



POLITECNICO
MILANO 1863

AOS Microcontroller Project

[Light Sensor & ADC Interfacing]

Student Christian Müller & Arno Virtanen
ID 894942 & 894986

Course Embedded Systems
Academic Year 2017-2018

Advisor Federico Terraneo
Professor William Fornaciari

December 27, 2017

Contents

1	Introduction	3
2	Design and implementation	5
2.1	Game logic	5
2.2	Peripheral drivers	6
2.2.1	ADC	6
2.2.2	Push button	7
2.2.3	LEDs & Timer	7
2.2.4	Serial	7
2.2.5	Connections	7
3	Experimental evaluation	7
3.1	Experimental setup	7
3.2	Results	7
4	Conclusions	7

1 Introduction

The goal of the project work was to implement a peripheral driver for the STM32f407 microcontroller. The implemented driver enables the use of the analog-to-digital converter (ADC). The microcontroller which contains the analog-to-digital converter is located on the development board (STM32 F4 Discovery kit) shown in figure 1. This development board contains components that can be interfaced with the microcontroller.

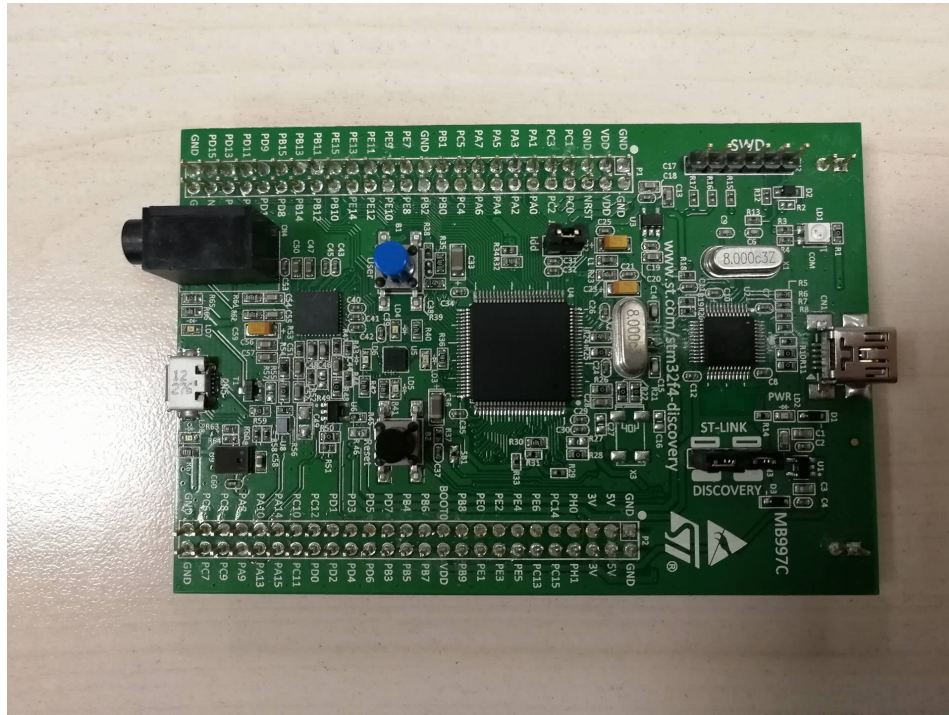


Figure 1: Discovery Board

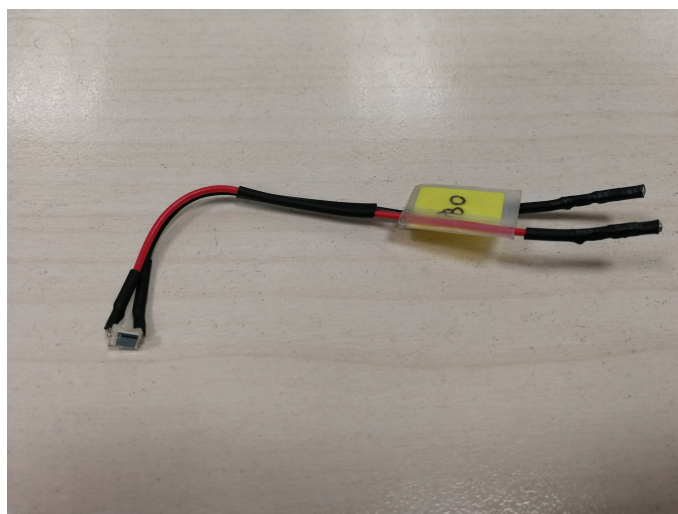


Figure 2: Photodiode

In this work the push buttons and LEDs were utilized. An external photodiode shown in figure 2 was

connected to the development board connection pins. These components were used to make a simple user reaction game.

2 Design and implementation

2.1 Game logic

The idea of the implemented game was to test the reaction time of a player. This game can be played by a defined number of players. The reaction time of a player is evaluated by indicating a start signal and after this signal the player has to cover the photodiode as fast as possible. The game logic is illustrated in figure 3.

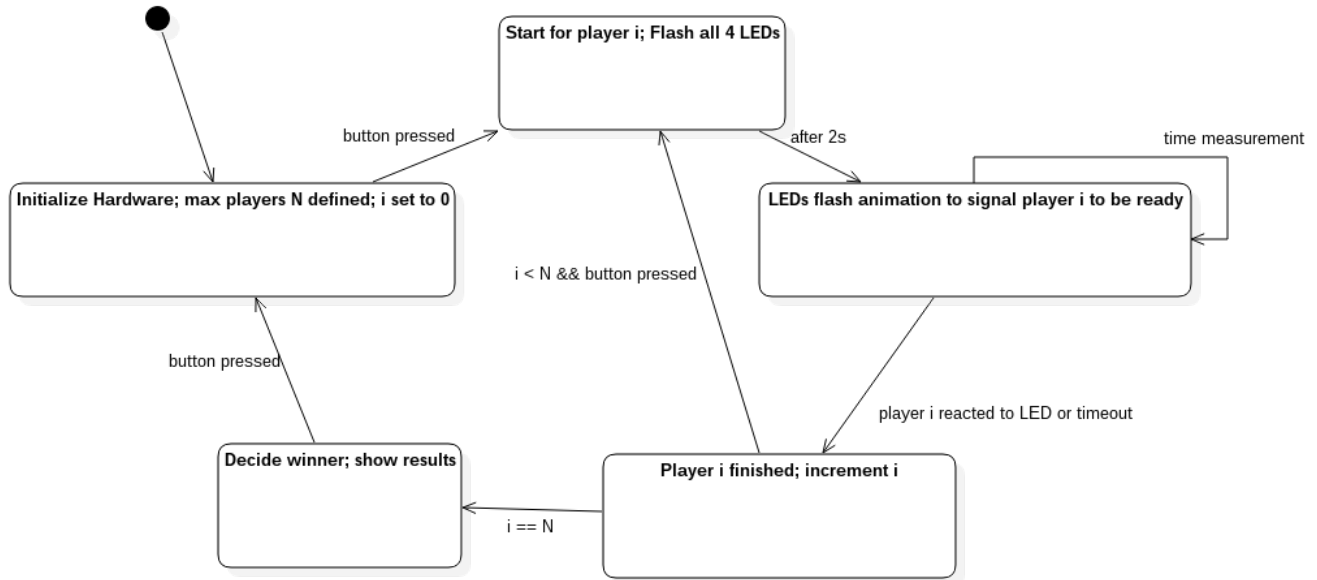


Figure 3: Game logic diagram

As illustrated in figure 3 the game is started after the hardware initialization has been completed. This is done when the power of the board is turned on and the board's push button has been pressed. Information about the game status and corresponding instructions are printed on a serial monitor. These instructions are shown in chapter 3.

After the push button has been pressed it is the first player's turn and all of the user programmable LEDs start to flash rapidly for a short duration and then turn off. This is an indication to the player to get ready to get ready. After this the LEDs start to light up in a clockwise orientation one by one. Once the fourth LED has lightened up, the player has to cover the photodiode as quickly as possible. Player reaction time is measured from the fourth LED lighting up and to the point when the photodiode has been covered.

Then another player starts their turn by pressing the push button and has to interact in the same manner as the player before. This cycle is repeated until all of the players have had their turn. The reaction times are finally displayed and the player who had the lowest reaction time wins the game. A new game can be started by pressing the push button.

2.2 Peripheral drivers

For simplicity a design choice for this microcontroller project was to avoid the use of threads. Every game logic procedure is executed in a sequential order and only interrupts allow to break this cycle and trigger events concurrently.

In this project, interrupts are used by timers to toggle LEDs and for button detection. For the analog watchdog polling is utilized. This approach allows to sense if the user tries to cheat by covering the photodiode before the fourth LED start to flash. As an alternative to this it would be also possible to realize the watchdog with interrupts and toggle the LEDs with sleep delays.

The Miosix operating system ([1]) is included in the project workspace and is mainly used for a simplified accessing of the STM32F4 peripheral addresses by calling provided register names. Also the *Timer* class for measuring time and the *Random Number Generator* are exploited.

2.2.1 ADC

The control of the analog-to-digital converter builds the core of the project. A simple photodiode is connected to the board and introduces an analog input voltage level depending on the ambient light intensity. The voltage has to be detected and read out as a digital signal in defined time intervals. Since the actual values are not important for this project, there is no data processing necessary. The values simply have to be compared to a threshold voltage and if they exceed this threshold, further action should be initiated. The analog watchdog provided by the microcontroller is used to execute this comparison.

To enable the the analog-to-digital converter peripheral and to connect the general purpose input/output (GPIO) pin to the ADC peripheral a set of bits were needed to be set in different registers. The desired implementation was to connect the light sensor to the first GPIO of port B (PB0) and connect that GPIO to the ADC.

First the GPIO port where the light sensor was connected needs to be enabled. Since the light sensor was connected to PB0 pin, the port B clock must be enabled. This is achieved by setting the GPIOBEN bit in the RCC AHB1 peripheral clock enable register. In order to connect the GPIO to the ADC it was needed to configure the GPIO port to analog mode of the four possible operation modes. This was done by writing value of 11 to number of the pin number (0) in the GPIO port mode register.

The ADC peripheral is setup after connecting the GPIO port to ADC. From the user manual of the development board [?] the used GPIO could be connected to first two ADC peripherals of the three available ADC peripherals of the microcontroller and connected to the input channel 8 of the ADC. It was decided to use the ADC1 of the microcontroller thus the corresponding clock was needed to be enabled from the RCC APB2 peripheral clock enable register.

The following functionality of the ADC was chosen:

- Resolution: 12 bits
- Prescaler: PCLK2/8
- Sampling time of input channel (8): 84 cycles

- Data alignment: right data alignment
- Conversion mode: continuous conversion mode

All functionalities stated above could be set specifically for the ADC1 register except the prescaler value that was set in the common ADC register. Also the watchdog functionality of the ADC is utilized to see if the converted value is below a certain threshold that is calibrated everytime the ADC is initialized (board is powered on or reseted). The use of the watchdog functionality can be set specifically to be used in one of the three ADCs.

2.2.2 Push button

2.2.3 LEDs & Timer

Since the toggling of the LEDs is realized with interrupts, the LED management explicitly includes a hardware timer. The *Led* class knows three different operating modes: At the beginning of a game round all four LEDs flash rapidly, after that the number of blinking LEDs increase from 1 to 3 and finally after a random time interval also the fourth LED becomes active. For all these three modes only one timer (TIM3) needs to be utilized because the blinking procedures happen sequentially.

For every sequence the function *startLedTimer()* is called with a different time interval which sets the prescaler of the timer. Also the new LED mode is passed with the function. With the given mode every time the timer generates an overflow interrupt, the corresponding IRQ handler knows in which way the LEDs should be toggled.

2.2.4 Serial

2.2.5 Connections

The connections made are shown both in table 1 and figure 4.

3 Results

The operation of the system was tested multiple times in order to test the functionality of the system and also in different locations in order to test the analog watchdog threshold set correctness. After testing the system in different locations the system was operating similarly everytime and thus it was concluded that the system is working correctly.

4 Conclusions

This project was about implementing an analog-to-digital converter driver for the STM32f407 microcontroller and utilizing it with a light sensor. Also the STM32 F4 Discovery kit development board was utilized to produce a user reaction time game with using the boards LEDs and pushbutton.

The most emphasis was on implementing the ADC driver. The most challenging part of constructing the driver was how to interface the used GPIO and the used ADC register.

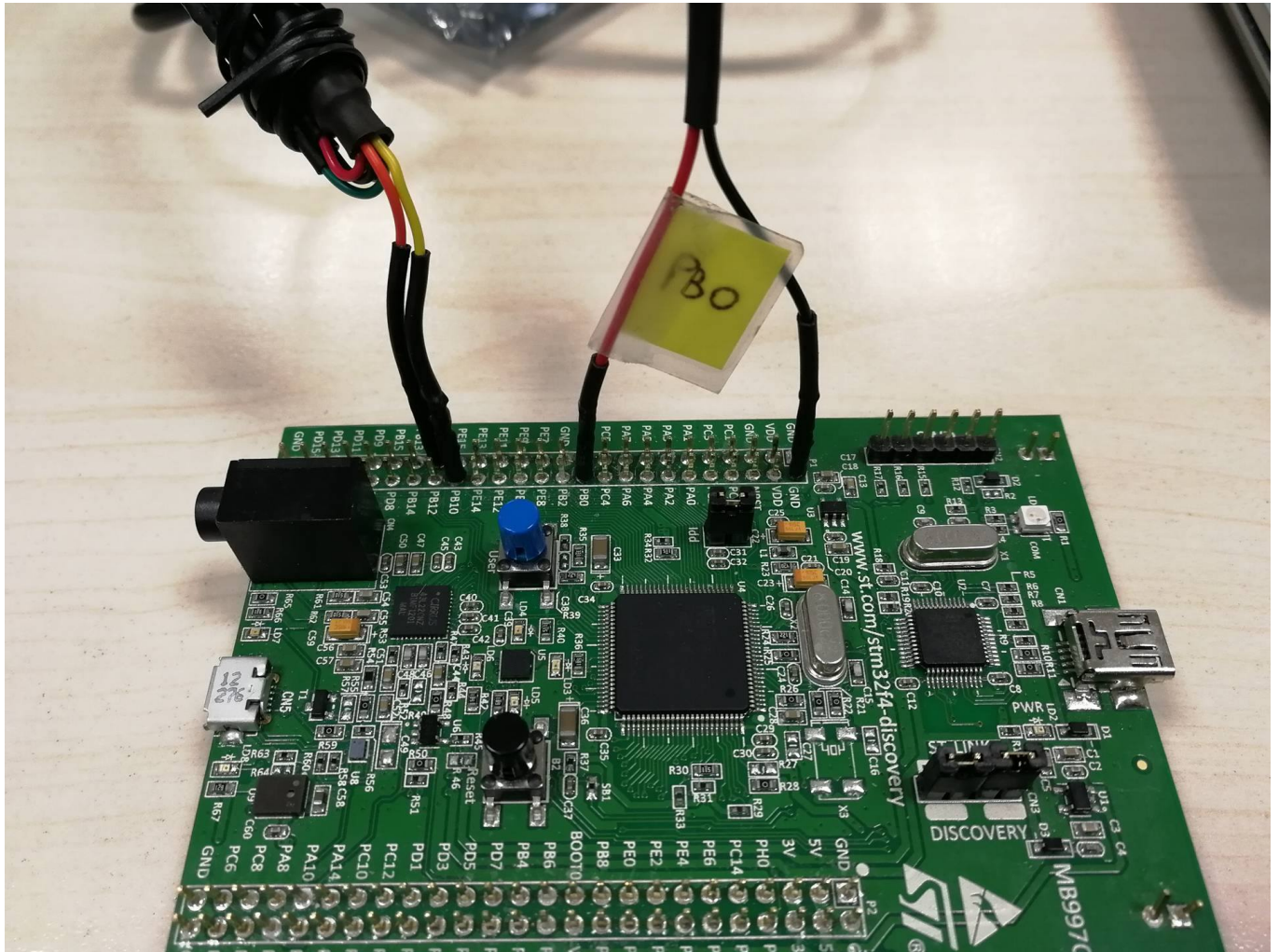


Figure 4: Connections

References

- [1] Federico Terraneo. Miosix operating system. Website. <https://miosix.org/index.html>.
- [2] STMicroelectronics. Reference Manual RM0090 for STM32F407. 2017.