

# Mikrokontrolerski sustav s grafičkim sučeljem za upravljanje raznim uređajima

---

Žugaj, Ivan

Master's thesis / Diplomski rad

2024

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:168:327424>

*Rights / Prava:* [In copyright](#) / [Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2025-03-20**



*Repository / Repozitorij:*

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 506

**MIKROKONTROLERSKI SUSTAV S GRAFIČKIM SUČELJEM  
ZA UPRAVLJANJE RAZNIM UREĐAJIMA**

Ivan Žugaj

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 506

**MIKROKONTROLERSKI SUSTAV S GRAFIČKIM SUČELJEM  
ZA UPRAVLJANJE RAZNIM UREĐAJIMA**

Ivan Žugaj

Zagreb, lipanj 2024.

## DIPLOMSKI ZADATAK br. 506

Pristupnik: **Ivan Žugaj (0036516286)**

Studij: Računarstvo

Profil: Računalno inženjerstvo

Mentor: prof. dr. sc. Mladen Vučić

Zadatak: **Mikrokontrolerski sustav s grafičkim sučeljem za upravljanje raznim uređajima**

### Opis zadatka:

Proučiti mikrokontrolere familije STM32F4 i pokaznike s tekućim kristalima s dodirnim zaslonom koji se temelje na pogonskom sklopu ILI9341. Odabrati mikrokontroler iz spomenute familije te razviti sklopovlje i programsku podršku sustava za upravljanje različitim uređajima. Korisničko sučelje sustava treba sadržavati pokaznik s dodirnim zaslonom, rotacijski enkoder i šest namjenskih tipki. Za komunikaciju s uređajima predvidjeti USB i UART konektore te konektor koji sadrži digitalne ulazno izlazne linije. Napajanje sustava izvesti dvojakom - preko USB konektora uređaja kojim se upravlja i preko vanjskih stezaljki. Cjelokupno sklopovlje objediniti na jednoj tiskanoj pločici te izraditi prototip. Upravljanje uređajem izvesti preko pokaznika i dodirnog zaslona, a odabir uređaja kojim se upravljanja izvesti preko namjenskih tipki. Razviti programsku podršku koja demonstrira rad sustava te programsku podršku za upravljanje danim programski definiranim prijamnikom.

Rok za predaju rada: 28. lipnja 2024.

Želio bih se zahvaliti mojoj obitelji i djevojci Luciji Budinski na podršci tijekom studiranja i izrade diplomskog rada. Volio bih se zahvaliti kolegi Mislavu Mandiću za pomoć tijekom izrade sklopovlja i svim kolegama tijekom studija koji su mi uljepšali ovaj period života.

Zahvaljujem se i mentoru Mladenu Vučiću tijekom diplomskog studija na strpljenju i mentoriranju.

## Sadržaj

1.	Uvod .....	3
2.	Sklopovska platforma .....	4
2.1.	Mikrokontroler STM32F407 .....	4
2.1.1.	Arhitektura .....	4
2.1.2.	ARM procesorska jezgra .....	5
2.1.3.	Memorija .....	6
2.1.4.	Upravljačko sklopovlje .....	8
2.1.5.	Komunikacijsko sklopovlje .....	10
2.1.6.	Brojila .....	14
2.2.	ILI9341 .....	14
2.3.	Rotacijski enkoder .....	16
3.	Sklopovlje računalnog sustava .....	17
3.1.	Napajanje sustava .....	17
3.2.	Mikrokontroler STM32F407 .....	18
3.3.	Sklopovlje za pogon serijske veze .....	20
3.4.	Rotacijski enkoder .....	22
3.5.	LCD prikaznik .....	22
3.6.	Priključak za uhodavanje i ispitivanje .....	23
3.7.	Svijetleće diode i tipke .....	24
3.8.	Ulazno-izlazni priključci .....	25
4.	Programska podrška .....	27
4.1.	FreeRTOS .....	28
4.2.	Organizacija i opis zadataka u FreeRTOS sustavu .....	31
4.2.1.	Enkoder zadatak .....	31
4.2.2.	Tipka zadatak i ulazno-izlazno prekidi .....	33

4.2.3.	LCD zadatak .....	33
4.2.4.	Biblioteka za crtanje i pisanje na grafičkom prikazniku .....	37
5.	Implementacija sustava.....	39
5.1.	Prototip tiskane pločice .....	39
5.2.	Izmjene na postojećem sklopovlju .....	40
5.3.	Rad s računalnim sučeljem .....	43
	Zaključak .....	49
	Literatura .....	50
	Sažetak.....	52
	Summary.....	53
	Privitak A – Električne sheme .....	54
	Privitak B – nacrti tiskane pločice .....	57
	Privitak C – tablica materijala .....	59

# 1. Uvod

Mikrokontroleri su doživjeli brzu evoluciju u posljednjih nekoliko godina, postajući sve snažniji i jeftiniji. To je dovelo do eksplozije u broju i raznolikosti aplikacija u kojima se koriste, od pametnih telefona i automobila do nosivih uređaja i kućne automatizacije.

Mikrokontroler je elektronička komponenta koja u svakom ugradbenom sklopu čini središnji dio spajajući procesorsku jezgru, memoriju i periferne sklopove. Mikrokontroleri se međusobno razlikuju po arhitekturi procesorske jezgre, koja može raditi s podacima širine 8, 16 ili 32 bita. Danas se na tržištu nalazi veliki izbor mikrokontrolera raznih proizvođača, kao što su Renesas Electronics, Texas Instruments, NXP Semiconductors, Microchip Technology i STMicroelectronics.

U okviru ovog rada razvijen je sustav koji sadrži mikrokontroler porodice STM32F407 [1] proizvođača STMicroelectronics temeljen na jezgri ARM Cortex M4. Korisničko sučelje sastoji se od LCD prikaznika s ILI9341 [2] sklopom, sklopa za dodir zaslona XPT2046 [3], enkodera PEC11R [4], 6 tipki i 6 svjetlećih dioda (*Light Emitting Diode*, LED). S vanjskim svijetom povezan je preko USB i RS232 serijskog sučelja te USB B veze.

Drugo poglavlje opisuje arhitekturu ARM procesorske jezgre te sklopove koji se koriste u dizajnu. Treće poglavlje opisuje dizajn sklopovlja i njihovu izvedbu, dok u četvrto poglavlje opisuje programsku podršku. Peto poglavlje daje opis prototipa i način uporabe sustava. U prilogu rada dana je cjelovita tehnička dokumentacija koja obuhvaća elektroničke sheme, nacрте tiskane pločice i popis materijala.



## 2. Sklopovska platforma

Sklopovska platforma sastoji se od mikrokontrolera STM32F407, LCD prikaznika s ILI9341 sklopom i XPT2046 sklopa za dodir te rotacijskog enkodera PEC11R. U sljedećim poglavljima detaljno je opisana arhitektura i rad mikrokontrolera te potom i rad prikaznika ILI9341 i rotacijskog enkodera.

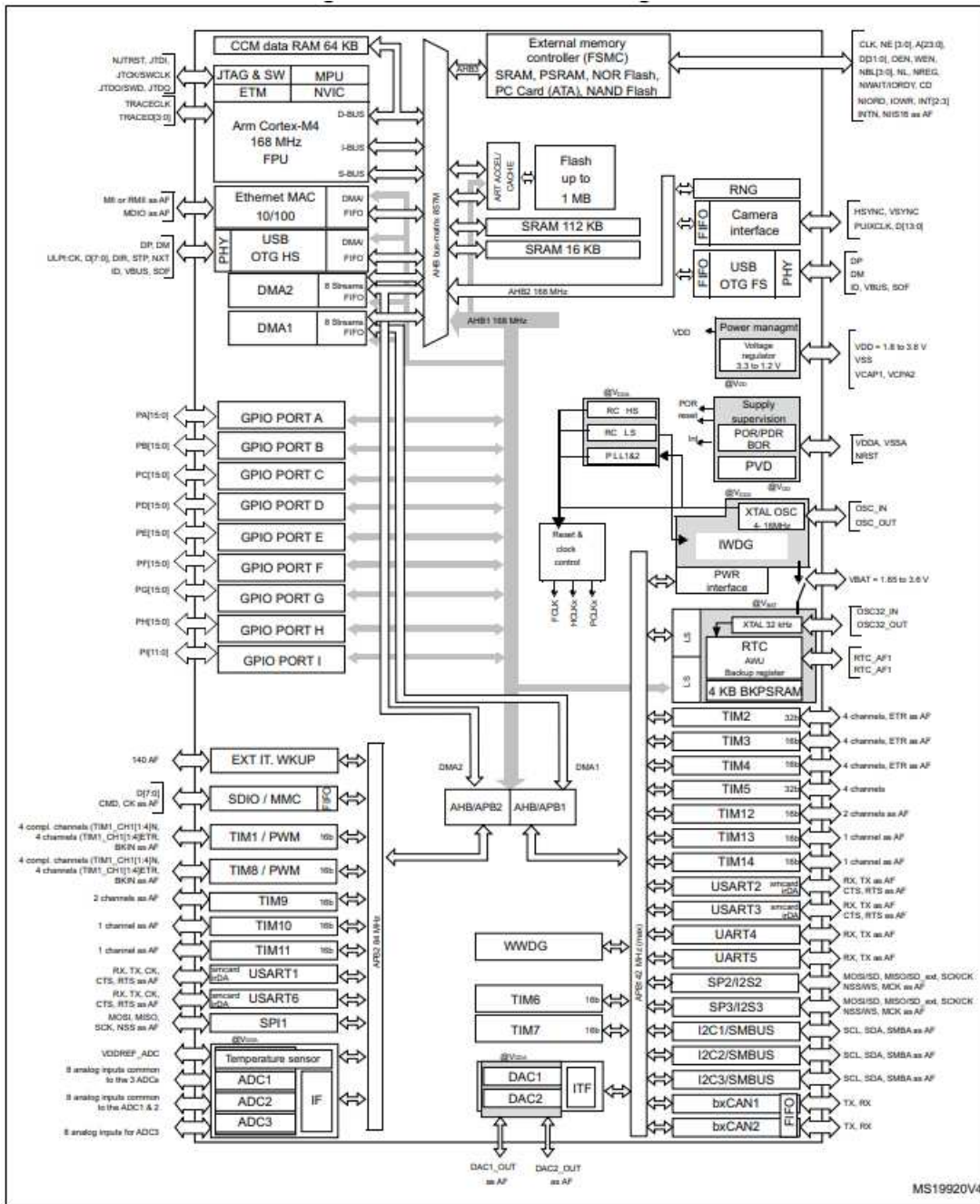
### 2.1. Mikrokontroler STM32F407

Mikrokontroler STM32F407 [1] proizvođača STMicroelectronics pripada porodici mikrokontrolera s 32 bitnom ARM RISC procesorskom jezgrom s jedinicom za rad u pomičnom zarezu (*Floating Point Unit*, FPU). Sadrži memoriju tipa Flash EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) veličine 512 kilobajta i statičku RAM memoriju (*Static Random-Access Memory*, SRAM) veličine 192 + 4 kilobajta.

Mikrokontroler se nalazi u LQFP100 kućištu koje se sastoji od 100 priključaka, od kojih su 82 priključka slobodna za korištenje kao ulazno-izlazni priključci ili za korištenje funkcije nekih od perifernih sklopova. Mikrokontroler radi u rasponu napajanja od 1.8 V do 3.3 V, te u sebi sadrži oscilator frekvencije 16 MHz koji preko PLL (*Phased Locked Loop*) sklopa omogućuje rad sustava između 8 MHz i 168 MHz.

#### 2.1.1. Arhitektura

Blokovska shema, prikazana slikom 2.1, prikazuje procesor, memoriju te periferno i upravljačko sklopovlje. Prikazani su i signali unutar mikrokontrolera koji vode na vanjske priključke, budući da se ovi priključci mogu koristiti za spoj unutrašnjih perifera mikrokontrolera.



Slika 2.1 Blokovska shema mikrokontrolera STM32F407 [1]

## 2.1.2. ARM procesorska jezgra

ARM procesorska jezgra prvi put je razvijena sredinom 80-ih godina 20. stoljeća u Engleskoj u firmi „Acorn Computers Limited“, koja se danas zove „ARM Holding plc“. Razvoj jezgri traje i danas. Glavne odlike poput jednostavnosti arhitekture, velike

učinkovitosti izvođenja instrukcija i mala potrošnje čine ih idealnim za ugradnju u male računalne sustave.

Mikrokontroler STM32F407 sadrži ARM Cortex-M4 procesorsku jezgru koja se temelji na 32-bitnoj arhitekturi sa smanjenim skupom instrukcija (*Reduced Instruction Set Computer*, RISC). Smanjeni skup instrukcija sadrži jednostavnije instrukcije za rad s podacima što dovodi do bržeg dekodiranja i izvršavanja. Uz jednostavne instrukcije, jezgra sadrži i protočnu strukturu. Protočnost omogućava brže izvođenje instrukcija čime se performance jezgre povećavaju u usporedbi s jezgrama koje nemaju protočnost. Instrukcije se mogu podijeliti u 6 skupina: učitavanje i spremanje podataka, obrada podataka, grananje, prijenos registara stanja, koprosorske instrukcije i generiranje iznimki. Procesor omogućava brzo izvođenje iznimki pomoću prekidnog kontrolera (*Nested Vectored Interrupt Controller*, NVIC).

Procesor je temeljen na Harvard arhitekturi, s memorijama kojima se pristupa preko 32-bitnih sabirnica. Procesor je 32-bitni, no on podržava i rad sa 16 bitnim podacima koje nazivamo poluriječi te 8 bitnim podacima tj. bajtovima. Unutar memorijskog prostora stoga imamo riječi od 32 bita poravnate s adresama koje su djeljive s četiri, poluriječi od 16 bitova na adresama djeljivim s dva dok bajtove možemo pronaći bilo gdje u memorijskom prostoru. Podatci se pohranjuju u *little-endian* formatu.

### **2.1.3. Memorija**

Slika 2.2 prikazuje memorijski prostor mikrokontrolera STM32F407 koji je organiziran u nekoliko područja, svaki sa specifičnom namjernošću. Na dnu memorijskom prostora, od adresa 0x00000000 do 0x1FFFFFFF, nalazi se kodni prostor unutar kojeg su alocirani Flash memorija, SRAM i sistemska memorija. Flash memorija ovdje je ključna budući da se u njoj nalazi programski kod koji mikrokontroler izvršava. SRAM se koristi za pohranu podataka.

Gornji dio memorijskog prostora, adrese od 0x20000000 do 0xFFFFFFFF, pokriva različite periferije i upravljačke registre. Pojedini segmenti memorije uključuju periferijske blokove, FSMC (*Flexible Static Memory Controller*) registre te interne periferije Cortex-M4 jezgre. Primjerice adrese od 0x40000000 do 0x5FFFFFFF odgovaraju periferijama spojenim na AHB1 sabirnicu (*Advanced High-performance Bus*), gdje se nalaze kontroleri



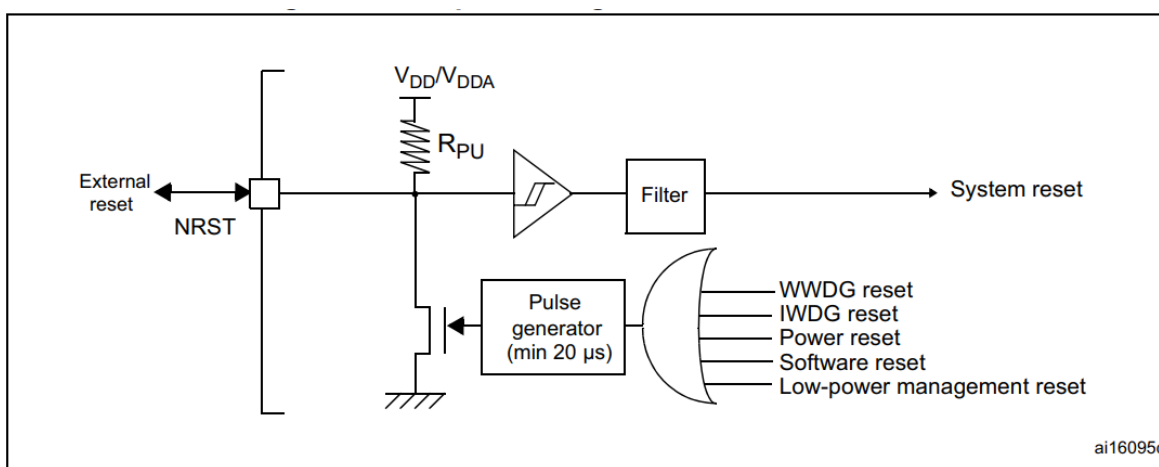
## 2.1.4. Upravljačko sklopovlje

Upravljačko sklopovlje upravlja svim perifernim jedinicama i funkcijama mikrokontrolera. U ovom poglavlju opisano je sklopovlje za upravljanje resetom, generator takta, upravljanje prekidima te upravljanje ulazno-izlaznim priključcima.

Reset može zatražiti:

- jedinica za nadzor napajanja
- vanjski priključak
- unutrašnje sklopovlje

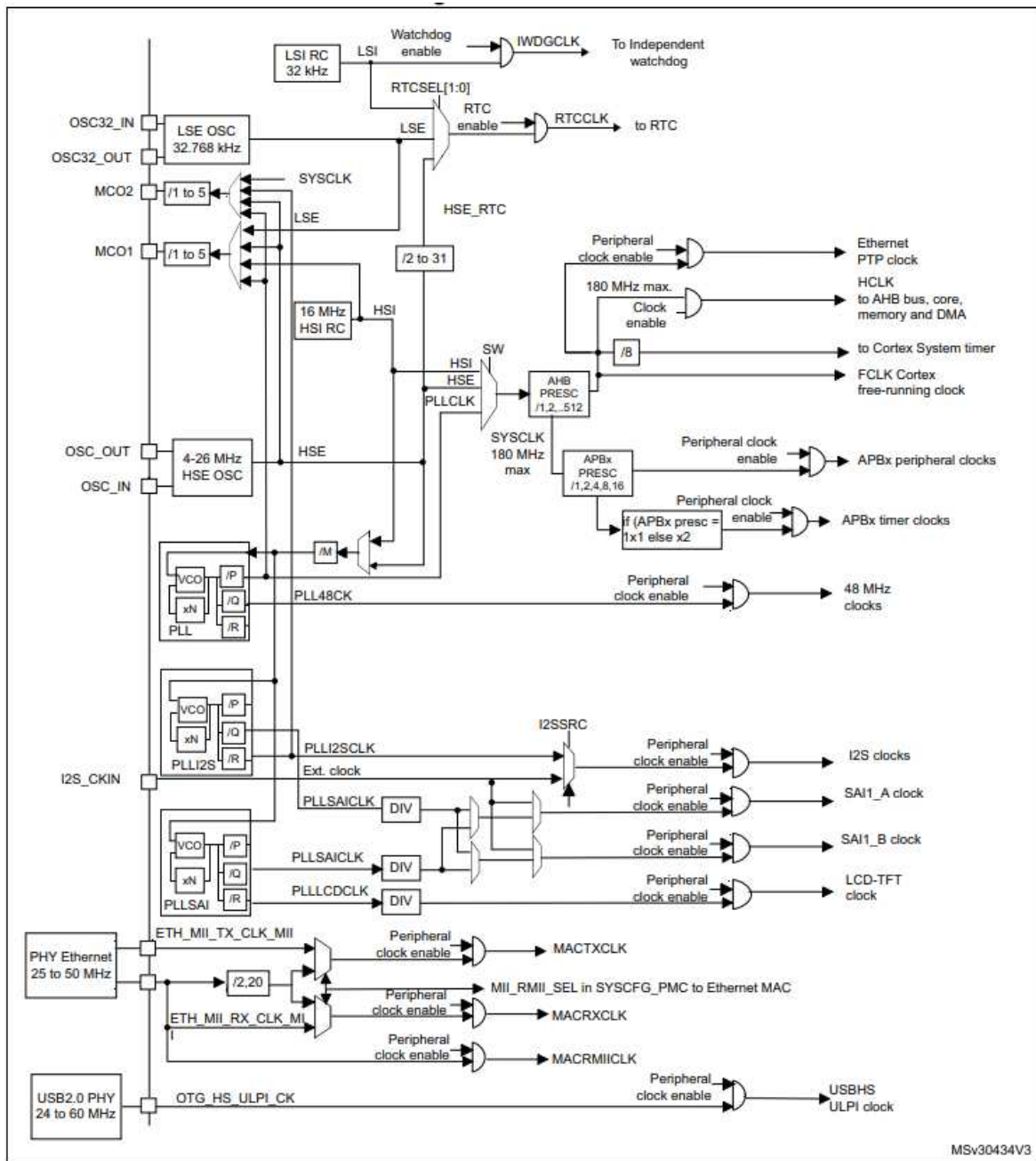
Slika 2.3 prikazuje blokovsku shemu sklopovlja za upravljanje resetom. Jedinica za nadzor napajanja generira reset prilikom uključivanja (*Power-On Reset, POR*) te reset prilikom pojave niskog napona napajanja (*Brownout Detection, BOD*). Reset koji dolazi iz vanjskog priključka može zatražiti neka vanjska jedinica ili se na ovaj priključak može spojiti tipka. Unutrašnji izvori reseta obuhvaćaju sklopove za nadzor rada (*watch dogs*), programske izvore, i sl. Reset mora trajati minimalno 20 mikrosekundi, o čemu brine ugrađena logika.



Slika 2.3 Sklopovlje za upravljanje RESET-om mikrokontrolera STM32F407 [5]

Generator takta za mikrokontroler STM32F407 ključan je za rad i sinkronizaciju svih dijelova sustava. On sadrži nekoliko izvora takta: niskofrekvencijski (*Low-Speed Internal, LSI*) i visokofrekvencijski (*High-Speed Internal, HSI*) unutrašnji oscilator te niskofrekvencijski (*Low-Speed External, LSE*) i visokofrekvencijski (*High-Speed External, HSE*) izvora takta koji koriste vanjske kristale kvarca. Ovi oscilator ne daju

dovoljno visoku frekvenciju takta za rad sustava, već se ona dobiva korištenjem PLL (*Phase Lock Loop*) sklopa. Slika 2.4 prikazuje sustav za upravljanje taktom. Sustav omogućava množenje i dijeljenje frekvencije osnovnog takta kako bi se postigla željena frekvencija rada za različite podsustave mikrokontrolera.



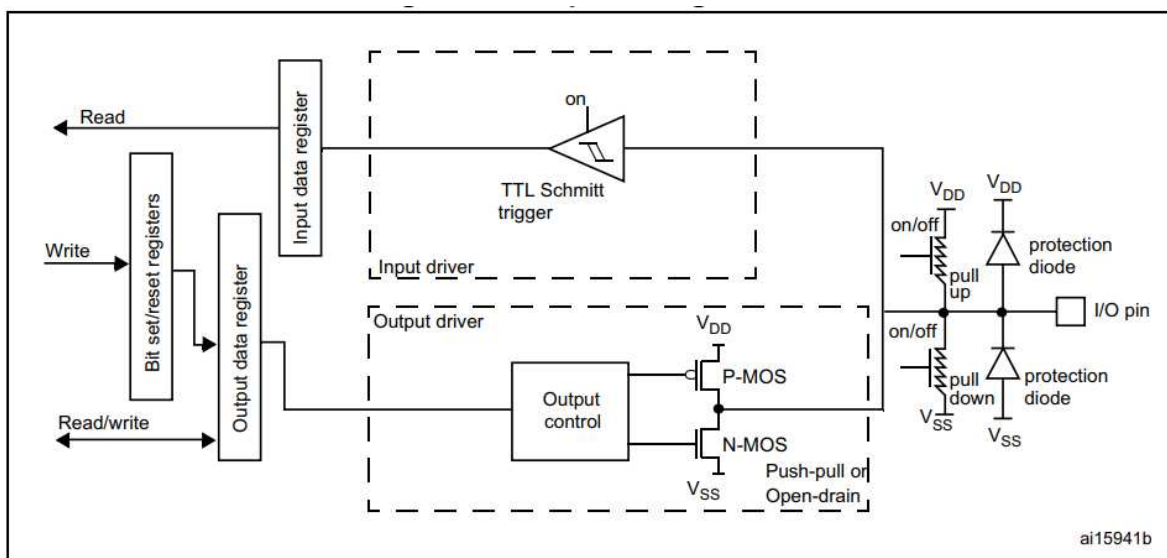
Slika 2.4 Sklopovlje za upravljanje taktom mikrokontrolera STM32F407 [5]

Sklop za upravljanje prekidima (*Nested Vector Interrupt Controller, NVIC*) obrađuje 82 prekida sa 16 razina prioriteta. Prekidi mogu biti vanjski i unutrašnji. Vanjski



prekidi dolaze preko priključaka koje je potrebno definirati kao ulazne. Programski je moguće definirati hoće li se prekid aktivirati na rastući, padajući ili pak na oba brida signala. Kada se dogodi prekid, NVIC automatski usmjerava programski tok na odgovarajuću prekidnu rutinu, smanjujući tako potrebu za softverskim upravljanjem prekidima.

Sklopovlje za upravljanje ulazno-izlaznim priključcima (*General Purpose Input/Output*, GPIO) omogućava definiranje priključaka kao ulaznih, izlaznih ili kao priključaka za periferni jedinice, na primjer PWM (*Pulse Width Modulation*), SPI (*Serial Peripheral Interface*), I2C (*Inter Integrated Connection*) i slično. Svaki izlazni priključak može imati dva stanja, otvoreni odvod ili stanje visoke impedancije. Izlazni priključci predviđeni su za struju do 25 mA, te je moguće specificirati brzinu rada kako bi se optimirala potrošnja i elektromagnetska zračenja. Kad se priključak koristi kao ulazni, izlazno sklopovlje mora se postaviti u stanje visoke impedancije.



Slika 2.5 Shema priključka mikrokontrolera STM32F407 [5]

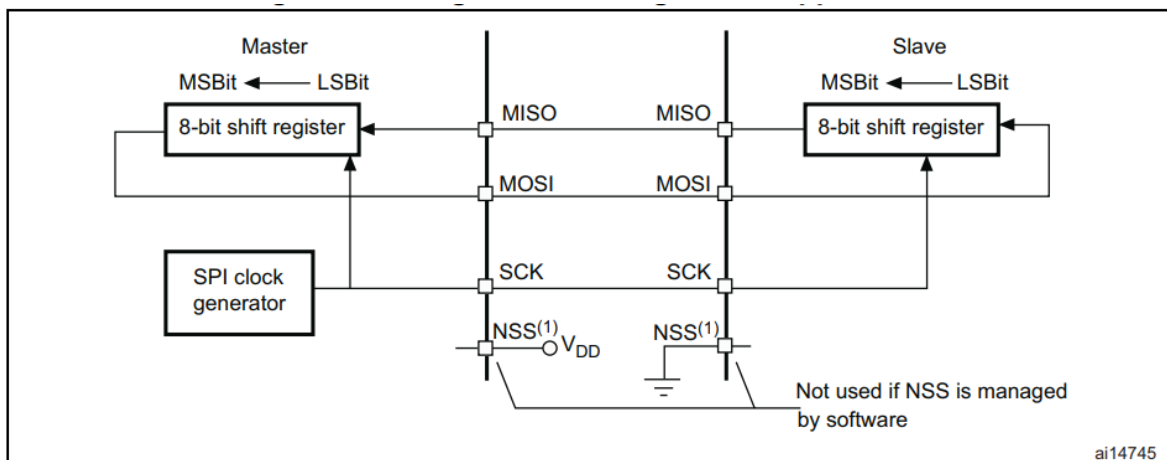
## 2.1.5. Komunikacijsko sklopovlje

Mikrokontroler STM32F407 ima veliki spektar perifernog sklopovlja za serijsku komunikaciju s drugim uređajima. U nastavku ovog poglavlja opisani su sklopovi i protokoli koji su korišteni u ovom radu.

SPI (*Serial Peripheral Interface*) je sinkroni serijski komunikacijski protokol koji povezuje mikrokontroler i vanjske periferijske uređaje. SPI omogućava brzu i učinkovitu razmjenu podataka preko četiri linije:

- SCK (*Serial Clock*)
- MISO (*Master Input Slave Output*)
- MOSI (*Master Output Slave Input*)
- SS/CS (*Slave Select / Chip Select*)

Spajanje ovih linija prikazano je slikom 2.6

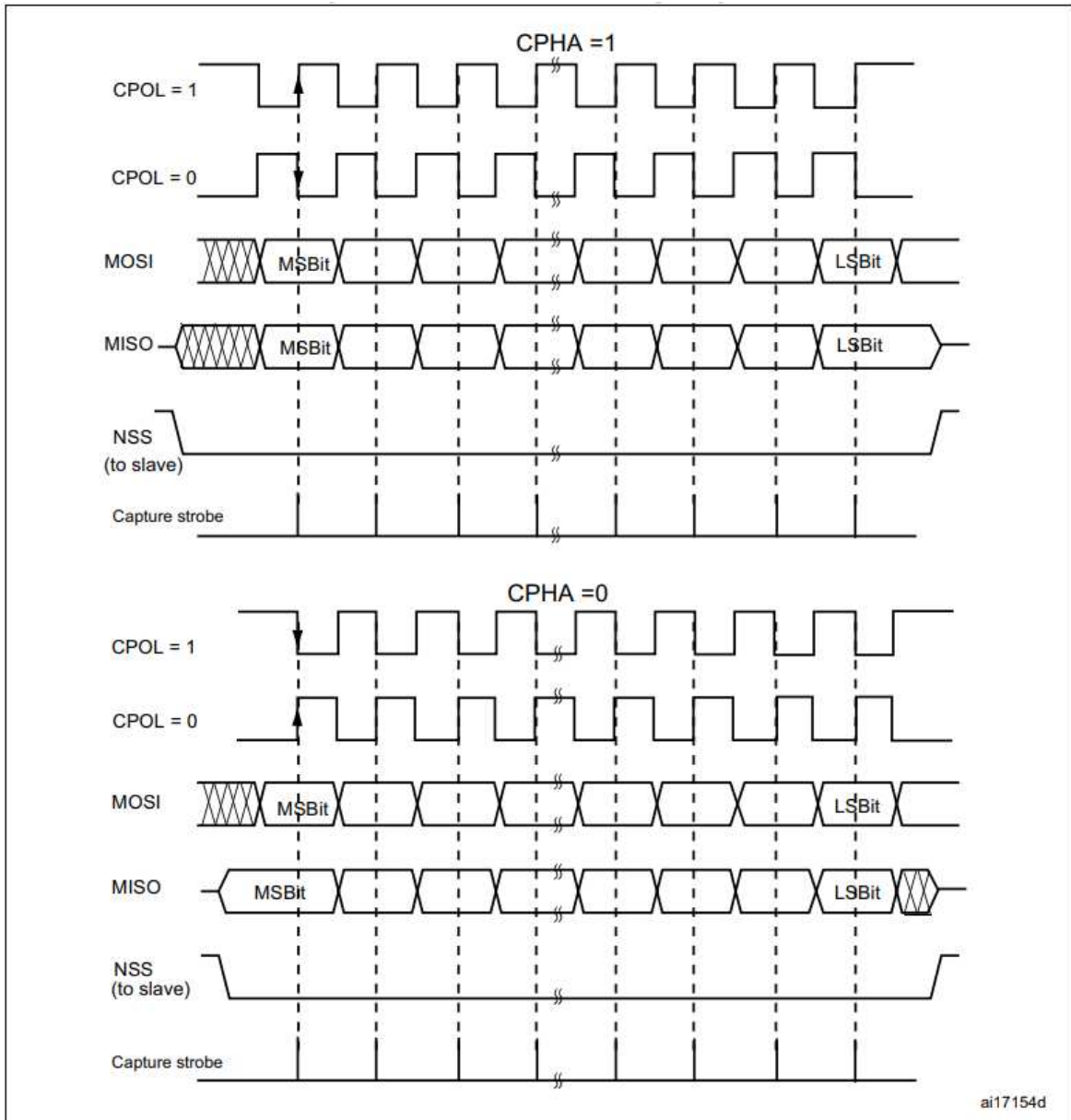


Slika 2.6 Prikaz spajanja upravljača i izvršioca SPI sučelja [5]

SPI sučelje može prenositi blokove podatka duljine 8 ili 16 bitova, unutar kojih se prvo prenosi MSB (*Most Significant Bit*). Prijenos je upravljan taktom (SCK). Moguće je konfigurirati njegov polaritet i fazu. Moguća su četiri načina rada, označena kao način (*mode*) 0, 1, 2 i 3, koji se razlikuju po kombinaciji polariteta takta (*Clock Polarity*) i njegove faze (*Clock Phase*). Slika 2.7 predstavlja dijagram prijenosa podataka ovisno o ovim parametrima. Polaritet takta određuje stanje linije SCK kada je komunikacija neaktivna. Pritom polaritet CPOL=0 označava nisko, a CPOL=1 visoko stanje. Faza takta određuje kada se podatci uzorkuju. Pritom CPHA=0 označava uzorkovanje podataka na prvom bridu takta. Radi li se o rastućem ili padajućem bridu, određuje parametar CPOL. CPHA=1 označava uzorkovanje podataka na drugom bridu takta. Ove kombinacije omogućavaju konfiguraciju SPI sučelja ovisno o specifikaciji periferije s kojom je mikrokontroler povezan. SPI mode 0 označava da koristimo nisko stanje takta za vrijeme



neaktivnosti te da uzorkujemo podatke na prvom bridu takta. Njemu odgovaraju parametri  $CPOL = 0$  i  $CPHA = 0$ .

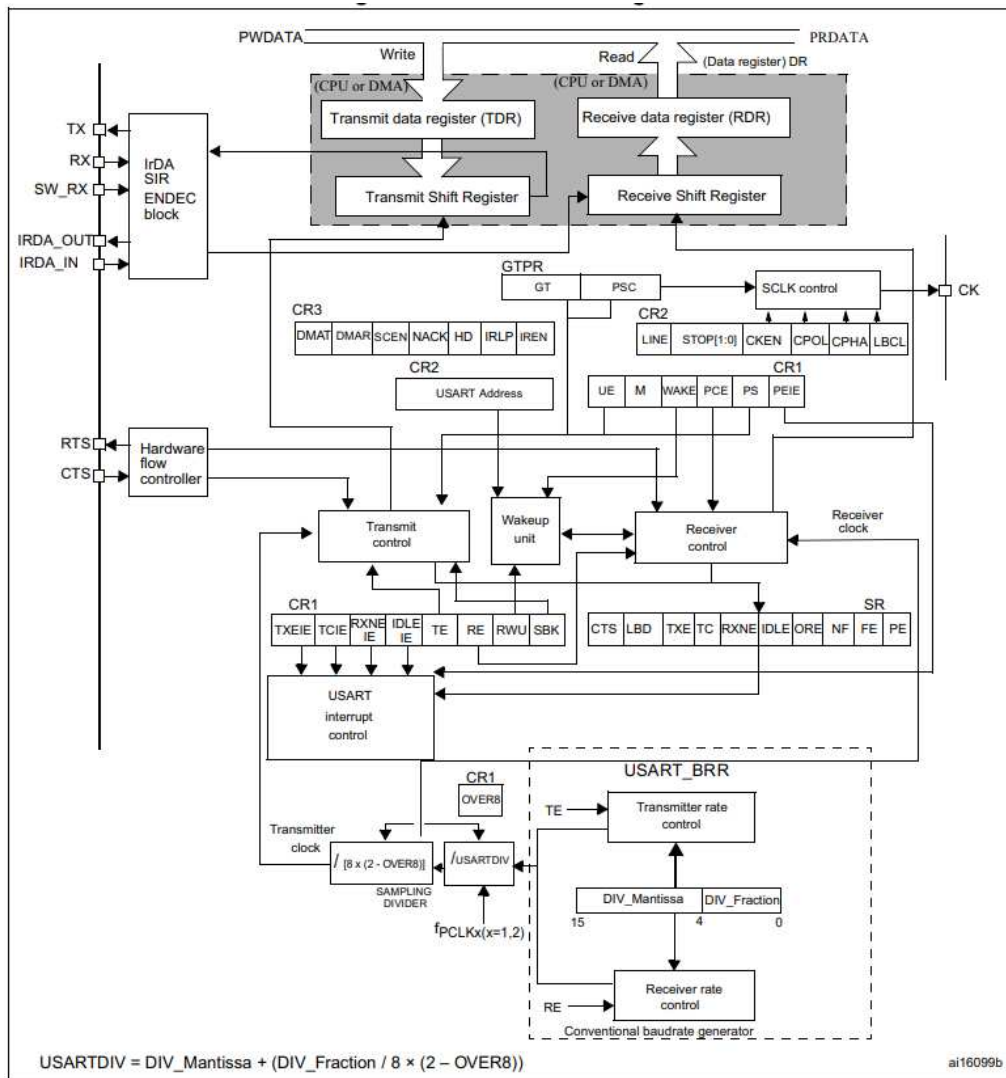


Slika 2.7 Dijagram prijenosa podataka u SPI komunikaciji [5]

USART (*Universal Synchronous/Asynchronous Receiver/Transmitter*) i UART (*Universal Asynchronous Receiver/Transmitter*) su komunikacijski protokoli koji omogućuju serijski prijenos podataka između mikrokontrolera i periferijskih uređaja. USART može raditi u oba načina rada, sinkronom i asinkronom, dok UART radi isključivo u asinkronom načinu rada. Asinkroni način rada koristi dva osnovna signala: TX (*Transmit*) i RX (*Receive*), čime se omogućuje prijenos i prijам podataka bez potrebe za

zajedničkim taktnim signalom. Moguće je koristiti još dva signala: CTS (*Clear to Send*) i RTS (*Request to Send*). CTS signal signalizira spremnost uređaja za primanje podatka, a RTS signal signalizira uređaju da se pripremi za prijem podataka.

Slika 2.8 prikazuje blokovski prikaz USART sklopovlja. Svako USART i UART sklopovlje moguće je konfigurirati za različite brzine prijenosa (*baud rate*), koje se izražavaju u bitovima po sekundi (*bits per second, bps*). Brzina prijenosa se postavlja upisom u *baud rate* registar kojeg je moguće postaviti ovisno o specifikaciji uređaja s kojim ostvarujemo konekciju. Tipične brzine prijenosa su 9600, 19200, 38400, 57600, 115200 *bps*.



Slika 2.8 Blokovska prikaz USART sklopovlja [5]

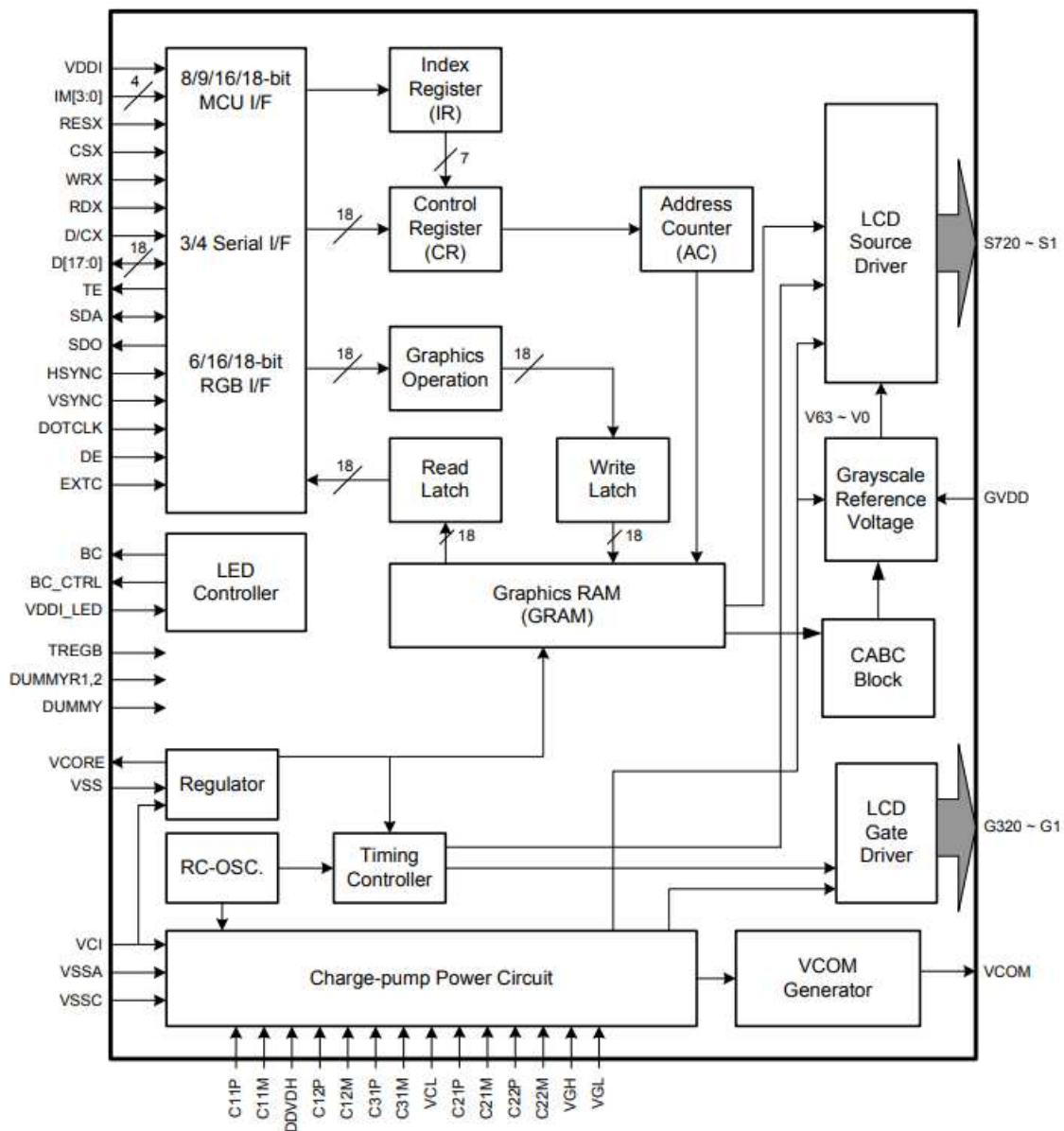
STM32F407 mikrokontroler ima USB (*Universal Serial Bus*) serijsku komunikaciju. Mikrokontroler podržava USB 2.0 full-speed (12 Mbps) i high-speed (480 Mbps) operacije. STM32F407 uključuje USB OTG (*On-The-Go*) kontroler, USB OTG FS (*Full-Speed*) i USB OTG HS (*High-Speed*), koji omogućuju mikrokontroleru da djeluje kao USB domaćin (*host*) ili kao vanjska jedinica (*device*). STM32F407 može komunicirati s drugim USB uređajima ili računalima preko različitih konfiguracija, poput jedinca za pohranu podataka (*Mass Storage*) i komunikaciju (*Communications Device*), čineći ga vrlo fleksibilnim za različite primjene.

### **2.1.6. Brojila**

Brojila su važan element za mjerenje vremena i sinkronizaciju u ugradbenim sustavima. Unutar STM32F407 mikrokontrolera nalazi se 17 brojila, od toga 12 sa 16-bitnom rezolucijom i dva sa 32-bitnom rezolucijom. Brojila pružaju mnogo funkcionalnosti, uključujući generiranje periodičkih prekida, mjerenje trajanja određenih događaja i upravljanje PWM (*Pulse Width Modulation*) signalima. U ovom radu brojilo je korišteno posebice za rad s rotacijskim enkoderom što je detaljnije opisano u poglavlju 2.3.

## **2.2. ILI9341**

ILI9341 [2] je LCD (*Liquid Crystal Display*) kontroler koji se koristi u TFT (*Thin-Film Transistor*) prikaznicima za primjenu u ugradbenim sustavima. Ovi prikaznici sastoje se od tekućih kristala koji pomoću pozadinskog svjetla emitiraju boje i proizvode sliku. Svaka točka koju nazivamo piksel sastoji se od najmanje tri točke kojima možemo upravljati zasebno kao RGB (*Red Green Blue*) kako bismo njihovom kombinacijom dobili jedinstvenu boju. Kontroler podržava paralelno i serijsko SPI sučelje, čime osigurava izrazito laku integraciju s mikrokontrolerima. ILI9341 upravlja TFT prikaznicima rezolucije do 240 x 320 piksela, što omogućuje kvalitetan prikaz alfanumeričkih podataka i grafike.



Slika 2.9 Blokovska shema ILI9341 kontrolera [2]

Slika 2.9 prikazuje blokovsku shemu ILI9341 kontrolera. Njegovi glavni blokovi su:

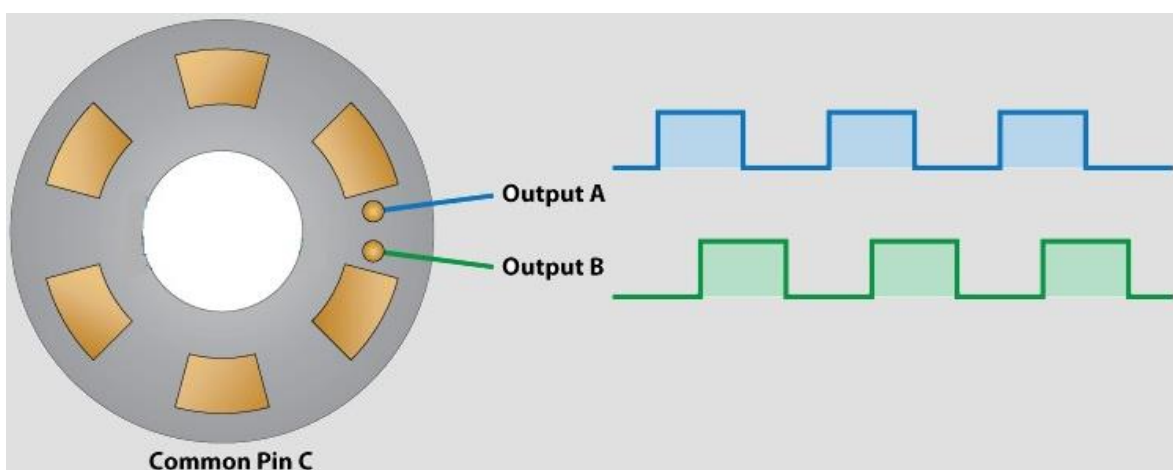
- MCU sučelje (*Micro Controller Unit*) – blok omogućuje komunikaciju između mikrokontrolera i ILI9341 preko 8/9/16/18-bitnog paralelnog sučelja korištenjem 3 ili 4 žičnog SPI sučelja koji se koristi za slanje naredbi i podataka za upravljanje prikazom.

- Grafički RAM (GRAM) – interna memorija koja pohranjuje podatke za prikaz. Podaci iz mikrokontrolera ili RGB sučelja se pohranjuju u GRAM te zatim prikazuju.
- LCD *Source Driver* i LCD *Gate Driver* – upravljaju fizičkim pikselima na TFT prikazniku. *Source Driver* kontrolira horizontalne linije dok *Gate Driver* upravlja vertikalnim linijama.

## 2.3. Rotacijski enkoder

Rotacijski enkoder je uređaj dizajniran za pretvaranje rotacijskog gibanja u digitalne signale. Rotacijski enkoder korišten u ovom radu je BOURNS PEC11R-4015F-S0024 [4] koji generira 24 impulsa po okretu (*Pulses Per Rotation*). Enkodere dijelimo na inkrementalne i apsolutne. Apsolutni enkodere prenose signal za apsolutnu poziciju dok inkrementalni prenosi promjenu pozicije.

Slika 2.10 prikazuje princip generiranja signala koje daje enkoder. Signali se dobivaju tijekom rotacije diska koji sadrži bakrene kontakte. Prolaskom bakra preko dviju elektroda generiraju se pravokutni impulsi međusobno pomaknuti za 90 stupnjeva. Kako bismo utvrdili promjenu stanja i smjer rotacije uspoređujemo stanje signala A naspram signala B. Ako okrećemo enkoder u smjeru kazaljke na satu imamo promjenu stanja signala A, koji je nakon promjene različit od stanja signala B, dok su okretanjem u smjeru suprotnom od kazaljke na satu stanja nakon promjene jednaka.



Slika 2.10 Princip rada rotacijskog enkodera [7]

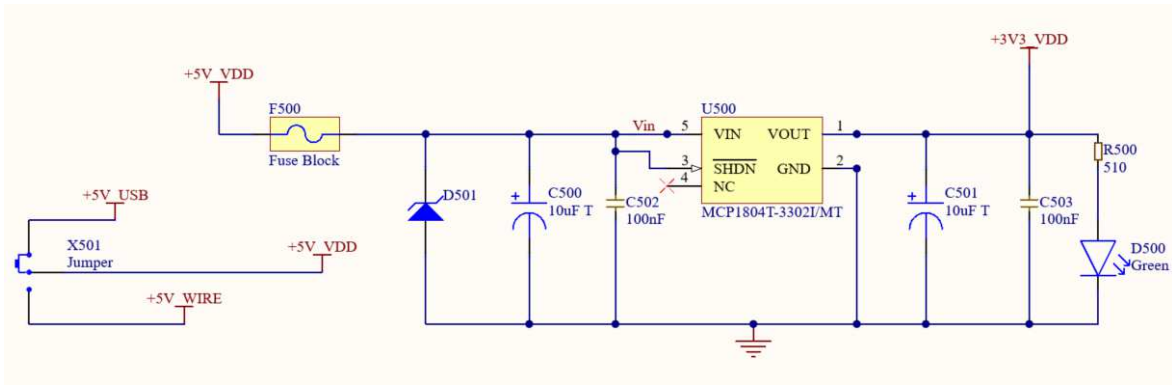
### 3. Sklopovlje računalnog sustava

Sklopovlje razvijenog računalnog sustava sastoji se od mikrokontrolera STM32F407VET6, 7 tipki, 9 svjetlećih dioda, rotacijskog enkodera BOURNS PEC11R-4015F-S0024, linearnog regulatora napajanja MCP1804-3302 [8], DE9 priključka, USB priključaka tipa micro B i B, konektora za ST-Link/V2 [9], konektora za vanjske GPIO priključke, pretvornika UART/RS232 ST3232BDR [10] i pretvornika USB/UART FT232R [11]. Dodano, sklopovlje sadrži ženske jednoreodne konektore s 14 i 4 priključaka za spajanje LCD grafičkog prikaznika. Sustav je izrađen u alatu Altium Designer [12]. Napravljene su dvije revizije sklopovlja. U rada je opisana druga revizija sklopovlja unutar koje su uklonjene greške koje su nastale tijekom izrade prvog prototipa.

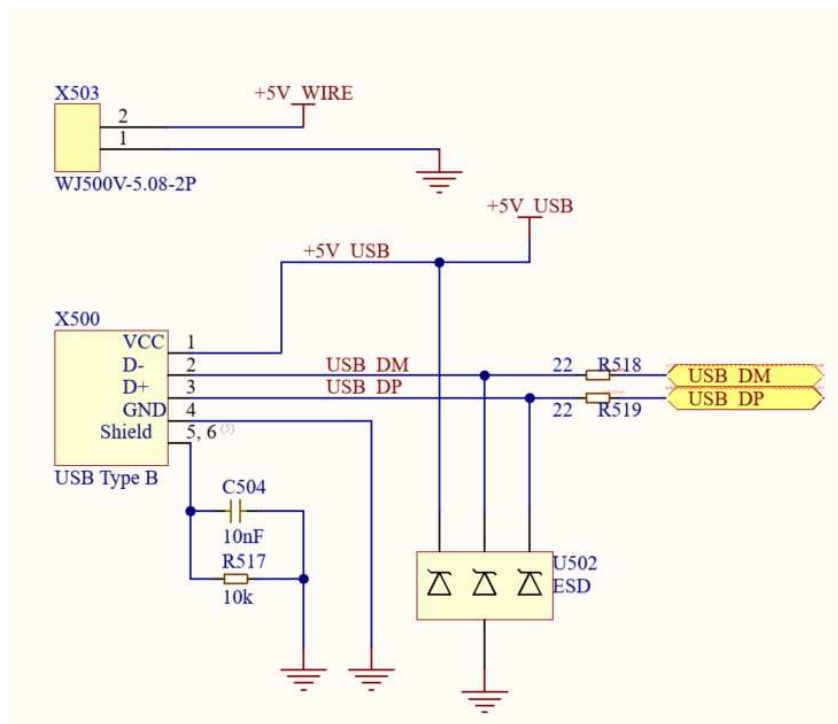
#### 3.1. Napajanje sustava

Napajanje sustava izvedeno je pomoću linearnog regulatora napajanja MCP1804-3302 [8], koji za ulazni napon u rasponu od 2.0 V – 28.0 V daje izlazni napon od 3.3 V uz struju do 150 mA. Uz linearni regulator imamo i Zenerovu diodu koja za ulazne napone veće od 5.6 V povuče struju dovoljnu za pregaranje osigurača F500. Kao ulazni napon linearnog regulatora moguće je dovesti 5 V iz namjenskog priključka za napajanje ili iz USB konektora. Odabir se izvodi preko premosnika X501. Svjetleća dioda na izlazu linearnog regulatora svijetli kako bi korisnik imao indikator da linearni regulator daje izlazni napon za sustav.

Električna shema sustava za napajanje prikazana je slikom 3.1. Slika 3.2 prikazuje USB tip B koji služi za napajanje i konekciju s mikrokontrolerom, kojim ostvarujemo *Full-Speed* (12 Mbit/s) komunikaciju između vanjskih uređaja i mikrokontrolera. Uz USB postavljene su diode koje štite od elektrostatičnog izbijanja. Uz USB konektor imamo i konektor X503 koji je predviđen za spajanje žicama laboratorijskog izvora napajanja.



Slika 3.1 Električna shema sustava za napajanje



Slika 3.2 Ulazni priključci za napajanje sustava

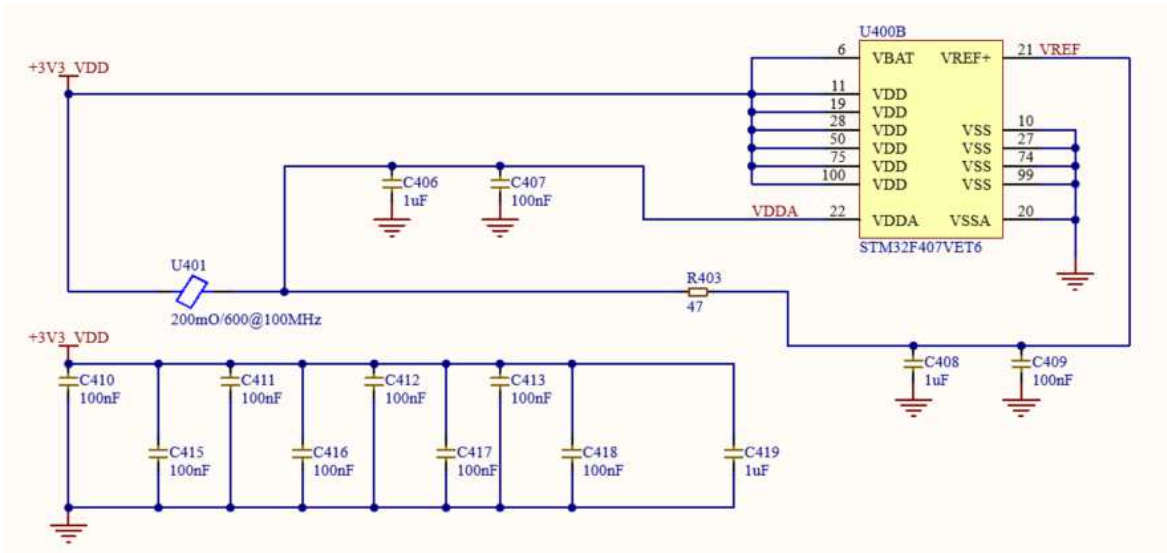
## 3.2. Mikrokontroler STM32F407

Mikrokontroler STM32F407VET6 sa svojim priključcima prikazan je na slici 3.3. Uz priključke mikrokontrolera nalaze se kristalni oscilator Y400 frekvencije 32.768 kHz i kristalni oscilator Y401 frekvencije 8 MHz. Oscilatori su povezani s mikrokontrolerom, a korišteni su za generiranje takta u LSE i HSE oscilatorima.





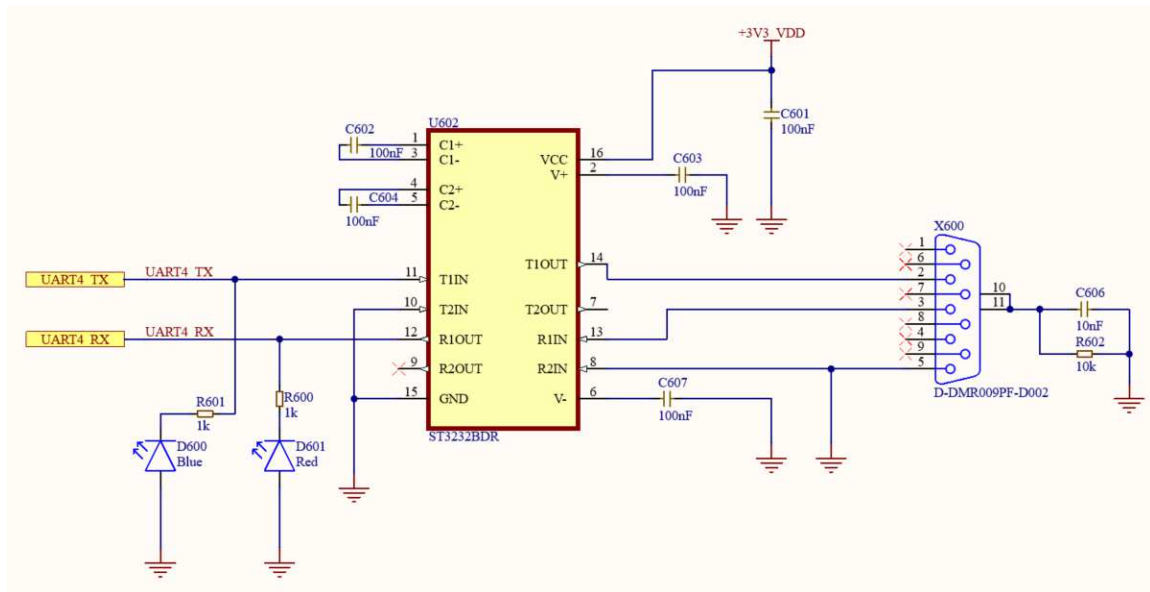




Slika 3.4 Napajanje mikrokontrolera

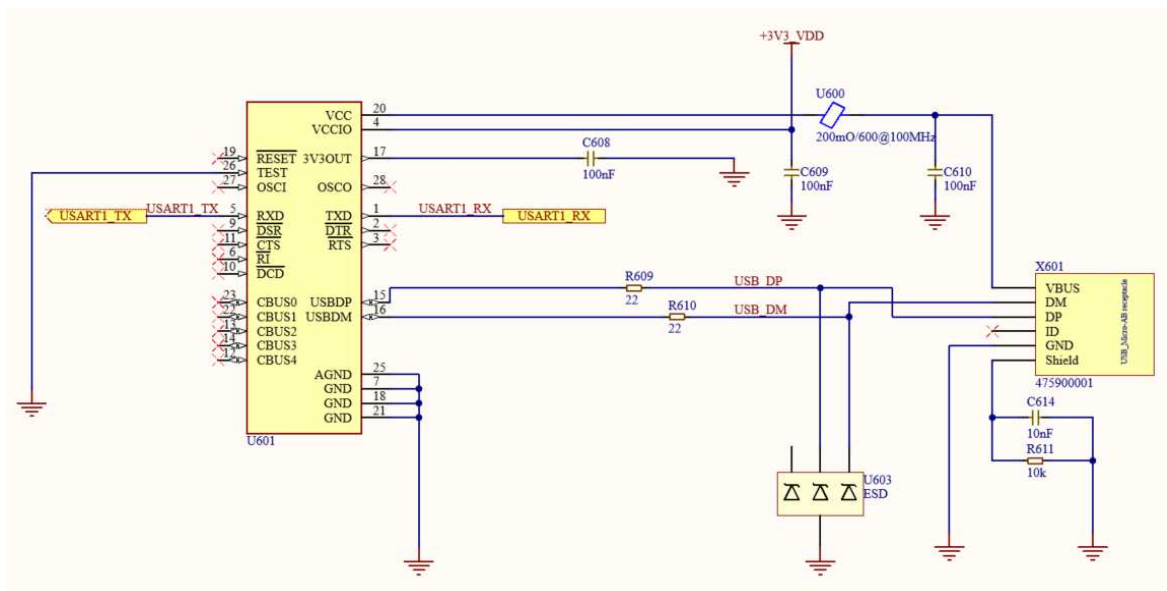
### 3.3. Sklopovlje za pogon serijske veze

Razvijeni računalni sustav sadrži dva asinkrona serijska sučelja. Prvo sučelje izvedene je pomoću UART sklopa mikrokontrolera. Potrebnu pretvorbu naponskih razina izvodi sklop ST3232 [10]. Korištena je samo jedna ulazna i jedna izlazna linija, dok je drugi par linija neaktivan. Na UART linije povezane su svjetleće diode plave i crvene boje koje svijetle tijekom prijenosa podataka. Slika 3.5 prikazuje shemu spajanja sklopovlja za pogon serijske veze s DE9 konektorom.



Slika 3.5 Serijska veza s RS232 konektorom

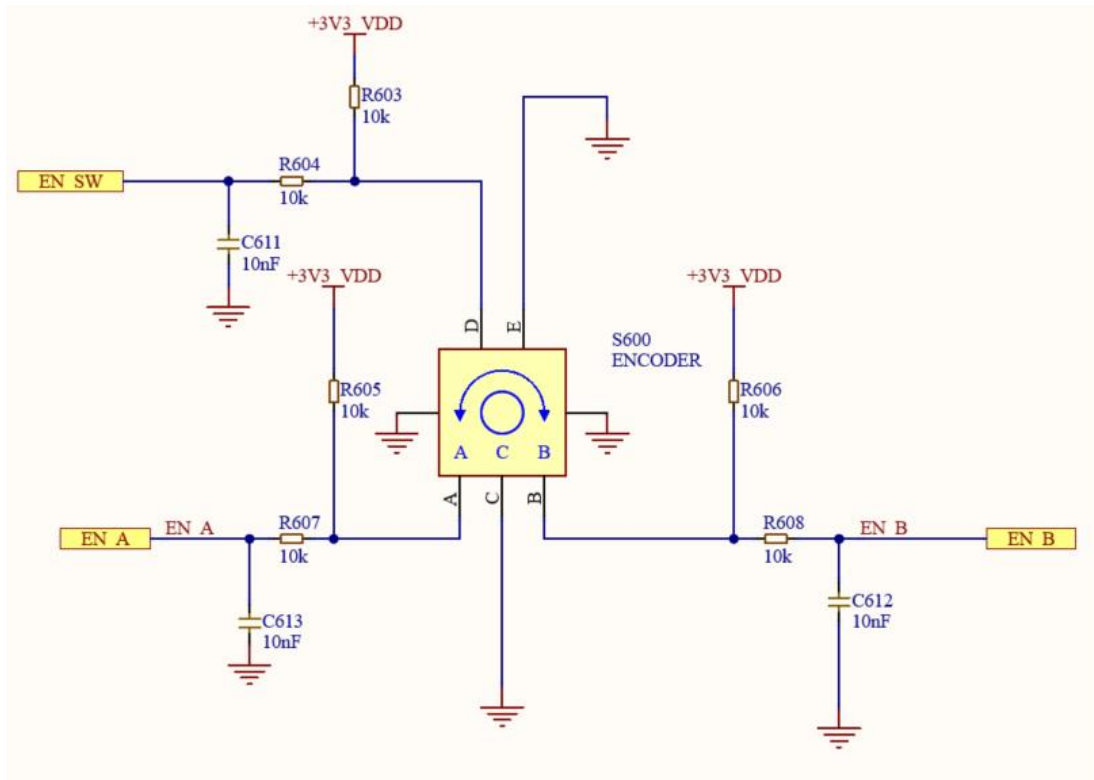
Drugo serijsko sučelje ostvaruje vezu preko micro B USB konektora. Za pretvorbu naponski razina ovdje se koristi sklop FT232R [11] kojeg prikazuje slika 3.6. Na konektor USB spojene su diode za elektrostatičku zaštitu. Napajanje od 5 V iz USB konektora spojeno je na VCC (*Voltage Common Collector*) sklopa FT232R kako bi on mogao ostvariti potrebnu pretvorbu naponskih razina.



Slika 3.6 Serijsko sklopvlje s micro B USB konektorom

### 3.4. Rotacijski enkoder

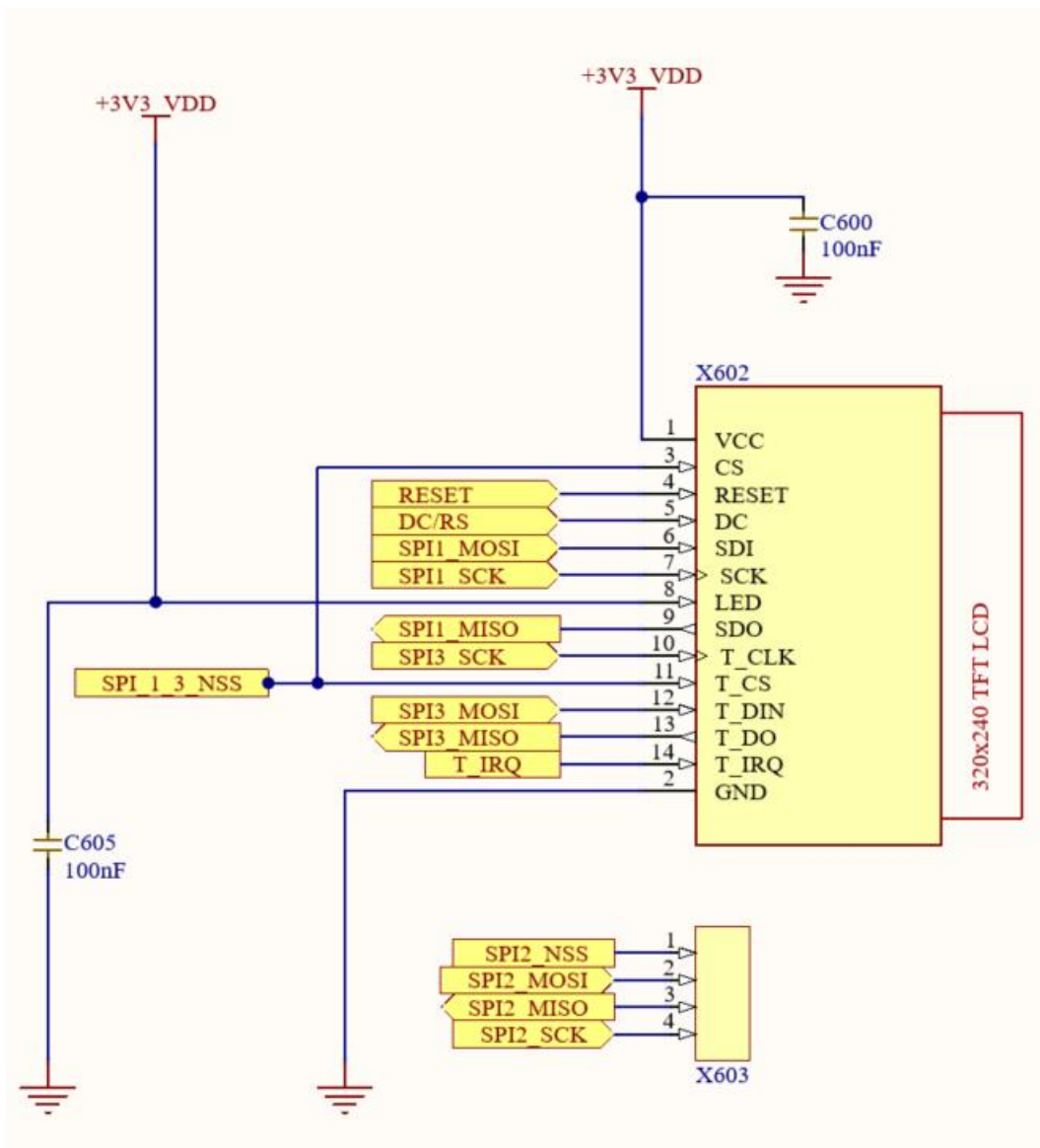
Izvodi rotacijskog enkodera PEC16-4020F-S0024 [4] A i B spojeni su priteznim otpornicima na napajanje VDD (*Voltage Drain*), a svaki od njih spojen je na RC filter za filtriranje kontaktnog šuma koje usporava vrijeme pada i porasta signala tijekom rotacije enkodera. Uz kanale A i B enkoder sadrži i tipku koja se aktivira pritiskom njegove osovine.



Slika 3.7 Sklopovlje rotacijskog enkodera

### 3.5. LCD prikaznik

Slika 3.8 prikazuje povezivanje mikrokontrolera STM32F407 i LCD prikaznika s ILI9341 kontrolerom, sklopom za upravljanje dodirnim zaslonom, XPT2046 te SD karticom. SPI 1 sučelje koristi se za upravljanje ILI9341 kontrolerom, SPI 2 sučelje za upravljanje SD karticom, a SPI 3 za upravljanje sklopom za dodirni zaslon. Sučelja SPI 1 i 3 dijele isti priključak za odabir čipa. LED priključak spojen je na 3.3 V čime zaslon ostvaruje maksimalnu svjetlinu.



Slika 3.8 Povezivanje mikrokontrolera s LCD prikaznikom, dodirnim zaslonom i SD karticom

### 3.6. Priključak za uhadavanje i ispitivanje

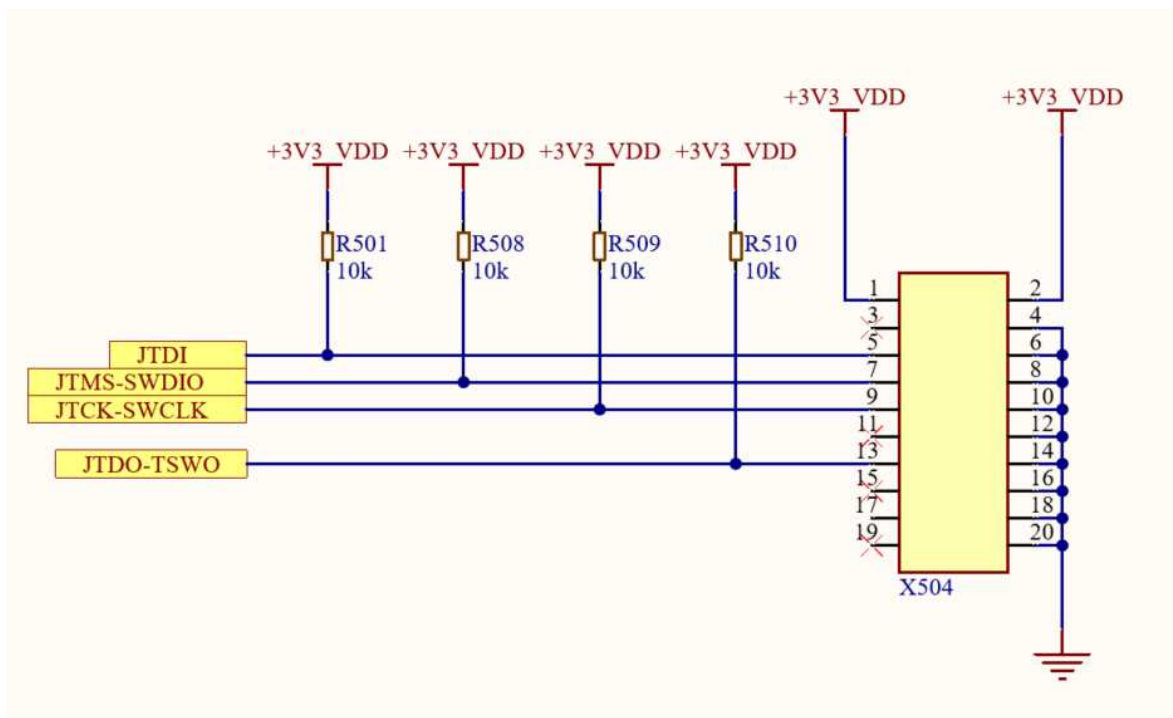
Učitavanje i uhadavanje programa na sklopovlju moguće je izvesti uz korištenje JTAG (*Joint Test Action Group*) ili SWD (*Serial Wire Debug*) sučelja koje spajamo na ST-LINK/V2 [9]. JTAG priključak sa slike 3.9 prikazuje četiri osnovna priključka:

- TCK – Test Clock
- TMS – Test Mode Select

- TDI – Test Data In
- TDO – Test Data Out

Isti priključci koriste se i za spajanje SWD sučelja:

- SWDIO – Serial Wire Data Input/Output
- SWCLK – Serial Wire Clock
- SWO – Serial Wire Optional Output za dodatne informacije ispitivanja

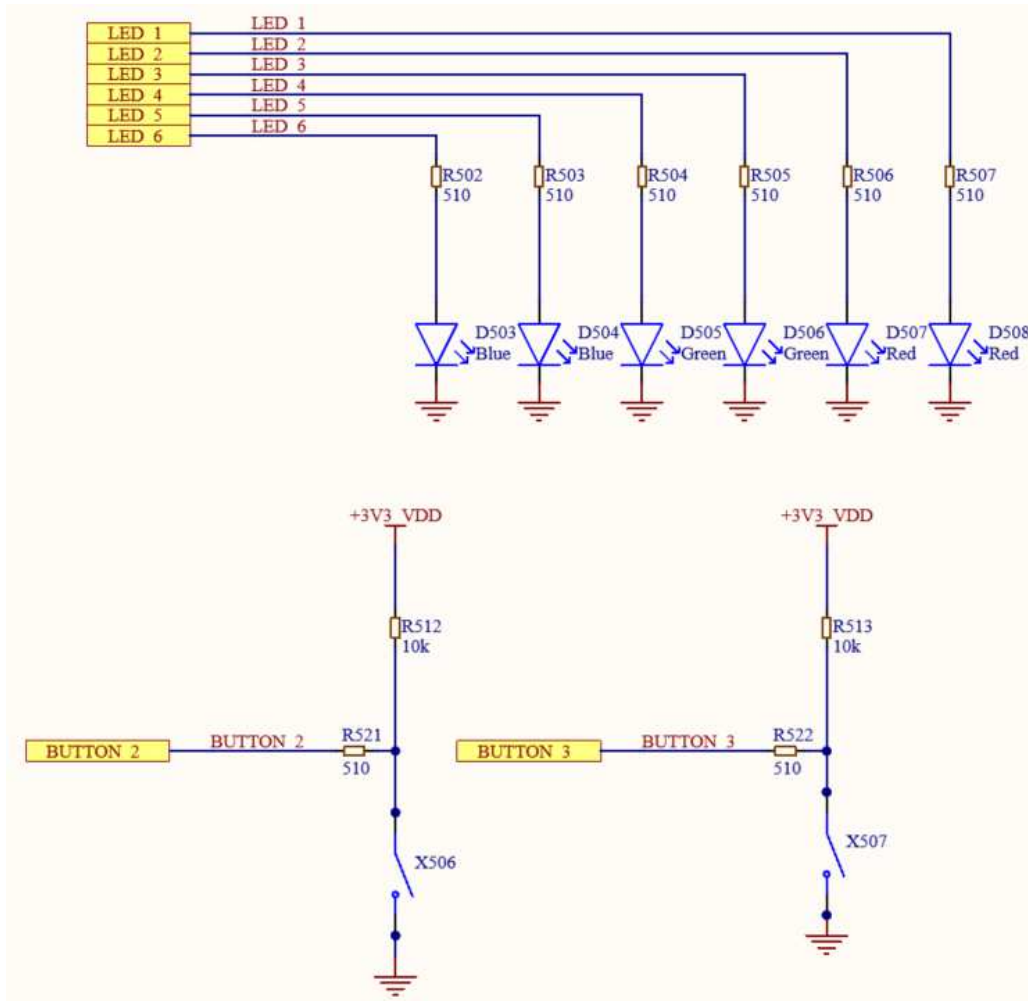


Slika 3.9 Priključci za uhadavanje sklopovlja

### 3.7. Svjetleće diode i tipke

Slika 3.10 prikazuje spoj svjetlećih dioda i tipki s priključcima mikrokontrolera. Svjetleće diode koriste struju od 1.5 mA, što je određeno otpornicima otpornici od 510 Ohma. Tipke su spojene prema masi, s priteznom otpornikom prema napajanju.

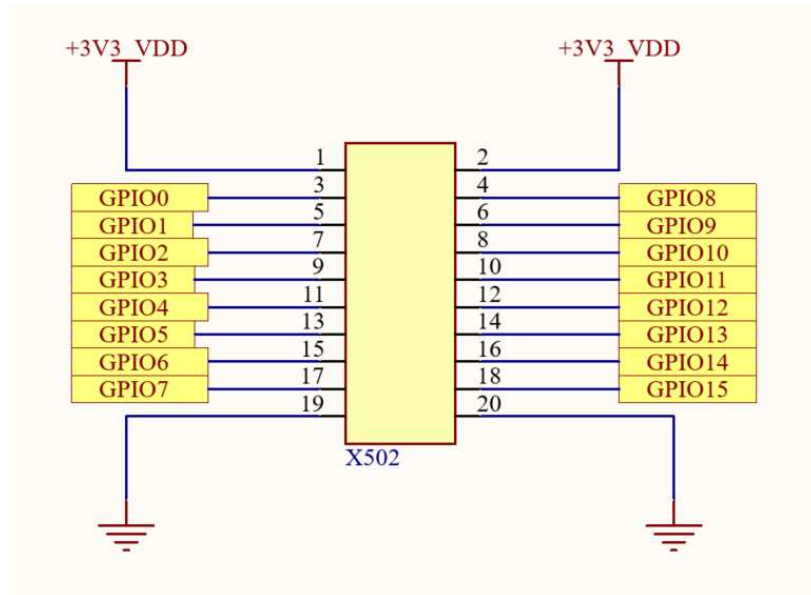
Također, spojeni su otpornicima prema mikrokontroleru koji služe kao zaštita u slučaju pogrešnog postavljanja priključka tijekom inicijalizacije.



Slika 3.10 Spoj svjetlećih dioda i tipki

### 3.8. Ulazno-izlazni priključci

Slika 3.11 prikazuje konektor na koji je spojeno 16 priključaka mikrokontrolera za proširenje sustava ili spajanja novih komponenti. Za ovu namjenu, na konektor je spojen cijeli niz priključaka D. Priključke je moguće definirati kao ulazne/izlazne ili neku od alternativnih funkcija sustava, pružaju struju do maksimalnih 20 mA te podržavaju rad frekvencijom takta do 84 MHz.



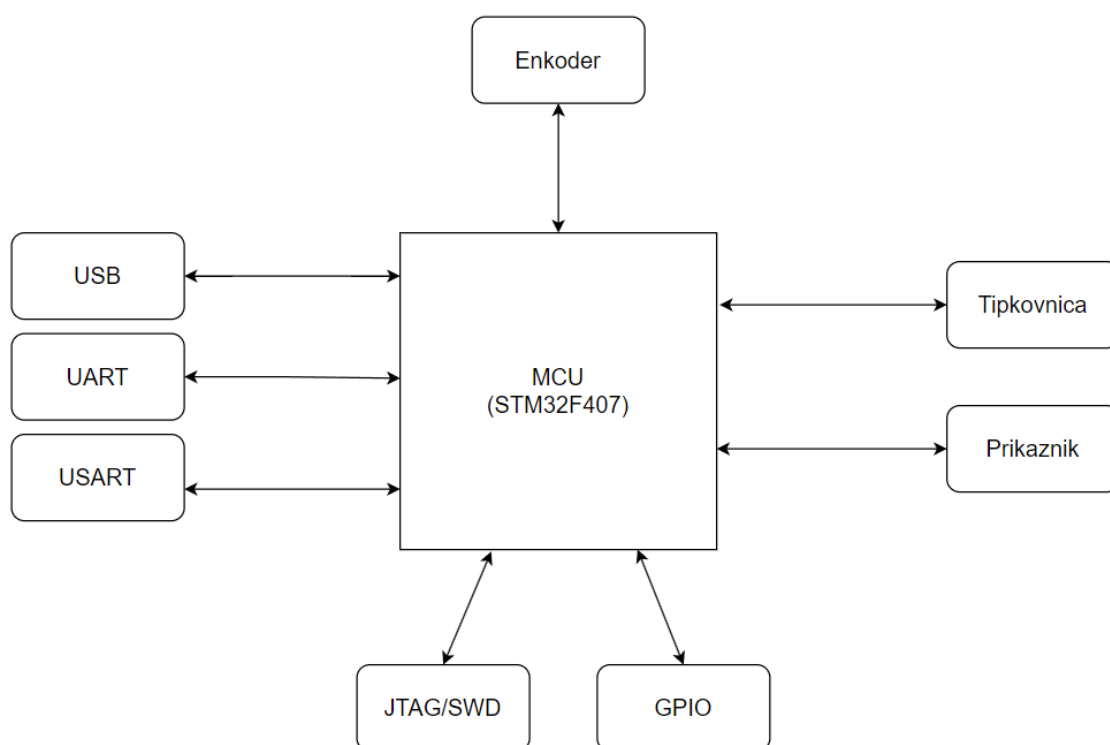
Slika 3.11 Ulazno-izlazni priključci

## 4. Programska podrška

Za računalni sustav opisan u prethodnom poglavlju, razvijena je programska podrška. Ona se može podijeliti na dva dijela:

1. Biblioteke za crtanje i pisanje po grafičkom prikazniku koristeći SPI sučelja
2. Programski kod koji se izvršava na mikrokontroleru koristeći FreeRTOS [13]

Cjelokupna programska podrška koja se izvršava na mikrokontroleru upravlja sučeljima prikazanim na slici 4.1. UART, USART i USB sučelja ostvaruje serijsku vezu s vanjskim uređajima, a JTAG/SWD konekcija korištena je za uhadavanje i ispitivanje sustava. Glavna namjena sustava je upravljanje frekvencijom programski definiranog radija. Pritom se unos frekvencije izvodi preko dodirnog zaslona i namjenskih tipki, a kontinuirana promjena izvodi se pomoću enkodera.



Slika 4.1 Blokovski prikaz sučelja mikrokontrolera

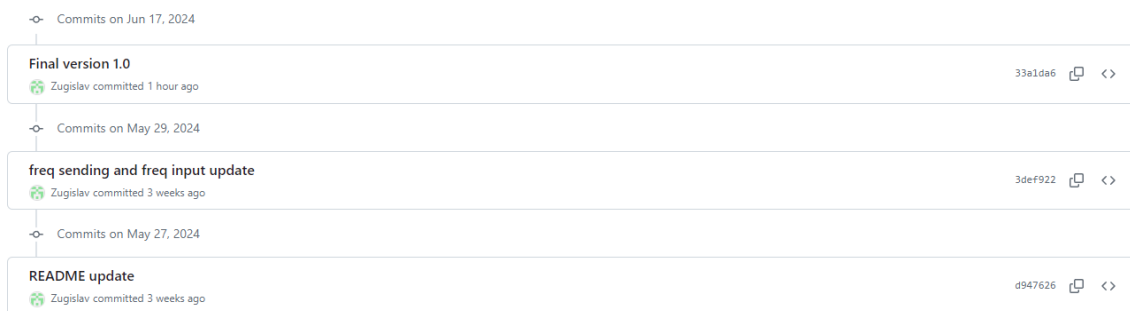


Programska podrška napisana je koristeći razvojno okruženje *STMicroelectronics STM32CubeMX 6.11.0* [6], a programski kod napisan je u C programskom jeziku koristeći *Visual Studio Code* [14] s ekstenzijom *STM32 for VSCode* [15]. Kod je preveden koristeći *GCC* [16] prevoditelj za ARM jezgru.

Tijekom izrade programske podrške korišten je Git sustav. Git je distribuirani sustav korišten za praćenje promjena koda u kodnoj bazi koji omogućuje suradnju s drugim programerima za održavanje i razvoj projekta. Sav kod dostupan je u Github repozitoriju [17] za korištenje. Projekt sadrži jednu granu kod koje se prilikom razvijanja određene komponente sustava kod prenosi u kodnu bazu. Nakon što su promjene u kodu napravljene, za prijenos svih promjena u glavnu kodnu bazu potrebno je izvršiti komadne u terminalu:

```
Git add .
Git commit -m „poruka“
Git push
```

Komande prenose promjene u kodu na glavnu kodnu bazu ako nije bilo konflikta s postojećim kodom te nastaje povijest verzija. Slika 4.2 prikazuje povijest verzija koda tijekom razvoja gdje svaka promjena ima svoju oznaku i poruku koju korisnik unosi kako bi naznačio trenutno stanje sustava.



Slika 4.2 Povijest verzija koda u kodnoj bazi

## 4.1. FreeRTOS

FreeRTOS (*Free Real-Time Operating System*) [13] je operacijski sustav za rad u stvarnom vremenu dizajniran za ugradbene sustave i mikrokontrolere. FreeRTOS sustav je poznat po svojoj maloj memorijskoj potrošnji, što ga čini posebno pogodnim za sustave s

ograničenim resursima. Ima široku prenosivost na različite sklopovske platforme s različitim mikrokontrolerskim arhitekturama poput ARM, RISC-V itd.

Dizajn FreeRTOS-a osigurava fleksibilnost i učinkovitost raspoređivanja zadataka u stvarnom vremenu. Središnji dio FreeRTOS sustava čini glavni raspoređivač zadataka (*Task scheduler*) koji omogućuje tri načina raspoređivanja:

- Preemptivno – zadaci višeg prioriteta izvršavaju se prekidanjem rada zadataka nižeg prioriteta
- Kooperativno – zadaci tijekom izvršavanja sami odlučuju kada će prepustiti procesorsko vrijeme drugim zadacima
- Hibridno – korištenje preemptivnog i kooperativnog načina raspoređivanja zadataka

Zadaci koji se izvršavaju imaju oznaku prioriteta izvršavanja, veličinu stoga i argumente koje primaju tijekom izvršavanja. Svaki zadatak može se naći u nekom od stanja:

- Spreman (*Ready*) – zadatak je spreman za izvođenje, no trenutno se ne izvodi i čeka u redu raspoređivača zadataka
- Izvodi se (*Running*) – zadatak se izvodi i koristi procesor za izvršavanje svog koda
- Blokiran (*Blocked*) – zadatak je privremeno zaustavljen zbog, primjerice, pristupanja zajedničkom resursu
- Suspendiran (*Suspended*) – zadatak je uklonjen iz redoslijeda izvršavanja i ne koristi resurse ako se ponovo ne aktivira
- Izbrisan (*Deleted*) – zadatak je izvršio svoj kod i ne postoji u sustavu te oslobađa memoriju za druge zadatke

Jedan od načina komunikacije zadataka je korištenjem redova (*Queue*). Queue djeluje kao FIFO (*First In, First Out*) struktura, gdje zadaci mogu postavljati poruke u red i dohvaćati ih. Osim komunikacije između zadatka putem redova, moguće je i komunikacija putem događaja (*Event*) koji međusobno obavještavaju zadatke o pojedinim događajima slanjem signala ili slanjem određene vrijednosti uz signal za događaj. Ovi mehanizmi omogućuju sinkronizaciju različitih zadataka omogućujući dijeljenje podataka bez sukoba.

Ako zadaci pristupaju zajedničkom memorijskom prostoru FreeRTOS koristi sučelje semafora i *mutex* objekata. Semafori mogu biti binarni ili brojeći, njihovo se stanje mijenja ako zadatak pokušava dohvatiti semafor u dostupnom stanju i odraditi operacije nad zajedničkim resursima, gdje pri izlazu iz regije pušta semafor za druge zadatke koji su u međuvremenu bili u blokiranom stanju. *Mutex* objekt je posebna vrsta binarnog semafora koji uključuju prioritarno nasljeđivanje, čime je spriječen problem prioritete inverzije do koje dolazi kada zadatak nižeg prioriteta blokira zadatak višeg prioriteta, budući da je zadatak nižeg prioriteta zauzeo zajednički resurs. Problem je rješiv prioritarnim nasljeđivanjem gdje zadatak nižeg prioriteta povisuje prioritet na razinu zadatka koji želi koristiti resurs čime sprječavamo blokiranje zadatka s visokim prioritetom. Ovi mehanizmi omogućuju sigurno dijeljenje resursa između zadataka sprečavajući istovremeni pristup i potencijalne sukobe.

Memorijska alokacija u FreeRTOS sustavu omogućava čak pet načina rada alokacije hrpe (*Heap*):

1. Heap 1 – najjednostavnija implementacija koja koristi statičku alokaciju. Izvodi se tijekom prevođenja koda i nakon toga se ne mijenja. Jednostavan i deterministički pristup zbog nepostojanja mehanizma za oslobađanje memorije ili defragmentacije.
2. Heap 2 – uvodi dinamičku alokaciju memorije pomoću slobodne liste tijekom izvršavanja zadataka. Kada se memorija alocira, heap 2 traži prvi dovoljno veliki blok memorije iz slobodne liste memorije. Problem nastaje pri oslobađanju memorije i ponovnim zauzimanjem koje može uzrokovati fragmentacije memorije.
3. Heap 3 – implementacija koja koristi standardne C funkcije poput *malloc* i *free* za upravljanje memorijom. Implementacija u sebi sadrži korištenje standardnog alokatora memorije koji dolazi s C bibliotekom. Najveći problem je nedeterminističko ponašanje u stvarnom vremenu budući da funkcije *malloc* i *free* nisu dizajnirane za sustave u stvarnom vremenu.
4. Heap 4 – koristi dinamičko alociranje memorije i oslobađanje memorije s mogućnosti spajanja slobodnih blokova memorije kako bi smanjila fragmentaciju memorije. Glavni nedostatak je veća složenost izvedbe i korištenje resursa, no učinkovita metoda za održavanje velikih slobodnih blokova memorije.

5. Heap 5 – nadograđuje heap 4 definiranjem neprekinutih memorijskih regija za korištenje dinamičkog alociranja memorije. Korisna metoda upravljanja memorije za sustave koji imaju memorijske regije s različitim karakteristikama podijele resursa.

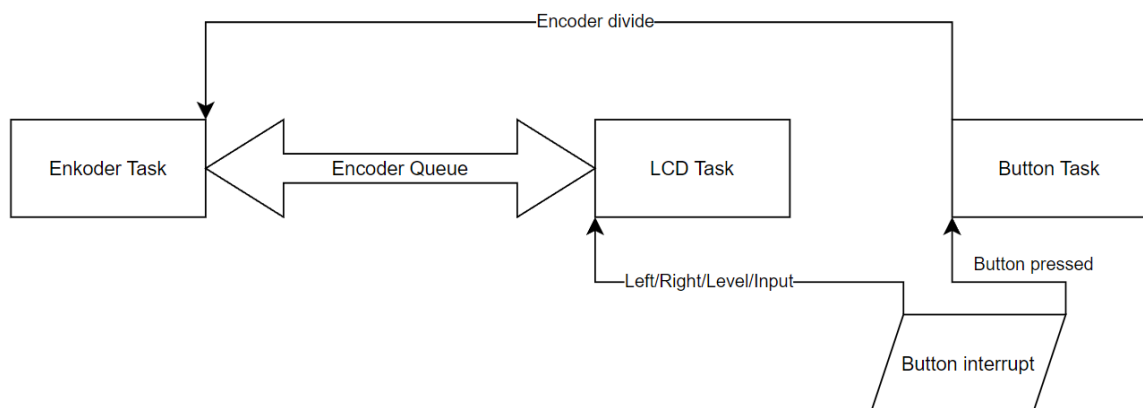
Heap 4 korištena je u radu kao memorijska alokacija.

## 4.2. Organizacija i opis zadataka u FreeRTOS sustavu

Kod koji se izvršava na mikrokontroleru unutar FreeRTOS sustava sastoji se od tri zadatka:

- Enkoder Task
- LCD Task
- Button Task

Svaki od zadataka zadužen je za upravljanjem različitim sustavima na pločici. Slika 4.3 opisuje relaciju zadataka i njihovo međusobno komuniciranje.

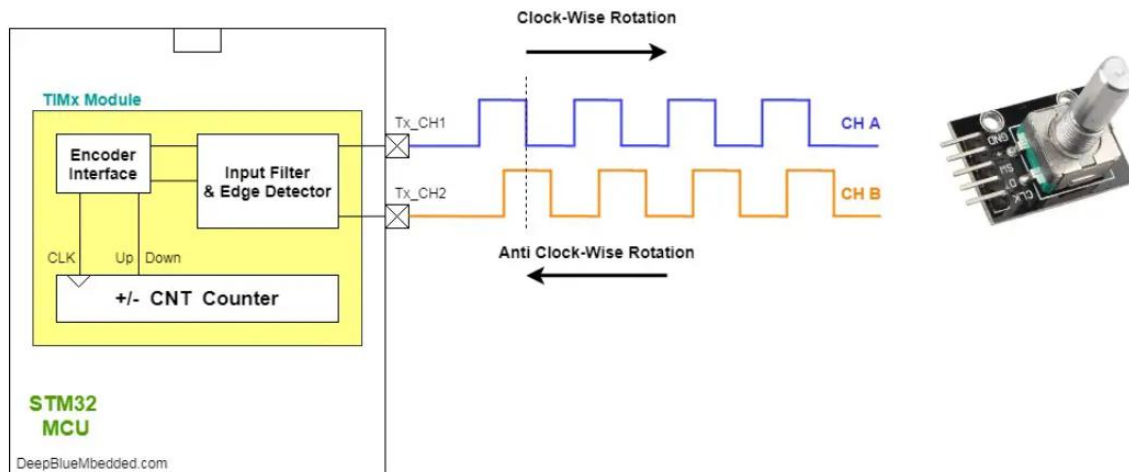


Slika 4.3 Raspored i komunikacija zadataka u sustavu FreeRTOS

### 4.2.1. Enkoder zadatak

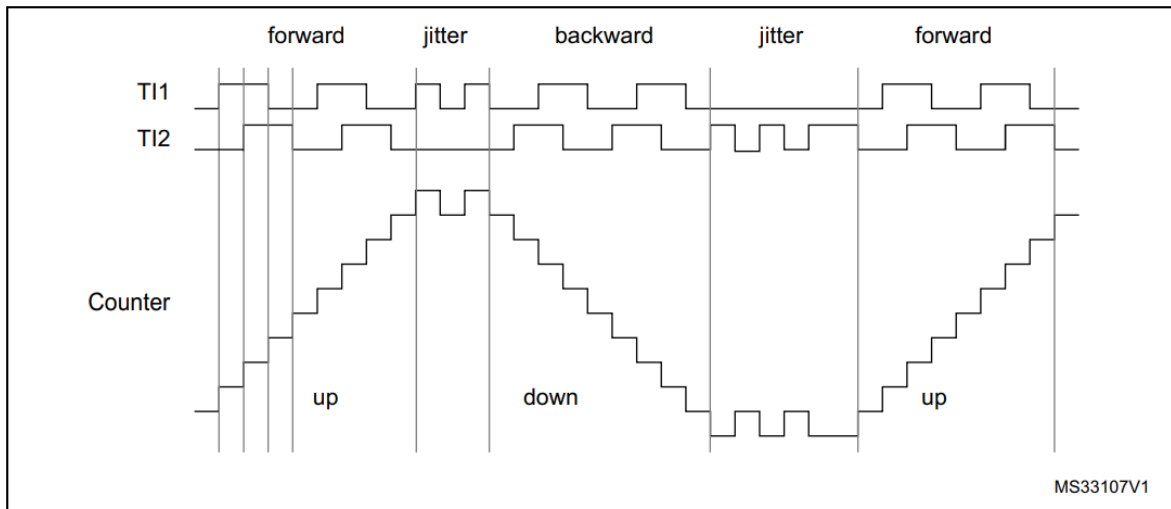
Enkoder zadatak sastoji se od poziva funkcije `void Encoder_init()` u kojoj se uspostavljaju parametri enkodera i brojilo TIM1 postavljen u STMCubeMX u Encoder modu. Kanali A i B enkodera povezani su s kanalima T1 i T2 na brojilo mikrokontrolera.

Brojilo se povećava ili smanjuje nakon obrade signala, čime se određuje broj pulseva i smjer signala. Slika 4.4 prikazuje povezivanje enkodera i brojila mikrokontrolera.



Slika 4.4 Povezivanje enkodera i brojila mikrokontrolera [18]

Ovisno o sekvenci, brojilo broji prema gore ili dolje što određuje bit DIR u TIM1\_CR1 registru. DIR bit se izračunava na svaki prijelaz signala T1 i T2. Slika 4.5 pokazuje ponašanje enkoderskog sučelja pri okretanju enkodera. Brojilo kontinuirano broji između 0 i automatski unesene vrijednosti u TIM1\_ARR registru koja je postavljena na maksimalnu vrijednost od 65535. U ovom načinu rada, brojilo se automatski mijenja u skladu s brzinom i smjerom inkrementalnog enkodera. Funkcija `void Encoder_count(encoderHandle *encoder)` broji stanje enkodera i njegovu promjenu u vremenu. Nakon brojanja promjene stanja, vrijednosti se šalju koristeći red `encoderValueHandle` LCD zadatku koji potom mijenja frekvenciju. Ovisno o stanju `Encoder_Divide` varijable koju mijenja Tipka zadatak omogućeno je da brojilo bilježi preciznije promjene frekvencije tako što promjenu enkodera dijeli s četiri kako bi korisnik preciznije definirao željenu frekvenciju.



Slika 4.5 Opis rada brojila u enkoder načinu rada [5]

## 4.2.2. Tipka zadatak i ulazno-izlazno prekidi

Tipka zadatak sastoji se od primanja događaja koji se šalju prilikom prekida. Prekidi su omogućeni na tipkama F1 i F2 te tipkom koja se nalazi na enkoderu.

Tipke F3-F6 na prekid šalju LCD zadatku događaj s brojčanom vrijednošću kako bi zadatak ovisno o prekidu mogao registrirati koja je tipka pritisnuta. Brojilo TIM2 zaduženo je za obradu lažnih signala. Ovi signali mogli bi se ukloniti sklopovski korištenjem otpornika i kondenzatora. Međutim, u radu je korišteno njihovo programsko uklanjanje postavljanjem stanja signala koje se mijenja ako tipka nije pritisnuta dulje od 50 ms. Nakon što je tipka pritisnuta duže od 50 ms, pali se svjetleća dioda iznad tipke koja korisniku označava da je ona pritisnuta.

Za ispis teksta koristeći micro B USB vezu s vanjskim uređajima, zadatci koriste funkciju `void printSerial(const char* msg, ...)` unutar koje se koristi `osMutexId_t printMutexHandle`, budući da zadaci pristupaju zajedničkom sučelju.

## 4.2.3. LCD zadatak

LCD zadatak je najveći i središnji zadatak koji je zaslužan za podršku prikaza frekvencije, prikaz tipkovnice te interakciju s korisnikom. Slika 4.6 prikazuje tijekom izvođenja LCD zadatka. Zadatak započinje inicijalizacijom objekta za upravljanje

prikaznikom. Nakon inicijalizacije varijabli vezanih za prikaznik, korištena je biblioteka za prikaz zaslona [19] koju je osmislio *Daniel Martensson*. Za podršku dodirnog zaslona prilagođena je biblioteka [20] koju je osmislio korisnik *afiskon*. U poglavlju 4.2.4 opisana je cijela prilagođena biblioteka za rad s prikaznikom i podrška za dodirni zaslon.

Za inicijalizaciju i dodjeljivanje sučelja za rad s prikaznikom, funkciji `void STM32_PLC_LCD(SPI_HandleTypeDef *lcdSpi, GPIO_TypeDef *LCD_CS_PORT, uint16_t LCD_CS_PIN, GPIO_TypeDef *LCD_DC_PORT, uint16_t LCD_DC_PIN, GPIO_TypeDef *LCD_RST_PORT, uint16_t LCD_RST_PIN)` prosljeđuje se SPI sučelje za rad s prikaznikom te ulazno-izlazni priključci za omogućavanje LCD kontrolera, slanje komadnih podataka i resetiranje prikaznika.

Funkcija `Void STM32_PLC_LCD_Show_Main_Frame(uint8_t *frame_id)` prikazuje glavni zaslon koji se sastoji od prikazane inicijalne frekvencije od 100 MHz i 4 tipke za rad korisnika. Ako je tipka na zaslonu pritisnuta, pozivom funkcije `bool ILI9341_TouchGetCoordinates(uint16_t* x, uint16_t* y)` ili jedne od četiriju tipki na pločici, događaj primljen korištenjem funkcije `xTaskNotifyWait()` uzrokuje sljedeći niz događaja:

- Pritisnuta strelica ulijevo ( ← ) ili tipka s oznakom F3 – dolazi do pomicanja u lijevu stranu strelice koja pokazuje jednu od 10 znamenki frekvencije, počevši od koje će biti ažurirana frekvencija tijekom okretanja enkodera. Pritom dolazi do povećanja pokazivača na vrijednost `i` i poziva funkciju `printPointerToValue()`.
- Pritisnuta strelice udesno ( → ) ili tipka s oznakom F5 – dolazi do pomicanja u desnu stranu strelice koja pokazuje jednu od 10 znamenki frekvencije, počevši od koje će biti ažurirana frekvencija tijekom okretanja enkodera. Pritom dolazi do povećanja pokazivača na vrijednost `i` i poziva funkciju `printPointerToValue()`.
- Pritisnuta tipka LEVEL ili tipka s oznakom F4 – dolazi do postavljanja svih znamenki desno od trenutne lokacije na koju pokazuje strelica na 0 pozivajući funkciju `LevelALLRightOfPointerToZero()`.

- Pritisnuta tipka INPUT ili tipka s oznakom F6 – odlazak s glavnog zaslona na zaslon s tipkovnicom, koji omogućuje ručni unos frekvencije pozivom funkcije `STM32_PLC_LCD_Show_Numpad_Frame()`.

Nakon provjere za dodir zaslona i pritisnute tipke, slijedi čitanje iz reda `encoderValueHandle` koristeći FreeRTOS funkciju `xQueueReceive()`, pri čemu funkcija za primanje nove vrijednosti enkodera čeka 10 ms. Promjena frekvencije se potom događa ulijevo od strelice koja prikazuje na jednu od 10 znamenki frekvencije. Nakon svake promjene frekvencije prikazuje se nova frekvencija sustava na zaslonu koristeći funkciju `printEncoderValue()`. Postavljanje frekvencije sustava moguće je do maksimalne vrijednosti cjelobrojnog broja od 32 bita (4,294,967,295), koja se za prikaz na zaslonu prikazuje kao lista od 10 članova decimalnih brojeva. Ako frekvencija pređe maksimalnu vrijednost, zaslon poprima crvenu boju na 5 sekundi, s ispisom OUT OF RANGE bijelim slovima. Potom se na zaslonu nalazi glavni prikaz s vrijednošću znamenke gigaherca postavljene na 0. Koristeći serijsku vezu na DE9 konektoru, frekvencija se šalje radio prijamniku u zadanom formatu. Funkcija `calculateSendFrequency()` pretvara listu znamenki u cijeli broj, zatim pretvara frekvenciju u varijablu `uint32_t delta` koristeći format:

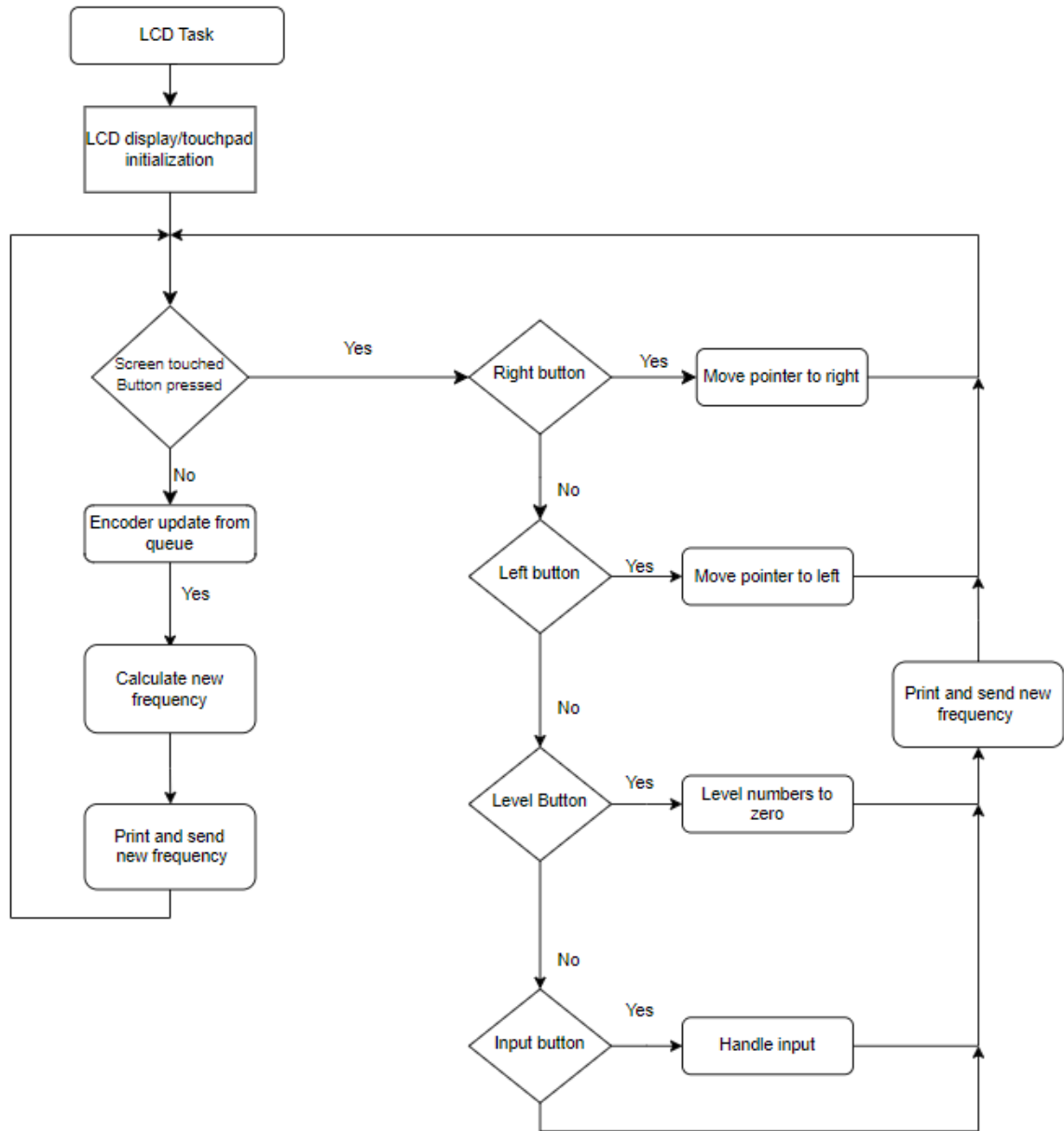
```
uint32_t delta = floor(frequency * pow(2,N) / FS + 0.5);
```

Za pretvorbu frekvencije koristimo konstante  $N$  i  $FS$ .  $FS$  predstavlja frekvenciju uzrokovanja programski definiranog radija od 50MHz koju je potrebno pretvoriti u Hz. Vrijednost `delta` pohranjuje se u listu od četiri bajta koji se šalju prijamniku. Ovi bajtovi se šalju u petlji koja čeka zastavicu u statusnom registru UART-a i nakon toga puni podatkovni registar bajtom  $c$ .

Pritiskom tipke F6 ili odabirom tipke INPUT, poziva se funkcija za unos frekvencije, `STM32_PLC_LCD_Show_Numpad_Frame()`, koja iscrtava novi zaslon s brojevima 0-9, decimalnom točkom i odabirom frekvencije u GHz, MHz i kHz. Korisnik proizvoljno unosi brojke koje se ispisuju na zaslon uz pomoć funkcije `printInputValue()`. Nakon što korisnik unese željenu frekvenciju, pritiskom na tipku za kHz, MHz ili GHz poziva se funkcija `bool checkInputValue()` za provjeru unosa. Ako korisnik unese ispravnu vrijednost nove frekvencije, ona se prikazuje funkcijom `printNewEncoderValue()` te, nakon 2 sekunde, dolazi do povratka na glavni zaslon s novom frekvencijom koja će biti poslana programsko definiranom radiju.



Prilikom unosa frekvencije koju nije moguće prikazati poziva se funkcija `void printInvalidInput()`, koja ispisuje tekst `INVALID INPUT` te zahtjeva novi unos frekvencije.



Slika 4.6 Tijek izvođenja LCD zadatka

## 4.2.4. Biblioteka za crtanje i pisanje na grafičkom prikazniku

Biblioteka za crtanje i pisanje na grafičkom prikazniku dizajnirana od Daniela Martenssona [19] sastoji se dvije datoteke ILI9341.h, ILI9341.c. Datoteka ILI9341.c sastoji se od gotovih funkcija poput void ILI9341\_send\_command(uint8\_t com), a komande koje se mogu poslati definirane su u zaglavlju ILI9341.h. U radu su korištenje sljedeće funkcije:

- void ILI9341\_send\_command(uint8\_t com) – slanje komandi ILI9341 kontroleru preko SPI sučelja. Komande su definirane u ILI9341.h zaglavlju
- void ILI9341\_send\_data(uint8\_t data) - slanje podataka kao argumenata komandi
- void ILI9341\_set\_rotation(uint8\_t rotate) – određivanje rotacije zaslona, postavljanje koordinata zaslona te postavljanje orijentacije teksta za ispis
- void ILI9341\_fill\_screen(uint16\_t color) – ispunjavanje cijelog zaslona željenom bojom, pri čemu se boje i njihove vrijednosti nalaze u zaglavlju ILI9341.h
- ILI9341\_draw\_horizontal\_line(int16\_t x, int16\_t y, int16\_t h, uint16\_t color) i ILI9341\_draw\_vertical\_line(int16\_t x, int16\_t y, int16\_t h, uint16\_t color) – ispis vertikalnih i horizontalnih linija koje koriste argumente x i y za označavanje početka linije i h za duljinu linije koja će biti nacrtana u boji definiranoj argumentu color
- ILI9341\_fill\_rect(int16\_t x0, int16\_t y0, int16\_t x1, int16\_t y1, uint16\_t color) – prikazuje pravokutnik ispunjen željenom bojom, pri čemu su x0 i x1 početak i kraj pravokutnika na x osi, a y0 i y1 početak i kraj pravokutnika na y osi
- void ILI9341\_print\_text(char text[], int16\_t x0, int16\_t y0, uint16\_t font\_color, uint16\_t background\_color, uint8\_t font\_size) – ispis teksta na zaslon, pri čemu je prvi argument

funkcije pokazivač na tekst koji se ispisuje,  $x_0$  i  $y_0$  lokacija na zaslonu, a preostali argumenti određuju boju slova, pozadinsku boju te veličina fonta slova

Podrška za dodir zaslona prilagođena je projektu koristeći biblioteku *stm32-ili9341* [20] korisnika *afiskon*. Ostvaruje se definiranjem konstanti u *ILI9341\_touch.h* datoteci. Korisnik povezuje SPI 3 sučelje za rad te definira ulazno-izlazne priključke za ostvarivanje dodira zaslona.

```
#define ILI9341_TOUCH_SPI_PORT      hspi3
extern SPI_HandleTypeDef ILI9341_TOUCH_SPI_PORT;
#define ILI9341_TOUCH_IRQ_Pin      T_IRQ_Pin
#define ILI9341_TOUCH_IRQ_GPIO_Port T_IRQ_GPIO_Port
#define ILI9341_TOUCH_CS_Pin       LCD_CS_Pin
#define ILI9341_TOUCH_CS_GPIO_Port LCD_CS_GPIO_Port
```

Jedina funkcija spomenute biblioteke je:

- `bool ILI9341_TouchGetCoordinates(uint16_t *x, uint16_t *y)` – zabilježava u argumente funkcije koordinate dodira zaslona.

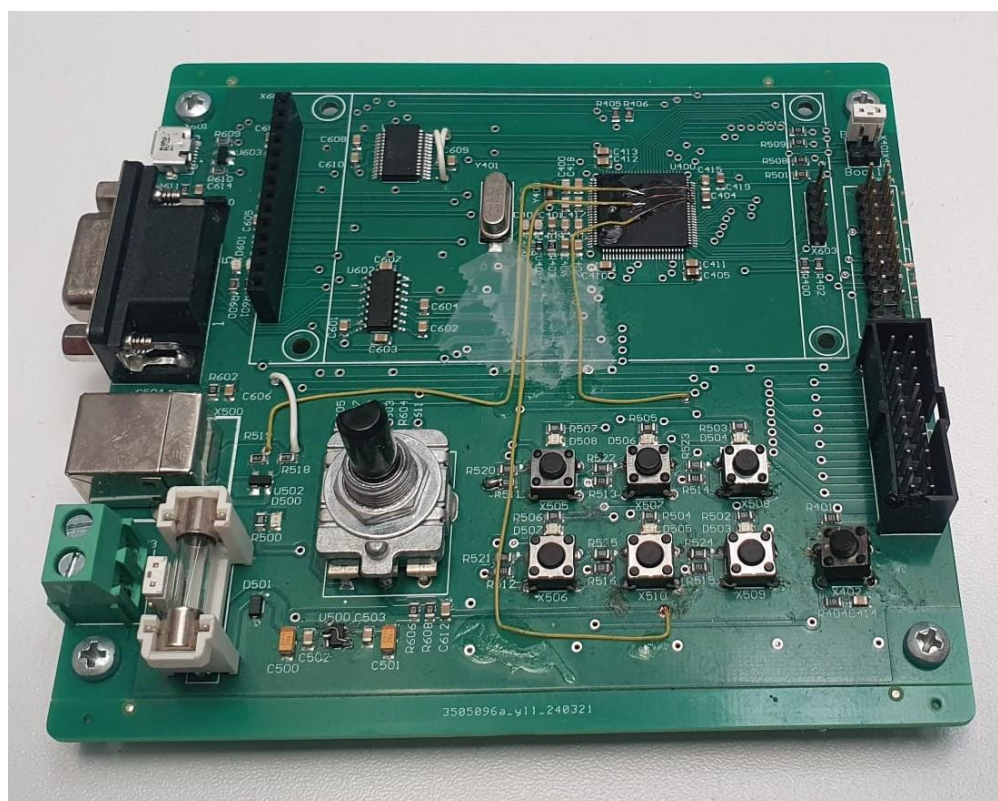
Za čitanje točne koordinate 16 puta se uzorkuje čitanje  $x$  i  $y$  koordinate koje se pohranjuju u zbroj vrijednosti  $x$  i  $y$ . Potom se vrijednost dijeli sa 16 i pohranjuje u argumente funkcije koji su proslijeđeni kao pokazivači. Ako se koordinate dodira zaslona dogode unutar definiranih vrijednosti za tipku, poziva se odgovarajuća funkcija.

## 5. Implementacija sustava

U ovom poglavlju opisan je prvi radni prototip tiskane pločice razvijenog sustava, zajedno s potrebnim izmjenama u sklopovlju. Također, opisan je rad s korisničkim sučeljem.

### 5.1. Prototip tiskane pločice

Cjelokupno sklopovlje računalnog sustava implementirano je na dvoslojnoj tiskanoj pločici dimenzija 116.3 mm x 100 mm. Korištene su SMD (*Surface Mount Device*) komponente veličine 0803. Prilikom razmještaja komponenti posebna pozornost posvećena je rasporedu elemenata koji čine korisničko sučelje. Tiskana pločica i montaža napravljene su u tvrtki JLC PCB [21]. Slika 5.1 prikazuje fizičku realizaciju sustava s izmjenama koje čine reviziju 2.

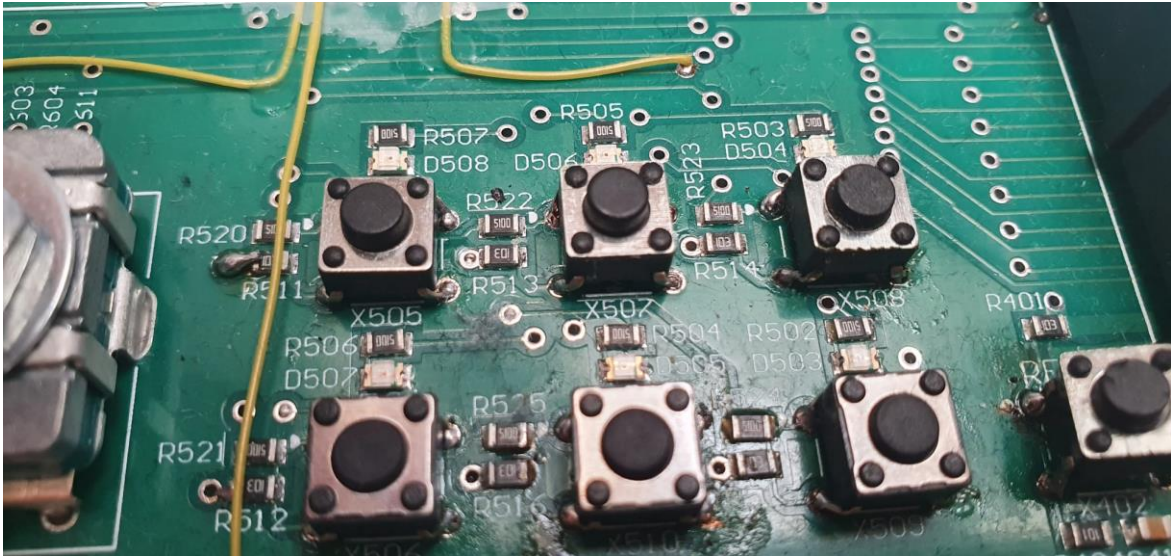


Slika 5.1 Prototip pločice





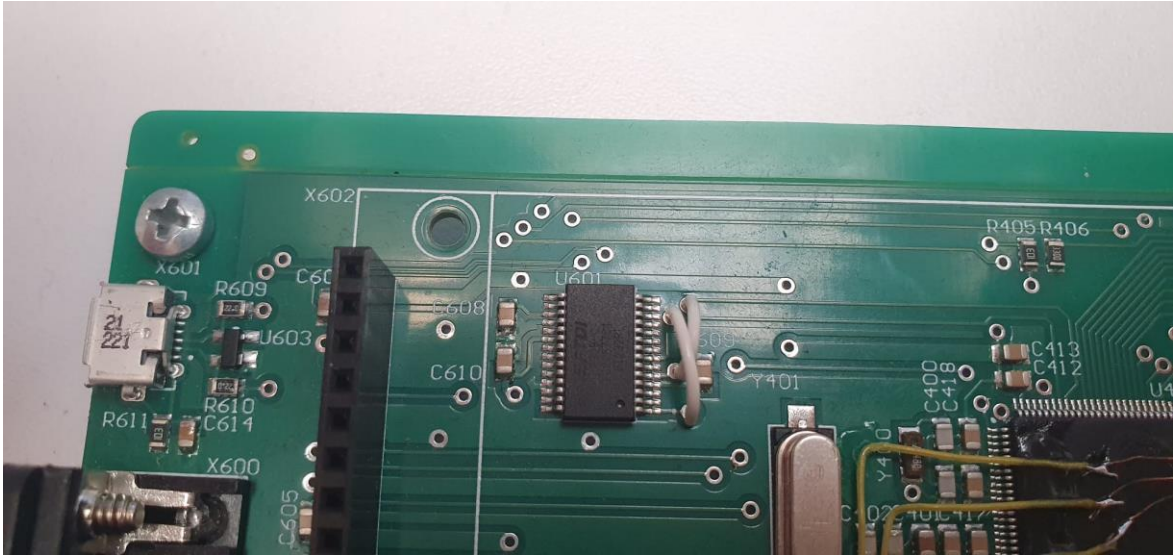
Prilikom ispitivanja i uhodavanja sustava uočeno je da su tipke greškom zarotirane za 90 stupnjeva. Slika 5.4 prikazuje ispravljen položaj tipki.



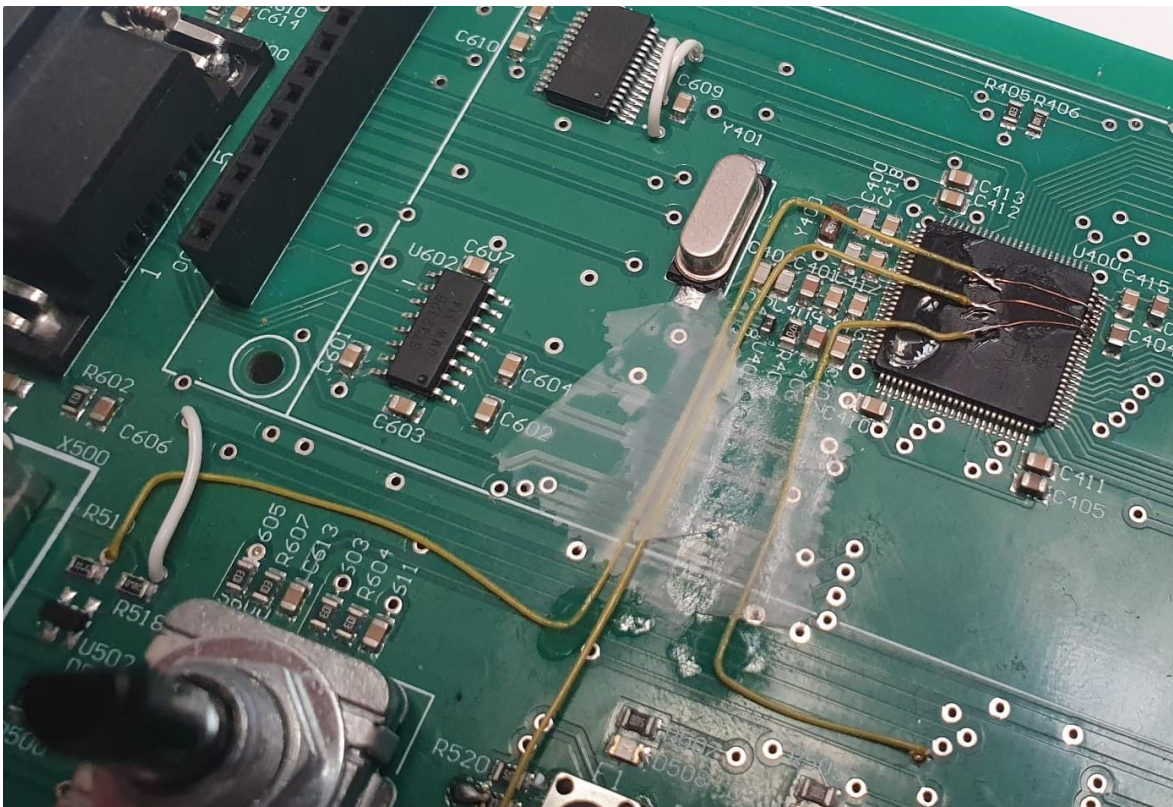
Slika 5.4 Izmjena u spajanju tipki

Prilikom rada i spajanja serije na micro B USB kabel s osobnim računalom, konekcija se nije mogla uspostaviti. Prema dokumentacije sklopa FT232RL [11] u poglavlju 8.4 linije RX i TX između sklopa i mikrokontrolera su nepravilno spojene, zajedno sa napajanjem sklopa FT232RL. U drugoj reviziji sklopa napajanje s micro B USB priključeno je na VCC priključak kako bi sklop mogao naponske razine USART sučelja adekvatno pretvoriti u naponske razine rada USB sučelja. Slika 5.5 prikazuje zamjenu RX i TX linija za spajanje mikrokontrolera i sklopa FT232RL.

Za ostvarivanje USB konekcije i povezivanje enkodera na brojilo TIM1, zaključeno je kako su USB Data Plus i Data Minus priključeni na pogrešne priključke mikrokontrolera zajedno sa spajanjem enkoderovih kanala 1 i 2 na priključke brojila TIM1. Pogreške su uočene tijekom izrade programske podrške te su uklonjene na prototipu pločice kao što prikazuje slika 5.5



Slika 5.5 Korekcija povezivanja mikrokontrolera i sklopa FT232RL

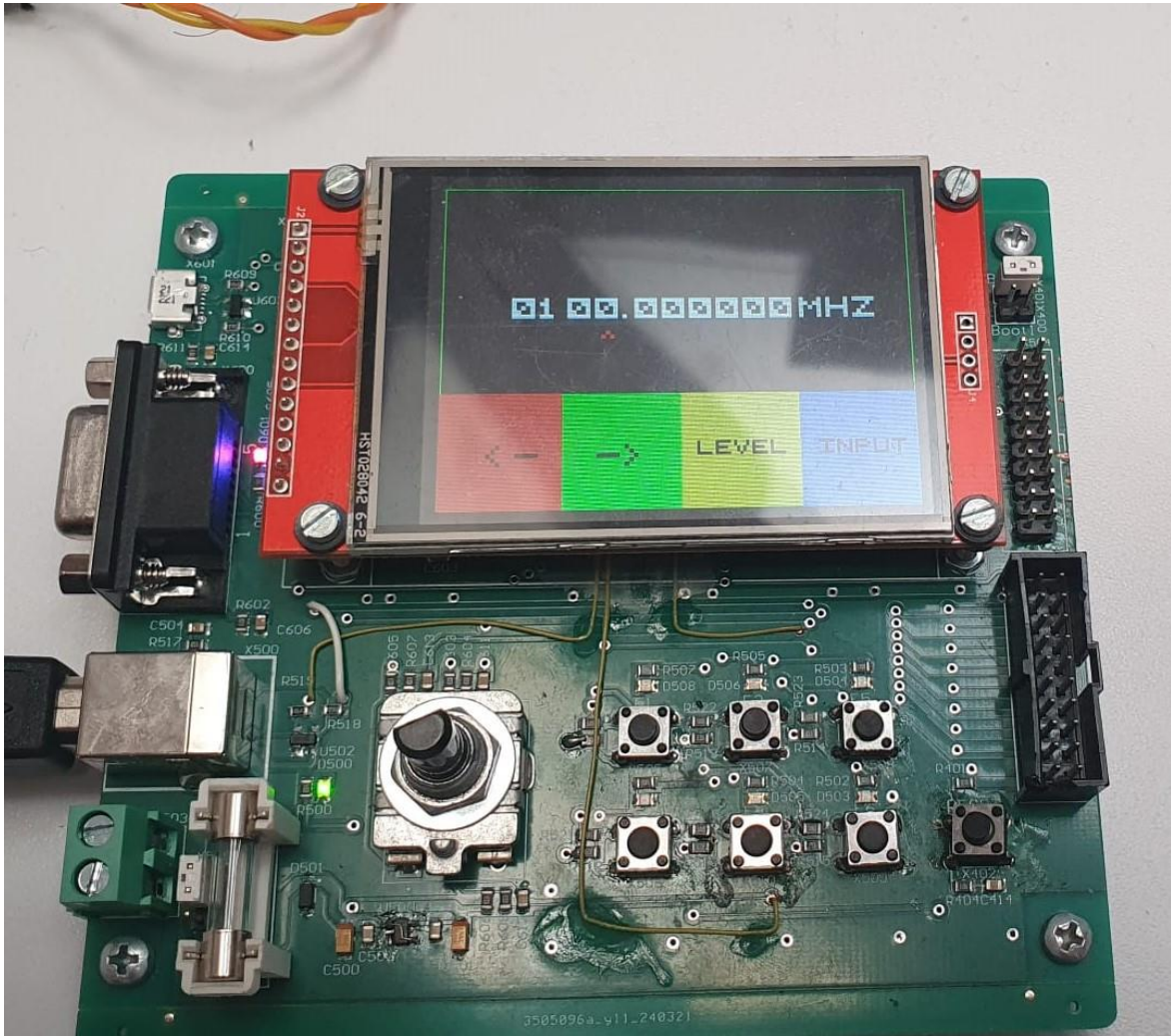


Slika 5.6 Korekcija povezivanja podatkovnih linija USB konektora i enkodera s mikrokontrolerom



### 5.3. Rad s računalnim sučeljem

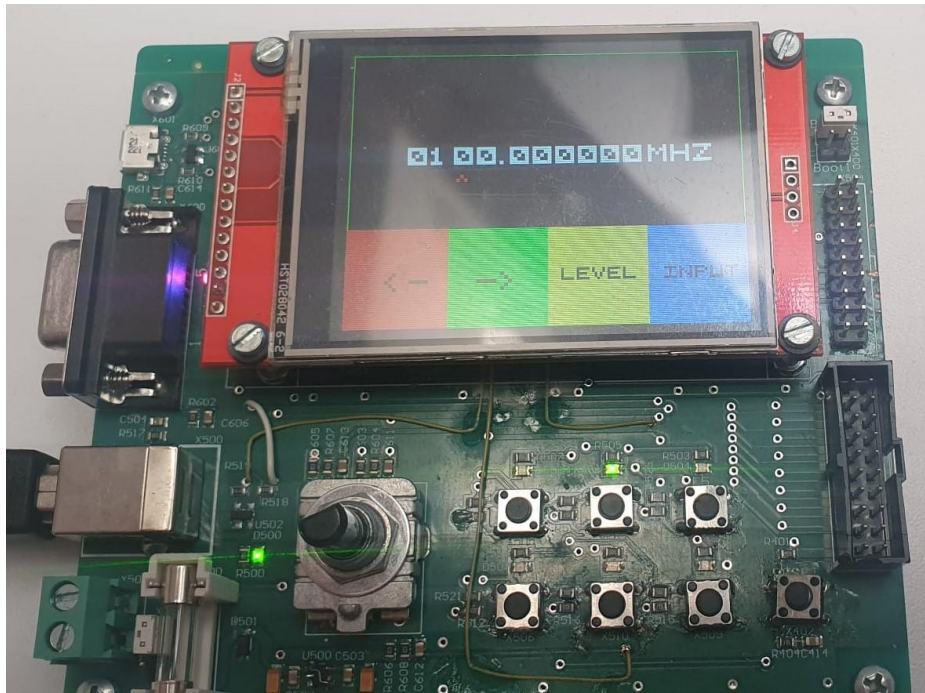
Interakcija za rad s računalnim sučeljem ostvaruje se preko rotacijskog enkodera, prikaznika s podrškom za dodir i četiri tipke F2-F6. Slika 5.7 prikazuje osnovno stanje sustava prilikom paljenja, s početnom frekvencijom od 100MHz.



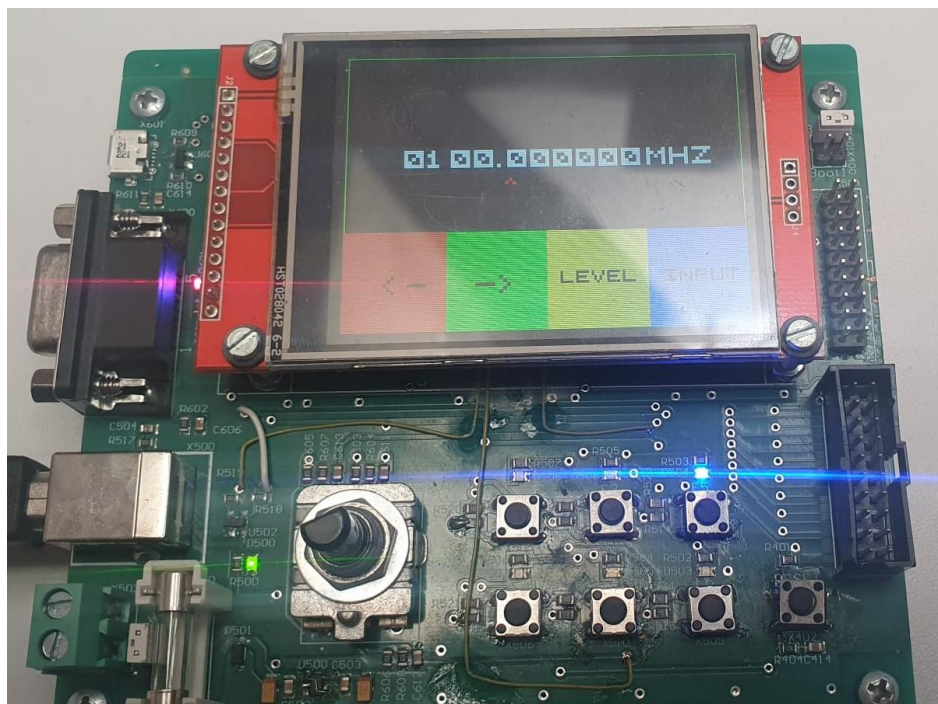
Slika 5.7 Početno stanje sustava

Korisnik nakon paljenja sustava može koristiti tipke na zaslonu za pomicanje strelice u lijevu ili desnu stranu kako bi prilikom okretanja enkodera ažurirao frekvenciju. Slika 5.8 prikazuje pomicanje strelice u lijevu stranu nakon pritiska tipke F3 ili odabira crvene tipke na zaslonu. Slika 5.9 prikazuje pomicanje strelice u desnu stranu nakon pritiska tipke F5 ili odabira zelene tipke na zaslonu.



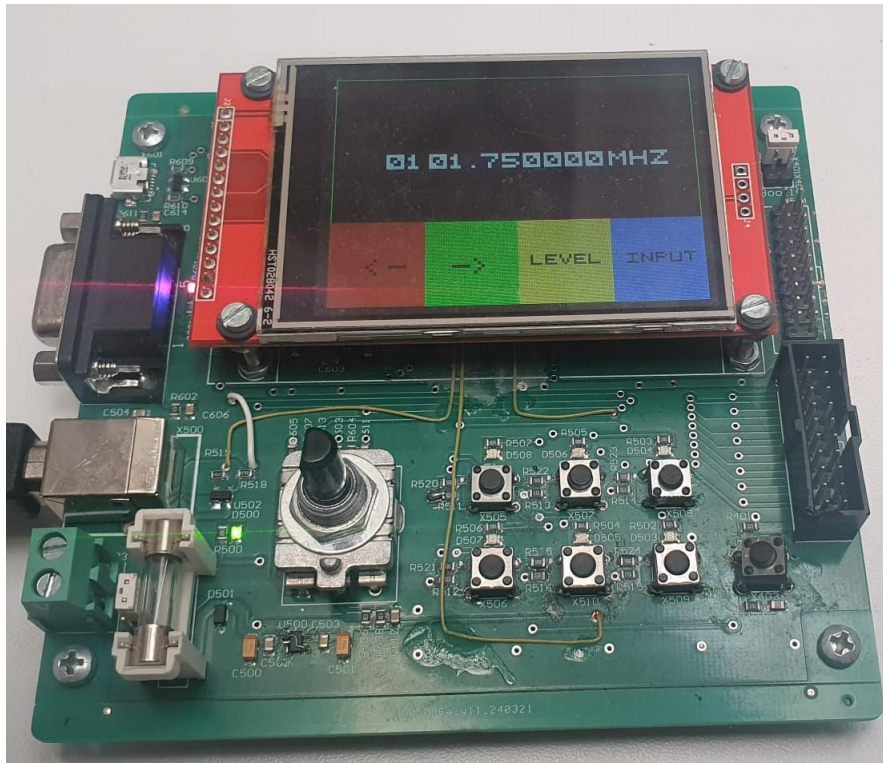


Slika 5.8 Pomicanje strelice ulijevo



Slika 5.9 Pomicanje strelice udesno

Prilikom zakretanja enkodera u smjeru kazaljke na satu ostvaruje se povećanje frekvencije počevši od indeksa strelice na frekvenciji prema ulijevo. Suprotno tome, okretanjem enkodera u kontra smjeru kazaljke na satu ostvaruje smanjivanje frekvencije do indeksa strelice. Korisnik za ostvarivanje preciznijeg izračuna nove frekvencije može pritisnuti osovinu enkodera kako bi ostvario preciznije dobivanje željene frekvencije. Korisnik može upravljati frekvencijom u rasponu od 1 Hz do 4.29 GHz. Slika 5.10 prikazuje promjenu frekvencije prilikom okretanja enkodera u smjeru kazaljke na satu.

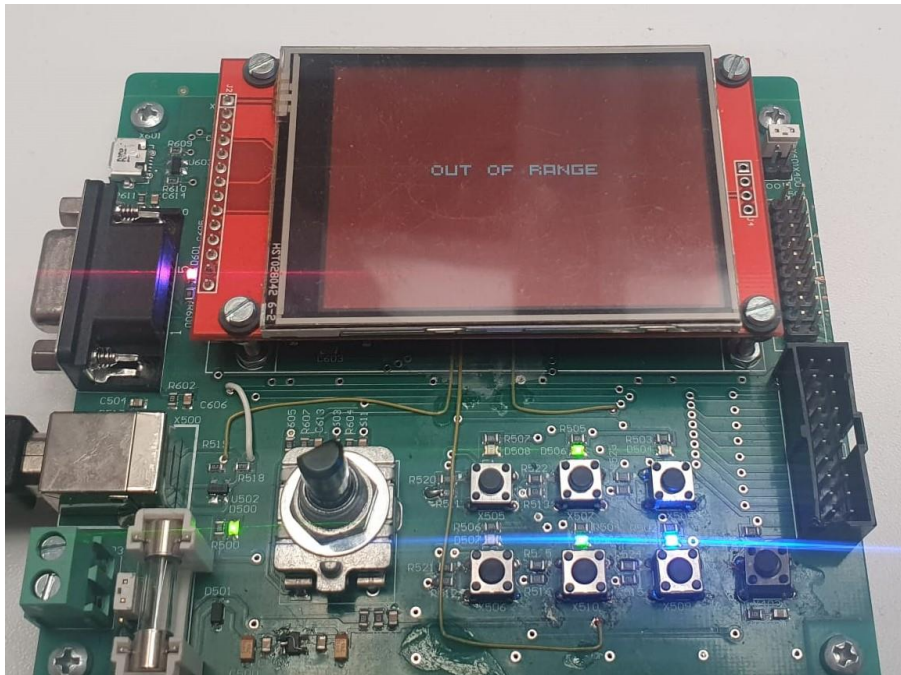


Slika 5.10 Rotiranje enkodera

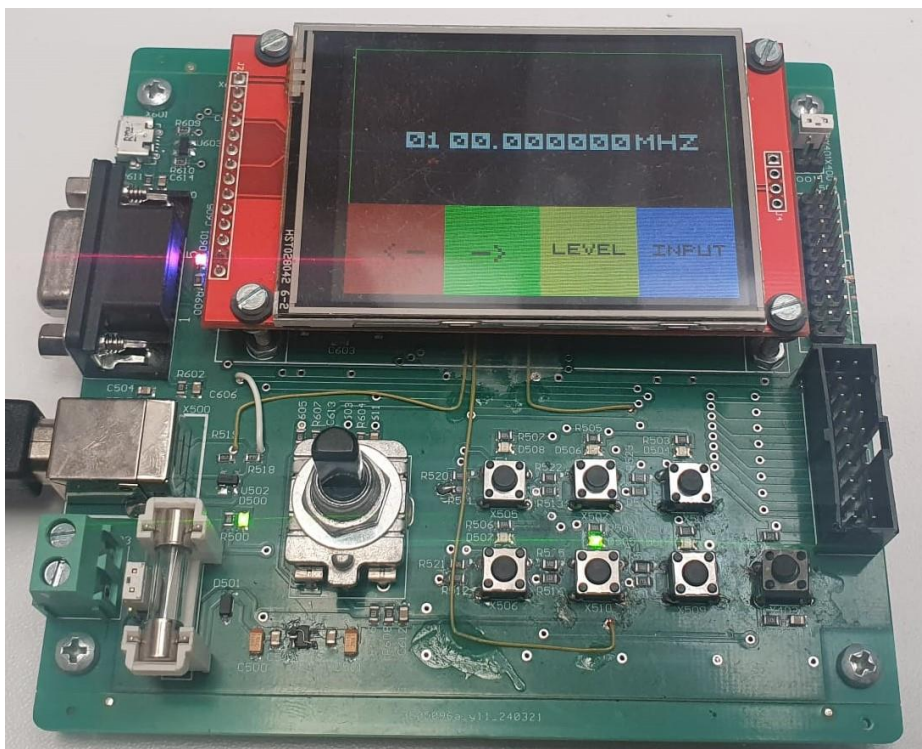
Korisnik prilikom okretanja enkodera može ostvariti vrijednost iznad maksimalno definirane frekvencije sustava nakon čega se prikazuje zaslon na slici 5.11, koji označava korisniku da je prekoračio maksimalnu vrijednost. Nakon 5 sekundi sučelje briše vrijednost u području GHz postavljajući je u nulu dok ostale znamenke frekvencije ostaju netaknute.

Slika 5.10 prikazuje stanje frekvencije od 101.75 MHz, ako korisnik pomakne strelicu prikaza na stotice MHz, pritiskom tipke F4 ili odabirom tipke LEVEL na zaslonu događa se izjednačavanje svih brojki desno od strelice na vrijednost nula. Strelicu je moguće pomaknuti van GHz područja i sve vrijednosti sustava postaviti na vrijednost nula. Slika 5.12 prikazuje stanje sustava nakon odabira tipke LEVEL.





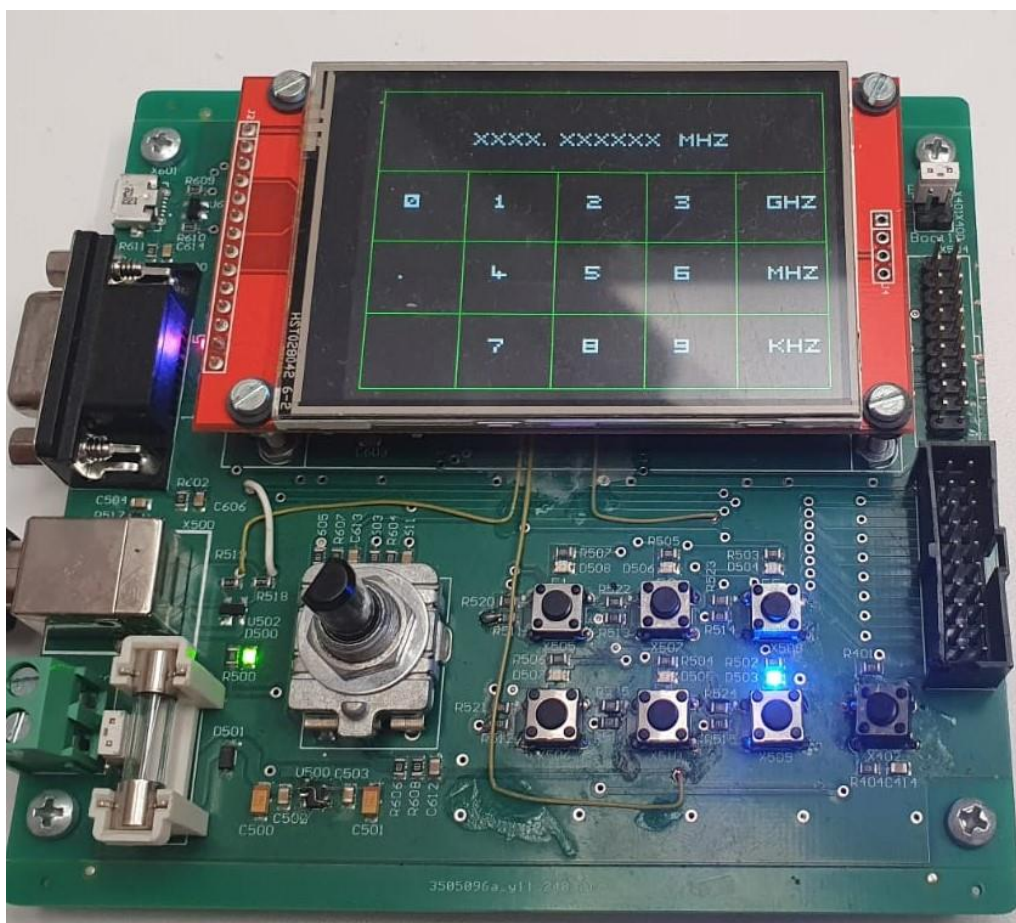
Slika 5.11 Prikaz sučelja prilikom unosa prevelike frekvencije



Slika 5.12 Promjera frekvencije korištenjem tipke LEVEL

Ako korisnik poželi unijeti frekvenciju bez okretanja enkodera moguće je pritisnuti tipku F6 ili odabrati tipku INPUT na zaslonu. Slika 5.13 prikazuje novi zaslon s tipkovnicom za unos frekvencije, točkom za označavanje decimalne točke i tipkama za GHz, MHz i kHz koje se koriste za potvrdu unesene frekvencije.

Korisnik unosi željenu frekvenciju pritiskom na brojkke koje se s pritiskom ispisuju s lijeva na desno. Ispravnost unesene frekvencije ispituje se prije prihvaćanja. Ako je unos neispravan, sučelje će na mjestu prikaza nove frekvencije ispisati INVALID INPUT zahtijevajući ponovni unos frekvencije. Nakon ispravnog unosa nove frekvencije, prikazat će se sučelje sa slike 5.14 te će se frekvencija ispisati na glavnom zaslonu sučelja.



Slika 5.13 Prikaz sučelja tijekom unosa željene frekvencije





Slika 5.14 Prikaz sučelja nakon uspješnog unosa željene frekvencije

## Zaključak

U ovom radu razvijen je ugradbeni računalni sustav temeljen na mikrokontroleru STM32F407. Sustav se sastoji od tiskane pločice na kojoj se nalazi mikrokontroler STM32F407, enkoder PEC11R, LCD prikaznik s ILI9341 kontrolerom za prikaz i XPT2046 sklopom za dodir te USB micro B, USB tip B i DE9 konektor. S pomoću SWD i JTAG sučelja moguće je uhodavati i ispitivati rad sustava. Na pločici se nalazi šest tipki i za interakciju sa sustavom, iznad kojih se nalaze svjetleće diode. Za postojeći sustav napisana je i programska podrška korištenjem STMCubeMX s FreeRTOS zadacima koju je moguće dohvatiti preko GitHub repozitorija. Također, dana su uputstva za rad sa sustavom. Korišten LCD prikaznik pokazao se pogodnim za izvedbu korisničkog sučelja programski definiranog radija. Kompaktnost upravljanja postignuta je primjenom dodirnog zaslona. Sustav je moguće unaprijediti dodavanjem memorijske kartice te povezati s drugim sustavima koristeći slobodne ulazno-izlazne priključke.

## Literatura

- [1] STM, „STM32F405xx STM32F407xx datasheet“, STMicroelectronics (kolovoz, 2020) Poveznica: <https://www.st.com/resource/en/datasheet/dm00037051.pdf> ; pristupljeno: listopad 2023.
- [2] „ILI9341 datasheet“, ILITEK, Poveznica: <https://www.digikey.com/htmldatasheets/production/1640716/0/0/1/ili9341-datasheet.html> ; pristupljeno travanj 2024.
- [3] „XPT2046 datasheet“, Xptek (svibanj, 2007), Poveznica <https://grobotronics.com/images/datasheets/xpt2046-datasheet.pdf> ; pristupljeno svibanj 2024.
- [4] „PEC11R Series datasheet“, Bourns (svibanj, 2018), Poveznica: <https://eu.mouser.com/datasheet/2/54/PEC11R-777457.pdf> ; pristupljeno siječanj 2024.
- [5] “STM32F407/417 Reference manual”, STMicroelectronics (lipanj, 2024), Poveznica: [https://www.st.com/resource/en/reference\\_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf](https://www.st.com/resource/en/reference_manual/rm0090-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf) ; pristupljeno ožujak 2024.
- [6] STM32CubeMX, STMicroelectronics, <https://www.st.com/en/development-tools/stm32cubemx.html>
- [7] Dejan Nedelkovski, *How Rotary Encoder Works and How To Use It with Arduino*, How To Mechatronics, Poveznica: <https://howtomechatronics.com/tutorials/arduino/rotary-encoder-works-use-arduino/> ; pristupljeno lipanj 2024.
- [8] “MCP1804 datasheet”, Microchip (kolovoz, 2013), Poveznica: <https://hr.mouser.com/datasheet/2/268/20002200D-3103159.pdf> ; pristupljeno travanj 2024.
- [9] “ST-Link/V2 user manual”, STMicroelectronics (travanj, 2024), Poveznica: [https://www.st.com/resource/en/user\\_manual/um1075-stlinkv2-incircuit-debuggerprogrammer-for-stm8-and-stm32-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um1075-stlinkv2-incircuit-debuggerprogrammer-for-stm8-and-stm32-stmicroelectronics.pdf) ; pristupljeno siječanj 2024.
- [10] “ST3232B datasheet”, STMicroelectronics (siječanj, 2016), Poveznica: <https://www.st.com/resource/en/datasheet/st3232b.pdf> ; pristupljeno siječanj 2024.
- [11] “FT232RL datasheet”, FTDI (2005), Poveznica: [https://www.lcsc.com/datasheet/lcsc\\_datasheet\\_1810311811\\_FTDI-FT232RL-REEL\\_C8690.pdf](https://www.lcsc.com/datasheet/lcsc_datasheet_1810311811_FTDI-FT232RL-REEL_C8690.pdf) ; pristupljeno siječanj 2024.
- [12] Altium Designer, Altium, Poveznica: <https://www.altium.com/altium-designer>
- [13] FreeRTOS, Poveznica: <https://freertos.org/index.html>
- [14] Visual Studio Code, Microsoft, Poveznica: <https://code.visualstudio.com/>

- [15] STM32 for VSCode, Bureau Moeilijke Dingen , Poveznica:<https://github.com/bmd-studio/stm32-for-vscode> ; pristupljeno ožujak 2024.
- [16] ARM GNU Toolchain, ARM Developer, Poveznica <https://developer.arm.com/downloads/-/gnu-rm>
- [17] DipRadSw, Poveznica <https://github.com/Zugislav/DipRadSW>
- [18] Khaled Magdy, *STM32 Timer Encoder Mode – STM32 Rotary Encoder Interfacing*, DeepBlue mBedded, Poveznica: <https://deepbluembedded.com/stm32-timer-encoder-mode-stm32-rotary-encoder-interfacing/> ; pristupljeno lipanj 2024.
- [19] STM32-Libraries, Daniel Martensson, (veljača 2024.) Poveznica: <https://github.com/DanielMartensson/STM32-Libraries> ; pristupljeno travanj 2024.
- [20] stm32-ili9341, afiskon, 20.8.2022. Poveznica: <https://github.com/afiskon/stm32-ili9341> ; pristupljeno svibanj 2024.
- [21] JLCPCB, Poveznica: <https://jlcpcb.com/>



## Sažetak

U ovom radu razvijen je ugradbeni računalni sustav temeljen na mikrokontroleru STM32F407. Sklopovlje sustava implementirano je na dvoslojnoj pločici dimenzije 116.3 x 100 mm. Sadrži linearni regulator napona, dva USB sučelja, RS232 sučelje, LCD prikaznik s ILI9341 kontrolerom za prikaz i XPT2046 sklopom za dodirni zaslon. Uhodavanje i ispitivanje sustava moguće je preko SWD/JTAG konektora. Na pločici se nalazi šest tipki sa svjetlećim diodama i rotacijski enkoder. Programska podrška razvijena za ovaj sustav omogućuje interaktivno rukovanje sa sustavom te upravljanje programski definiranim danim prijamnikom.

### **Ključne riječi:**

Ugradbeni računalni sustav, STM32F4, USB, SWD, JTAG, pokaznik s tekućim kristalima, inkrementalni enkoder, dodirni zaslon

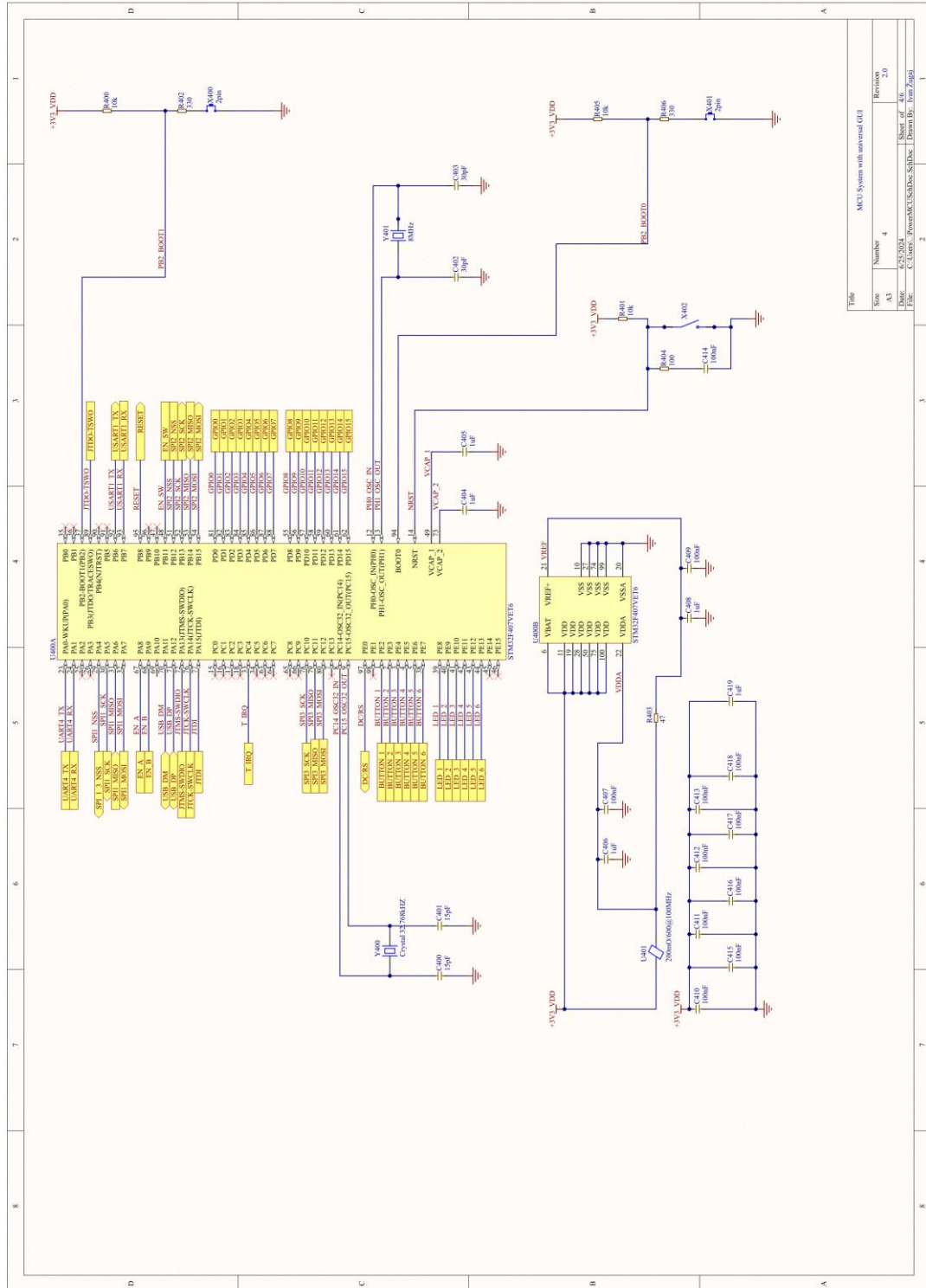
## Summary

In this work, an embedded computer system based on the STM32F407 microcontroller is developed. The system's hardware is implemented on a two-layer board measuring 116.3 x 100 mm. It contains a linear voltage regulator, two USB interfaces, an RS232 interface, an LCD display with the ILI9341 controller for display, and the XPT2046 chip for the touch screen. Debugging and testing of the system are possible through the SWD/JTAG connector. The board features six buttons with LED indicators and a rotary encoder. The software developed for this system allows interactive handling of the system and control of the software-defined given receiver.

### Keywords:

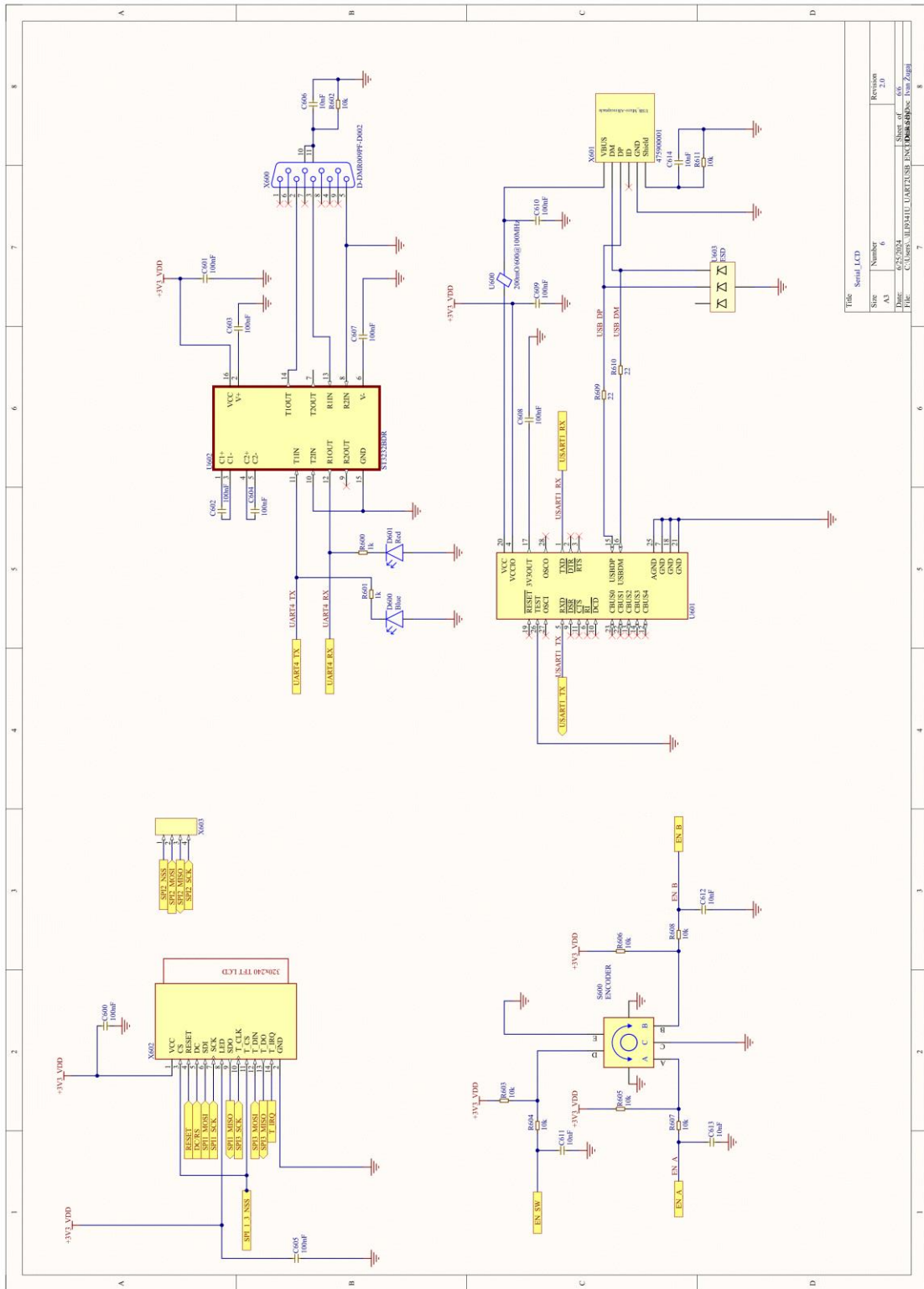
Embedded system, STM32F4, USB, SWD, JTAG, liquid crystal display, incremental encoder, touch screen

# Privitak A – Električne sheme



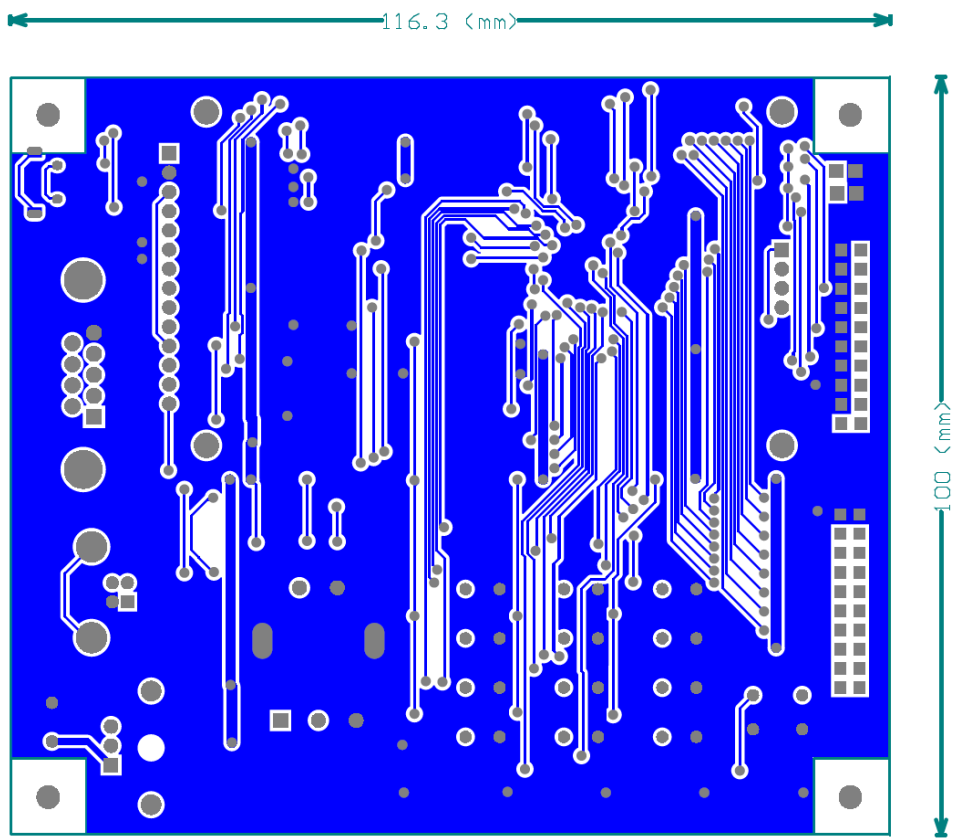
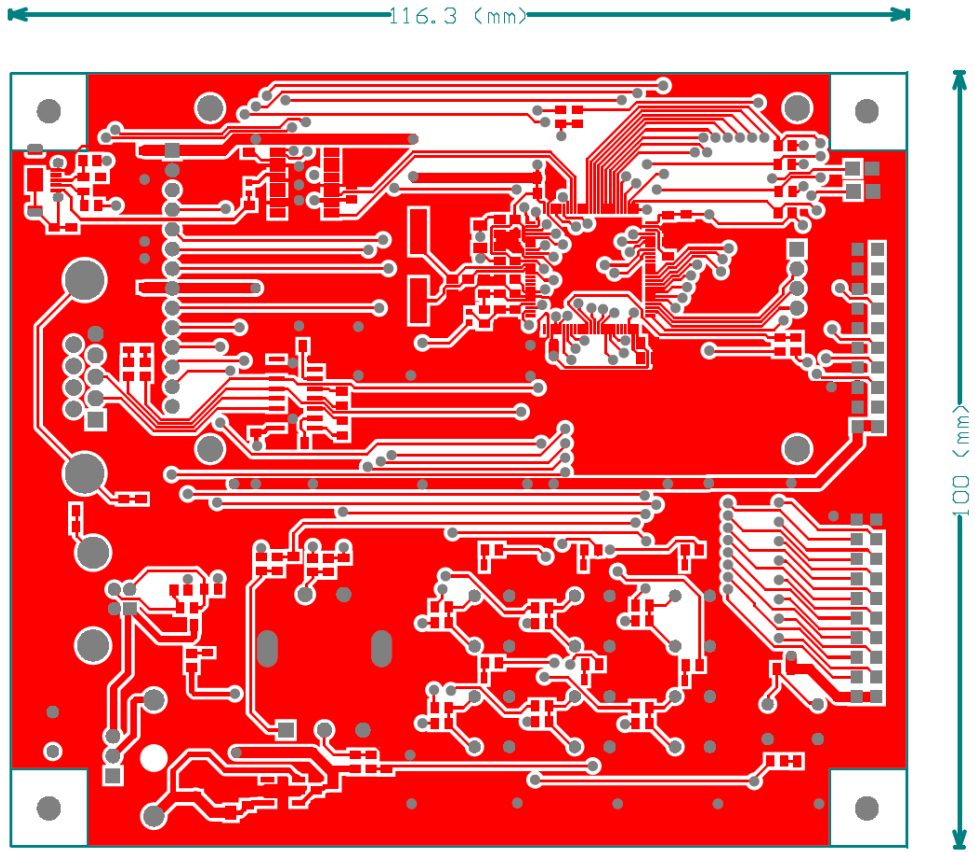
Title			
MCL System with universal GUI			
Size	Number	Revision	
A3	4	20	
Date:	6/25/2024	Sheet of:	46
File:	C:\Users\PowerDCU\Sketches\Privitak	Drawn By:	Ivan Zupnik





Title	Serial_LCD
Size	Number
A3	6
Date	6/25/2024
File	C:\Users\JED94HE\OneDrive\Desktop\Serial_LCD
Sheet of	66
Revision	2.0





## Privitak C – tablica materijala

Komentar	Šifra proizvoda	Oznaka	Količina
15pF	C1794	C400, C401	2
30pF	C1809	C402, C403	2
1uF	C377773	C404, C405	2
1uF	C28323	C406, C408, C419	3
100nF	C1711	C407, C409, C410,C411, C412, C413, C414, C415, C416, C417, C418, C502, C503, C600, C601, C602, C603, C604, C605, C607, C608, C609, C610	23
10uF T	C7171	C500, C501	2
10nF	C1710	C504, C606, C611, C612, C613, C614	6
Green	C205449	D500, D505, D506	3
Zener diode	C152307	D501	1
Blue	C84259	D503, D504, D600	3
Red	C84256	D507, D508, D601	3
Fuse Block	C206888	F500	1



10k	C25612	R400, R401, R405, R501, R508, R509, R510, R511, R512, R513, R514, R515, R516, R517, R602, R603, R604, R605, R606, R607, R608, R611	22
330	C17630	R402, R406	2
47	C25315	R403	1
100	C25277	R404	1
510	C17734	R500, R502, R503, R504, R505, R506, R507, R520, R521, R522, R523, R524, R525	13
22	C416949	R518, R519, R609, R610	4
1k	C204136	R600, R601	2
ENCODER	C143803	S600	1
STM32F407VET6	C28730	U400	1
200mO/600@100MHz	C780186	U401, U600	2
MCP1804T-3302I/MT	C150783	U500	1
ESD	C7074	U502, U603	2
FT232RL-REEL	C8690	U601	1
ST3232BDR	C2987706	U602	1
2pin	C492401	X400, X401	2
Round Tactile Switch	C2765189	X402, X505, X506, X507, X508, X509, X510	7
USB Type B	C254704	X500	1

Jumper	C706875	X501	1
2x10	C492444	X502, X504	2
WJ500V-5.08-2P	C8465	X503	1
D-DMR009PF-D002	C141882	X600	1
micro B USB	C136001	X601	1
B-2200S14P-A120	C124407	X602	1
4FP header	C5144876	X603	1
Crystal 32.768kHz	C32346	Y400	1
8MHz	C12674	Y401	1