



UNIVERSIDADE FEDERAL DE MATO GROSSO – UFMT
CAMPUS UNIVERSITÁRIO DE VÁRZEA GRANDE
FACULDADE DE ENGENHARIA
ENGENHARIA DE COMPUTAÇÃO

Filipe Chagas Ferraz

**Relatório Executivo do Projeto de SE:
GERADOR DE SINAIS**

DISCIPLINA: *SISTEMAS EMBARCADOS*

Prof. Dr. Jesus Franco Bueno

Cuiabá
2021

Filipe Chagas Ferraz

Relatório Executivo do Projeto de SE:
Gerador de Sinais

Trabalho apresentado na disciplina de Sistemas Embarcados como parte dos requisitos de avaliação na disciplina.

Cuiabá
2021

Resumo

Filipe Chagas Ferraz . **Relatório Executivo do Projeto de SE: Gerador de Sinais.** 27 p. Projeto – Engenharia de Computação, Faculdade de Engenharia, Universidade Federal de Mato Grosso, Brasil, 2021.

Este documento é um relatório de um projeto de gerador de sinais utilizando o microcontrolador STM32F411CE, um DAC 1334A, um display Nokia 5110 e um encoder KY-040. O firmware foi desenvolvido em C com auxílio dos software STM32CubeMX e STM32CubeIDE.

Palavras-chave: Sistemas. Embarcados. Microcontrolador. Gerador. Sinais.

Abstract

Filipe Chagas Ferraz . **ES Project Report: Signal Generator.** 27 p. Project – Computing Engineering, Engineering Faculty, Federal University of Mato Grosso, Brazil, 2021.

This document is a report of a signal generator project using the STM32F411CE microcontroller, a 1334A DAC, a Nokia 5110 display and a KY-040 encoder. The firmware was developed in C language with the STM32CubeMX and STM32CubeIDE softwares.

Keywords: Embedded. Systems. Microcontroller. Signal. Generator.

Lista de ilustrações

Figura 1 – Arquitetura do projeto	13
Figura 2 – Montagem do projeto na <i>protoboard</i>	14
Figura 3 – Placa de desenvolvimento Black Pill	15
Figura 4 – <i>Pinout</i> do microcontrolador	16
Figura 5 – Módulo CJMCU-1334	17
Figura 6 – Sinais do protocolo I2S	18
Figura 7 – <i>Pinout</i> do módulo KY-040 (fonte: < https://www.rcscomponents.kiev.ua/datasheets/ky-040-datasheet.pdf >)	21
Figura 8 – Padrões gerados pelo módulo KY-040 (fonte: < https://www.rcscomponents.kiev.ua/datasheets/ky-040-datasheet.pdf >)	21
Figura 9 – Diagrama de estados da IHM	22
Figura 10 – Fluxograma da IHM com imagens do <i>display</i>	23
Figura 11 – Exemplo de operação de mudança de frequência	23
Figura 12 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SIN, HARM=50)	25
Figura 13 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SQR, HARM=50)	26
Figura 14 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=TRI, HARM=50)	26
Figura 15 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SAW, HARM=50)	27

Lista de tabelas

Tabela 1 – Conexões básicas do protocolo I2S	18
--	----

Sumário

1	INTRODUÇÃO	13
2	MICROCONTROLADOR E FIRMWARE	15
2.1	Placa de desenvolvimento	15
2.2	Microcontrolador STM32F411CEU6	15
2.3	Programação	16
2.4	<i>Pinout</i>	16
3	CONVERSÃO DIGITAL-ANALÓGICO	17
3.1	O conversor digital-analógico 1334A	17
3.2	Protocolo I2S	17
3.3	Transmissão com DMA	18
4	GERAÇÃO DE SINAIS	19
4.1	Senoide (SIN)	19
4.2	Onda quadrada (SQR)	19
4.3	Onda triangular (TRI)	20
4.4	Onda dente-de-serra (SAW)	20
4.5	Otimização	20
5	INTERFACE HOMEM-MÁQUINA	21
5.1	<i>Encoder</i> KY-040	21
5.2	Estados da IHM	22
6	CONCLUSÃO	25
6.1	Resultados	25
6.2	Problemas encontrados	27

CAPÍTULO 1

Introdução

Um gerador de sinais ou gerador de função é um dispositivo eletrônico digital ou análogo que se propõe a gerar sinais com frequência, amplitude e forma de onda específicas. Neste trabalho, foi desenvolvido um gerador de sinais capaz de gerar sinais de 20Hz a 20kHz, amplitude de no máximo 1V pico-a-pico e formas de onda senoidal, quadrada, triangular e dente de serra. O gerador de sinais possui uma interface homem-máquina composta por um *display* LCD Nokia 5110 e um *encoder* rotativo KY-040. Foi utilizado um DAC de áudio 1334A, com comunicação **I2S!** (**I2S!**). A frequência de amostragem utilizada foi 44kHz.

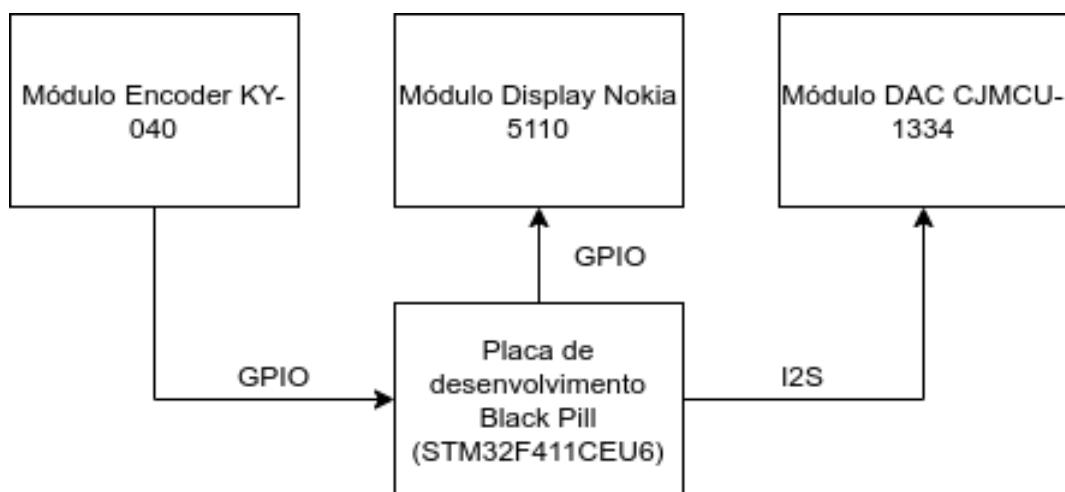


Figura 1 – Arquitetura do projeto

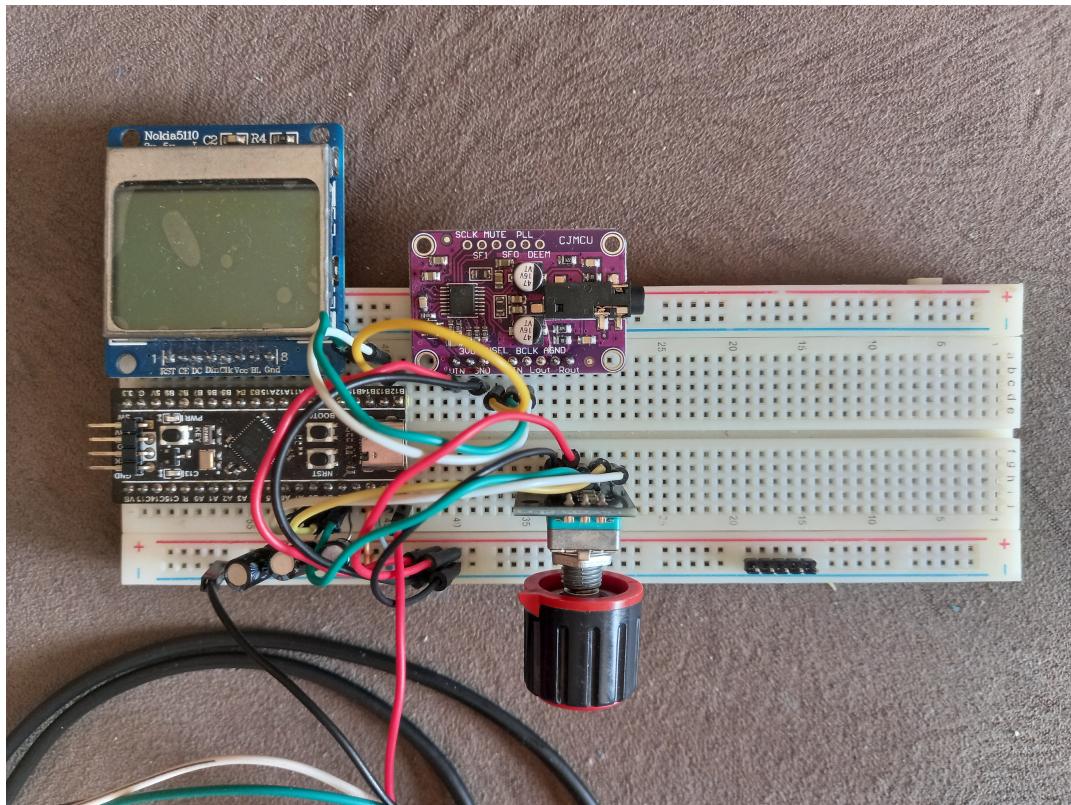


Figura 2 – Montagem do projeto na *protoboard*

CAPÍTULO 2

Microcontrolador e Firmware

2.1 Placa de desenvolvimento

A placa de desenvolvimento utilizada é uma Black Pill genérica de baixo custo (fabricante desconhecido). Esta placa possui um microcontrolador STM32F411CEU6 e componentes essenciais para o funcionamento deste (como cristais osciladores, capacitores de desacoplamento, reguladores de tensão, etc). Ela pode ser facilmente inserida em uma *protoboard* para prototipação.

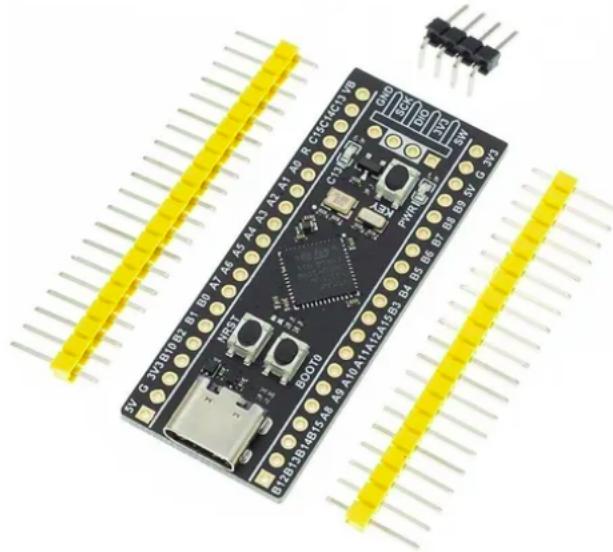


Figura 3 – Placa de desenvolvimento Black Pill

2.2 Microcontrolador STM32F411CEU6

STM32F411CEU6 é um microcontrolador da linha STM32, da fabricante ST. Este microcontrolador possui 512 Kbytes de memória *flash*, 128 Kbytes de memória RAM e

um processador ARM Cortex-M4 com FPU.

2.3 Programação

O *firmware* foi desenvolvido em linguagem C com o auxílio dos softwares STM32CubeMX e STM32CubeIDE.

O software STM32CubeMX é um editor visual responsável por gerar um código base chamado **HAL!** (**HAL!**). Este código possui as funções básicas e configurações iniciais do microcontrolador necessárias para o projeto.

2.4 Pinout

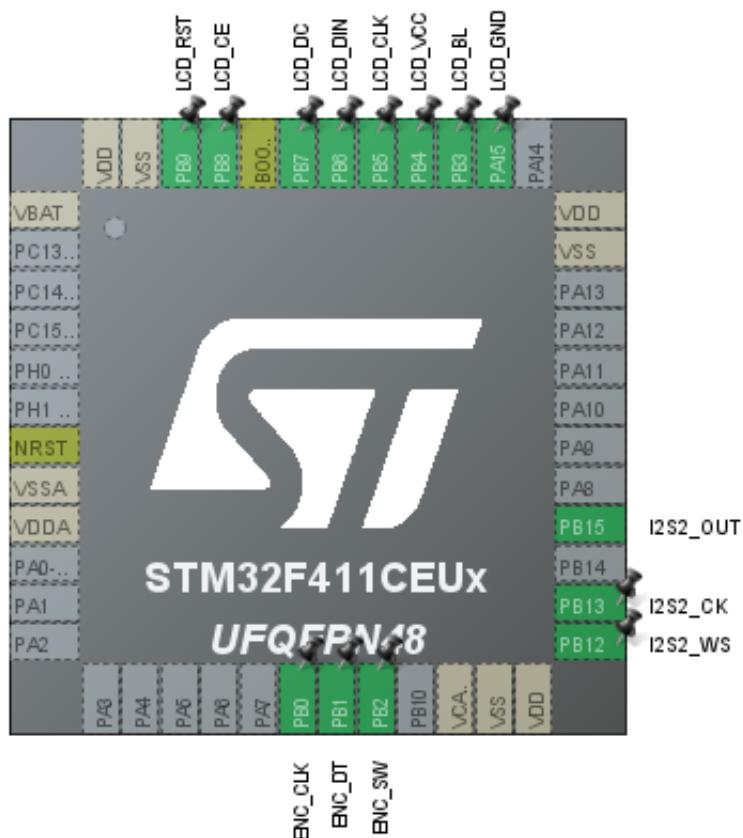


Figura 4 – Pinout do microcontrolador

CAPÍTULO 3

Conversão digital-analógico

3.1 O conversor digital-analógico 1334A

O 1334A é um **DAC!** (**DAC!**) de áudio com comunicação **I2S!**. Neste projeto, foi utilizado um módulo CJMCU-1334 com este conversor. A frequência de amostragem utilizada é 44kHz.

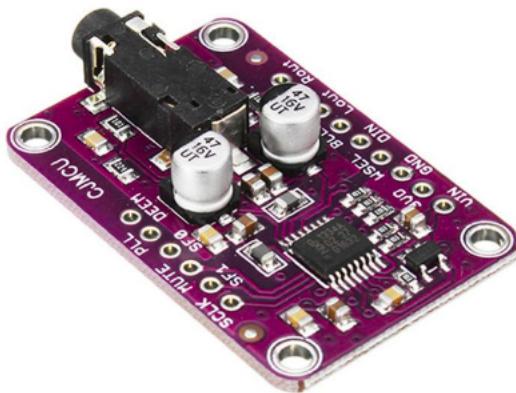


Figura 5 – Módulo CJMCU-1334

3.2 Protocolo I2S

I2S é um padrão de interface de barramento serial usado para conectar dispositivos de áudio digital juntos. A especificação é da Philips Semiconductor.

Este protocolo possui as seguintes conexões:

Bit Clock (BCLK)	O clock que sincroniza a transmissão dos bits de áudio
Word Select (WS ou LRCLK)	Indica se os bits transmitidos são do canal L ou R.
Serial Data (SD)	Fio por onde os bits de áudio são transmitidos.

Tabela 1 – Conexões básicas do protocolo I2S

Neste projeto, o DAC é utilizado em modo *slave*. Neste caso, os sinais BCLK e WS do DAC são gerados pelo MCU e o sinal SD é gerado pelo DAC.

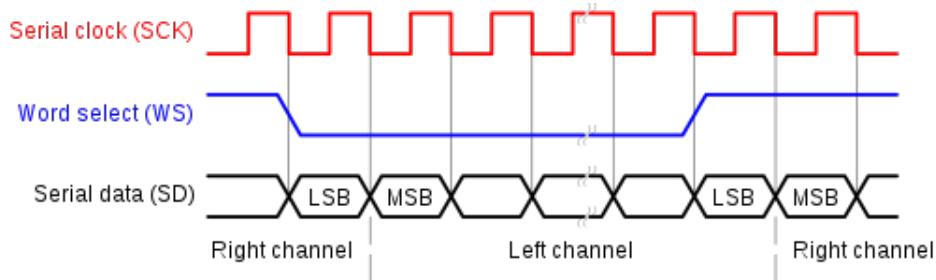


Figura 6 – Sinais do protocolo I2S

3.3 Transmissão com DMA

Para poupar tempo do processador, os dados são transmitidos para o DAC com DMA. Básicamente, o *firmware* registra periodicamente os *samples* de áudio em um *buffer* de 1024 *half-words*, o controlador DMA acessa esse *buffer* e transmite os samples para o DAC paralelamente à atividade do processador. Quando a transmissão dos 1024 *samples* é concluída, uma interrupção de *hardware* é acionada e uma rotina é executada para gerar uma nova sequência de 1024 *samples*.

CAPÍTULO 4

Geração de sinais

4.1 Senoide (SIN)

Esta é a função mais básica do gerador de sinais. Um sinal senoidal contínuo é definido da seguinte forma:

$$s(t) = A \sin(2\pi ft) \quad (1)$$

Onde f é a frequência do sinal em Hz, t é o tempo em segundos e A é a amplitude do sinal.

A fórmula utilizada para produzir o sinal discreto é a seguinte:

$$s[i] = A \sin(2\pi f(i/f_a)) \quad (2)$$

Onde i é o índice do *sample* e f_a é a frequência de amostragem em Hz (44kHz neste caso).

4.2 Onda quadrada (SQR)

Uma onda quadrada pode ser obtida por um somatório de harmônicos. A definição com tempo contínuo é a seguinte:

$$s(t) = A \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin((2k-1)2\pi ft)}{(2k-1)} \quad (3)$$

A formula utilizada para produzir o sinal discreto é a seguinte:

$$s[i] = A \frac{4}{\pi} \sum_{k=1}^H \frac{\sin((2k-1)\frac{2\pi f_i}{f_a})}{(2k-1)} \quad (4)$$

Onde H é a quantidade de harmônicos no sinal.

4.3 Onda triangular (TRI)

Assim como a onda quadrada, a onda triangular pode ser obtida por um somatório de harmônicos. A definição com tempo contínuo é a seguinte:

$$s(t) = A \frac{8}{\pi^2} \sum_{k=1}^{\infty} \sin\left(\frac{k\pi}{2}\right) \frac{\sin(2k\pi ft)}{k^2} \quad (5)$$

A formula utilizada para produzir o sinal discreto é a seguinte:

$$s[i] = A \frac{8}{\pi^2} \sum_{k=1}^H \sin\left(\frac{k\pi}{2}\right) \frac{\sin(2k\pi f \frac{i}{f_a})}{k^2} \quad (6)$$

4.4 Onda dente-de-serra (SAW)

Assim como a onda quadrada e a onda triangular, a onda dente de serra pode ser obtida por um somatório de harmônicos. A definição com tempo contínuo é a seguinte:

$$s(t) = A \frac{2}{\pi} \sum_{k=1}^{\infty} (-1)^k \frac{\sin(2\pi kft)}{k} \quad (7)$$

A formula utilizada para produzir o sinal discreto é a seguinte:

$$s(t) = A \frac{2}{\pi} \sum_{k=1}^H (-1)^k \frac{\sin(2\pi kf \frac{i}{f_a})}{k} \quad (8)$$

4.5 Otimização

Para reduzir o processamento necessário para gerar os sinais, foi utilizada a seguinte técnica de otimização: quando o *firmware* é inicializado, ou quando há uma alteração de frequência, forma de onda ou quantidade de harmônicos, uma rotina é executada para gerar apenas um ciclo completo de onda e armazenar em um *buffer*. A partir disso, a rotina de geração de sinal, que é executada periodicamente para gerar sequências de 1024 samples, faz apenas a replicação do ciclo de onda armazenado no *buffer*.

CAPÍTULO 5

Interface Homem-Máquina

5.1 Encoder KY-040

Um *encoder* rotativo é sensor mecânico responsável por identificar movimentos de rotação nos sentidos horário e anti-horário. O *encoder* KY-040 também conta com *push-button* interno, responsável pela ação *click*. Os movimentos de rotação produzem padrões diferentes de pulsos nos terminais CLK e DT do módulo.

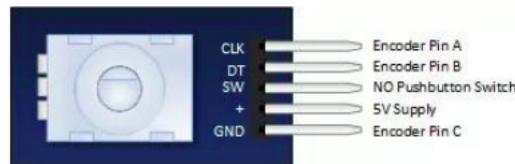


Figura 7 – *Pinout* do módulo KY-040 (fonte: <<https://www.rcscomponents.kiev.ua/datasheets/ky-040-datasheet.pdf>>)

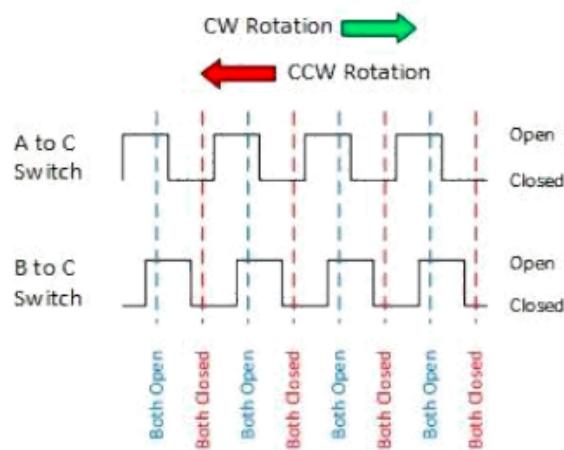


Figura 8 – Padrões gerados pelo módulo KY-040 (fonte: <<https://www.rcscomponents.kiev.ua/datasheets/ky-040-datasheet.pdf>>)

5.2 Estados da IHM

A interface homem-máquina do gerador de sinais foi implementada como uma máquina de estados. Cada movimento do *encoder* (rotação no sentido horário, rotação no sentido anti-horário e *click*) induz uma mudança de estado na IHM. O objetivo desta IHM é permitir que o usuário altere os parâmetros do sinal (frequência, amplitude, forma de onda e quantidade de harmônicos).

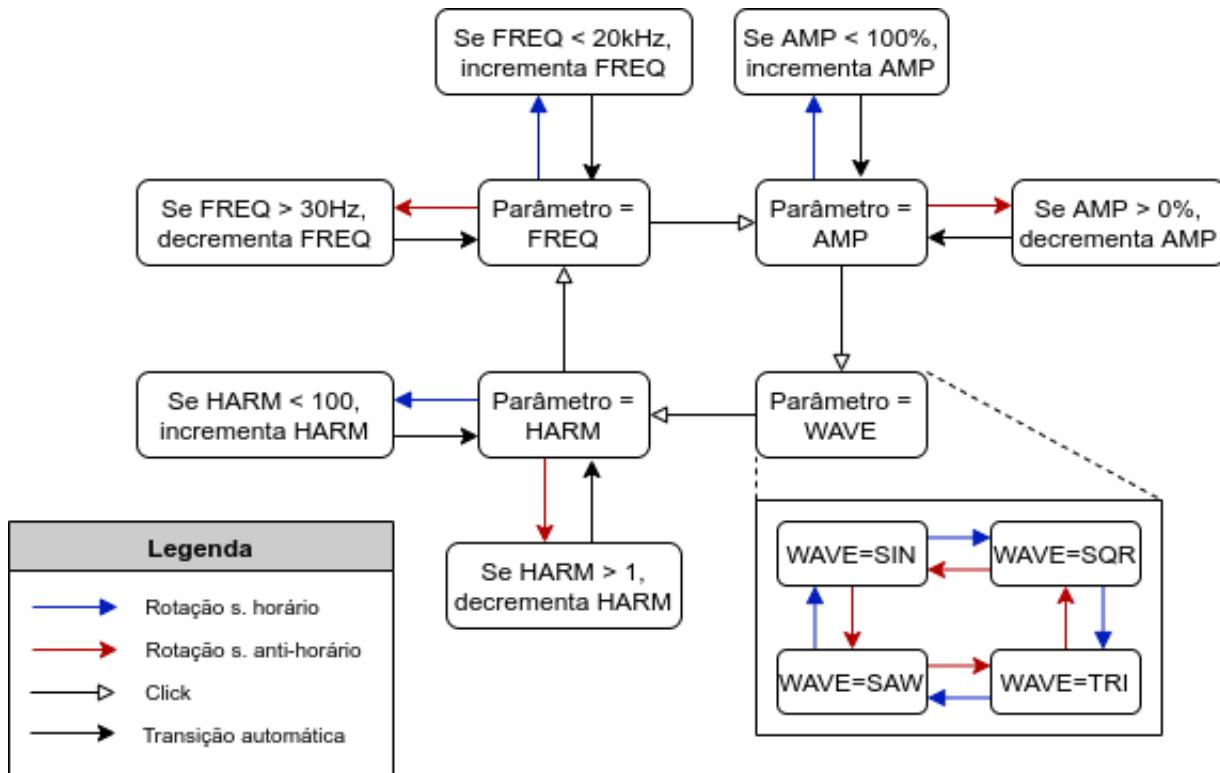


Figura 9 – Diagrama de estados da IHM

Durante a operação do gerador de sinais, o parâmetro selecionado é indicado por um asterisco. O usuário de usar o *click* do *encoder* para alterar o parâmetro selecionado e as rotações para alterar o valor do parâmetro.

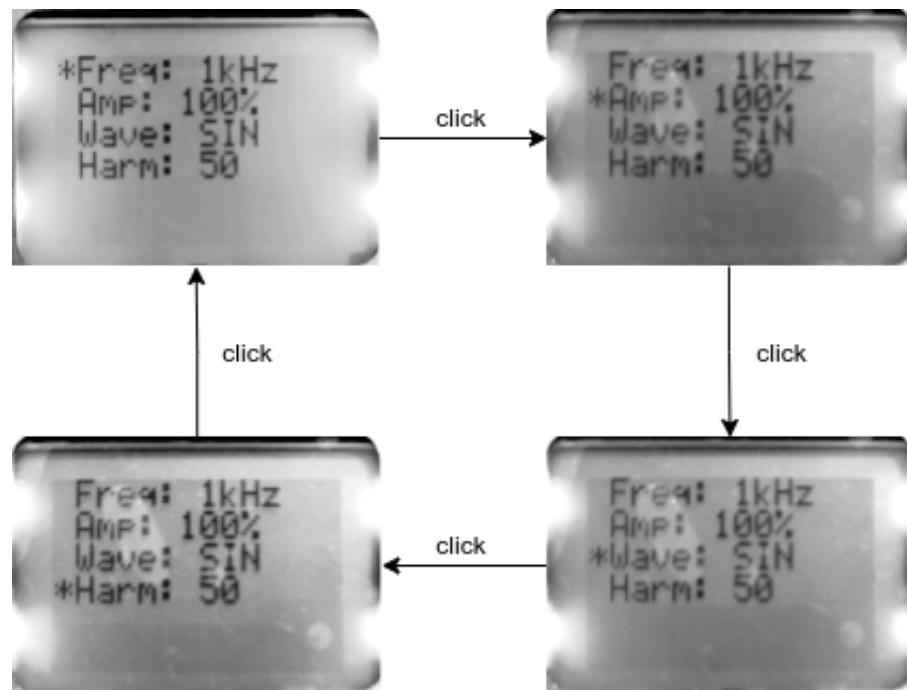
Figura 10 – Fluxograma da IHM com imagens do *display*

Figura 11 – Exemplo de operação de mudança de frequência

CAPÍTULO 6

Conclusão

6.1 Resultados

Seguem alguns exemplos de sinais gerados:

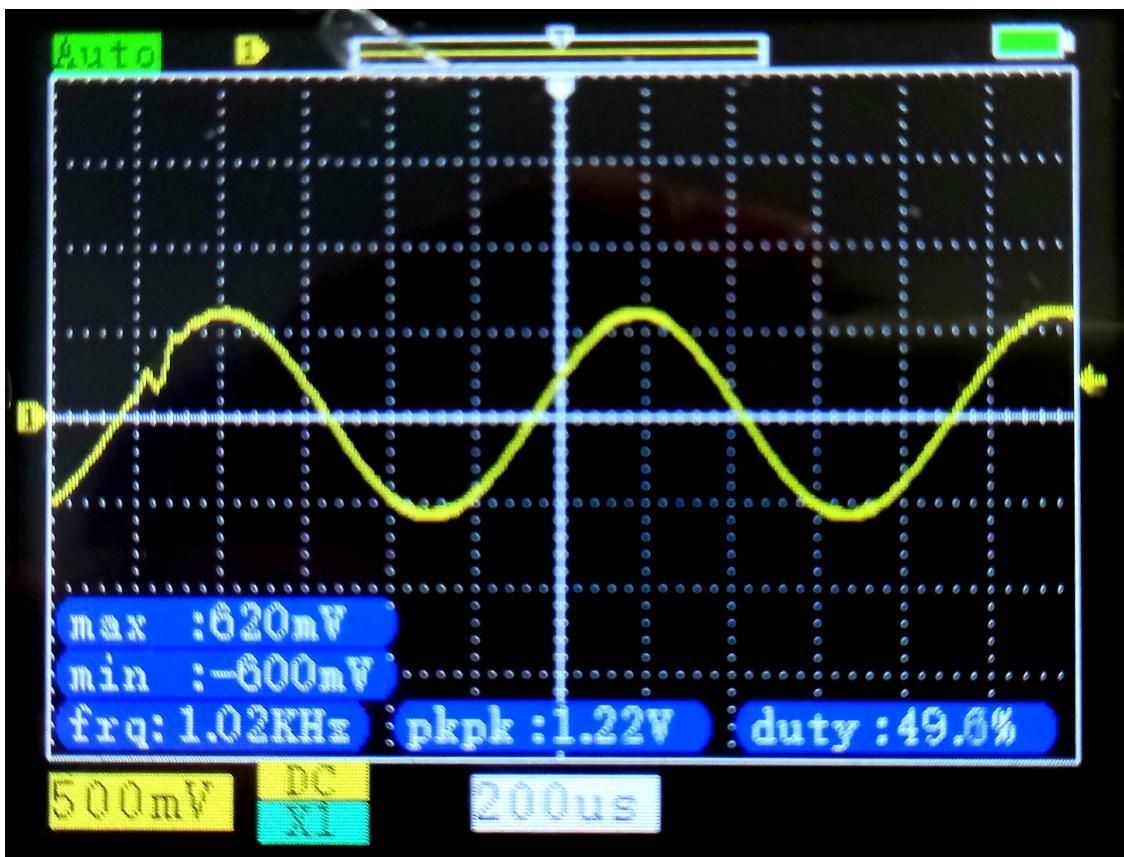


Figura 12 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SIN, HARM=50)



Figura 13 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SQR, HARM=50)

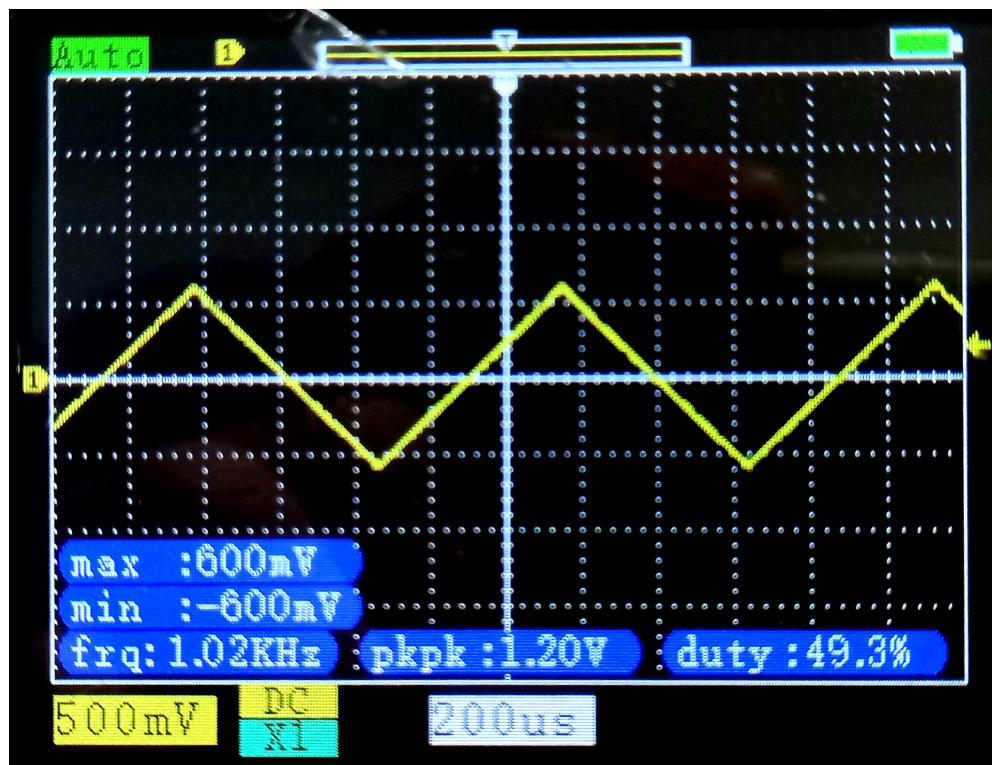


Figura 14 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=TRI, HARM=50)

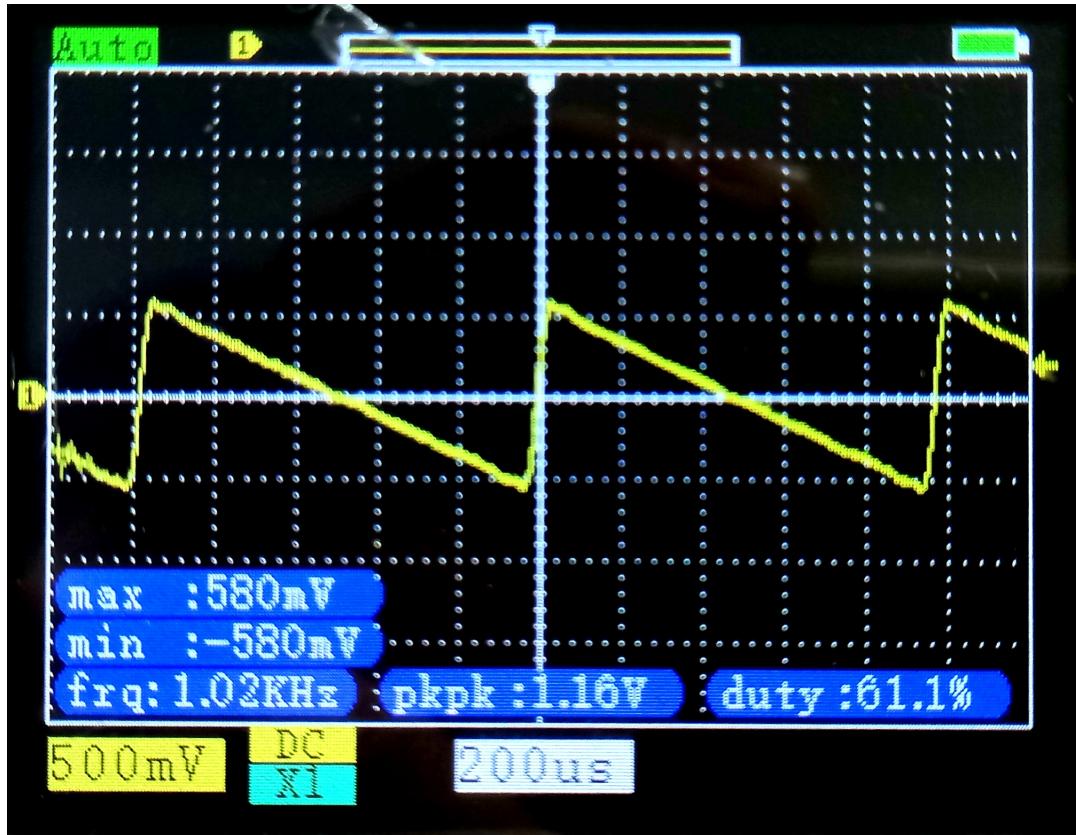


Figura 15 – Exemplo de sinal gerado (FREQ=1kHz, AMP=100%, WAVE=SAW, HARM=50)

6.2 Problemas encontrados

Os principais problemas identificados na IHM foram:

- ruído no *encoder* (que causa falhas no processamento dos movimentos)
- visibilidade ruim no *display*

Estes problemas podem ser resolvidos com o uso de um encoder de melhor qualidade e um display TFT ou OLED.

Na geração de sinal, os principais problemas foram:

- Pequena diferença entre a frequência definida pelo usuário e a frequência gerada
- Componente CC indesejada no sinal
- ruídos causados por falhas no processo de conversão D-A

Estes problemas podem ser resolvidos com o uso de MCUs mais poderosos (como os STM32H7), cálculos mais precisos no software e com um DAC de melhor qualidade.