AOS project

A.Y. 2015/2016

Politecnico di Milano

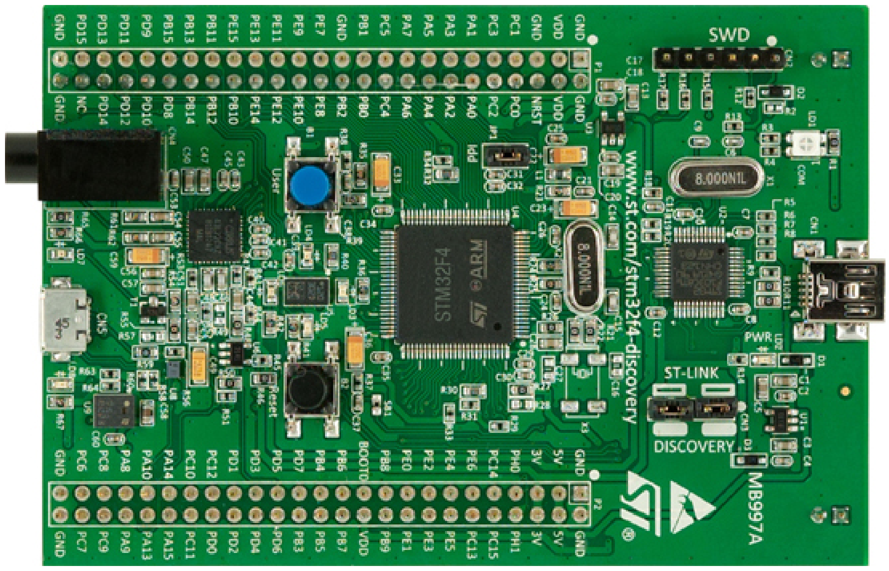
Barlocco Mattia, matr. 873801
Belotti Nicola, matr. 793419
Colombo Andrea, matr. 853381

September, 2016

Contents

1 Introduction

The goal of this project is to recognize which button of a remote control is pushed by using the BPW34 photodiode and the stm32f407. This board has a Analogic to Digital Converter (ADC) built-in that converts the anologic signal received from the BPW to a digital signal. This board operating system is Miosix.



2 BPW34

The BPW34 is a photodiode that is sensitive to visible and infrared radiation. Its carateristics are the following:

- It is sensitive at most 34KHz
- Fast response times
- Angle of half sensitivity: = 65
- High photo sensitivity
- Suitable for visible and near infrared radiation

For any other information on this photodiode visit: `http://www.vishay.com/docs/81521/bpw34.pdf`

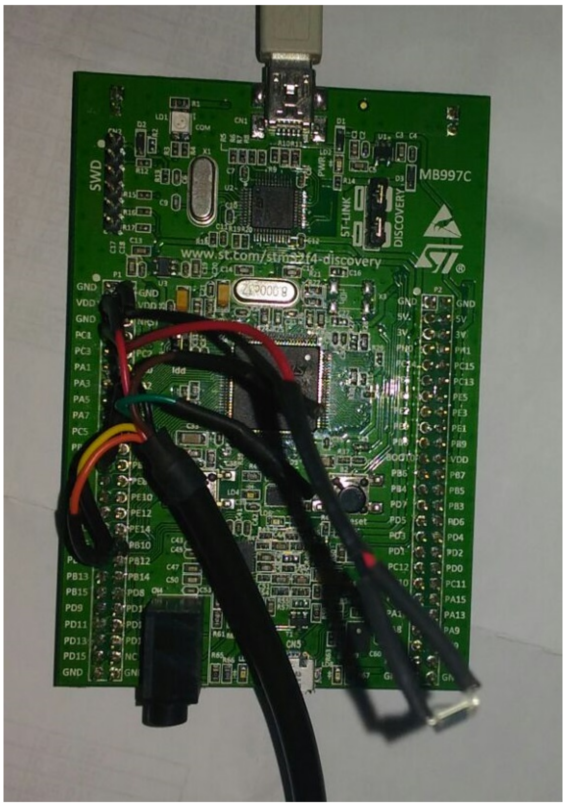
3 How it works

3.1 Board configuration

Used pins are the following:

- GND: BPW34 black wire
- GND: FTDI black wire
- PC1: BPW34 red wire
- PB10:FTDI yellow wire
- PB11:FTDI orange wire
- VDD: FTDI red wire

The BPW34 sends the value that it sees to the pin PC1. This pin is connected to the ADC, it converts the value in digital so we can use it easily.
The FTDI is used to print the values, that the photodiode receives, on the Arduino serial monitor.



3.2 Code description

We wrote three files of code:

- main.cpp: source file where we wrote the main
- adc.h: header file where we define the new class Adc with the constructor, the method start() starts a conversion and the method read() waits for the end of the conversion and it returns the digital value
- adc.cpp: source file where we wrote the code of the method defined in the adc.h

In the main.cpp the program starts and it takes 200 values of the light of the surrounding space to fix a threshold. After that it monitors the light, and when it receives a value greater then the treshold, it starts to save 30000 values in a buffer (the results are illustrated in the next chapter). In this buffer is memorized the button code of the remote control that we pressed. Whenever we press the blue button of the board, if the program is not saving values in the buffer, we calculate a new threshold.

3.3 Results

The goal has not been reached. However we have been able to see whether a button is pushed or not.
Obviously our project works when a consumer infrared is used, such as remote control for a tv, or a dvd player.
We found our system doesn't works on home gate remote controls or car keys because those kind of remote controls use signals out of the infrared trasmission(such as Megaherz transmission).
Following images show our sampling on different buttons of the remote control Samsung AK59-00149A.

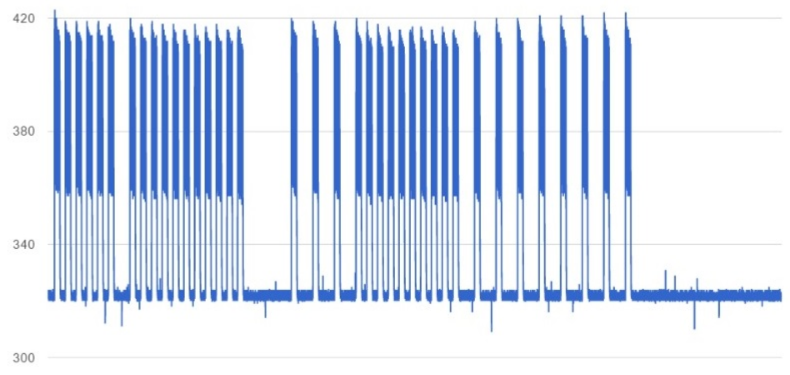


Figure 1: button "ON/OFF"

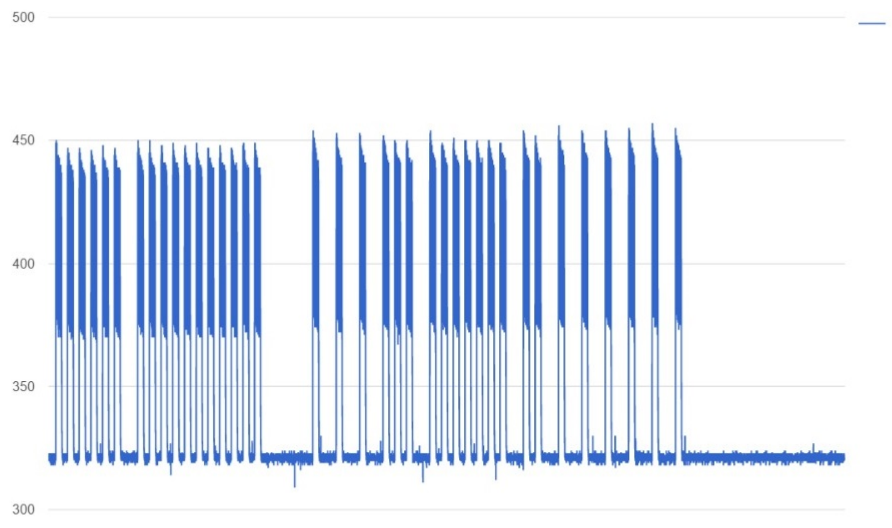


Figure 2: button "Channel 1"

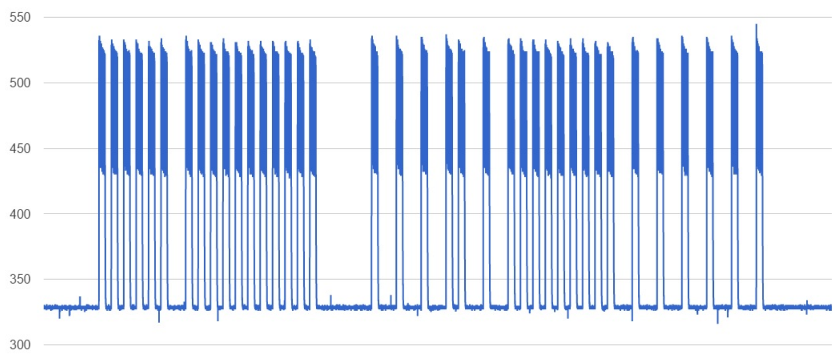


Figure 3: button "Channel 2"

How we can see there exist a part at the beginning of the code, the header, that is equal for every button of the same remote control, but the central part is different to recognize several button.

4 Used Software

- QSTlink2: used to program the board.
- Github: to save every code changes online.
- Latex: to write this document.
- Notepad++: to write code.
- Miosix Toolchain: to compile the project
- ArduinoIDE: serial Arduino monitor
- Spread sheets of Google Drive: to make graphs