**LAB NOTE**

**Subject: Hardware/Software Interfacing**

**Lab 3: ADC**

**Student: Minh Quan Tran**

**Sep 26th, 2024**

Table of Contents

[1. Objectives 4](#_Toc178227413)

[2. Problems and Solutions 5](#_Toc178227414)

[2.1 Problems 5](#_Toc178227415)

[2.2 Solutions 5](#_Toc178227416)

[3. Software Design 6](#_Toc178227417)

[3.1 List of function 6](#_Toc178227418)

[3.2 While loop 6](#_Toc178227419)

[4. Result 7](#_Toc178227420)

**TABLE OF FIGURES**

[Figure 2‑1: Changing Clock Prescaler 5](#_Toc178227425)

[Figure 2‑2: Changing the sampling time 5](#_Toc178227426)

[Figure 2‑3: Changing PCLK2 5](#_Toc178227427)

[Figure 4‑1: Lab3's result 7](#_Toc178227428)

# Objectives

* Using STM32F411 board and STM32CubeIDE in Windows, create code to:
* Read a voltage from one or more ADC-configured STM32 pins
* Show use in a simple application

Advance:

* Code created that:
* Uses 3 ADC pins for input on different channels
  + The different channels will need sampling one at a time, as there is only one physical ADC on the F411 board.
* Using the ADC conversion complete interrupt that signals conversion completion and triggers.
  + Update on the terminal the raw value of the 3 ADCs.
  + Displays the converted raw ADC values in voltage to 1 decimal place accurately.
  + You may optionally use DMA as well.

# Problems and Solutions

## Problems

* Using interrupt with ADC somehow interrupt got execute too often lead to the main while loop can’t be executed.

## Solutions

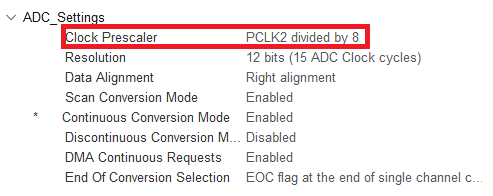
* So far, there are 3 ways to change how frequent the conversion occur:
* Changing the Clock Prescale
* 

Figure 2‑1: Changing Clock Prescaler

* Changing the sampling time

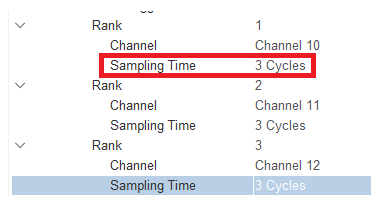


Figure 2‑2: Changing the sampling time

* Changing the frequency of PCLK2

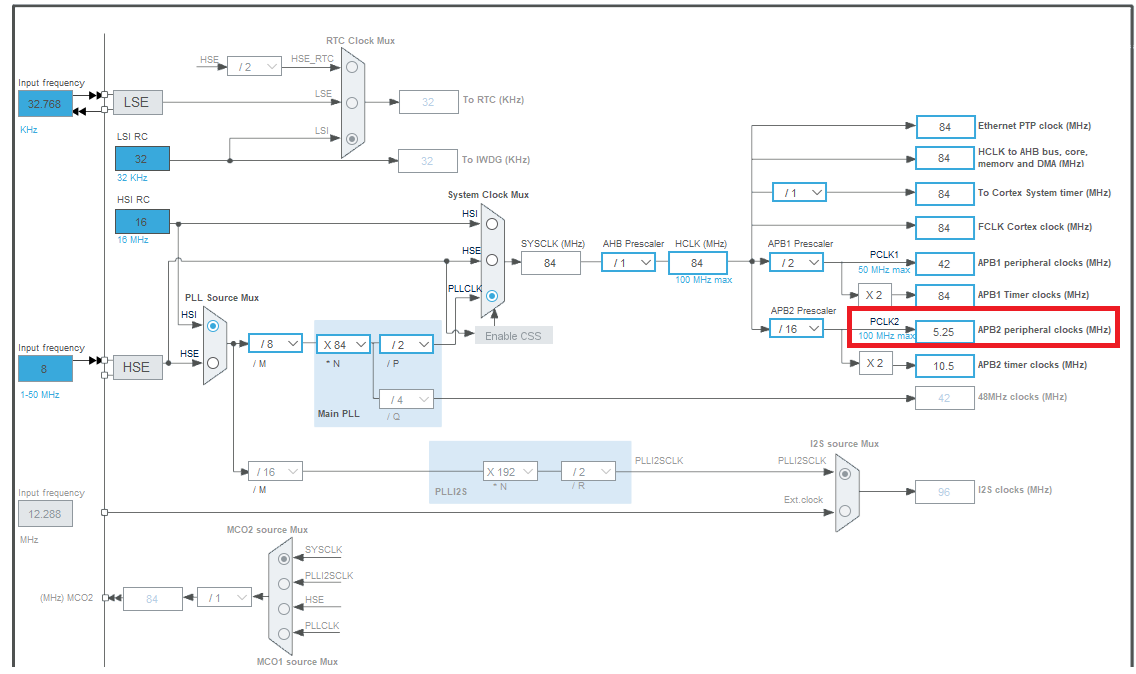


Figure 2‑3: Changing PCLK2

# Software Design

## List of function

* This function is used to convert the adcValue (12 bits value) to voltage value.

**float adcConvertVoltage(uint32\_t adcVal)**

**{**

**return adcVal \* 3.3 / 4095;**

**}**

* This function is used to get the internal temperature of the Nucleo board, the formula to calculate is: ((Vsense - V25)/AVG\_SLOPE) + 25 (refer [STM32F411 reference manual p222](https://www.st.com/resource/zh/reference_manual/rm0383-stm32f411xce-advanced-armbased-32bit-mcus-stmicroelectronics.pdf))
* V25 and AVG\_SLOPE value refers to [STM32F411CE datasheet p120](https://www.st.com/resource/en/datasheet/stm32f411ce.pdf)

**flofloat getTemp(float Vsense)**

**{**

**return ((Vsense - V25)/AVG\_SLOPE) + 25;**

**}**

* This function is used to get the ADC value and convert to Voltage value whenever conversion occur.

**void** **HAL\_ADC\_ConvCpltCallback**(ADC\_HandleTypeDef \*hadc)

{

// Get ADC value from DMA and convert the value to Voltage

adc1Val = buffer[0];

adc1Vol = adcConvertVoltage(adc1Val);

adc2Val = buffer[1];

adc2Vol = adcConvertVoltage(adc2Val);

adc3Val = buffer[2];

adc3Vol = adcConvertVoltage(adc3Val);

tempVal = buffer[3];

tempVol = adcConvertVoltage(tempVal);

}

## While loop

**while** (1)

{

temp = getTemp(tempVol); // get temperature value

**printf**("ADC10: %lu(%.1f), ADC11: %lu(%.1f), ADC12: %lu(%.1f), Temp: %lu(%.1fC)\r\n", adc1Val, adc1Vol,

adc2Val, adc2Vol,

adc3Val, adc3Vol,

tempVal, temp);

HAL\_Delay(100); // Send every 100ms

}

# Result

A screenshot of a computer

Description automatically generated

Figure 4‑1: Lab3's result

**REFERENCES**