Poročilo 2. domače naloge pri predmetu OR

# Ideja projekta

Osnovna ideja je implementirati aktivni brenčač, ki igra pesmi poslane preko usart sprejemnika na stm32h750bdk discovery plošči.

# Kam povezati brenčač

Slika, ki vsebuje besede vezje, elektronska komponenta, elektronsko inženirstvo, pasivna komponenta vezja

Vsebina, ustvarjena z UI, morda ni pravilna.Brenčač sem povezal na pin D11, saj je primerno oddaljen od GND -> lahko brenčač povežemo direktno brez ostalih žic in »breadboard-ov«. Pin D11 je tudi interno povezan na GPIO pin PB15, ki pa se lahko nastavi direktno za generiranje PWM signalov iz časovnika 12 na kanalu 2 (tim12 channel 2).

# Slika, ki vsebuje besede besedilo, diagram, posnetek zaslona, vrstica Vsebina, ustvarjena z UI, morda ni pravilna.



# Koda

Kodo sem naredil s pomočjo orodja CubeMX, kjer lahko vizualno nastavimo željene funkcionalnosti mikrokrmilnika. Zato sem nastavil usart3 in pa tim12, ki mi omogočata komunikacijo z zunanjo napravo (računalnik) preko serijskega vmesnika in pa generacijo PWM signala za brenčač. Glavna logika kode je taka:

* Sprejem vseh znakov do vključno '\n' in shranjevanje v tabelo
* Predvajanje pesmi glede na znake, ki so shranjeni v tabeli (c,d,e,f,g,h, ' ')
* Končanje pesmi
* Ponovna pripravljenost na sprejem

Sprejem znakov:

* void HAL\_UART\_RxCpltCallback(UART\_HandleTypeDef \*huart)
* {
* if(songPlaying == 1)
* return;
* if (huart->Instance == USART3)
* {
* song[songEnd++] = rxData;
* if(rxData == '\n')
* {
* songEnd = 0;
* songPlaying = 1;
* }
* HAL\_UART\_Receive\_IT(&huart3, &rxData, 1);
* }
* }

Proizvajanje zvoka:

void Buzzer\_ProcessSong(uint8\_t c[])

{

    int i = 0;

    while (c[i] != '\n'){

        switch (c[i])

            {

                    case 'c': Buzzer\_PlayNote(NOTE\_C, 300); break;

                    case 'd': Buzzer\_PlayNote(NOTE\_D, 300); break;

                    case 'e': Buzzer\_PlayNote(NOTE\_E, 300); break;

                    case 'f': Buzzer\_PlayNote(NOTE\_F, 300); break;

                    case 'g': Buzzer\_PlayNote(NOTE\_G, 300); break;

                    case 'a': Buzzer\_PlayNote(NOTE\_A, 300); break;

                    case 'h': Buzzer\_PlayNote(NOTE\_H, 300); break;

                    case ' ': HAL\_Delay(300); break;

                    case '\n': break;

                    default: break;

            }

    i++;

    }

}

void Buzzer\_PlayNote(uint16\_t freq, uint16\_t duration)

{

    if (freq == 0) return;

    uint32\_t prescaler = 80 - 1;

    uint32\_t period = (1000000 / freq) - 1;

    \_\_HAL\_TIM\_SET\_PRESCALER(&htim12, prescaler);

    \_\_HAL\_TIM\_SET\_AUTORELOAD(&htim12, period);

    \_\_HAL\_TIM\_SET\_COMPARE(&htim12, TIM\_CHANNEL\_2, period / 2);

    HAL\_TIM\_PWM\_Start(&htim12, TIM\_CHANNEL\_2);

    HAL\_Delay(duration);

    HAL\_TIM\_PWM\_Stop(&htim12, TIM\_CHANNEL\_2);

}

Glavna zanka:

**while** (1)

{

/\* USER CODE END WHILE \*/

/\* USER CODE BEGIN 3 \*/

**if**(songPlaying == 1){

**Buzzer\_ProcessSong**(song);

songPlaying = 0;

}

}

# Primer delovanja:

Za ta primer sem preko serijskega vmesnika poslal pesem »Čuk se je oženil«, ki je sledeče oblike:

* »c d e f gg gg aa aa gggg aa aa gggg f f f f ee ee dd dd gggg f f f f ee ee dd dd cccc »

