

**»Mladi za napredek Maribora 2016«  
33. srečanje**

**DIGITALNA URA Z RADIEM**

Raziskovalno področje: ELEKTROTEHNIKA, ELEKTRONIKA

Raziskovalna naloga

PROSTOR ZA NALEPKO

Avtor: ROK DOLENC  
Mentor: MILAN IVIČ  
Šola: SREDNJA ELEKTRO-RAČUNALNIŠKA ŠOLA

**2016, Maribor**

## Kazalo vsebine

1. UVOD.....	4
2. URA.....	5
2.1 SONČNA URA.....	5
2.2 VODNA IN PEŠČENA URA.....	6
2.3 MEHANSKA URA.....	7
2.4 DIGITALNA URA.....	7
3. ARDUINO.....	8
3.1 ARDUINO UNO.....	9
4. RTC.....	10
5. IR sprejemnik in oddajnik.....	11
6. TEMPERATURNI SENZOR.....	12
7. LCD in tipke.....	14
8. RADIO.....	14
9. ALARM.....	15
10. PROGRAM.....	15
11. NAPAJANJE.....	17
12. DELOVANJE.....	18
12.1 TIPKE.....	18
12.2 UPRAVLJANJE URE.....	19
13. ZAKJUČEK.....	20
14. VIRI.....	21
14.1 VIRI SLIK.....	21

Slika 1: Sončna ura Obelisk (Vir: skyscanatomicclock) .....	5
Slika 2: Peščena ura (Vir: iconfinder) .....	6
Slika 3: Vodna ura Egipčanov (vir: beforeitsnews) .....	6
Slika 4: Mehanska ura (vir: tuxgraphics).....	7
Slika 5: Digitalna ura ( Vir: Conrad).....	7
Slika 6: Arduino program (Vir: Avtor naloge) .....	8
Slika 7: Arduino UNO ploščica (Vir: arduino).....	9
Slika 8: RTC DS1307 (Vir: atomsindustries ) .....	11
Slika 9: TL 1838 (Vir: webshopapp).....	11
Slika 10: Temperaturni senzor ADT75 (Vir: alicdn ) .....	12
Slika 11: Temperaturni register v ADT75 (Vir: Analog ) .....	13
Slika 12: Primeri izmerjenih temperatur (Vir: Analog ).....	13
Slika 13: LCD 4x20 (Vir: embeddedadventures ).....	14
Slika 14: Radio modul (Vir: alicdn) .....	15
Slika 15: Deklarirane knjižnice in spremenljivki (Vir: Avtor naloge).....	16
Slika 16: Program zapisan znotraj void loop(a) (Vir: Avtor naloge) .....	16
Slika 17: Primer zapisa kode (Vir: Avtor).....	17
Slika 18: Dioda (Vir: Avtor naloge).....	17
Slika 19: Shema napajalnega dela vezja( Vir: Wikipedia) .....	18
Slika 20: Pogledi napetosti z osciloskopom (Vir: Avtor naloge).....	18

## 1. UVOD

Namen te raziskovalne naloge je bilo spoznati delovanje digitalne ure, radia in programiranje v C jeziku, hkrati pa sem sam želel izdelati nek uporaben izdelek, ki mi bo praktično služil do konca življenja, saj bo po mojih sklepanjih čas vedno pomemben podatek.

Že dolgo časa sem razmišljal o izdelavi ure, katera me bo vsako jutro zbudila in mi sporočala pomembne podatke kot so čas, datum, dan in temperatura. In prišel je čas ko sem dobil priložnost in zagon za izdelavo te digitalne ure, pri kateri sem se imel namen marsikaj naučiti kot tudi izdelati uporaben izdelek. Uro sem želel opremiti tudi z radiem, ki bi ga lahko poslušal ob prostem času.

## 2. URA

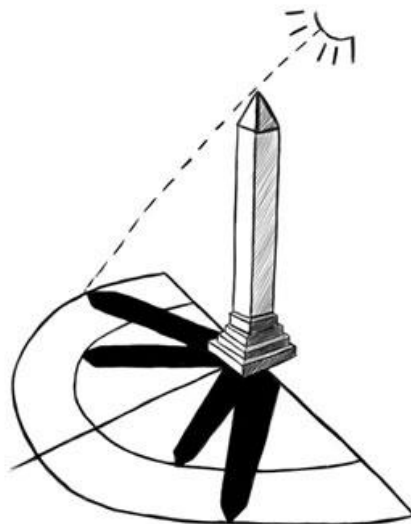
Ura je naprava, ki v glavnem kaže čas s kazalcem na številčnici ali pa numerično na LCD zaslonu. Poznamo več vrst ur:

- Peščena ura,
- Sončna ura,
- Vodna ura,
- Mehanska ura in
- Digitalna ura, katera se predvsem uporablja danes.

Ura je ena najstarejših človeških izumov. V prazgodovinskem času so ljudje uporabljali sonce za določanje časa v dnevu. Ko je sonce zjutraj vzhajalo, so vedeli, da se je dan pričel in nastopil je čas za njihovo prebujanje. Vedeli so tudi kdaj je opoldne dneva, ko je bilo sonce na nebu na najvišji točki. Ko je sonce pričelo zahajati, so vedeli, da je čas za vrnitev v prenočišče in pripravo za nočitev.

### 2.1 SONČNA URA

Prva ura je bila vsekakor sončna ura, ki je delovala s pomočjo sence. To je bila ura po imenu Obelisk, ki je nastala okoli leta 3500 pr. n. št. po zaslugi Egipčanov. To je bila mogočna ura v višino sestavljena iz velikih kamnitih kock in je tehtala več ton. Do sedaj je ostalo le še 26 Egipčanskih obeliskov.



Slika 1: Sončna ura Obelisk (Vir: skyscanatomicclock)

## 2.2 VODNA IN PEŠČENA URA

Sončna ura, ki je lahko prikazovala točen čas le po dnevu, je postala stvar preteklosti. Razvila se je vodna ura, ki je merila čas skozi dan in noč. Poleg 24-urnega merjenja časa, je imela prednosti tudi v boljši natančnosti. Vodna ura se je razvila v Egiptu in postala zelo priljubljena v Grčiji, kjer je bila deležna rednih razvojev. Grki so z leti razdelili takratne merske enote časa v enote, ki jih poznamo danes - mesci, tedni in dnevi (minute in sekunde izvirajo iz sumerskega matematičnega sistema). Vendar, ker dan in noč nista enako dolga, grški sistem ni bil popoln in popolnoma točen, zato je bilo vodno uro potrebno vsak dan ponastaviti. Vodna ura je delovala na podobnem principu kot peščena ura. Voda je iztekala iz velikega vrča in na posameznih gladinah vode je bilo označeno število, ki je ponazarjalo čas.



Slika 2: Peščena ura (Vir: iconfinder)



Slika 3: Vodna ura Egipčanov (vir: beforeitsnews)

## 2.3 MEHANSKA URA

Prvi mehanski merilci časa so se pojavili okoli leta 1500 pr. n. št. v Egiptu, prve velike mehanske ure pa so izdelali italijanski menihi, da bi ljudem oznanili čas molitve. Te ure še niso imele pravih kazalcev, čas so sporočali za to namenjeni posebni zvončki.



Mehanska ura danes je sestavljena iz velikega števila mehanizmov in zobnikov. Mehanska ura deluje s pomočjo vzmeti, katero je potrebno navijati. Prve prenosne ure so se pojavile v 16. stoletju izumili pa so jih v Angliji. Bile so izredno velike in okorne, zato so jih nosili kar okoli vratu. Imele so le urni kazalec, navijati pa jih je bilo potrebno dvakrat dnevno. V tem času so se začeli pojavljati tudi prvi žepni časomeri, a so bili dovolj izpopolnjeni šele v 17. stoletju.

*Slika 4: Mehanska ura (vir: tuxgraphics)*

## 2.4 DIGITALNA URA

Digitalna ura je ura, kjer ni več prisotna mehanika temveč elektronika. Gre za popolnoma elektronsko obliko naprave (ure). Digitalna ura ima numeričen prikaz števil (časa) na LCD-ju. Prikazan je lahko tudi datum, dan, leto kot pri mehanski uri. Digitalna ura pa omogoča še pomnjenje in merjenje več podatkov kot so starost, dolžina koraka, temperatura, merjenje korakov in hkrati prehojene dolžine, ipd.

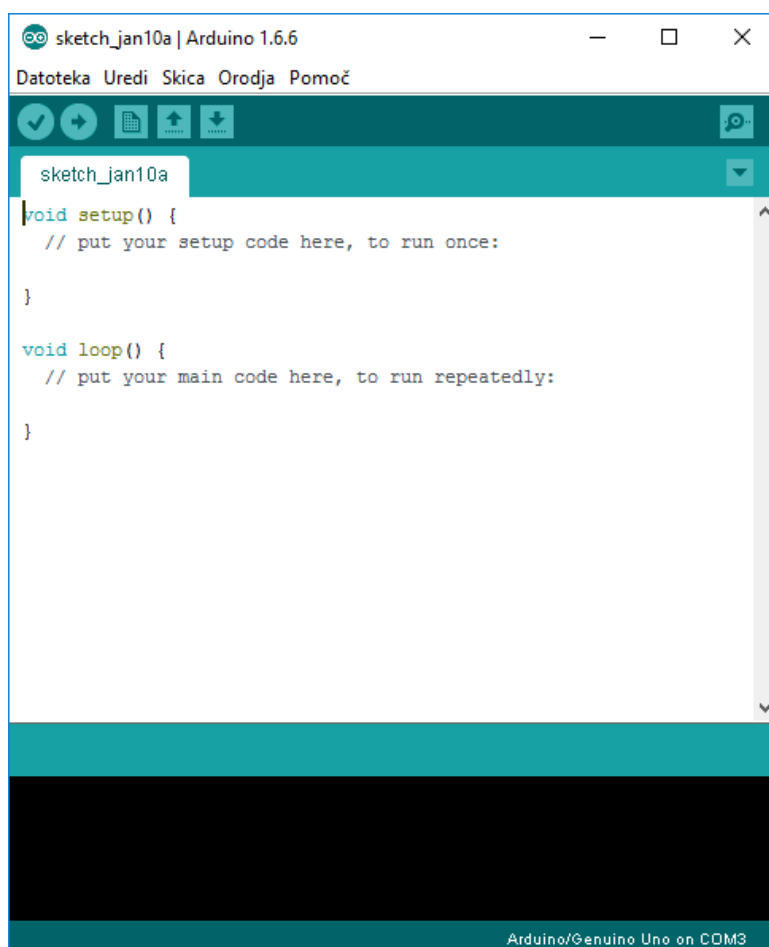


*Slika 5: Digitalna ura (Vir: Conrad)*

### 3. ARDUINO

Za osnovo izdelave ure sem uporabil Arduino Uno ploščico. To je eksperimentalna ploščica, namenjena za razvoj nekega vezja, oz. kasneje nastanek nekega novega produkta. Arduino je podjetje, ki nam ponuja veliko število razvojnih ploščic (NANO, UNO, MEGA) in nam nudi brezplačno uporabo Arduino programa. V bistvu celotni koncept Arduina temelji na odprtokodni programski in strojni opreми.

Program Arduino je prilagojen začetnikom in omogoča enostavno kot tudi zahtevnejšo uporabo oz. programiranje. Programiranje v programu Arduino temelji na C ali C++ jeziku. Osnovno okno urejevalnika z osnovno programsko strukturo zglada takole:



```
sketch_jan10a | Arduino 1.6.6
Datoteka Uredi Skica Orodja Pomoč

sketch_jan10a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

Arduino/Genuino Uno on COM3
```

Slika 6: Arduino program (Vir: Avtor naloge)

V začetku programa vidimo vrstico »VOID SETUP«. Pove nam da bo program, kar je zapisano znotraj zavrtih oklepajev, izvedel le enkrat.

Nekaj vrstic spodaj pa vidimo napisano »VOID LOOP« kar v dobesednem prevodu pomeni zanka. Vse kar bo napisano znotraj teh zavrtih oklepajev se bo ponavljalo v nedogled.



Arduino program uporablja tudi tako imenovane knjižnice, ki zagotavljajo dodatne funkcionalnosti za uporabo v skicah (programu v Arduinu rečemo skica), npr. delo s strojno opremo ali manipuliranje podatkov.

### 3.1 ARDUINO UNO

Arduino UNO je verjetno najbolj priljubljena med vsemi Arduino razvojnimi ploščicami. Kot osnovo ima Atmelov mikrokontroler ATmega328P. Na ploščici UNO je na voljo 14 digitalnih vhodno/izhodnih (I/O) priključkov (pinov) od tega 6 s funkcijo PWM (Pulzno širinska modulacija). Dodatno je na razpolago 6 analognih vhodnih pinov, ki pa se prav tako lahko definirajo in uporabljajo kot digitalni I/O pini. Takt delovanja mikrokontroler je določen s kristalom in se navaja v MHz ali GHz in pomeni število taktov, ki jih procesor izvede v sekundi. Takt tega mikrokontroler je 16 MHz. Arduino UNO ploščico lahko napajamo preko USB priključka s petimi volti ali pa preko DC priključka z 7-12 volti (maksimalno 20V). Največji možni tok posameznega pina je 20 mA.

Arduino UNO je deljen na dva osnovna dela:

Prvi del sestavlja Atmelov mikrokontroler ATmega328P, katerega vhodno/izhodni pini so speljani na priključne letvice (ženske) na katere lahko priključimo zunanje elemente.

Drugi del pa sestavlja programator. ATmega 16U2 je glavni programatorski čip, ki je vmesnik med Atmelovim mikrokontrolerom 328P in USB priključkom računalnika.



Slika 7: Arduino UNO ploščica (Vir: arduino)

Arduino Uno ploščico sem uporabil samo za razvoj ure z FM radiem, tako da sem z Arduino programom programiral čip in ga za ta namen vzpostavil z računalnikom. Kasneje sem za uro zasnoval samostojno tiskano vezje, ki sem ga narisal s pomočjo Eagle programa za

načrtovanje ploščic tiskanega vezja. Ploščico tiskanega vezja sem izdelal s foto-postopkom in jedkanjem.

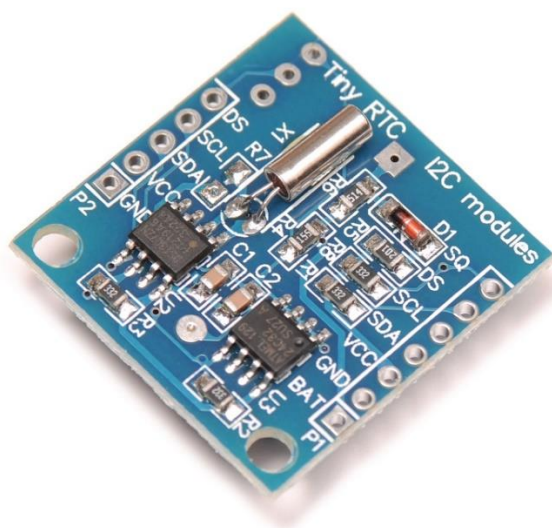
#### 4. RTC

RTC ali »Real Time Clock« je računalniška ura v kateri je zapisan čas, ki ga lahko spreminjamo in po želji beremo. Je integrirano vezje, ki ima svoje napajanje, saj je montiran na posebnem tiskanem vezju, ki ima vgrajeno baterijo, tako da čas teče neodvisno od osnovnega napajanja ostalih komponent. Prebrani čas je torej vedno realni čas, odvisen od začetne nastavitve časa in točnosti samega RTC-ja. Točnost pa je v veliki meri odvisna od priključenega kristala z vrednostjo 32.768 Hz.

Sam sem uporabil RTC DS1307, ki ima za namen komunikacije (vpisovanje in branje parametrov) vgrajeno tudi I<sup>2</sup>C vodilo. Koledar v DS1307 zagotavlja sekunde, minute, ure, dneve, dneve v mesecu, mesece in leta. Datum se pravilno prestavi ne glede na število dni v posameznem mesecu in upošteva tudi prestopno leto. Ura deluje s 24 ali 12-urnim formatom z AM (dopoldan) in PM (popoldan) indikatorjem. Njegova tipična napajalna napetost je 5 V. Stanje ure oziroma čas vpišemo ali preberemo preko 2 pinov imenovanih SCL (serial clock input) in SDA (serial data output/input). Skupaj z maso tvorijo ti pini t.i. I<sup>2</sup>C vodilo (Inter-Integrated Circuit). Vodilo je bilo razvito s strani podjetja Philips v začetku 80-ih let z namenom serijske povezave (izmenjava podatkov) več integriranih vezij na nekem manjšem območju. Na vodilu je lahko več integriranih vezij tipa »master« ali »slave«, ki se med seboj ločijo po naslovu oziroma adresi.

**SDA** je vhodno/izhodni priključek, preko katerega se pretakajo podatki, v našem primeru parametri časa in ostali podatki. Fizično gre za »open drain« (odprti ponor) priključek, ki zato zahteva stalni upor proti pozitivnemu napajanju. V praksi to pomeni upor, vezan med SDA in +5V, vrednosti med nekaj kohm-i do 10 kohm-i.

**SCL** je vhodni priključek, ki je namenjen sinhronizaciji pretoka podatkov za serijski vmesnik. Gre prav tako za »open drain« priključek, ki za pravilno delovanje potrebuje priključitev t.i. »pool-up« upora.



Slika 8: RTC DS1307 (Vir: atomsindustries )

V arduino programu sem za programiranje RTC – modula uporabil knjižnico »RealTimeClockDS1307.h«, ki sem jo našel na spletni strani »instructables«. V navodilih za uporabo knjižnice sem poiskal ukaze za vpisovanje (nastavljanje) in branje časa, datuma, ... ter na podlagi tega sprogramiral komunikacijo med RTC-jem in Arduino ploščico. RTC sem z Arduino ploščico povezal preko analognih pinov A4 (SDA) in A5 (SCL).

## 5. IR sprejemnik in oddajnik

Uro je možno nastavljati s pomočjo tipk, ker pa nam ura ni ves čas v dosegu, da bi jo lahko nastavljali s tipkami, sem dodal še IR oddajnik in sprejemnik za nastavljanje ure z daljave.

Za IR brezžično upravljanje ure sem uporabil mp3 car oddajnik in sprejemnik, ki sem ju naročil preko spleta. Sprejemnik ima 3 pine. Dva sta namenjena napajanju in eden za signal. Oddajnik pa pošilja kodo, katero preberemo z Arduino programom in na podlagi tega sprožimo nek ukaz, ki se izvrši. To v programu napišemo oz. definiramo podobno kot tipke.



Slika 9: TL 1838 (Vir: webshopapp)

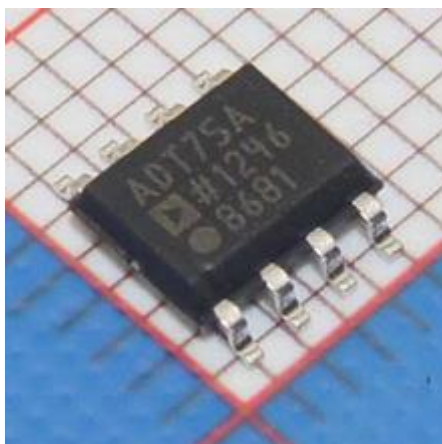
Za IR sprejemnik sem uporabil HX 1838 modul. Njegova delovna napetost je med 3,3 in 5,5 V in porabi 1,4 mA kar je izredno malo. Sprejemnik deluje z 38 KHz kar je frekvenca nosilca IR signala. V Arduino programu sem za IR oddajnik in sprejemnik uporabil knjižnico »IRremote.h« katero sem našel na Arduino spletni strani. V knjižnici je že definirano, da kodo katero sprejme od oddajnika, IR sprejemnik pretvori iz binarne v heksadecimalno kodo saj binarna pride veliko daljša za napisat.

PRIMER:

FFE01F (HEX) = 1111 1111 1110 0000 0001 1111 (BIN)

## 6. TEMPERATURNI SENZOR

Temperatura je prav tako kot čas pomemben podatek, zato sem v uro z radiem vgradil še temperaturni senzor, ki bo meril temperaturo v prostoru, kjer se bo ura nahajala. Za temperaturni senzor sem uporabil integrirano vezje ADT75, ki z Arduino ploščico prav tako kot RTC, komunicira preko I<sup>2</sup>C vodila torej preko SDA in SCL pinov na arduino ploščici. Kot že omenjeno, lahko preko I<sup>2</sup>C vodila komunicira več komponent hkrati.



Slika 10: Temperaturni senzor ADT75 (Vir: alicdn )

ADT75 zraven temperaturnega senzorja vsebuje še 12-bitni analogno-digitalni pretvornik, ustrezne registre za hranjenje parametrov in potrebno krmilno logiko. Vse skupaj je v 8 nožnem SMD ohišju, omogoča pa meritve temperature od -55 do +125 °C. Napaja se z napetostjo od 2,7 do 5,5 V, za delovanje pa pri napetosti 3,3 V porabi ca. 200 µA. Izmerjena temperatura se po pretvorbi v AD pretvorniku shrani v 16-bitni register, od koder jo lahko preberemo preko I<sup>2</sup>C vodila.

MSB												LSB			
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	N/A <sup>1</sup>	N/A <sup>1</sup>	N/A <sup>1</sup>	N/A <sup>1</sup>

<sup>1</sup> N/A = not applicable.

Slika 11: Temperaturni register v ADT75 (Vir: Analog )

V registru je temperatura zapisana v dvojnem komplementu, pri čemer biti D0 do D3 niso uporabljeni in so vedno na logični ničli, bit D15 pa pomeni predznak vrednosti temperature. V tabeli, v nadaljevanju, je prikazanih nekaj primerov zapisa vrednosti temperature v 16-bitnem registru in pripadajoča vrednost temperature v °C.

**Table 6. 12-Bit Temperature Data Format**

Temperature	Digital Output (Binary) DB15 to DB4	Digital Output (Hex)
-55°C	1100 1001 0000	0xC90
-50°C	1100 1110 0000	0xCE0
-25°C	1110 0111 0000	0xE70
-0.0625°C	1111 1111 1111	0xFFF
0°C	0000 0000 0000	0x000
+0.0625°C	0000 0000 0001	0x001
+10°C	0000 1010 0000	0x0A0
+25°C	0001 1001 0000	0x190
+50°C	0011 0010 0000	0x320
+75°C	0100 1011 0000	0x4B0
+100°C	0110 0100 0000	0x640
+125°C	0111 1101 0000	0x7D0

### **Temperature Conversion Formulas**

Slika 12: Primeri izmerjenih temperatur (Vir: Analog )

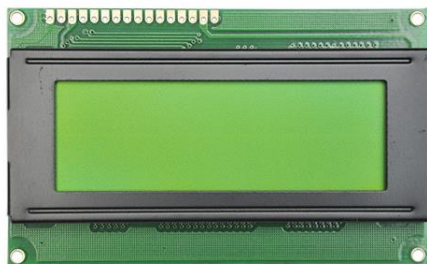
Pri pretvorbi prebranega temperaturnega registra v programu Arduino sem najprej preveril bit D15 – v primeru, da je D15 logična 1, je odčitana temperatura negativna. V tem primeru je za pravilni rezultat temperature potrebno izračunati še dvojni komplement prebranega temperaturnega registra (biti D4 do D15).

Biti D4 do D7 so decimalna vrednost temperature z natančnostjo 0,0625 °C.

Pomembnejši je še 8-bitni konfiguracijski register, v katerem se skonfigurirajo osnovni parametri delovanja vezja ADT75.

## 7. LCD in tipke

LCD (Liquid Crystal Display) ali zaslon s tekočimi kristali je elektronski prikazovalnik. Dvovrstični LCD zaslon s 16-imi znaki na vrstico je osnovni modul in se zelo pogosto uporablja v različnih napravah in vezjih. Ti moduli imajo kar nekaj prednosti: so ekonomični saj porabijo izredno majhne tokove, omogočajo enostavno programiranje, omogočajo tudi prikazovanje posebnih znakov ali celo znakov po meri.



Slika 13: LCD 4x20 (Vir: *embeddedadventures* )

Sam sem za digitalno uro uporabil nekoliko večji 20x4 LCD, kar pomeni, da lahko prikaže 20 znakov v 4-ih vrsticah. Na tem LCD-ju se prikaže vsak znak v 5x7 točkovni matrici. LCD ima dva registra - ukazni in podatkovni.

Ukazni register shranjuje navodila oziroma ukaze, ki jih LCD izvaja. Ukaz je navodilo za LCD, da naredi vnaprej določeno nalogo, kot na primer inicializacijo LCD-ja, brisanje (clear) zaslona, določitev položaja kurzorja, nadzor zaslona itd.

Register podatkov shranjuje podatke, ki se prikažejo na LCD zaslonu (vnaprej sprogramirani znaki).

## 8. RADIO

Za radio sem uporabil radijski FM modul s čipom TEA5767, ki ima, tako kot RTC modul in temperaturni senzor, vgrajeno I<sup>2</sup>C vodilo. Torej dva komunikacijska pina (SDA in SCL). TEA5767 čip je povsem integriran in vsebuje vse potrebne komponente za sprejem stereo FM signala. Iskanje postaj na UKV območju med 87,5 in 108,0 MHz temelji na konvencionalnem PLL (*phase lock loop*) konceptu. Poraba električne energije sprejemnika je zelo nizka. Porabi približno 13 mA, napajalna napetost pa se lahko spreminja med 2,5 in 5V.

Na izhodu ima radio modul zelo šibek signal, zato je potrebno ta signal za predvajanje na zvočniku ustrezno ojačati. Za ta namen sem izdelal ojačevalec nizke moči z znanim integriranim vezjem LM386. Le ta omogoča poslušanje MONO radia. Glasbo z ure pa je

možno poslušati tudi z zunanjo STEREO napravo, kot so na primer računalniški zvočniki z vgrajenim ojačevalnikom.

V programu Arduino sem sprogramiral upravljanje radia tako s tipkami, kot tudi z IR daljinskim oddajnikom. Možno je vnašanje želene frekvence radijskih postaj na našem UKV področju.



Slika 14: Radio modul (Vir: alicdn)

## 9. ALARM

Pri uri sem hotel imeti tudi bujenje, zato sem dodal še alarm. Za alarm sem uporabil brenčoč (buzzer), ki deluje na 5 V. Alarm sem sprogramiral na ta način, da sem določil, ko bodo ure in minute na uri in alarmu enake, se bo ta sprožil. Pri uri je mogoče nastaviti 2 alarma za določene dneve, ki sta med seboj povsem neodvisna. Parametri alarma (dan, ura in minuta) se vpišejo v EEPROM pomnilnik, tako da se ne izgubijo tudi, v kolikor ura ostane brez napajanja.

## 10. PROGRAM

Program sem v celoti napisal sam. Informacije na internetu so mi bile poleg mojega očeta glavno vodilo za pisanje programa. Program ne vsebuje zahtevnejših vsebin, zato ker je sprogramiran na enostaven način. S pisanjem programa sem začel z določanjem spremenljivk in vstavljanjem različnih knjižnic ob začetku programa. Nekatere knjižnice se že nahajajo znotraj Arduino programa, nekatere pa sem našel na spletu. Spodnja slika prikazuje primere definiranih knjižnic in dveh spremenljivk. V programu sem definiral okoli 20 spremenljivk.

```

#include "IRremote.h"
#include <Wire.h>
#include <RealTimeClockDS1307.h>
#include <LiquidCrystal.h>

float tempC;
int tempPin = A1;

```

Slika 15: Deklarirane knjižnice in spremenljivki (Vir: Avtor naloge)

Del programa, ki je zapisan znotraj »VOID SETUP« se bo izvedel le enkrat. Znotraj tega dela sem najprej deklariral hitrost serijske komunikacije z računalnikom. Nato sem določil LCD in v oklepaju zapisal število vrstic in stolpcev. Deklariral sem tudi zagon IR sprejemnika. Digitalni pini na Arduino ploščici, kot sem že omenil, se lahko uporabljajo kot vhodi in izhodi. Ker s tipko uro nastavljamo, jo je potrebno definirati kot vhod (INPUT). Kot primer uporabe analognega pina za digitalnega lahko vidimo na spodnji sliki, kjer preprosto napišemo vhod za A3. »pinMode (A3, INPUT);

```

void setup() ///////////////////////////////////////////////////
{
  Serial.begin(9600);
  lcd.begin(20,4);
  irrecv.enableIRIn(); //Zaženi IR

  pinMode(Tmode, INPUT); //TIPKA MODE
  pinMode(Tset, INPUT); //TIPKA SET
  pinMode(Tplus, INPUT); //TIPKA +
  pinMode(Tminus, INPUT); //TIPKA -

  pinMode(A3, INPUT);
  //digitalWrite(A3, HIGH);
  //pinMode(A2, OUTPUT);
  //digitalWrite(A2, LOW);
  pinMode(12, OUTPUT);
}

```

Slika 16: Program zapisan znotraj void loop(a) (Vir: Avtor naloge)

Del programa, ki bo zapisan znotraj »VOID LOOP« se bo ponavljal v nedogled. Večina programa (lahko rečem kar polovica programa) je zapisana s tako imenovanimi IF ELSE stavki. Mikrokrmilnik kot tak sam je popolnoma »ne pameten« in ne zna narediti ničesar, dokler mu mi ne vnesemo nekega ukaza kot npr.

»Če bo tipka pritisnjena, vklopi Svetlečo diodo, če ne bo, ne naredi ničesar«

Zgoraj opisana logika je zapisana na zelo enostaven in nam razumljiv način, kako se uporablja IF ELSE stavek. Spodaj si lahko ogledamo kako bi to izgledalo zapisano v programu, če bi žarnica bila priključena na pin 3.



```

void loop()
{
  if(Tipka == 1)
  {
    digitalWrite(3, HIGH); // Vklopi SVETLEČO DIODO
  }

  else
  {
    digitalWrite(3, LOW); // Izklopi SVETLEČO DIODO
  }
}

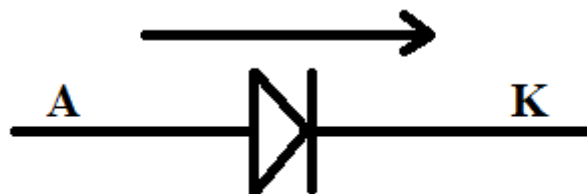
```

Slika 17: Primer zapisa kode (Vir: Avtor)

## 11. NAPAJANJE

Na začetku, ko sem programiral, sem Arduino ploščico in ostale komponente napajal z računalniškim USB portom 5 V. Čez čas, ko pa sem vezje opremil z ohišjem sem se odločil, da bom naredil ločen napajalni del vezja, tako da bom uro enostavno priključil na omrežno napetost 230 V.

Za napajanje sem uporabil transformator, ki zniža napetost iz 230 V na 9 V. Ker pa imamo še vedno izmeničnih 9 V, sem moral napetost usmeriti. Uporabil sem diodni mostič. Pri diodnem mostiču imamo mostično vezavo 4 diod. Vedeti moramo, da dioda prevaja samo v eno smer, tako kot kaže slika 18 spodaj.



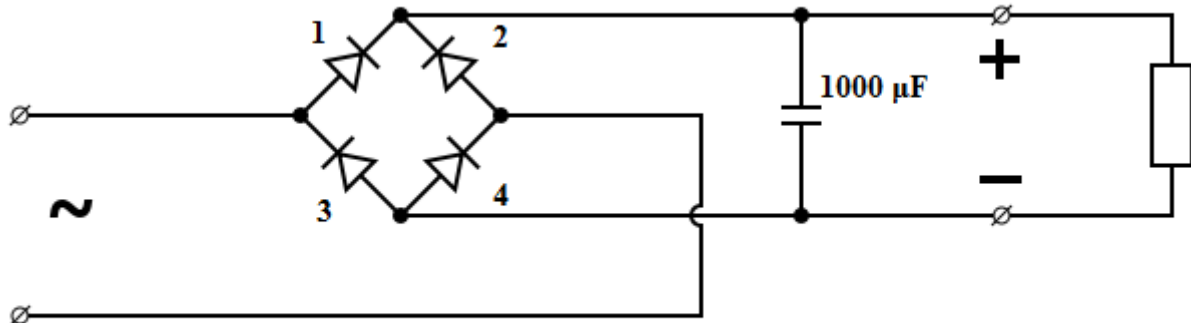
Slika 18: Dioda (Vir: Avtor naloge)

Delovanje diodnega mostiča:

Če + (PLUS) izmenične napetosti priteče po zgornjem vodniku do mostiča bo pozitivna napetost stekla skozi diodo 1 in ne skozi diodo 3 saj dioda prevaja le v smeri od anode do katode. Če plus izmenične napetosti priteče po spodnjem vodniku bo spet pozitivna napetost prevajala skozi diodo 2 in ne skozi diodo 4. Kar pomeni da smo plus usmerili na zgornji vodnik enosmerne napetosti.

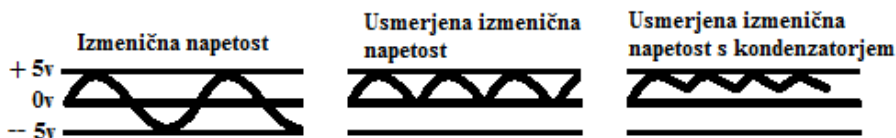
Če – (MINUS) izmenične napetosti priteče po zgornjem vodniku do mostiča bo negativna napetost stekla skozi diodo 3 in ne skozi diodo 1. Če minus izmenične napetosti priteče po

spodnjem vodniku bo spet negativna napetost prevajala skozi diodo 4 in ne skozi diodo 2. Kar pomeni da smo minus usmerili na spodnji vodnik enosmerne napetosti.



Slika 19: Shema napajalnega dela vezja( Vir: Wikipedia)

Ker pa napetost niha, v vezje dodamo še kondenzator.



Slika 20: Pogledi napetosti z osciloskopom (Vir: Avtor naloge)

Zdaj sem na izhodu dobil 9 V enosmerne napetosti, ki še ne zadoščajo napajanju Arduino ploščice. Napetost na Arduino ploščici preko USB-ja mora biti 5 V, zato sem uporabil 5 V regulator napetosti L7805. Na izhodno napetost sem še dodal manjši kondenzator za glajenje napetosti in preprečenje motenj.

## 12. DELOVANJE

Ura ima 5 različnih mode(ov) in pet različnih tipk za upravljanje oz. nastavljanje.

### 12.1 TIPKE

S prvo tipko »MODE« spreminjamo različne mode(e) od 1 do 5.

S tipko »SET« preklapljammo možnosti nastavljanja npr. med urami, minutami in sekundami pri času.

S tipko »PLUS« povečujemo neke faktorje in z tipko »MINUS« zmanjšujemo neke faktorje.

S tipko »STOP« ustavimo delovanje sproženega alarma.

## **12.2 UPRAVLJANJE URE**

Kot osnovni in hkrati prvi mode nam prikazuje čas, datum, dan v tednu, temperaturo in če sta alarm 1 in 2 vklopljena, kdaj bosta sprožena.

V drugem mode(u) nastavljammo čas (ure, minute in sekunde).

V tretjem mode(u) nastavljammo datum (dneve, mesece in leto).

V četrtem mode(u) nastavljammo alarm 1 (ure, minute in kateri dneve v tednu bo sprožen).

V petem mode(u) nastavljammo alarm 2 (ure, minute in katere dneve v tednu bo sprožen).

### 13. ZAKJUČEK

Pri svoji raziskovalni nalogi sem spoznal in se naučil veliko novega. Kot prvo sem se naučil iz morja informacij poiskati tisto, kar sem za raziskovalno nalogo potreboval. Začel sem tudi uporabljati tako imenovan Data sheet, v katerem so zapisani vsi tehnični podatki o neki komponenti. V Arduino programu sem se naučil osnovnega programiranja, kar je bil eden od mojih ciljev. V okviru te raziskovalne naloge sem se naučil tudi foto postopka izdelave tiskanega vezja. Do sedaj sem namreč tiskana vezja izdeloval s CNC rezkalnikom. Pri raziskovalni nalogi sem se spoznal med drugim tudi Arduino UNO ploščico, RTC modul, LCD in radio modul.

Vsa izdelava ure od ohišja do vezja je potekala nemoteno, pri programiranju pa sem naletel na manjši problem. Ko sem že imel okrog 600 vrstic programa sem moral še sprogramirati IR komunikacijo, za nastavljanje ure in radia. Tukaj sem naletel na problem. Problem je bil, da mi ni program zaznal IR signala od oddajnika in sem moral pritisniti večkrat eno tipko da je program naredil naslednji korak. Čez čas sem program popolnoma preuredil in to naredil na podoben način kot to:

»Če mi hočemo priti na neko mesto v določeni državi moramo najprej priti na mejo te države, da nas spustijo vanjo. Če mi nimamo ne karte ali navigacije moramo na meji vprašat v kateri smeri moramo peljati, da pridemo do te točke in slediti tablam na katerih je zapisan ta kraj.«

Na takšen način sem program preuredil in deloval je boljše in zanesljivo.

Ko sem končal uro, sem bil vesel svojega produkta, saj mi bo ura služila za bujenje zjutraj, ko bom hodil v šolo in mi sporočala čas ter temperaturo, ob tem pa bom lahko poslušal glasbo preko radia.

V uro sem vložil ogromno časa za izdelavo, testiranje in raziskovanje. Marsikdo si ne zna predstavljati, koliko vložnega časa in dela potrebujemo za izdelavo preproste ure in radia, ki si ju v trgovini lahko kupimo po izredno nizki ceni. Izdelava ure je bil dober projekt, pri katerem sem pridobil veliko izkušen glede elektrotehnike in programiranja. Zahvaljujem se vsem, ki so mi bili v pomoč pri projektu, predvsem mojemu očetu in mentorju!

## 14.VIRI

[https://sl.wikipedia.org/wiki/Ura\\_\(naprava\)](https://sl.wikipedia.org/wiki/Ura_(naprava))

<http://www.tocnaura.si/zgodovina-ure-tocen-cas>

<http://www.minutka.si/zgodovina-ur.html>

[http://www.seeedstudio.com/wiki/Arduino\\_UNO#Schematic](http://www.seeedstudio.com/wiki/Arduino_UNO#Schematic)

<http://datasheets.maximintegrated.com/en/ds/DS1307.pdf>

<http://www.engineersgarage.com/electronic-components/16x2-lcd-module-datasheet>

<https://www.arduino.cc/>

<https://arduino-info.wikispaces.com/IR-RemoteControl>

[http://www.quickwiki.com/sl/Amplitudna\\_modulacija](http://www.quickwiki.com/sl/Amplitudna_modulacija)

Knjižnica za RTC: <http://www.instructables.com/id/Arduino-Real-Time-Clock-DS1307/step4/Checking-and-Setting-The-Module/>

### 14.1 VIRI SLIK

(23.12.2016) Skyscanatomicclock: <http://skyscanatomicclock.zaleplon.info/wp-content/uploads/2011/06/Time-Keeping-Obelisk.jpg>

(24.12.2016) Beforeitsnews: <http://beforeitsnews.com/mediadrop/uploads/2015/43/c522afcd8f5803be34601113a104f32be54bc95f.png>

(27.12.2016) Iconfinder: <https://cdn3.iconfinder.com/data/icons/classic-icons-1/512/13.png>

(27.12.2016) Webshopapp: <http://static.webshopapp.com/shops/019462/files/014937389/ir-reciever-vs1838b.jpg>

(27.12.2016) Embeddedadventures: [http://www.embeddedadventures.com/images/uploaded\\_images/LCD-2004\\_2\\_600.jpg](http://www.embeddedadventures.com/images/uploaded_images/LCD-2004_2_600.jpg)

(29.12.2016) Tuxgraphics: <http://tuxgraphics.org/~guido/molnija-pocket-watch/rolex-1940.jpg>

(29.12.2016) Arduino: [https://www.arduino.cc/en/uploads/Main/ArduinoUno\\_R3\\_Front\\_450px.jpg](https://www.arduino.cc/en/uploads/Main/ArduinoUno_R3_Front_450px.jpg)

(1.1.2016) Atomsindustries: <http://www.atomsindustries.com/assets/images/items/1078/1078.jpg>

(2.1.2016)Conrad:

[http://media.conrad.com/medias/global/ce/5000\\_5999/5800/5840/5846/640712\\_RB\\_00\\_FB.EPS\\_250.jpg](http://media.conrad.com/medias/global/ce/5000_5999/5800/5840/5846/640712_RB_00_FB.EPS_250.jpg)

(2.1.2016)Wikipedija:

[https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Diode\\_bridge\\_smoothing.svg/600px-Diode\\_bridge\\_smoothing.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/1/1d/Diode_bridge_smoothing.svg/600px-Diode_bridge_smoothing.svg.png)

(2.1.2016) Wikipedia: [https://sl.wikipedia.org/wiki/Frekven%C4%8Dna\\_modulacija](https://sl.wikipedia.org/wiki/Frekven%C4%8Dna_modulacija)

(3.1.2016)Alicdn:

<http://g01.a.alicdn.com/kf/HTB1t7LIHVXXXXcPXFXXq6xXFXX2/Free-shipping-10pcs-lot-TEA5767-Programmable-Low-power-font-b-FM-b-font-font-b-Module.jpg>

(10.1.2016)Analog:<http://www.analog.com/media/en/technical-documentation/data-sheets/ADT75.pdf>