# Project-Report

## Problem Statement:

**Implement signal generator system.**

1. Define lookup tables for various waveforms.

2. Use interrupt-based push button as user input to change wave shape.

3. Also provide option to change frequency using another external push button.

4. Use LEDs as indicators of operations.

## Algorithm:

*All logical steps involved in implementing the project.*

### I. Initialization:

**Enable the clocks for GPIOA, GPIOC, and DAC.**

**Configure GPIO pins:**

- Set PA4 to analog mode.
- Set PA5 and PA6 to output mode (for LED indicators).
- Set PA7 and PC13 to input mode with pull-up resistors (for buttons).

**Initialize the DAC (Channel 1):**

- Enable DAC and disable its buffer.
- Set DAC trigger to software trigger.
- Reset DAC value.

**Initialize EXTI for PC13 and PA7:**

- Enable SYSCFG clock.
- Configure EXTI13 to use PC13 and EXTI7 to use PA7.
- Unmask EXTI13 and EXTI7.
- Set falling edge trigger for both EXTI13 and EXTI7.
- Enable interrupts for EXTI13 and EXTI7 in the NVIC.

## II. Interrupt Handlers:

### EXTI15_10_IRQHandler:

- Check if EXTI13 triggered.
- Clear pending bit for EXTI13.
- Increment waveform_mode.
- Reset waveform_mode if it exceeds 2.
- Toggle PA5 to indicate waveform change.

### EXTI9_5_IRQHandler:

- Check if EXTI7 triggered.
- Clear pending bit for EXTI7.
- Increase frequency by 10,000.
- Reset frequency to 50,000 if it exceeds 200,000.
- Toggle PA6 to indicate frequency change.

## III. Waveform Generation Functions:

### SineWave:

- Calculate DAC value using sine function.
- Load calculated value into DAC.
- Trigger DAC.
- Increment angle for sine calculation.
- Reset angle if it reaches 360 degrees.
- Call Delay function with frequency.

### SquareWave:

- Calculate DAC value based on step (high or low).
- Load calculated value into DAC.
- Trigger DAC.
- Increment step.
- Reset step if it reaches 4096.
- Call Delay function with frequency.

**TriangleWave:**

- Calculate DAC value based on step (rising or falling).
- Load calculated value into DAC.
- Trigger DAC.
- Increment step.
- Reset step if it reaches 4096.
- Call Delay function with frequency.

**IV. Delay Function:**

- Configure SysTick for the specified delay.
- Start SysTick.
- Wait for the COUNTFLAG to be set.
- Stop SysTick.

**V. Main Loop:**

- Initialize GPIO, DAC, and EXTI.
- Initialize SysTick for 1 ms tick.
- Enter infinite loop:

**Generate waveform based on waveform_mode:**

- SineWave if waveform_mode is 0.
- SquareWave if waveform_mode is 1.
- TriangleWave if waveform_mode is 2.

# Connection details:

| Sl No | Peripheral | Port & Pin No | Type | Description | Mode |
|---|---|---|---|---|---|
| 1 | LED1 | PA5 | OUTPUT | Used to drive LEDs to indicate changes in waveform mode. | OUTPUT |
| 2 | LED2 | PA6 | OUTPUT | Used to drive LEDs to indicate changes in waveform frequency. | OUTPUT |
| 3 | ON-BOARD PUSH BUTTON | PC13 | INPUT | Button to change the waveform type (sine, square, triangle). | INPUT |
| 4 | EXTERNAL PUSH BUTTON | PA7 | INPUT | Button to change the frequency of the waveform | INPUT |
| 5 | DAC (Digital-to-Analog Converter) | PA4 | | Converts digital values to analog signals to generate the desired waveform. | ANALOG INPUT |
| 6 | EXTI (External Interrupt) | PC13 & PA7 | | Triggers an interrupt when the button on PC13 is pressed to change the waveform type.<br><br>Triggers an interrupt when the button on PA7 is pressed to change the frequency. | EXTERNAL INTERRUPT ON FALLING EDGE |
| 7 | SysTick TIMER | | | Provides a delay mechanism to control the timing of waveform generation. | TIMER |

## Future scope:

**A. Enhanced User Interface:**

- LCD Display: Integrate an LCD to display the current waveform type, frequency, and other relevant parameters.
- Rotary Encoders: Use rotary encoders for more precise control over frequency and waveform selection.

**B. Additional Waveforms:**

- Arbitrary Waveforms: Implement the ability to generate user-defined arbitrary waveforms.
- Noise Generation: Include white noise or other types of noise generation for testing and analysis purposes**.**

**C. Data Logging and Analysis:**

- SD Card Logging: Implement data logging to an SD card for later analysis of generated waveforms.
- Real-time Analysis: Add features for real-time analysis and display of waveform characteristics like frequency, amplitude, and harmonics.

## CONCLUSION:

This project successfully shows how to generate different waveforms (sine, square, and triangle) with the STM32F446xx microcontroller. We created a flexible waveform generator by combining the microcontroller's DAC, GPIO, EXTI, and SysTick timer. The solution allows users to interact with the waveform type and frequency via buttons, with LED indications giving visible feedback. This project demonstrates the STM32F446xx's capacity to handle analog signal production and real-time user inputs, making it an invaluable tool for signal processing, testing, and teaching applications.