Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*

Projekt:

Arduino Timer

Avtor: Jakob Baumgartner

Asistent: Jernej Olenšek

Profesor: Tadej Tuma

Modul B,

Fakulteta za elektrotehniko,

Univerza v Ljubljani

2019/2020

CONTENTS

UVOD

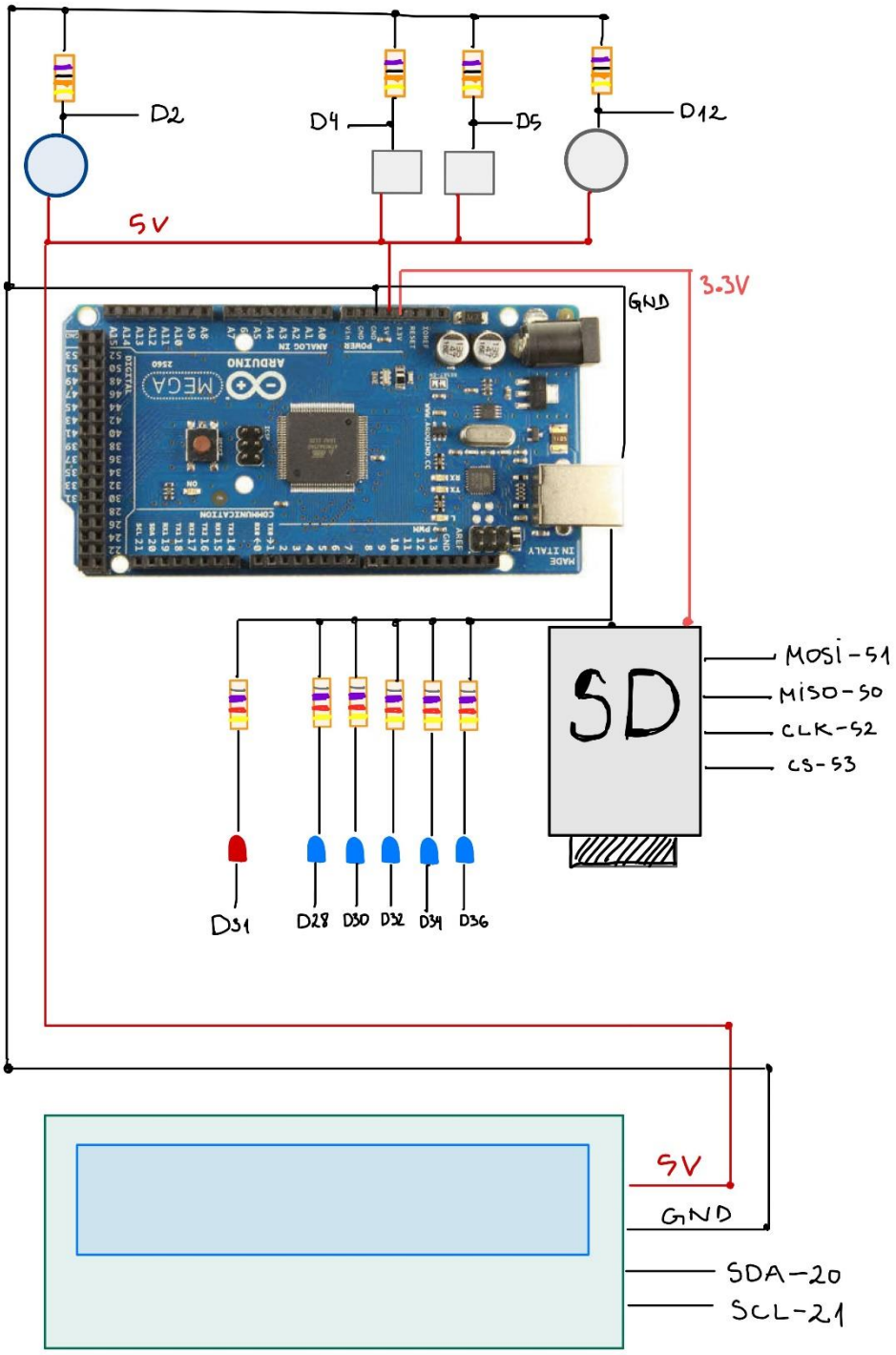
Projekt je bil narejen kot obvezni del laboratorijskih vaj pri predmetih Načrtovanje vgrajenih sistemov in Programiranje vgrajenih sistemov, ki skupaj sestavljata Modul B v tretjem letniku univerzitetnega študija na Fakulteti za Elektrotehniko, UNI LJ.

Moj cilj je bil izvesti fizično različico merilca časa, ki ga uporabljam pri študiju, da vidim realno količino dela, ki sem ga naredil, občutek nas namreč pogosto vara. Pomembno pri dizajnu mi je bilo tudi, da se podatki shranjujejo na SD kartico, od koder jih lahko potem prenesem na računalnik in naprej na platformo Toggl, ki jo običajno uporabljam za spremljanje časa. Dodano vrednost pa sem naredil z LED diodami, ki mi s svetlobo pokažejo pretekel čas dela / študija. Prižigajo se namreč v časovnih intervalih (npr. na 9min ena).

ZMOGLJIVOSTI:

- Merjenje časa za različne projekte,
- Prikaz merjenega časa z LED diodami,
- Prikaz na 2x16 znakovnem zaslonu,
- Prikaz statistike vsega časa za posamezni projekt,
- Večopravilnost (prikaz statistike med samim merjenjem),
- Implementiran daljši potrebni čas za potrditev začetka in konca merjenja (register),
- Shranjevanje naslednjega prostega ID za sejo na SD,
- Sprotno shranjevanje (vsako minuto) časa na SD,
- Shranjevanje Statistike vsega merjenega časa za posamezni projekt,
- Rdeča LED dioda, ki sporoči kdaj se izvajajo operacije na SD kartici,
- Utripanje LED diode, ko pride do napak pri delovanju SD kartice,
- Preprost zapis shranjevanja omogoča enostavno nadaljno obdelavo na PC.

SHEMA VEZAVE



SEZNAM DELOV

- ARDUINO MEGA 2560,
- SD čitalec kartic LC Technology (priporočen drug del),
- SanDisk microSD 2GB + PLATINET microSD adapter,
- Serial LCD Blue Screen 1602A HD44780,
- 4F57 I2C TWI 1602 kontroler,
- 4x Tactile Push Buttons 12x12x7.3mm,
    - 1x modro okroglo pokrivalo,
    - 1x belo okroglo pokrivalo,
    - 2x belo kvadratno pokrivalo,
- 6x 4.7 kΩ upornik,
- 4x 10kΩ upornik,
- navadne žice,
- 1x USB TypeB to USB TypeA kabel,
- Proto-board.

MENI STRUKTURA

MODE = 1 (RECORDING MODE)

MENU = 0 (COUNTER MODE)

(**START BUTTON** = START / STOP RECORDING)

(**PREVIOUS / NEXT BUTTON** = CHANGE PROJECT)

MODE = 2 (STATISTICS MODE)

(**PREVIOUS / NEXT BUTTON** = CHANGE PROJECT)

OPIS STRUKTURE PROGRAMA

Main

Main program je sestavljen iz vključenih knjižic, prototipov funkcij, inicializacije globalnih spremenljivk, ter funkcij setup in loop.

GLOBALNE SPREMENLJIVKE SO RAZDELJENE NA SKUPINE:

- DEMONSTRATION,
- MENU,
- SCREEN,
- PROJECTS,
- RECORDING,
- STATISTICS,
- SD,
- LEDS.

SETUP()

Prvo inicializiramo Serijski port, za uporabo pri debug-iranju. Nato inicializiramo display.

Iz SD kartice nato preberemo seznam projektov, podatke o statistikah projektov, ter ID, ki ga potrebujemo za zaporedno shranjevanje sej.

Vključimo še vhodne in izhodne pine.

LOOP()

Loop funkcija se izvaja zaporedoma v neskončni zanki.

Prvo pogledamo, če je pritisnjen kak gumb. Nato v primeru, da se izvaja snemanje posodobimo shranjene minute, če je potrebno dodamo zapis na SD in prižgemo kako od LED diod.

Sedaj pogledamo če je bil pritisnjen kak od gumbov in v primeru, da je bil izvedemo klic na ustrezno funkcijo, v udvisnosti od tega v katerem meniju se nahajamo.

Na koncu še posodobimo izpis na LCD zaslonu, če se je spremenil zapis v eni od dveh String spremenljivk za vrstici.

Čisto na zadnje pa še preverimo, če se je pojavila kaka napaka na SD kartici in v primeru, da napaka je utripamo z rdečo LED diodo.

DATA

Definicije vseh funkcij, ki berejo ali pišejo na SD kartico so v tej datoteki.

STRING PROJECTSREAD()

Funkcija prebere stanje v datoteki »projects.txt« na SD kartici. V primeru, da je bila poizvedba uspešna vrne String z vsebino datoteke, v nasprotnem primeru pa javi napako in prižge utripanje rdeče LED.

Med izvajanjem operacij na SD kartici sveti rdeča LED dioda.

INT GETNEXTID()

Funkcija prebere vsebino datoteke »pid.txt« in vrne ID, ki ga uporabimo za naslednje merjenje časa.

Prav tako poveča število shranjeno na SD kartici za ena.

Med izvajanjem operacij na SD kartici sveti rdeča LED dioda.

V primeru, da je bila poizvedba uspešna vrne int z PID, v nasprotnem primeru pa javi napako in prižge utripanje rdeče LED.

STRING *PROJECTSLIST(STRING PROJEKTI)

Funkcija kot parameter sprejme String, ki ga dobimo iz funkcije ProjectsRead in vrne (kazalec) do Array of Strings, imen projektov.

INT NUMOFPROJECTS(STRING PROJEKTI)

Funkcija kot parameter sprejme String, ki ga dobimo iz funkcije ProjectsRead in vrne (kazalec) int števila projektov, ki jih imamo na seznamu.

VOID SAVESESSIONSTATUS()

Funkcija shrani zapis o trenutnem merjenju časa v datoteko »sessions.txt«, v novo vrstico, v sledečem formatu:

PID?PROJECTNAME;PROJECTID$TIMEINSECONDS#

Med izvajanjem operacij na SD kartici sveti rdeča LED dioda.

V primeru SD napake to sporoči in prižge rdečo LED diodo.

STRING STATISTICS(BOOL SAVE = FALSE)

Funkcija prebere statistiko časov na vseh projektih iz datoteke »stats.txt« in vrne String z vsemi časi.

1;175#

2;16#

3;88#

4;300#

Default parameter za save je enak false, v primeru, da ga spremenimo na true, bo tudi povečala statistiko minut trenutno merjenega projekta za eno minuto.

Med izvajanjem operacij na SD kartici sveti rdeča LED dioda.

V primeru SD napake to sporoči in prižge rdečo LED diodo.

INPUT

Preko input funkcij se izvaja vse zaznavanje pritiskov gumbov.

INT BUTTONPRESSED()

Funkcija preveri, če je vrednost na katerem od pinov na vhodu pozitivna. V primeru, da je bo vrnila int številke gumba. V nasprotnem primeru bo vrnila 0.

Če je pritisnjenih več gumbov, bo vrnjena številka gumba v naslednjem vrstnem redu: 12 > 5 > 4 > 3 > 2 .

Po pritisku gumba se izvede manjši delay, ki ni opazen v delovanju programa, vendar pa prepreči »odbijanje gumbov.«

Pri gumbu številka 12 je nekoliko drugače. Implementiran je register z 100 polji, katerih vsota mora biti >90, za zaznan pritisk gumba. To pomeni, da moramo gumb držati dovolj časa, brez, oz. z minimalnimi prekinitvami, da je zaznan pritisk. Gumb številka 12 je namreč namenjen začetku in koncu merjenja časa in tako ne želimo pritiskov po pomoti. Med držanjem gumba se na zaslonu pojavi odštevalnik, ki nam pove koliko časa moramo gumb držati in v primeru uspešnega pritiska dobimo izpis na zaslonu:

REC. STARTING ...

Oz.

REC. STOPPING ...

LEDS

Preko LED funkcij se izvaja upravljanje modih LED diod in rdeče LED diode.

VOID BLUELEDSTIME (BOOL DEMO = FALSE)

Funkcija skrbi za vklop modrih LED diod v določenem časovnem intervalu (običajno ena dioda na 9 min oziroma na 12 min). V primeru, da je parameter demo = true, se diode prižigajo na 10 sekund. To je namenjeno demonstracijskim namenom, ko smo časovno omejeni.

VOID BLUELEDSOFF ()

Funkcija izklopi vse modre diode.

VOID SDERROR()

Funkcija poskrbi za utripanje rdeče diode v primeru napake.

RECORDING

VOID RECORDINGMODE ()

Funkcija poskrbi za navigacijo, ko smo v načinu (mode) recording; za menjanje projektov, za pričetek snemanja in potrebne operacije za začetek snemanja. Za konec snemanja in potrebne operacije za izvedbo konca snemanja. Poskrbi tudi za posodabljanje izpisa na zaslonu.

VOID CHANGEPROJECT(INT CHANGEDIRECTION)

Ta precej enostavna funkcija samo zamenja indeks trenutnega projekta, na katerem se nahajamo. V primeru, da smo na končnih indeksih (0 ali numberOfProjects-1), ne naredi nič.

VOID RECORDINGUPDATE ()

Ta funkcija je klicana na vsakem krogu programa, da poveča čas, ki ga merimo. Uporablja millis(), da lahko teče ne glede na izvajanje druge kode.

Ure, minute in sekunde shranjuje v spremenljivko unsigned long recordedTime[3]. Poveča tudi števec lastSavedMIN, ko se čas poveča za minuto.

V primeru, da se snemanje projekta izvaja več kot 6 ur (600min), se snemanje prekine in na zaslonu se izpiše opozorilo. To je namenjeno situacijam, ko pozabimo ustaviti časovnik.

STATISTICS

VOID STATSMODE()

Funkcija se izvaja za navigacijo po mode = statistics. V tem primeru se na zaslonu v zgornji vrstici izpisuje:

STATS: PROJECTNAME

V spodnji vrstici pa:

h ## min

ZUNANJE KOMPONENTE

LED DIODE

Prvo nastavimo pine:

```
pinMode(#, OUTPUT);
```

Nato pišemo na pine z:

```
digitalWrite(#, HIGH);
```

Led diode so povezane med digitalne pine in ground, vmes pa je vezan še 4.7kΩ upor, ki skrbi za omejitev toka v diodo.

GUMBI

Prvo nastavimo pine:

```
pinMode(#, INPUT);
```

Nato beremo z:

```
digitalRead(#);
```

Gumbi so priklopljeni na pullDown način, kar pomeni, da je 10kΩ register vezan tako, da v primeru izklopljenega gumba pin poveže na GND, v primeru staknjenega stikala pa se pin poveže na 5V.

SD BRALNIK

SD bralnik je generične vrste, poleg tega je zelo nezanesljiv. Deluje le z nekaterimi karticami in pogosto povzroča napake.

Za komunikacijo z njim uporabljam knjižico SD, ki je že vključena v Arduino okolje, ter ukaze, ki jih knjižica vsebuje. Naprava za komunikacijo z Arduinotom uporablja SPI protokol.

Serial Peripheral Interface (SPI) je sinhron protokol, ki je namenjen komunikaciji mikrokontrolerjev z zunanjimi napravami (ali med dvema mikrokontrolerjema) preko krajših razdalij.

Na pine SPI lahko priklopimo na vodnike več naprav, deluje po principu MASTER-SLAVE, vsaka pa je izbrana preko SS (Slave-Select), ko je v uporabi, SS mora biti za vsako napravo specifičen.

Povezan je na pine:

MOSI - pin 51

MISO - pin 50

CLK - pin 52

CS - pin 53

Napajamo ga lahko z 3.3V ali z 5V. Izbral sem 3.3V.

ZASLON

Zaslon je zelo pogosto uporabljen dvovrstični in 16 znakov dolg HD44780, ki pa je na mikrokontroler povezan preko vmesnika, oz. kontrolerja, ki nam omogoča I2C komunikacijo z kontrolerjem. Prednost tega, ki je pri tem projektu niti nisem potreboval je, da imamo namesto velike količine potrebnih komunikacijskih pinov, potrebo le po dveh.

Za samo komunikacijo z njim nisem uporabljal knjižice, ki bi jo vseboval Arduino program, pač pa sem dodal knjižico, ki jo dobimo na naslovu: https://github.com/johnrickman/LiquidCrystal_I2C.

I2C je namenjen komunikaciji med komponentami, ki so običajno na istem vezju. Potrebne sta le dve liniji, komunikacija je vedno master-slave, vsaka naprava na povezavi ima svoj naslov.

Z zaslonom sem imel precejšen problem, kajti v specifikacijah so omenjeni drugi naslovi, na katerih se naj bi zaslon dejansko uporabljal, kot je naslov, na katerega je narejen ta zaslon. To sem rešil s pomočjo wire library, ki med drugim omogoča, da zaznamo vse naprave na I2C povezavi.

https://github.com/johnrickman/LiquidCrystal_I2C

Povezan je na pine:

SDA - pin 20

SCL - pin 21

Napajamo ga z 5V.