Projeto Piano de 4 Teclas

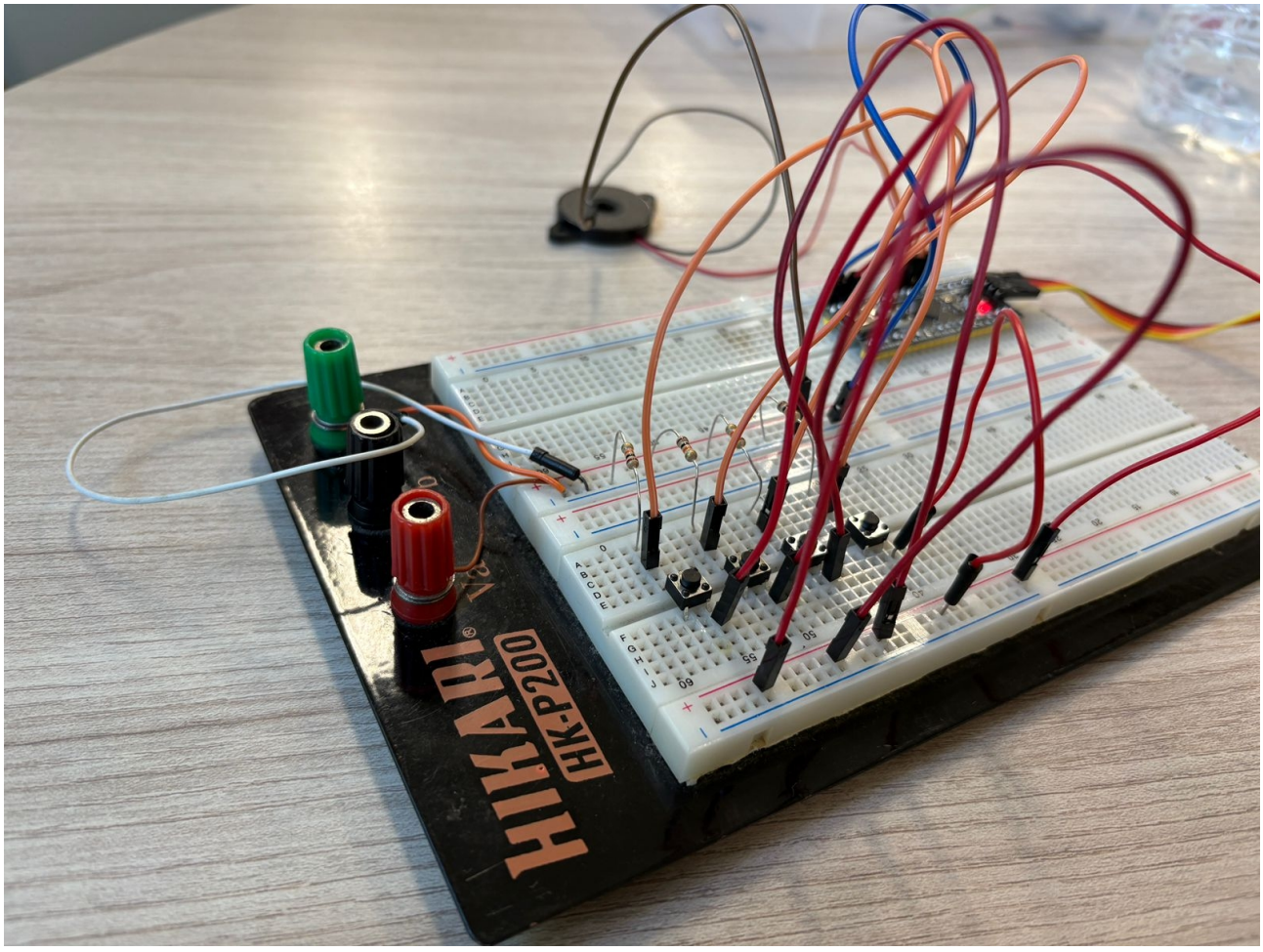
Projeto final da disciplina de Sistemas Microcontrolados turma S22 2025

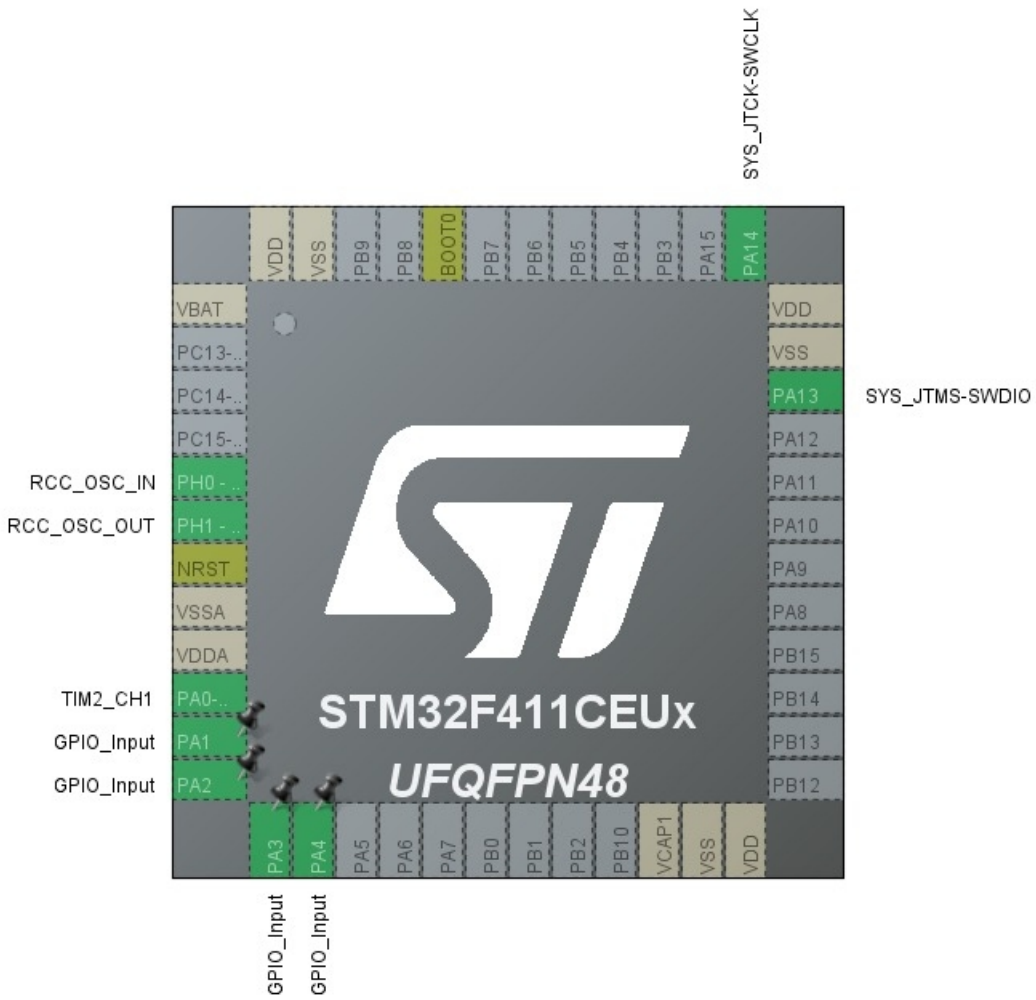
Ferramentas utilizadas

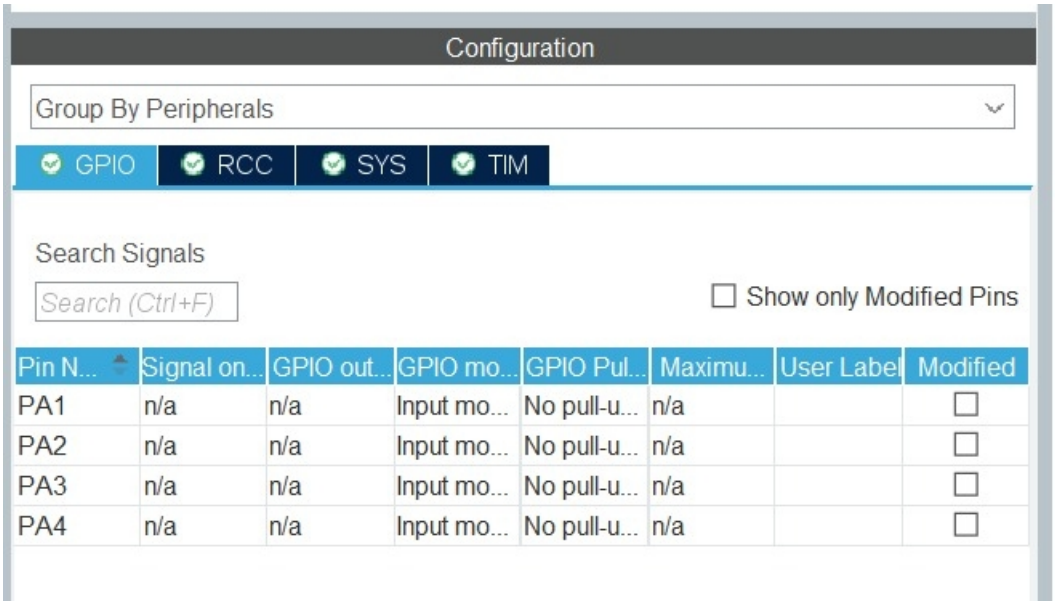
-1 microcontolador STM32F411CEU6
-4 botões
-4 resistores de 1k Ω
-1 buzzer passivo

Processo

Os botões foram conectados às entradas PA1 a PA4 do STM32, as quais foram configuradas como “input”, e à entrada 3V3 do microcontrolador. O modo foi definido como “no pull-up and no pull-down”, e cada botão foi conectado ao ground através de um resistor externo. Dessa forma, quando se aperta o botão, o buzzer toca e quando solta o botão o buzzer é silenciado.



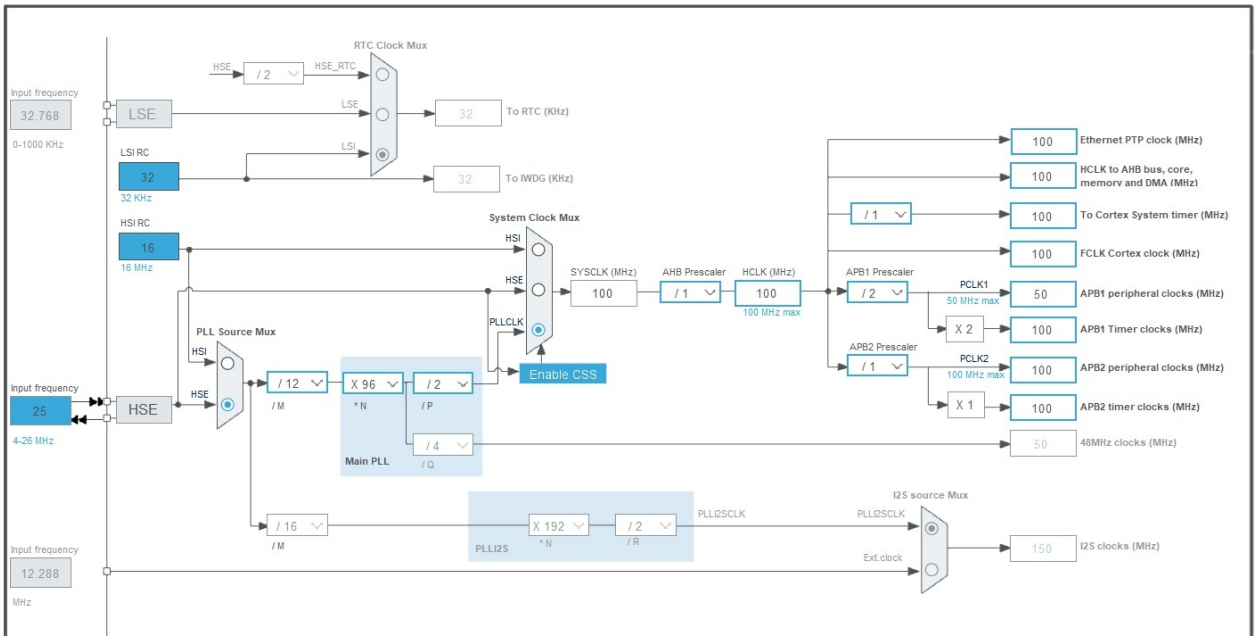




Já o buzzer foi conectado à entrada PA0, configurada como um output, para controlar a frequência do som emitido junto com as seguintes configurações de clock:

Configuration | Clock Configuration | Project Manager | To

Resolve Clock Issues



TIM2 Mode and Configuration

Mode

Slave Mode: Disable

Trigger Source: Disable

Clock Source: Internal Clock

Channel1: PWM Generation CH1

Channel2: Disable

Configuration

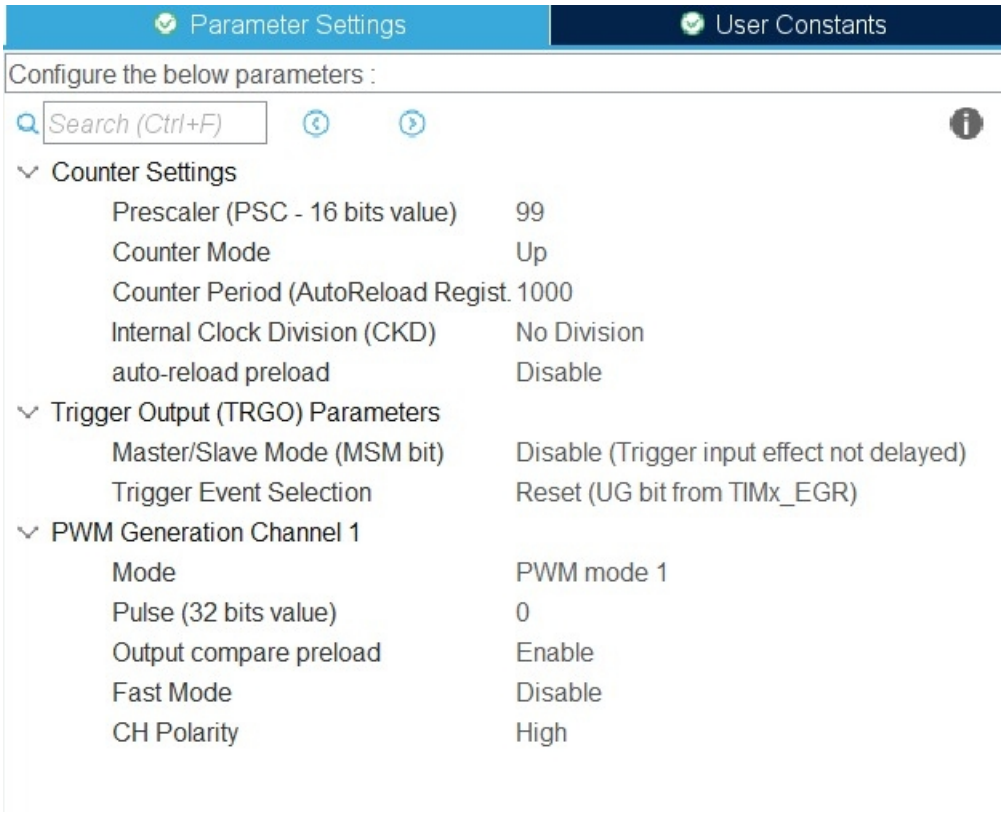
Reset Configuration

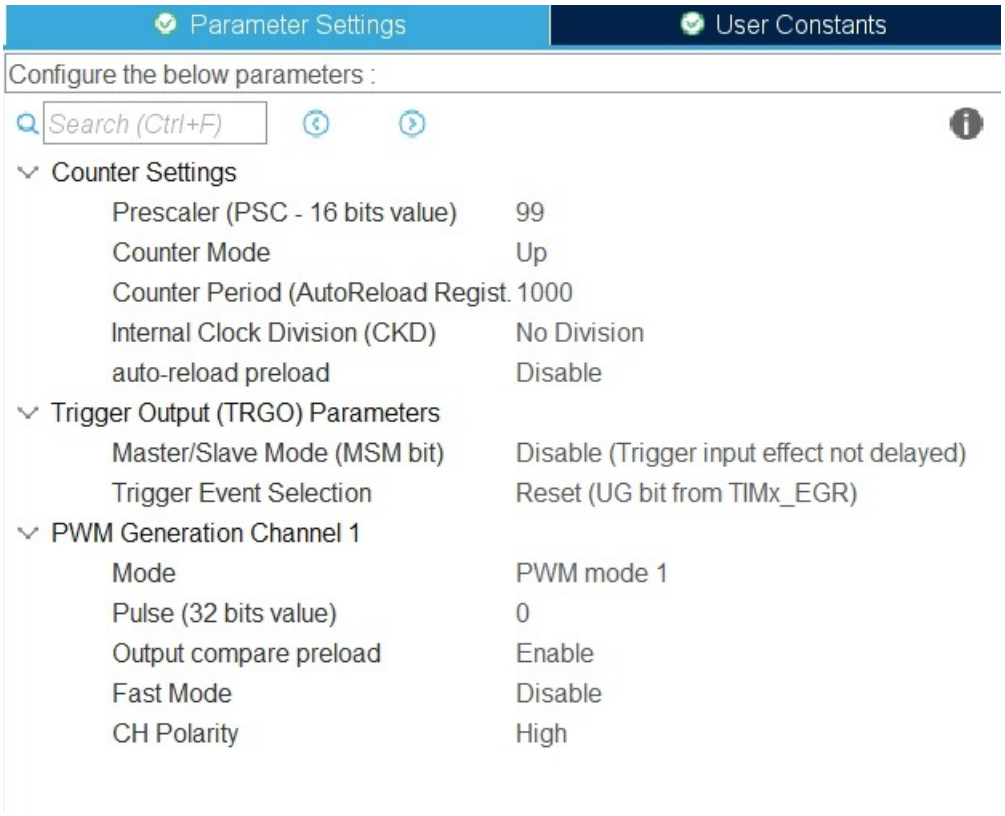
NVIC Settings | DMA Settings | GPIO Settings | Parameter Settings | User Constants

Search Signals

Search (Ctrl+F)

Pin Name	Signal on Pin	GPIO output l...	GPIO mode	GPIO Pull-up/...	Maximum
PA0-WKUP	TIM2_CH1	n/a	Alternate Fun...	No pull-up an...	Low


Parameter Settings
User Constants
Configure the below parameters :
Search (Ctrl+F)
Counter Settings
Prescaler (PSC - 16 bits value) 99
Counter Mode Up
Counter Period (AutoReload Regist. 1000
Internal Clock Division (CKD) No Division
auto-reload preload Disable
Trigger Output (TRGO) Parameters
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)
Trigger Event Selection Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1
Mode PWM mode 1
Pulse (32 bits value) 0
Output compare preload Enable
Fast Mode Disable
CH Polarity High


Parameter Settings
User Constants
Configure the below parameters :
Search (Ctrl+F)
Counter Settings
Prescaler (PSC - 16 bits value) 99
Counter Mode Up
Counter Period (AutoReload Regist. 1000
Internal Clock Division (CKD) No Division
auto-reload preload Disable
Trigger Output (TRGO) Parameters
Master/Slave Mode (MSM bit) Disable (Trigger input effect not delayed)
Trigger Event Selection Reset (UG bit from TIMx_EGR)
PWM Generation Channel 1
Mode PWM mode 1
Pulse (32 bits value) 0
Output compare preload Enable
Fast Mode Disable
CH Polarity High

Código

As variáveis “NOTA_DO, NOTA_RE, NOTA_MI e NOTA_FA” foram definidas para modificar a frequência do som que o buzzer emite, correspondendo a aproximadamente 261, 294, 329 e 249 Hz.

```
/* USER CODE BEGIN PV */
#define NOTA_DO  261
#define NOTA_RE  294
#define NOTA_MI  329
#define NOTA_FA  349
/* USER CODE END PV */
```

No loop principal, foi criada uma função que toca aquela nota específica dependendo do botão que é apertado.

```
/* USER CODE BEGIN 0 */
void tocar_nota(uint32_t frequencia) {
    if (frequencia == 0) {
        __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, 0); // Silêncio
    } else {
        uint32_t novo_arr = 1000000 / frequencia;
        __HAL_TIM_SET_AUTORELOAD(&htim2, novo_arr);
        __HAL_TIM_SET_COMPARE(&htim2, TIM_CHANNEL_1, novo_arr / 2);
        __HAL_TIM_SET_COUNTER(&htim2, 0); // Reinicia contador para resposta rápida
    }
}
/* USER CODE END 0 */
```

```
if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1) == GPIO_PIN_RESET) { // Mudou de SET para RESET
    tocar_nota(NOTA_DO);
}
else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_2) == GPIO_PIN_RESET) {
    tocar_nota(NOTA_RE);
}
else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_3) == GPIO_PIN_RESET) {
    tocar_nota(NOTA_MI);
}
else if (HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_4) == GPIO_PIN_RESET) {
    tocar_nota(NOTA_FA);
}
else {
    tocar_nota(0);
}

HAL_Delay(10);
```