Name: Hazem Mohamed Mahmoud

ID:19015567

STM project

Objective:

The objective of this assignment is to create an ADC (Analog-to-Digital Converter) driver for the STM32F401VE microcontroller without using any libraries. The driver will provide two essential functions: analogRead to read analog values from specified pins and analogReference to set the ADC reference voltage.

Introduction:

The STM32F401VE is a high-performance microcontroller with built-in ADC functionality. The ADC is crucial for converting analog signals into digital values for processing within the microcontroller.

Implementation:

Hardware Configuration:

Microcontroller: STM32F401VE

Analog Pins: The assignment assumes a specific set of analog pins, e.g., PA0, PA1, PA2, etc., configured for analog input.

Software Configuration:

ADC Initialization:

The ADC driver initializes the ADC module on the microcontroller using direct register access.

- Enable the ADC clock.
- Configure the desired GPIO pin(s) as analog mode for ADC input.
- Configure ADC settings such as resolution, clock, and reference voltage.

analogRead Function:

The **analogRead** function reads an analog value from a specified pin.

- It selects the appropriate channel for conversion.
- Starts the conversion.
- Waits for the conversion to complete.
- Retrieves the converted digital value.

analogReference Function:

The **analogReference** function sets the ADC reference voltage.

Code used:

```
#include "stm32f4xx.h"

void ADC_init(void);
void analog_reference(int reference);
int ADC_read(int pin);

int ADCdata;
long Output;
int reference = 5;

int main(void) {
   ADC_init();

   while (1) {
      analog_reference(5);
            Output = ADCdata * reference / 1024;
   }
}
void ADC_init(void) {
 RCC->AHB1ENR |= (1<<0);
```

```
   GPIOA->MODER |= (GPIO_MODER_MODER0 |
GPIO_MODER_MODER1 | GPIO_MODER_MODER2);
        RCC->APB2ENR |= (1<<8);
        ADC1->CR1 |= (1<<24);
        ADC1->CR2 &= ~(1<<0);
        ADC1->SQR3 |= 0;
        ADC1->CR2 |=  (1<<0);
}

void analog_reference(int reference) {
   ADC1->CR2 |= ADC_CR2_SWSTART;
   ADCdata = ADC_read(2);
}
int ADC_read(int pin) {
   ADC1->SQR3 = pin & 0x0F;

   ADC1->CR2 |= ADC_CR2_SWSTART;

   while (!(ADC1->SR & ADC_SR_EOC)){}

   return ADC1->DR;
}
```

Conclusion:

The implemented ADC driver for the STM32F401VE provides essential functionalities to read analog values and potentially set the analog reference for the ADC. However, due to hardware limitations, the analog reference setting might not be applicable for this specific microcontroller.

References:

- STM32F401VE datasheet and reference manual.
- ARM Cortex-M4 technical documentation