Comprehensive Analysis of STM32F0-Based Frequency Measurement System

Technical Documentation Team

November 22, 2024

Contents

1 System Overview

1.1 Architecture Specifications

The system is built around the STM32F051x8 microcontroller, operating at 48MHz through PLL multiplication. The core architecture implements:

$$f_{SYSCLK} = f_{HSI} \times \text{PLL}_{\text{multiplier}} = 8\text{MHz} \times 6 = 48\text{MHz} \tag{1}$$

1.2 Core Components

The system comprises the following fundamental blocks:

- **Input Processing Unit (IPU)**
  - NE555 Timer interface (PA1)
  - Function Generator interface (PA2)
  - Mode Selection Button (PA0)
  - Potentiometer ADC input (PA5)
- **Output Processing Unit (OPU)**
  - DAC output channel (PA4)
  - OLED display interface
- **Processing Core**
  - Timer-based frequency measurement
  - Real-time signal processing
  - Mode switching logic

2 Hardware Implementation

2.1 GPIO Configuration Details

The GPIO configuration follows the following register-level setup:

$$\text{MODER}_{\text{PA0}} = 00_2 \text{ (Input Mode)} \tag{2}$$

$$\text{MODER}_{\text{PA1}} = 00_2 \text{ (Input Mode)} \tag{3}$$

$$\text{MODER}_{\text{PA2}} = 00_2 \text{ (Input Mode)} \tag{4}$$

$$\text{MODER}_{\text{PA4,5}} = 11_2 \text{ (Analog Mode)} \tag{5}$$

Register configuration sequence:

Algorithm 1 Detailed GPIO Configuration

RCC $\rightarrow$ AHBENR $\leftarrow$ RCC $\rightarrow$ AHBENR $\vee$ RCC_AHBENR_GPIOAEN
GPIOA $\rightarrow$ MODER $\leftarrow$ GPIOA $\rightarrow$ MODER$\wedge\neg$(GPIO_MODER_MODER0)
GPIOA $\rightarrow$ MODER $\leftarrow$ GPIOA $\rightarrow$ MODER$\wedge\neg$(GPIO_MODER_MODER1)
GPIOA $\rightarrow$ MODER $\leftarrow$ GPIOA $\rightarrow$ MODER$\wedge\neg$(GPIO_MODER_MODER2)
GPIOA $\rightarrow$ MODER $\leftarrow$ GPIOA $\rightarrow$ MODER $\vee$ GPIO_MODER_MODER4
GPIOA $\rightarrow$ MODER $\leftarrow$ GPIOA $\rightarrow$ MODER $\vee$ GPIO_MODER_MODER5
GPIOA $\rightarrow$ PUPDR $\leftarrow$ GPIOA $\rightarrow$ PUPDR $\wedge\neg$(GPIO_PUPDR_PUPDR1 $\vee$ GPIO_PUPDR_PUPDR2)

2.2 Timer Configuration

2.2.1 TIM2 Setup Parameters

The timer configuration implements a free-running counter with the following characteristics:

$$T_{\text{count}} = \frac{1}{f_{\text{SYSCLK}}} = \frac{1}{48\text{MHz}} = 20.83\text{ns} \tag{6}$$

$$\text{Maximum Period} = 2^{32} \times T_{\text{count}} = 89.48\text{s} \tag{7}$$

Timer register configuration:

Algorithm 2 TIM2 Initialization Sequence

RCC $\rightarrow$ APB1ENR $\leftarrow$ RCC $\rightarrow$ APB1ENR $\vee$ RCC_APB1ENR_TIM2EN
TIM2 $\rightarrow$ CR1 $\leftarrow 0x008C$ ▷ Auto-reload, up-count, overflow stop
TIM2 $\rightarrow$ PSC $\leftarrow 0$ ▷ No prescaling
TIM2 $\rightarrow$ ARR $\leftarrow 0xFFFFFFFF$ ▷ Maximum period
TIM2 $\rightarrow$ EGR $\leftarrow 0x0001$ ▷ Update generation
NVIC_SetPriority(TIM2_IRQn, 0) ▷ Highest priority

3 Analog Systems

3.1 ADC Implementation

3.1.1 ADC Calibration Process

The ADC calibration sequence ensures accurate voltage measurements:

$$\text{ADC}_{\text{error}} = \text{ADC}_{\text{measured}} - \text{ADC}_{\text{actual}} \tag{8}$$

Calibration algorithm:

Algorithm 3 ADC Calibration Sequence

```
ADC1 → CR ← ADC_CR_ADCAL
while ADC1 → CR = ADC_CR_ADCAL do
    Wait for calibration completion
end while
ADC1 → SMPR ← 0x7                          ▷ Maximum sampling time
ADC1 → CHSELR ← ADC_CHSELR_CHSEL5
```

3.1.2 ADC Sampling Process

The ADC sampling follows:

$$V_{\text{measured}} = \frac{\text{ADC}_{\text{value}}}{4095} \times V_{\text{ref}} \tag{9}$$

$$R_{\text{measured}} = \frac{\text{ADC}_{\text{value}}}{4095} \times 5000\Omega \tag{10}$$

3.2 DAC Implementation

The DAC output voltage is calculated as:

$$V_{\text{out}} = \frac{\text{DAC}_{\text{value}}}{4095} \times V_{\text{ref}} \tag{11}$$

DAC initialization sequence:

Algorithm 4 DAC Initialization

```
RCC → APB1ENR ← RCC → APB1ENR ∨ RCC_APB1ENR_DACEN
DAC → CR ← DAC → CR ∧ ¬(0x7)
DAC → CR ← DAC → CR ∨ DAC_CR_EN1
```

4 Frequency Measurement System

4.1 Edge Detection and Timing

The frequency measurement process follows:

$$f_{\text{measured}} = \frac{f_{\text{SYSCLK}}}{\text{TIM2\_CNT}} \tag{12}$$

$$T_{\text{measured}} = \frac{\text{TIM2\_CNT}}{f_{\text{SYSCLK}}} \tag{13}$$

Edge detection algorithm:

Algorithm 5 Frequency Measurement Process

if (EXTI $\rightarrow$ PR $\wedge$ register_mask) $\neq 0$ **then**
  if (TIM2 $\rightarrow$ CR1 $\wedge$ TIM_CR1_CEN) $= 0$ **then**
    TIM2 $\rightarrow$ CNT $\leftarrow 0$
    TIM2 $\rightarrow$ CR1 $\leftarrow$ TIM2 $\rightarrow$ CR1 $\vee$ TIM_CR1_CEN
  else
    TIM2 $\rightarrow$ CR1 $\leftarrow$ TIM2 $\rightarrow$ CR1 $\wedge \neg$(TIM_CR1_CEN)
    count $\leftarrow$ TIM2 $\rightarrow$ CNT
    period $\leftarrow \frac{\text{count}}{\text{SystemCoreClock}}$
    frequency $\leftarrow \frac{1}{\text{period}}$
  end if
  EXTI $\rightarrow$ PR $\leftarrow$ register_mask
end if

5 Mode Switching System

5.1 Button Debouncing and Mode Control

The mode switching implements the following state machine:

$$\text{State}_{\text{next}} = \begin{cases} \text{NE555}, & \text{if State}_{\text{current}} = \text{FGen} \wedge \text{Button} = 1 \\ \text{FGen}, & \text{if State}_{\text{current}} = \text{NE555} \wedge \text{Button} = 1 \\ \text{State}_{\text{current}}, & \text{otherwise} \end{cases} \tag{14}$$

Mode switching algorithm:

Algorithm 6 Mode Switching Process

if (EXTI $\rightarrow$ PR $\wedge$ EXTI_PR_PR0) $\neq 0$ **then**
  if (GPIOA $\rightarrow$ IDR $\wedge$ GPIO_IDR_0) $\neq 0$ **then**
    while (GPIOA $\rightarrow$ IDR $\wedge$ GPIO_IDR_0) $\neq 0$ **do**
      Wait for button release
    end while
    funcGen_mode $\leftarrow \neg$funcGen_mode
    if $\neg$funcGen_mode **then**
      EXTI $\rightarrow$ IMR $\leftarrow$ EXTI $\rightarrow$ IMR $\wedge \neg$(EXTI_IMR_IM2)
      EXTI $\rightarrow$ IMR $\leftarrow$ EXTI $\rightarrow$ IMR $\vee$ EXTI_IMR_IM1
    else
      EXTI $\rightarrow$ IMR $\leftarrow$ EXTI $\rightarrow$ IMR $\wedge \neg$(EXTI_IMR_IM1)
      EXTI $\rightarrow$ IMR $\leftarrow$ EXTI $\rightarrow$ IMR $\vee$ EXTI_IMR_IM2
    end if
  end if
  EXTI $\rightarrow$ PR $\leftarrow$ EXTI_PR_PR0
end if

6 System Performance Analysis

6.1 Timing Characteristics

The system timing characteristics are defined by:

$$T_{\text{measurement}} = \frac{1}{f_{\text{input}}} \tag{15}$$

$$\text{Resolution} = \frac{1}{f_{\text{SYSCLK}}} = 20.83\text{ns} \tag{16}$$

$$f_{\max} = \frac{f_{\text{SYSCLK}}}{2} = 24\text{MHz} \tag{17}$$

6.2 Accuracy Analysis

The measurement accuracy is affected by several factors:

$$\text{Error}_{\text{total}} = \sqrt{\text{Error}_{\text{quantization}}^2 + \text{Error}_{\text{clock}}^2 + \text{Error}_{\text{trigger}}^2} \tag{18}$$

Where:

$$\text{Error}_{\text{quantization}} = \pm\frac{1}{2} \times \frac{1}{f_{\text{SYSCLK}}} \tag{19}$$

$$\text{Error}_{\text{clock}} = \pm\text{PPM}_{\text{clock}} \times T_{\text{measurement}} \tag{20}$$

7 Interrupt Priority and Management

7.1 Priority Hierarchy

The interrupt priority system follows:

$$\text{Priority}_{\text{effective}} = \begin{cases} 0, & \text{for TIM2\_IRQn} \\ 0, & \text{for EXTI0\_1\_IRQn} \\ 1, & \text{for EXTI2\_3\_IRQn} \end{cases} \tag{21}$$