

UNIVERZITET U BEOGRADU  
ELEKTROTEHNIČKI FAKULTET

Milica Vukajlović, 3015/2012

## Touchscreen kalkulator

*projekat iz predmeta 32-bitni mikrokontroleri i primjena*

mentor:  
doc. dr Dragan Vasiljević

Beograd, februar 2013.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>2</b>
1.1	Projektni zadatak . . . . .	2
<b>2</b>	<b>Opis sistema</b>	<b>4</b>
<b>3</b>	<b>Algoritam glavnog programa</b>	<b>5</b>
<b>4</b>	<b>Opis aplikacija</b>	<b>8</b>
4.1	Kalkulator . . . . .	8
4.2	Igrica memorija . . . . .	8
4.3	Igrica uhvati balon . . . . .	9
<b>5</b>	<b>Zaključak</b>	<b>10</b>

# Glava 1

## Uvod

*Touchscreen*, ekrani osjetljivi na dodir, zauzimaju veliki dio tržišta elektronskih uređaja. Prvi put se pojavljuju van laboratorija 1972. godine. Danas sve više uređaja koristi *touchscreen* displeje, telefoni, automobili, konzole za igranje, fabrike...

*Touchscreen* ekrani se sastoje od displeja i *touch panel*-a (površina osjetljiva na dodir). Na ekranu je prikazan skup funkcija koje ekran može "pokrenuti" na sistemu kojim upravlja. Ekran koji reaguje na dodir ukidaju potrebu korišćenja posebnog uređaja za unos podataka u sistemima u kojima se koriste. Oni objedinjuju funkcionalnost ulaznih i izlaznih uređaja, služe za prikaz rezultata, a istovremeno omogućavaju i unos podataka i manipulaciju njima. Danas postoje različite tehnološke implementacije *touch panel*-a, a najčešće korišćene su :

- otpornička tehnologija sa četiri žice,
- otpornička tehnologija sa pet žica,
- kapacitivna tehnologija,
- tehnologija površinskih talasa.

Pri izradi projekta korišćen je otpornički *touch panel* sa četiri žice. Sastoji se iz vanjskog staklenog dijela koji je presvučen otpornim (rezistivnim) slojem, a takav isti sloj se koristi i u unutrašnjosti. Vanjski otporni panela pri dodiru prsta sa ekranom dodiruje unutrašnji otporni sloj mijenjajući napon na pritisnutom dijelu. Očitavanjem napona određuje se tačka pritiska ekrana. Tehnologija sa četiri žice je najjeftinija tehnologija, osjetljiva na dodir bilo kojim objektom (ne samo prstom), a nije osjetljiva na vlagu, prašinu, svjetlost.. Nedostaci su lošija vidljivost i mogućnost oštećenja oštrim predmetima.

Projekat je rađen u IAR razvojnom okruženju, verzija 6.4. Kod je pisan u C-u. Prilikom pisanja koda korišćene su CMSIS biblioteke.

Prvobitna zamisao autora je bila da se napravi samo *touchscreen* kalkulator, ali je posle projekat proširen dodavanjem dvije jednostavne igrice. U nastavku će biti dat opis predloženog rješenja, opis hardvera, softvera i podešavanja mikrokontrolera.

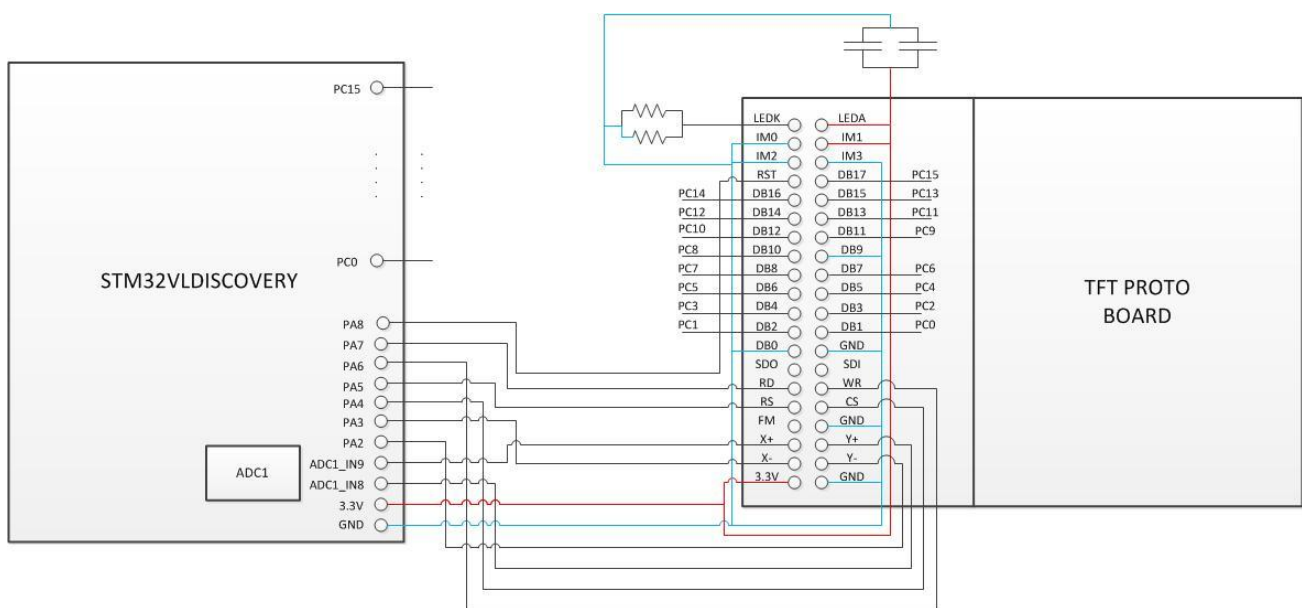
### 1.1 Projektni zadatak

Povezati STM32VLDISCOVERY razvojno okruženje i *TFT Proto Board* i realizovati *touchscreen* kalkulator. Kalkulator u realnom vremenu treba na svom ekranu da ispisuje pritisnute cifre, operacije i daje rezultat.

# Glava 2

## Opis sistema

Na slici 2.1 je prikazana električna šema sistema.



Slika 2.1: *Električna šema sistema*

*TFT Proto Board* je evalucionna ploča koja služi za prikaz grafičkih i tekstualnih sadržaja. Ima mogućnost povezivanja sa procesorom preko *SPI* ili paralelne magistrale. Potreban napon napajanja je 3.3V, ima rezoluciju 320x240 piksela. *TFT*-om upravlja *HX8347 – D* kontroler. On omogućava grafički i tekstualni prikaz slika, vertikalno skrolovanje, parcijalno prikazivanje ekrana. Kontroler ima 32KB SRAM memorije koja služi za čuvanje teksta, karaktera, i bit mapiranih crteža.

Korišćena evalucionna ploča je STM32VLDISCOVERY, na kojoj se nalazi mikrokontroler STM32F100RB, baziran na ARM Cortex-M3 jezgru. Napaja se preko USB-a, a ima i pinove za eksterno napajanje 3.3V i 5V. Mikrokontroler se debuguje i programira pomoću *ST-Link debugger*-a. Ovo je familija mikrokontrolera visokih performansi, 32-bitni RISC procesor sa maksimalnom frekvencijom od 24MHz. Ima tri *USART* periferije, jedan 12-bitni AD konvertor, dva 12-bitna DA konvertora, šest šestobitnih tajmera opšte namjene i jedan tajmer sa naprednim funkcijama. Pri izradi projekta korišćena su dva kanala AD konvertora za očitavanje napona sa displeja i *GPIO* pinovi za slanje komandi i podataka displeju. AD konvertor radi u *single channel* modu. *SysTick* tajmer služi da se podesi vrijeme zadržavanja otkrivene slike na ekranu u igri memorija.

# Glava 3

## Algoritam glavnog programa

Glavni program na početku poziva funkciju za inicijalizaciju korišćenih periferija. Prvo se resetuju sve korišćene periferije. Inicijalizuju se GPIO pinovi kao izlazni GPIO\_Output\_PP, za prenos podataka ka displeju. Brzina na svim pinovima je podešena na maksimalnu brzinu od 50Hz. Zatim se inicijalizuje AD konvertor.

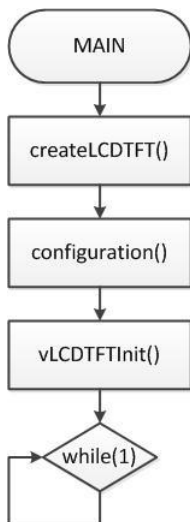
Brzina AD konvertora se podesi na 12MHz. Koristi se nezavistan mod rada, bez eksternih trigeru. ADC\_DataAlign\_Right smješta rezultat konverzije u nižih 12 bita. Start konverzije se poziva iz glavnog programa, eksterni trigeri su isključeni. Koriste se ADC\_Channel\_8 i ADC\_Channel\_9, pinovi 0 i 1 GPIOB porta. Oni se inicijalizuju kao GPIO\_Mode\_AIN. SysTick tajmer se podesi da daje prekid na 1ms.

```
RCC_ADCCLKConfig(RCC_PCLK2_Div6);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1, ENABLE);
ADC_DeInit(ADC1);
ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
ADC_InitStructure.ADC_ScanConvMode = DISABLE;
ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
ADC_InitStructure.ADC_NbrOfChannel = 1;
ADC_Init(ADC1, &ADC_InitStructure);
/* Enable ADC1 */ ADC_Cmd(ADC1, ENABLE);
ADC_ResetCalibration(ADC1);
while (ADC_GetResetCalibrationStatus(ADC1));
ADC_StartCalibration(ADC1);
while (ADC_GetCalibrationStatus(ADC1));

GPIO_StructInit(&GPIO_InitStructure);
GPIO_InitStructure.GPIO_Pin = Yp_PIN | Xp_PIN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init( GPIOB, &GPIO_InitStructure );

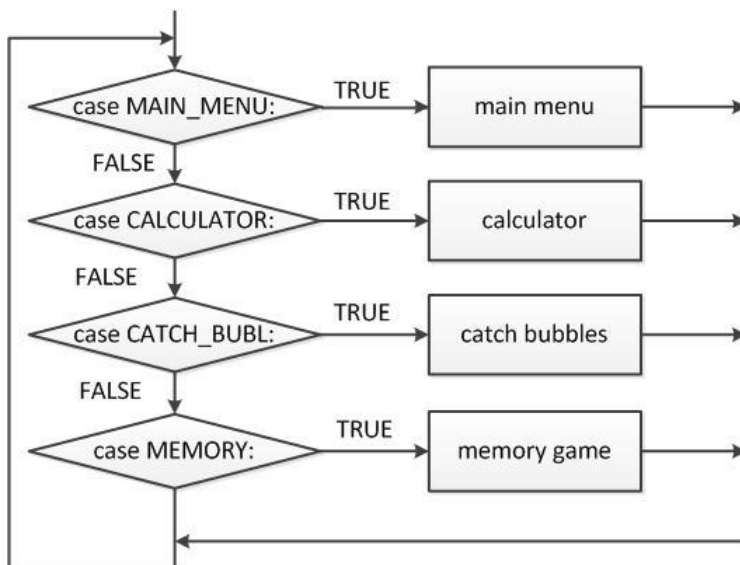
/* Setup SysTick Timer for 1 msec interrupts */
if (SysTick_Config(SystemCoreClock / 1000))
{
    while (1);
}
```

Nakon inicijalizacije periferija poziva se funkcija za inicijalizaciju displeja. Displej se inicijalizuje popunjavanjem registara. Sa displejem se komunicira preko funkcija koje su definisane u *LCDTFT.h* fajlu. Napisane su funkcije za ispis teksta na displej, crtanje kruga, pravougaonika, linije, za bojenje čitavog ekrana, i ispis slike formata .bmp. Pritisak detektujemo pozivanjem funkcije `get_touch()`.



Slika 3.1: *Algoritam main programa 1*

Funkcija `get_touch()` startuje AD konverziju. Prvo postavi X+ pin da bude izlazni i njegovu vrijednost na 1, a Y+ pin da bude ulazni i očita y koordinatu. Zatim postavlja Y+ pin da bude izlazni i njegovu vrijednost na 1, a X+ da bude ulazni i očita x koordinatu. Funkcija vraća podatak koji je tipa *point* (x, y) koordinate. Nakon toga se poziva funkcija koja provjerava da li je pritisak validan, tj. da li je napon iznad praga. Prag za detekciju pritiska se dobija u funkciji `calibrate()`, koja i kalibriše ekran. Kalibracijom odredimo minimalni i maksimalni napon koji dobijamo od displeja, pritiskanjem dva dijagonalna ćoška displeja. Jednostavnom funkcijom se napon konvertuje u tačku, pozivanjem funkcije `to_pixel()`.



Slika 3.2: *Algoritam main programa 2*

Funkcija `createLCDTFT()` kreira strukturu LCD, u kojoj se čuvaju podešavanja displeja. Nakon inicijalizacije program ulazi u beskonačnu petlju. Na slici 3.1 je prikazan algoritam rada

glavnog programa do ulaska u beskonačnu *while* petlju. Nakon što uđe u beskonačnu petlju, na displeju se ponude opcije za dostupne aplikacije, tj. glavni meni. Pritiskom odgovarajućeg ikonice ulazi se u željenu aplikaciju. Svaka aplikacija ima taster *back*, za povratak u glavni meni. Dostupne aplikacije su: kalkulator, *catch\_bubbles* i *memory*. Na slici 3.2 je prikazan algoritam unutar beskonačne petlje, *switchcase* struktura.

# Glava 4

## Opis aplikacija

### 4.1 Kalkulator

Izborom ikonice za kalkulator iz glavnog menija, na ekranu se prikazuje digitron. Prvobitna ideja autora je bila da se za svaku tipku digitrona učitaju po dvije slike .bmp, da bi se dobila aplikacija koja mijenja boju pritisnute tipke. Čitanje slike je mnogo sporo, tako da ne može da se postigne dovoljna brzina, jer korisnik vidi ispis slike liniju po liniju. Zbog toga su napravljene bitmape slika, u RBG formatu 565 i one se nalaze u *calculator\_resources.c* fajlu. Slike se čuvaju kao nizovi, ali i oni zauzimaju mnogo prostora. Zbog toga je učitana za svaku tipku samo po jedna slika i pritiskom tipke ne mijenja se boja. Prikaz digitrona se poziva funkcijom `draw_calculator()`.

S obzirom da je ispis na displej dovoljno brz, da korisnik ne vidi proces ispisivanja slike, kada se ona se učitava iz fajla napravljen je i digitron kod kojeg se tipke crtaju, a ne učitavaju iz slika. Time je dobijena mogućnost promijene boje pritisnute tipke. Ovaj digitron se prikazuje pozivanjem funkcije `draw_calculator2()`. Mijenjanje digitrona se vrši kada smo u toj aplikaciji pritiskom na tipku `set`.

Digitron od operacija podržava sabiranje, oduzimanje, množenje i dijeljenje. Radi i sa decimalnim brojevima. Preciznost digitrona je 6 cifara, rezultate veće od 999999 prikazuje sa eksponentom, tj. zaokružuje. To je zbog korišćene konverzije za ispis *floating point* brojeva `%g`. Takođe ima mogućnosti brisanja po unijetih cifri i brisanja memorije digitrona.

Program ostaje unutar ove aplikacije sve dok korisnik ne pritisne ikonicu *back*, koja ga vraća na glavni meni. Program stalno provjerava da li je displej pritisnut. Ako jeste, u glavnom programu odredi koja je tipka pritisnuta, i poziva funkciju za obradu `button_pressed()`. Iz te funkcije se vrši ispis na broja na displej, ili se poziva funkcija za računanje ako je pritisnuta neka od operacija.

### 4.2 Igrica memorija

Izborom ikonice igra memorija iz glavnog menija, bira se igra memorije. Igra je napravljena tako da se na ekranu pojave balončići sa brojevima, nakon 1s se zatamne balončići tako da se brojevi više ne vide. Korisnik treba da otkriva balončiće jedan po jedan, otkrivajući brojeve od najmanjeg do najvećeg. Ako uspješno otkrije sve balončiće prelazi se na sledeći nivo, a ako pogriši vraća se na prethodni nivo. Svaki sledeći nivo ima jedan balončić više. Igra se završava



nakon tri greške ili ako korisnik dođe do poslednjeg nivoa.

Funkcije za realizovanje igrice koriste *random* generator, tako da se balončići prikazuju nasumice po ekranu, a mogući brojevi unutar balončića su od 0..19, i one su definisane u fajlu *games.c*. Pritiskom na ikonicu *back*, korisnik se vraća u glavni meni.

### 4.3 Igrica uhvati balon

Izborom ikonice uhvati balon iz glavnog menija, ulazi se u ovu aplikaciju. Na ekranu se nasumično pojavljuju balončići raznih boja koje treba pritisnuti, prije nego istekne vrijeme. Svaki promašeni balončić se kažnjava jednim negativnim poenom, a pogođeni jednim poenom. Nakon 10 uhvaćenih balona vrijeme za koje korisnik treba da uhvati balon se smanjuje. I ovdje je korišćen *random* generator za određivanje mjesta prikaza sledećeg balona.

# Glava 5

## Zaključak

Realizovan je jednostavan sistem. Uspješno je povezan hardver i uspostavljena je komunikacija između mikrokontrolera i displeja. Autor je testirao sve tri aplikacije i nije primjetio greške u realizaciji. Izrada ovog projekta bazirana na procesorskom jezgru Cortex M3, vrlo je značajna zbog upoznavanja studenta sa načinom rada kontrolera, i njegovom primjenom u namjenskom sistemu.

Nadogradnja sistema je moguća dodavanjem novih aplikacija displeju. Dovođenjem eksternog napajanja na Vbat pin, moguće je osposobiti *real time clock*, i time bi dobili mogućnost dodavanja sata na displeju. Takođe, povezivanjem memorijske kartice mogu da se dodaju funkcije koje bi čuvale podatke sa displeja na karticu, ili koje bi učitavale slike sa kartice.

Trenutno je sistem napravljen tako da kontroler stalno provjerava da li je displej pritisnut. Ako uzmemo u obzir koliko brzo realan korisnik može da pritiska tipke, sistem bi se mogao preraditi da na prekid nekog od tajmera provjeri da li je displej pritisnut, a između prekida bi mogao da radi neku drugu obradu.

# Literatura

- [1] *<http://www.mikroe.com/add-on-boards/display/tft-proto/>*
- [2] *User manual za STM32F100xx familiju mikrokontrolera*
- [3] *User manual za STM32VLDISCOVERY razvojnu ploču*
- [4] *Datasheet za HX8347D kontroler*