

# R-2R Digitální - Analogový převodník (DAC)

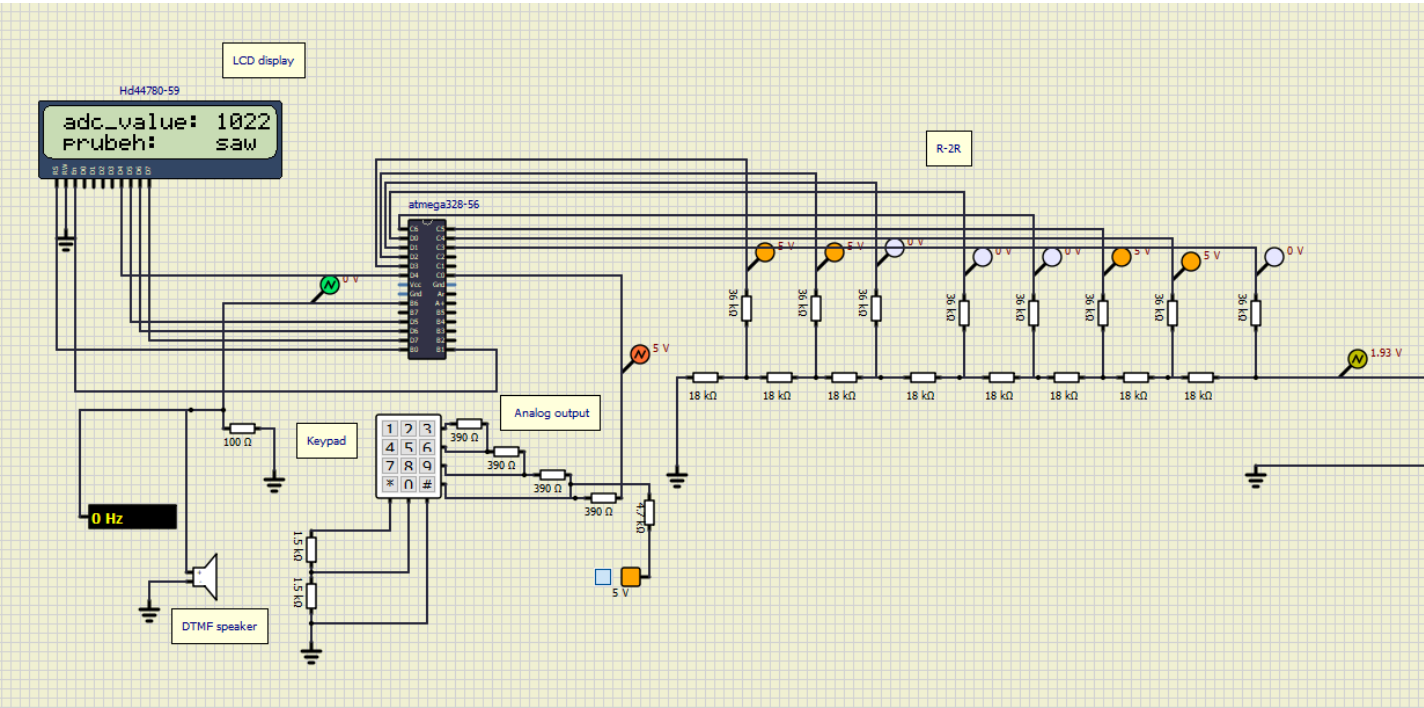
## Team members

Libor Matějek <https://github.com/libormatejek/Digital-electronics-2/tree/master/Labs/project>  
Tomáš Stupka <https://github.com/xstupk04/Digital-electronics-2/tree/master/Labs/Project/Project/project/DAC>

## Zadání projektu

Aplikace generátoru analogového signálu využívající 8bitový DAC; několik přednastavených typů signálů; displej; ovládání pomocí klávesnice 4x3; (možnost generování tónu DTMF, Dual-Tone Multiple Frequencies); odesílání zajímavých informací o stavu aplikace na UART.

## Popis hardwaru

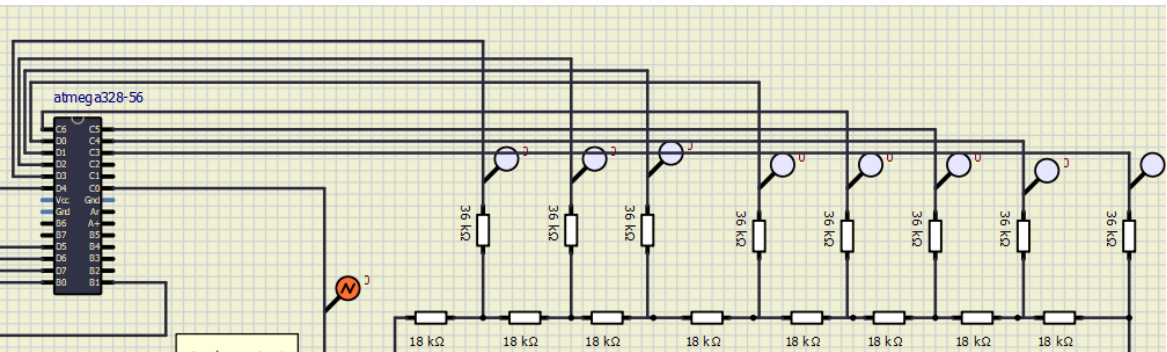


Projekt postavený na mikrokontroleru Atmega328p z rodiny AVR. K němuž je připojen LCD displej s řadičem HD44780 pro zobrazení typu posílaného signálu a informace ohledně stisknuté klávesnici. Nastavený typ signálu bude možné nastavit pomocí 4x3 klávesy. Pro převod signálu z digitální hodnoty na analogovou bude využit 8 bitový tzv. žebříkový systém odporů na jehož výstup se připojí osciloskop na kterém se zobrazí průběh signálu. Hodnota stisku konkrétního tlačítka je vysíláno pomocí UARTU.

### Tabulka zapojení

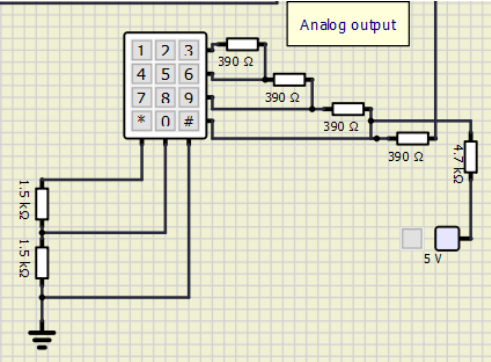
Zařízení	Port procesoru	Popis
R-2R převodník	PC[3:6], PD[0-3]	Připojení R2R DAC převodníku
Analogový výstup klávesy	C0	AD převodník pro klávesnici
RS (disp)	PB0	Výběr mezi instrukčním registrem (RS = 0) a datovým registrem (RS = 1)
R/W (disp)	GND	Read/write (Read=1, write=0)
E (disp)	PB1	Enable
D[7:4] (disp)	PD[7:4]	Obousměrná datová sběrnice, přenos dat se provádí jednou, přes D0 až D7, v případě, že délka dat je 8 bitů. Pokud pouze 4 bitová, jedná se o D4 až D7

## R-2R rezistorová síť



Jak ze schématu plyne jedná se o síť rezistorů, pomocí které jde jednoduše převést digitální hodnotu signálu na ekvivalentní analogovou hodnotu. Rozlišení takové sítě je závislé na počtu tzv. „žebříků“ (čím více žebříků, tím vyšší rozlišení). Síť rezistorů není nic jiného než propojení několika odporových děličů napětí za sebou a jejich výstup závisí na přivedené digitální hodnotě z vstupu (v našem případě generované hodnotě z mikroprocesoru).

## 4x3 keypad



Pomocí využití klávesnice lze přednastavit 12 analogových hodnot. Výstup klávesnice je přiveden na port C0 mikroprocesoru. Tento port má funkci AD převodníku, kdy v rámci programu rozlišuje stisknuté tlačítko. Každá klávesa kombinuje specifické rezistory a tvoří tak dělič napětí, který vytváří specifické napětí na výstupu. S tímto napětím lze určit, zda je aktuálně stisknut jakýkoli a který konkrétní klíč.

## Popis programu

Program je postaven na knihovnách ze cvičení- tedy timer.h, která zjednodušuje práci s časovači, gpio.h, jež umí ovládat i/o, lcd.h, která je převzata od Petera Fleuryho a umožňující snadnou práci s lcd displejem, uart.h, která je převzata od stejného autora, jako knihovna lcd.h. Navíc využíváme knihovnu math.h pro zpřístupnění matematických operací. DAC převodník je založen na stále se inkrementující hodnotě i pomocí interrupt rutiny, která je zpracovávána ve funkci fce. Tato funkce obsahuje 3 různé typy matematických výpočtů aktuálního vzorku pro nastavený průběh.

```
//function for generating signals values
unsigned int fce(unsigned int i ) {

    if(fceType==0)
        return (unsigned int)(255*(1 + sin(i * 6.28 * Tvz/T))/2); //returns sin values
    else if(fceType==1)
        return (unsigned int)(255 * fmod(i*Tvz,T)/T); //returns saw values
    else if(fceType==2) {
        int j = (int)(i*Tvz/(T/2)); //returns triangle values
        float a = 2*fmod(i*Tvz,T)/T;
        a = 255*(j%2==0? a : 1 - a);
        return (unsigned int)a;
    }
    else if(fceType==10) //stops signal generating
        return (uint8_t){0};
}
```

Hodnota je dále filtrována funkcí setByte, která zjišťuje, zda je bit na určité pozici aktivní, nebo ne. Pokud zjistí aktivitu, vydá instrukci pro aktivaci výstupu.

```
// function for recognition each bit values
unsigned setByte(unsigned int num) {
    if (num & (0x01 << 0)) GPIO_write_high(&PORTD,PD3); else GPIO_write_low(&PORTD,PD3);
    if (num & (0x01 << 1)) GPIO_write_high(&PORTD,PD2); else GPIO_write_low(&PORTD,PD2);
    if (num & (0x01 << 2)) GPIO_write_high(&PORTD,PD1); else GPIO_write_low(&PORTD,PD1);
    if (num & (0x01 << 3)) GPIO_write_high(&PORTD,PD0); else GPIO_write_low(&PORTD,PD0);
    if (num & (0x01 << 4)) GPIO_write_high(&PORTC,PC6); else GPIO_write_low(&PORTC,PC6);
    if (num & (0x01 << 5)) GPIO_write_high(&PORTC,PC5); else GPIO_write_low(&PORTC,PC5);
    if (num & (0x01 << 6)) GPIO_write_high(&PORTC,PC4); else GPIO_write_low(&PORTC,PC4);
    if (num & (0x01 << 7)) GPIO_write_high(&PORTC,PC3); else GPIO_write_low(&PORTC,PC3);
}
```

ADC interrupt, který je aktivován interruptem timeru 1, hlídá hodnotu ADC vstupu z keypadu a nastavuje typy průběhů + DTMF děličku frekvence a dobu příslušného tónu. DTMF je realizováno pomocí funkce GPIO\_toggle(), která je aktivována pokaždé, kdy timer2 "dopočítá" do určená hodnoty DTMF hodnoty nastavené keypadem. Zároveň je regulovaná doba znění a to pomocí hodnoty duration. Toggleování je totiž zastaveno v moment, kdy je napočítáno do právě hodnoty duration.

```
ISR(TIMER2_OVF_vect)
// DTMF generator
{
    x++;
    if (x>=(DTMF/2)&&duration>1){
        GPIO_toggle(&PORTB,PB6);
        x=0;
        duration=duration-1;
    }
}
```

## Video/Animation

[Link for video simulation of signals](#)

[Link for video simulation of DTMF](#)

## Diskuze Úkolem bylo zrealizovat DAC 8-bitový převodník pro přednastavené druhy signálů. Tato část úkolu se nám povedla splnit pomocí matematických funkcí. Metoda je však frekvenčně omezena. Dalším nedostatkem této metody je glitch u trojúhelníkového signálu, z důvodu dělení nulou.Tento problém se nám povedlo zminimalizovat. Výhodou našeho matematického řešení signálů je snadná rozlišitelnost na vícebitový převodník. Zároveň bylo snadné implementovat změnu frekvence signálu, což je nad rámec zadání. Signály a frekvence jsou voleny za pomoci 3x4 keypadu zapojeném pomocí ADC vstupu mikrokontroleru, zde se nevyskytují žádné problémy. Analogová přiváděná hodnota je zároveň s konverzí na digitální hodnotu vysílána přes UART. Tato aplikace je inspirována cvičením. Dalším úkol byla implementace DTMF na již zmíněný keypad. Tato funkce je provedena za pomoci děličky interrupt timeru. DTMF tedy není generován součtem dvou frekvencí, jak říká teorie. Zapojení je doplněno o LCD displej, který zobrazuje aktuální generovaný průběh. Displej je programován dle cvičení.(knihovna od Petera Fleuryho)

## Zdroje

1. [http://www.avr-asm-tutorial.net/avr\\_en/apps/key\\_matrix/keypad/resmatrix/resmatrix.html](http://www.avr-asm-tutorial.net/avr_en/apps/key_matrix/keypad/resmatrix/resmatrix.html)
2. <https://www.electronics-tutorials.ws/combinator/r-2r-dac.html>
3. <http://www.peterfleury.epizy.com>