
        speed_ = map(val, 0, 1023, 5, 100);
        stepper.setSpeed(speed_);
    }

    stepper.step(direction_);
}
// pomocna funkcija za debounce
bool debounce()
{
    byte count = 0;
```





```
for(byte i = 0; i < 5; i++) {  
    if (digitalRead(button) == 0)  
        count++;  
    delay(10);  
}  
if(count > 2) return 1;  
else        return 0;  
}
```

4



CPU ventilator kao indikator temperature

CILJEVI VJEŽBE:

- Spojiti **CPU ventilator** sa **3 žice** na Arduino.
- Očitavati temperaturu pomoću senzora **TMP36**.
- Generisati **PWM signal od 25 kHz** kojim upravljamo brzinom ventilatora na osnovu temperature.

UČENJE:

- Rad sa **analogno-digitalnim pretvaračem (ADC)**
- Upravljanje **PWM frekvencijama** (Timer1)
- Praktičnu primjenu **termičkog nadzora** kao kod pravog CPU-a

Dječak sa PC ventilatorima (AI Generisana)



Kroz ovu vježbu učenici i učenice će naučiti kako reciklirati dijelove iz starih računara i koristiti ih u novim elektronskim projektima. Konkretno, koristiti će **ventilator iz PC napajanja i senzor temperature TMP36 za pravljenje jednostavnog sistema hlađenja**. Važno je napomenuti da je za ovu vježbu potrebno koristiti dvožični PC ventilator.

Ventilator iz napajanja

Mali 12V ventilatori se nalaze u gotovo svakom PC napajanju. Oni se mogu iskoristiti u projektima za hlađenje, ali i kao pokazatelji temperature. Kao što se vidi na slici iznad, u jednom PC-u koji je planiran za otpis kao elektronički otpad, moguće je naći barem jedan od gore navedenih ventilatora. Što ventilator ima više vodiča na sebi to su njegove mogućnosti veće. Da bi naučili razliku navesti ćemo osnovne.



Slika 4.1. Tipovi ventilatora u PC-u

- **Dva vodiča** - najjednostavniji za upravljanje, može se aktivirati direktnim spajanjem na bateriju ili preko pomoćnog uređaja - tranzistora ili releja. Nema kontrole brzine niti očitavanja stanja. Ovo će biti opcija za našu vježbu.
- **Tri vodiča** - +12V, GND, i signal za očitavanje brzine (žuta žica – tzv. "tach signal"). Omogućava očitavanje broja obrtaja (RPM) pomoću matične ploče ili mikrokontrolera, ali nema mogućnosti kontrole brzine (radi punom snagom).
- **Četiri vodiča** - Dodana plava žica za PWM kontrolu brzine, odnosno mogućnost kontrole brzine putem modulacije širine impulsa (PWM, Pulse Width Modulation). I dalje omogućava očitavanje brzine (žuta). Najčešći je u modernim računarima.

Ponovit ćemo, više vodiča znači više kontrole i informacija – od prostog uključivanja, preko mjerenja obrtaja, do precizne kontrole brzine rada ventilatora.

Ventilator iz napajanja

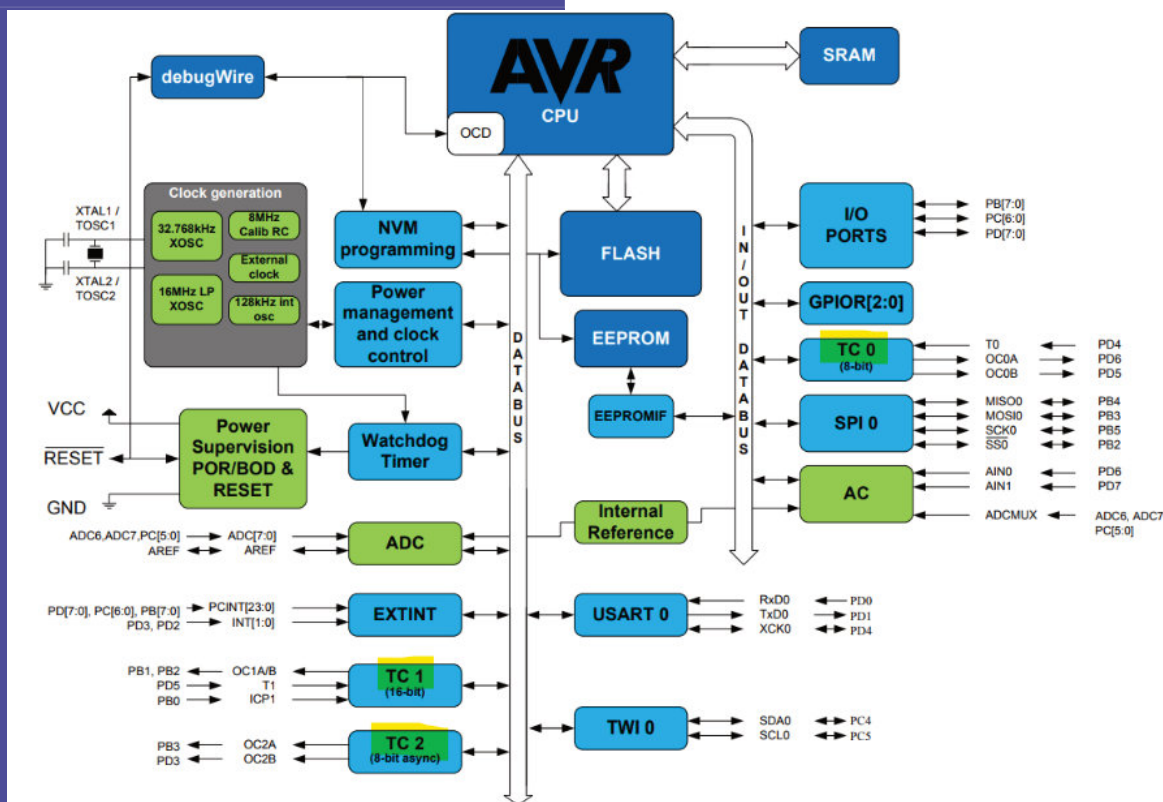
Ova vježba ima višestruku težinu jer ćemo kroz nju prvi put malo detaljnije upoznati sam hardver mikrokontrolera kao i malo detaljnije razumjeti koncept rada jedne od najvažnijih periferija mikrokontrolera, timer periferiji.

Naime jedan od prvih ozbiljnih problema kod izrade kompleksnih projekata s kojim se početnici i početnice u pisanju aplikacija za mikrokontrolere nađu je da različite biblioteke integrisane u projekat koriste iste hardverske resurse mikrokontrolera.

Pojednostavljeno to znači da dva koda od dvije različite biblioteke pokušavaju da istovremeno koriste jedan hardver mikrokontrolera, to je kao da pokušate na jedan VGA ulaz spojiti dva monitora, jednostavno imate konflikt i zbog toga se vaša aplikacija ne ponaša kako bi trebalo.

Obično ovo podrazumjeva učenje na višem nivou i istraživanje samog hardvera mikrokontrolera i njegovih periferija. Može se slobodno reći da na ovakvim projektima izlazite iz komfort zone Arduino razvojnog okruženja i postajete senior programer/programerka.

Pa da vidimo u čemu je problem. Na slici ispod je prikazan blok dijagram AVR mikrokontrolera. S označenim tajmerima TC 0, TC 1 i TC 2.



Slika 4.2. Blok dijagram AtMega328p

PINOVI	TIMER	FREKVENCIJA
5,6	Timer 0	~976 Hz
9,10	Timer 1	~490 Hz
3,11	Timer 2	~490 Hz

Na Arduino Uno, funkcija `analogWrite(pin, vrijednost)` koristi podešene PWM frekvencije, koje zavise od tajmera kako je to prikazano u tabeli iznad. Unutar Arduino razvojnog okruženja PWM biblioteka je podešena tako da pinovi obično koriste **490 Hz**, što je sasvim dovoljno za LED, buzzer i standardne ventilatore, ali u velikom broju slučajeva CPU ventilatori očekuju **~25 kHz**.

Rješenje ovog problema je fino podešavanje periferije Timer1 i precizna kontrola Timer1 je 16-bitni tajmer u AVR mikrokontroleru (npr. ATmega328P), te za razliku od Timer0 i Timer2 (koji su 8-bitni), Timer1 može da broji od 0 do 65535, što omogućava vrlo precizno podešavanje frekvencije!

Formula za PWM frekvenciju:

$$f = \frac{f_{clock}}{N \cdot (1 + TOP)}$$

Gdje je:

- f = željena frekvencija (25,000 Hz)
- f_{clock} = 16 MHz (standardni Arduino takt)
- N = prescaler (koliko se usporava tajmer)
- TOP = vrijednost do koje tajmer broji (ICR1-registar)

Ako uzmemo:

- $f = 25,000$
- $f_{clock} = 16 \text{ MHz}$
- $N = 1$ (Bez prescalera)

$$TOP = \left(\frac{16,000,000}{25,000 \cdot 1} \right) - 1 = 639$$

Znači, ako **tajmer broji od 0 do 639**, frekvencija će biti tačno **25 kHz**! Zato koristimo Timer1 (16-bitni tajmer) i direktno ga konfigurišemo za 25 kHz.

Za pin 9 koji smo odlučili da koristimo potrebno je da u dokumentaciji o samom mikrokontroleru, nađemo informaciju o opisu sljedećih registara :

- OCR1A = koliko dugo signal ostaje HIGH
- ICR1 = gornja vrijednost za brojanje (TOP)

Duty cycle ili radni ciklus se računa kao:

$$\text{Duty cycle} = \frac{OCR1A}{ICR1}$$

I za kraj najvažnije, ovu vježbu je moguće realizirati samo sa CPU ventilatorom s četiri priključna pina, slika ventilatora i pinout je prikazan na sljedećoj slici.

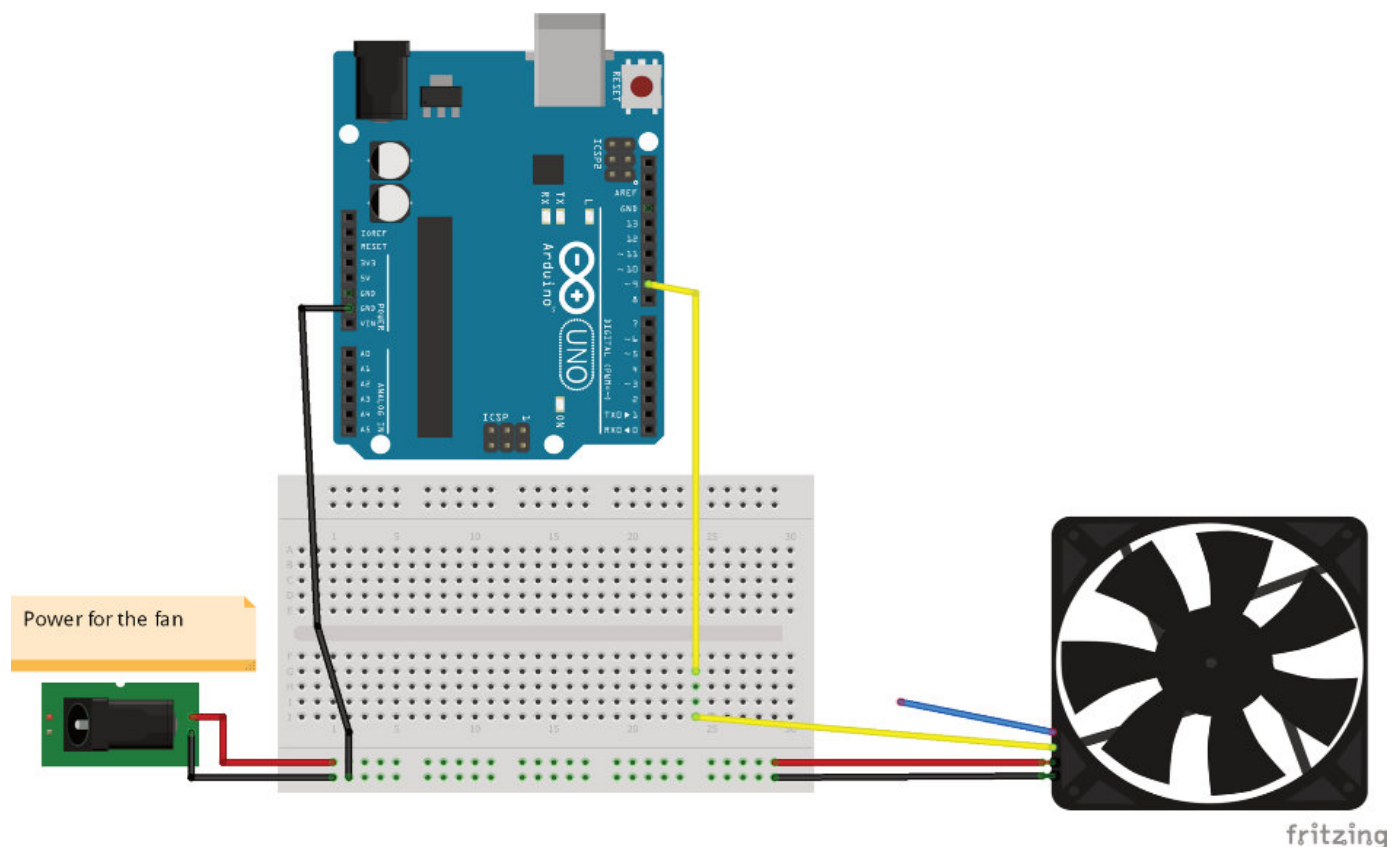
Kako je u prethodnim vježbama već korišten temperaturni senzor TMP36, šema spjanja će biti prikazana samo za četiri pinski CPU ventilator.



Slika 4.3. CPU ventilator pinout

Šema spajanja

Slika 4.4. Šema spajanja



NAPOMENA: prije realizacije vježbe provjeriti datasheet proizvođača ventilatora, provjeriti radnu frekvenciju i pinout.

Kodiranje

```
const int tempPin = A0; // TMP36 izlaz
const int fanPin = 9;   // PWM na D9

void setup() {
    pinMode(fanPin, OUTPUT);
    Serial.begin(9600);

    // Konfiguracija Timer1 za 25 kHz PWM na pin 9
    TCCR1A = _BV(COM1A1) | _BV(WGM11); // Fast PWM, non-
inverting
    TCCR1B = _BV(WGM13) | _BV(WGM12) | _BV(CS10); // Fast PWM,
prescaler 1
    ICR1 = 639; // TOP vrednost -> 16MHz / (639+1) = 25 kHz
}

void loop() {
    // Očitaj temperaturu
    int value = analogRead(tempPin);
    float voltage = value * 5.0 / 1023.0;
    float temperature = (voltage - 0.5) * 100;

    Serial.print("Temp: ");
    Serial.print(temperature);
    Serial.println(" °C");

    // Mapiraj temperaturu u duty cycle (25°C = 0%, 60°C =
100%)
    int pwmDuty = map(temperature, 25, 60, 0, ICR1); // ICR1
= 639 → puni radni ciklus
    pwmDuty = constrain(pwmDuty, 0, ICR1);

    OCR1A = pwmDuty; // Pošalji PWM na pin 9

    delay(1000); // Pauza 1s
}
```





Morzov kod iz zvučnika matične ploče

CILJEVI VJEŽBE:

- Spojiti PS/2 tastaturu na Arduino.
- Arduino očitava karaktere koje korisnik ili korisnica unosi.
- Za svaki karakter, Arduino ga pretvara u Morzov (Morse) kod.
- Morzov kod se:
 - Oglašava preko piezo zvučnika (tzv. "motherboard speaker")
 - Blinka LED-icom u istom ritmu

Morzov kod (AI Generisana)



I za kraj jedna jednostavna vježba sa komponentom čiji zvuk kada čujemo obično znači neki problem s našim PC-om. **Motherboard speaker** ili **PC beeper** je mali **piezoelektrični zvučnik** koji se ili povezivao direktno na matičnu ploču računara ili je s dva kablčića vezan na koentor na matičnoj ploči. Najčešće se koristi za:

- **Beep** kodove prilikom POST testa
- Signalizaciju grešaka (npr. RAM nije ubačen)
- Ponekad i za osnovne zvučne efekte u DOS igrama

Piezo zvučnik može biti aktivni ili pasivni, razlika se očituje u načinu upravljanja, aktivni ima ugrađen oscilator, to znači da mu je upravljanje dovoljna logička HIGH/LOW komanda. Pasivni nema oscilator, što znači da je potrebno koristiti funkciju tone() da se generiše zvuk. Napon napajanja iznosi 5V, te sam buzzer ima jako malu potrošnju.

Prosječan korisnik jako teško može iskoristiti neke elemente s matične ploče, ali je bitno da se učenicama i učenicima objasni da je svaka komponenta, ukoliko je ispravna, vrijedna reciklaže. Energija koje je potrebna da se ponovo izgradi ta komponenta bi se mogla iskoristiti za proizvodnju nečeg drugog.

Šta će biti novi život reciklirane komponente koja bi mogla postati e-otpad, direktno zavisi od mašte i vještine onoga ko je planira iskoristiti.

Stoga je namjerno za kraj ostavljena vježba koja nas vraća u 19. vijek jer kucamo SOS poruke kao pravi telegrafisti iz tog doba... ali s Arduinom!

Ideja je jednostavna Morzov kod predstavlja slova i brojeve kroz **kratke (dot)** i **duge (dash)** zvučne i/ili svjetlosne impulse:

- **Dot (.)** = kratak ton (npr. 200 ms)
- **Dash (-)** = duži ton (npr. 600 ms)
- Pauze između:
 - Dva simbola: 200 ms
 - Dva slova: 600 ms
 - Dvije riječi: 1400 ms

Primjer: **SOS** = ... --- ...



Slika 5.1. Zvučnik s matične ploče

Kodiranje

```
#include <PS2Keyboard.h>

const int DataPin = 5;
const int IRQpin = 6;
PS2Keyboard keyboard;
const int buzzerPin = 8;
const int ledPin = 7;
// Morse kod tabela
const char* morseTable[36] = {
    ".-", "-...", "-.-.", "-..", ".", "...-", "--.",
    "....", "..", "----", "-.-", "-...", "--", "-.",
    "---", ".---", "--.-", ".-.", "...", "-", "...-",
    "...-", ".---", "-...-", "-.-.-", "-...", // A-Z
};

void setup() {
    pinMode(buzzerPin, OUTPUT);
    pinMode(ledPin, OUTPUT);
    keyboard.begin(DataPin, IRQpin);
    Serial.begin(9600);
    Serial.println("Kucaj tekst za Morse kod:");
}

void loop() {
    if (keyboard.available()) {
        char c = keyboard.read();
        Serial.print(c);
        if (c == ' ') {
            delay(1400); // pauza između riječi
        } else if (c >= 'A' && c <= 'Z') {
            playMorse(morseTable[c - 'A']);
        }
    }
}

void playMorse(const char* code) {
    for (int i = 0; code[i] != '\0'; i++) {
        if (code[i] == '.') {
            signal(200);
        } else if (code[i] == '-') {

```





```
        signal(600);
    }
    delay(200); // razmak između znakova
}
delay(600); // razmak između slova
}
void signal(int duration) {
    digitalWrite(ledPin, HIGH);
    tone(buzzerPin, 1000); // 1 kHz ton
    delay(duration);
    noTone(buzzerPin);
    digitalWrite(ledPin, LOW);
}
```

Zaključak

Ako su vas ove vježbe nečemu naučile, onda je to da stari kompjuterski dijelovi nisu otpad – oni su zlatni rudnik za sve koji vole lemilicu, Arduino i malo eksperimentiranja. Umjesto da završe na nekoj elektroničkoj deponiji, stari CD-ROM-ovi, PS/2 tastature, hladnjaci i kablovi dobili su novu priliku da blistaju u svijetu pametnih projekata. Na taj način ne spašavamo samo par KM u budžetu, nego i planetu.

Kroz ovih nekoliko praktičnih primjera pokazali smo da uz malo znanja, puno entuzijazma i nešto starog hardvera, mogu nastati korisni, zanimljivi i edukativni projekti.

Dakle, sljedeći put kad vidite stari računar na putu prema otpadu, možda nećete samo proći pored njega – možda ćete ga ponijeti kući i pretvoriti u robota. Ili bar nešto što svijetli, zuji i spašava Zemlju. Pomalo.

