



Univerza *v Ljubljani*
Fakulteta *za računalništvo*
in informatiko

Večna Pot 113, 1000 Ljubljana

VIN Projekt – DIY Osciloskop

Mentor: Robert Rozman
Predmet: Vhodno-izhodne naprave

Avtor: Trop Žiga

Ljubljana, Avgust 2023

Povzetek

Namen mojega projekta je bil se bolj spoznati z delovanjem osciloskopa in ga izdelati.

Osciloskop je instrument, ki nam grafično prikaže relacijo električne napetosti s časom. To omogoča posamezniku vizualizacijo in analizo hitrih električnih signalov, kateri niso vidni drugače.

V preteklosti so bili osciloskopi izdelani iz katodnih cevi, podobne tisti, ki jih lahko najdemo v starih televizijah. Delali so tako, da so poslali elektronski žarek preko ekrana, ki je bil prevečen s fosforjem, ki je bil vertikalno odklonjen z vhodnim signalom. To bi povzročilo ekran da žari v vsaki točki, kjer je udaril žarek, omogčanje vizualizacije vhodnega signala kot val. Ta proces je bil narejen analogno, vse meritve so morale biti narejene na roko, fizično meriti val na ekranu.

Dandanes je večina osciloskopov digitalnih, ki nudijo shrambo valov. Na ta način imamo možnost analizirati signal na računalniku, kar pomeni napredne in avtomatične meritve.

Ključne besede: Digitalni in analogni osciloskop, vizualizacija električnih signalov

Summary

The point of my project was to get to know how the oscilloscope works and how to make it.

Oscilloscope is an instrument which graphically shows the relation between electrical voltage and time. This enables us the visualisation and analisation of fast eletrcial signals, which can't be seen otherwise.

In the past the osilosopes were made from cathode ray tubes, similar to the ones found in old TVs. They worked by sweeping an electron beam across a phosphor-coated screen, which was being vertically deflected by the input signal. This made the screen glow in every point where the ray hit, allowing us to view the input signal as a waveform. This process was analog, which means every measurement had to be done physically by hand.

Nowadays most of the oscilloscopes are digital which also allow us to store the waveforms. This way we have access to advanced and automatic measurements.

Key words: Digital and analog oscilloscope, visualisation of electrical signals

KAZALO

1.	DELOVANJE	1
A)	KREPITEV IN SLABITEV SIGNALA	1
B)	SPREMEMBA V ŠTEVILKE	2
C)	VZORČENJE SIGNALA	2
2.	VEZAVE	3
A)	POTREBNI DELI	3
B)	FRONTEND	3
C)	MIKROKONTROLER	4
3.	PROGRAMSKA KODA	5
A)	MODULI	5
B)	INICIALIZACIJA	6
4.	PRIMER DELOVANJA	7
5.	ZAKLJUČEK	8
6.	VIRI IN LITERATURA	9

KAZALO SLIK

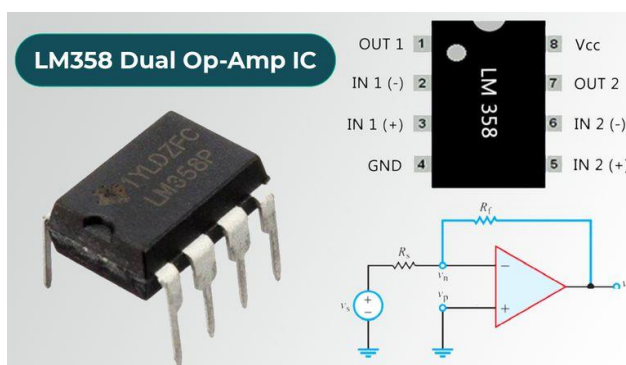
SLIKA 1:OPERACIJSKI OJAČEVALNIK.....	1
SLIKA 2:DELITELJ NAPETOSTI	1
SLIKA 3:ADC - ANALOG TO DIGITAL CONVERTER	2
SLIKA 4:HEMA FRONTEND.....	3
SLIKA 5:VEZAVA FRONTEND	3
SLIKA 6:PINOUT.....	4
SLIKA 7:IZGLED LCD.....	4
SLIKA 8:KODA #1	6
SLIKA 9:KODA #2	6
SLIKA 10:PRIKAZ DELOVANJA	7

1. Delovanje

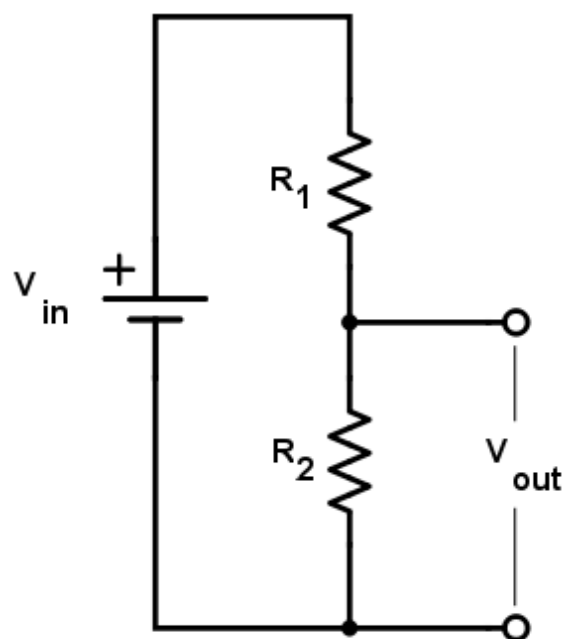
a) Krepitev in slabitev signala

Da lahko merimo večji predel napetosti moramo najprej krepiti in oslabiti signal, to naredimo z komponento imenovano »analogni frontend«.

S to komponento lahko povečamo področje napetosti iz 0 do 3,3V na -3,3V do 3,3V. Do tega pridemo tako, da oslabimo signal na polovico in ga centriramo na 1,65V. To storimo z uporabo delitelja napetosti in operacijskega ojačevalnika.



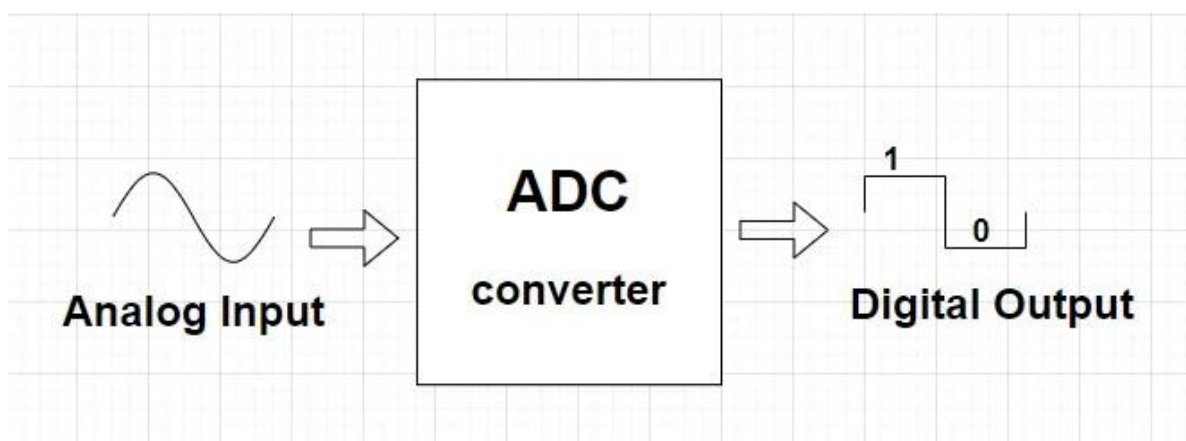
Slika 1: Operacijski ojačevalnik



Slika 2: Delitelj napetosti

b) Sprememba v številke

Ker napetost spada pod zvezne spremenljivke jo moramo spremeniti v številčno vrednost. To dosežemo z ADC(Analog to Digital Converter). ADC je v našem primeru še integriran v mikrokontroler. ADC deluje tako, da vzame vhodno napetost in jo primerja z različnimi napetostmi in vzame tisto napetost, ki je najbližje le tej. Glavni karakteristiki ADC-ja sta hitrost vzorčenja in rezolucija. Rezolucija nam pove maksimalno število bitov, ki jih lahko uporabljamo za predstavo številke. V našem primeru je 12 bitno, kar pomeni da obstaja 2^{12} različnih vrednosti, ki jih lahko prikaže. Hitrost vzorčenja pa nam pove kako hitro lahko ADC izmeri vhodni signal. V našem primeru je to 2.4 Msa/s (megasamples per second).



Slika 3:ADC - Analog to Digital Converter

c) Vzorčenje signala

Za digitalno rekonstrukcijo valov moramo vzorčiti signal v preciznih časovnih intervalih. To dosežemo z uporabo časovnikov našega mikrokontrolerja. Na ta način ADC zajame vzorec na vsako sprožitev časovnika. Po navadi osciloskopi uporabljajo naprave specifično za sproženje vzorcev na preciznih segmentih vhodnega signala. V tem projektu sem to poenostavil z programom, tako da poravnam izrisan val s sprožitvijo.

2. Vezave

a) Potrebni deli

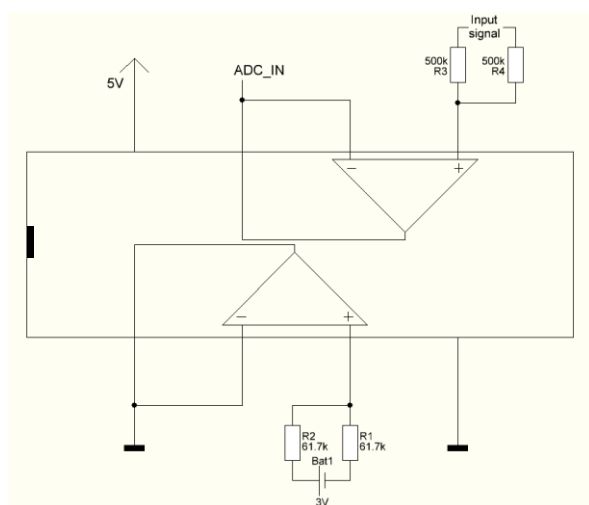
Za izdelavo osciloskopa sem potreboval naslednje:

- 2x61,7k Ω in 2x500k Ω upore,
- LM358 operacijski ojačevalnik,
- STM32F407VGT6,
- 3 Gumbi,
- ST7735 128x160 LCD ekran,
- 3V in 5V vir napetosti.

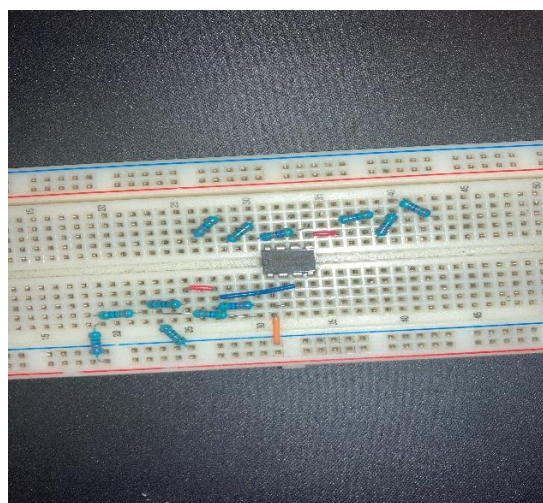
b) Frontend

Preden lahko začnemo meriti signal z osciloskopom moramo najprej predelati signal, tako da lahko dosežemo volte med -3.3V in 3.3V namesto od 0V do 3.3V.

Komponenta imenovana »analogni frontend« ima to vlogo, slabi in krepi signal. V našem primeru to naredimo tako, da oslabimo signal na neke polovico in ga centriramo na 1.65V. Z uporabo napetostnega delilnika in operacijskega ojačevalnika(LM358) nam to uspe.



Slika 4:Shema frontend



Slika 5:Vezava frontend

V moji vezavi sem zaporedno vezal več uporov, tako da sem lahko dosegel pravilno upornost. Do 500k Ω sem prišel tako, da sem zaporedno vezal 470k Ω in 2x 15k Ω . Do 61.7k Ω pa sem prišel tako, da sem zaporedno vezal 56k Ω in 2x po 2.2k Ω . 5V sem dovajal direktno iz STM32F407 mikrokontrolerja, 3V pa iz dveh 1.5V zaporedno vezanih baterij.

c) Mikrokontroler

Signal pride do mikrokontrolerja preko ADC_IN iz frontenda in se veže na PA0. LCD ekran pa je bil zvezan tekoče: LED – PC13, SCK – PB3, SDA – PB5, A0 – PB14, RST – PB13, CS – PB12.

CS- Chip Select, ki je uporabljen v primeru, da je več naprav povezanih na enak SPI vodilo,

RST- Reset, služi za nadzor napetosti, uporabljen je tudi lahko za resetiranje stanja mikrokontrolerja,

A0 – Address Select, služi za izbiro enega izmed 8 vnaprej definiranih IIC vodilnih naslovov,

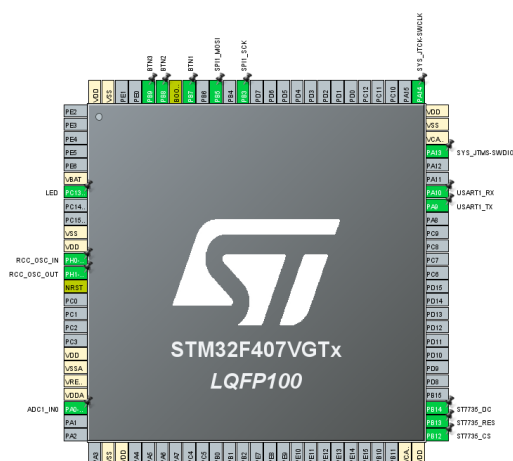
SDA – Serial Data, SPI komunikacija s podatki,

SCK – Serial Clock, časovna SPI komunikacija,

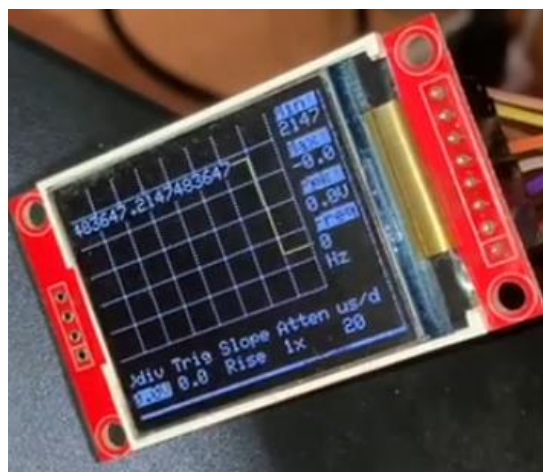
LED – Nam prižge LCD zaslon.

Za dovod napetosti pa sem uporabljal 3V pin, ki je na mikrokontrolerju, enako velja za ozemljitev.

Zraven tega pa še sem uporabljal 3 gumbе, ki so služili za spremembo različnih parametrov, npr velikost kvadrata itd. Na PB9 je bil zvezan gumb za povečanje posameznega parametra, PB8 za izbiro parametra in na PB7 za manjšanje parametra.



Slika 6: PinOut



Slika 7: Izgled LCD

3. Programska koda

a) Moduli

Za pravilno delovanje moramo v datoteki `stm32f4xx_hal_conf.h` imeti vključene naslednje module: `HAL_ADC`, `HAL_RTC`, `HAL_SPI`, `HAL_TIM`, `HAL_UART`, `HAL_GPIO`, `HAL_EXTI`, `HAL_RCC`, `HAL_FLASH`, `HAL_PWR` in `HAL_CORTEX`.

HAL – Hardware Abstraction Layer

ADC(Analog to Digital Converter)- pretvornik analognega signala v digitalnega.

RTC(Real Time Clock)- potek časa.

SPI (Serial Peripheral Interface)- sinhrona serijska povezava elektronskih naprav.

TIM(Advanced-control, general-purpose or basic timer) - sprožitev dogodkov glede na čas.

UART(Universal Asynchronous Receiver/Transmitter)- pošiljanje podatkov, bit po bit, prvo najpomembnejši.

GPIO(General Purpose Input/Output)- delo z vhodno izhodnimi napravami, ki so priklopljene.

EXTI(Extended Interrupt/Event Controller)- prekinitve pri GPIO preklapljanju.

RCC(Reset and Clock Control)- kontrola internih perifernih naprav, kot tudi za resetiranje signalov in časa

FLASH(Flash memory)- visoko nivojski vmesnik za upravljanje z flash spominom.

PWR(Power Controller)- poenostavi upravljanje upravljanje moči.

CORTEX- API za upravljanje z NVIC(Nested vectored interrupt controller) in SysTick (System tick Timer)

b) Inicializacija

Najprej se vsi ti moduli imenovani zgoraj inicilaizirajo, nato pridemo do funkcije scopeInit, ki inicializira LCD ekran, nastavi hitrost vzorčenja in pobriše vse kaj je na njem.

Potem pridemo v glavni del main funkcije, ki samo kliče funkcijo scopeLoop, kjer dobimo signal, katerega s pomočjo ADC-ja krepimo in slabimo. V ui funkciji pa vse izrišemo na LCD ekran. To se vse ponavlja dokler sprejemamo podatke, kar ugotovimo s finishedConversion spremenljivko.

```
int main(void)
{
    /* USER CODE BEGIN 1 */
    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and
    HAL_Init();

    /* USER CODE BEGIN Init */
    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */
    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_ADC1_Init();
    MX_SPI1_Init();
    MX_TIM3_Init();
    MX_USART1_UART_Init();
    /* USER CODE BEGIN 2 */
    scopeInit();
    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1)
    {
        /* USER CODE END WHILE */
        scopeLoop();
        /* USER CODE BEGIN 3 */
    }
    /* USER CODE END 3 */
}
```

Slika 8:Koda #1

```
void scopeInit()
{
    ST7735_initR(INITR_BLACKTAB, &hspi1); // Initialize the LCD
    setRotation(1);
    createFramebuf(); // Create the framebuffer for the LCD
    clearDisplay();
    splash(); // Splash screen

    // Set the sampling rate
    sampRate = (16000 * 1000) / tdiv;
    sampPer = tdiv / 16.0;
    setTimerFreq(sampRate);

    // Initialize the UART
    HAL_UART_Receive_IT(&huart1, uartBuf, 1);
}

// This function acquires one buffer worth of data
void sample()
{
    HAL_TIM_Base_Start(&htim3);
    HAL_ADC_Start_DMA(&hadc1, (uint32_t *)adcBuf, BUFFER_LEN);
    while (!finishedConversion);
    HAL_TIM_Base_Stop(&htim3);
    finishedConversion = 1;
}

// This is the main loop of the app
void scopeLoop()
{
    // Acquire one buffer
    sample();

    // Find the trigger point
    findTrigger(adcBuf);
    if (triggered)
        HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, 0); // light the on-board

    // Run the UI
    ui();
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, 1);
}
```

Slika 9:Koda #2

Zraven tega še imamo nekaj datotek, kjer definiramo izgled pisave, ki bo na LCD ekranu in risanje na LCD ekran.

4. Primer delovanja

Tukaj vidimo primer delovanja osciloskopa. Ker nimam generatorja napetosti sem imel nekaj problemov, ki sem jih razrešil na edini način, ki sem ga videl. 3V za frontend sem dobil z zaporedno vezavo dveh 1.5V baterij. Za 5V dovod napetosti do operacijskega ojačevalnika sem uporabil 5V, ki jih lahko dobim iz mikrokontrolerja. Za LCD ekran sem prav tako uporabil 3V, ki so na mikrokontrolerju. Edini problem, ki sem ga imel je bil da nisem imel naprav, ki delujejo med -3.3 do 3.3V in sem moral uporabiti 12V DC motor, kar pomeni da je signal konstanten in zmeraj na vrhu. Za dovod tega DC motorja sem uporabil star napajalnik v računalniku, enako bi tudi lahko naredil dovod 5V za operacijski ojačevalnik.



Slika 10: Prikaz delovanja

5. Zaključek

S svojim projektom sem zadovoljen. Naučil sem se osnove delovanja in programiranja mikrokontrolerjev. Zraven tega pa sem se tudi naučil kako deluje osciloskop. Ta projekt sem si izbral, ker sem se želel nekaj novega naučiti, kar se ne bom če si bom izbral nekaj lahkega. Iz istega razloga sem se tudi lotil tega projekta sam.

6. Viri in literatura

1. ElectroBOOM - What's an OSCILLOSCOPE. (Online). (Citirano 1.9.2023). Dostopno na naslovu: https://www.youtube.com/watch?v=DgYGRtkd9Vs&t=344s&ab_channel=ElectroBOOM
2. HACKADAY DIY STM32 Scope. (Online). (Citirano 1.9.2023). Dostopno na naslovu: <https://hackaday.com/tag/stm32f4/>
3. Tvlad1234 pillScopePlus. (Online). (Citirano 1.9.2023). Dostopno na naslovu: <https://github.com/tvlad1234/pillScopePlus/tree/main>
4. OverKill Projects Youtube. (Online). (Citirano 1.9.2023). Dostopno na naslovu: <https://www.youtube.com/@overkillprojects8091>
5. STM32F4 HAL User manual. (Online). (Citirano 1.9.2023). Dostopno na naslovu: https://www.st.com/resource/en/user_manual/um1725-description-of-stm32f4-hal-and-lowlayer-drivers-stmicroelectronics.pdf
6. Oscilloscope-wikipedia. (Online). (Citirano 1.9.2023). Dostopno na naslovu: <https://en.wikipedia.org/wiki/Oscilloscope>
7. Tyco electronics – The Oscilloscope (Online). (Citirano 1.9.2023). Dostopno na naslovu: <http://www.cromptonusa.com/oscilloscope.pdf>
8. ChatGPT – Explanation of STM32 Modules and code. (Online). (Citirano 1.9.2023). Dostopno na naslovu: <https://chat.openai.com/>