Working Electrical Engineering Projects with STM32 + Fun Kit

Daniel Beatty

October 24, 2020

Abstract

Abstract

1 Introduction

Why am I doing this? I have some examples provided by a client of mine. These might have worked in some STM32 development environments. However, software evolves to meet hardware and security needs. I suspect that I am not alone.

I would prefer to have a cross platform means to develop software. I would prefer the ability to analyze the kernels deployed on these devices. I would like to ensure that I can deliver a good product, and apply micro and macro architectures (aka design patterns) as well good algorithms.

So, I want to know how this mBed system work. I will start with the STM32 product line to develop this micro-architecture approach.

I plan to demostrate these techiniques on MacOS, Windows, and Linux (ARM).

2 The Base Line Integrated Development Environment Part 1

The Eclipse Foundation and Advanced RISC Machines(ARM) consortium work together to produce an Integrated Development Environment(IDE)[1] to fulfill needs of micro-controller developers. The development community needs build services and IDE systems to handle the complex builds.

The Advanced RISC Machines (ARM) Limited developed the mBedOS to exhibit standard features amongst various manufactures builds. Thus, I can develop a simple program and compile it for many different boards, and it still works.

Mbed Studios (a modified Eclipse Theia) released in June 2019. It is derived from the Eclipse 'Theia' line shown in a YouTube video[2].

ARM made the mBed Suite [3] based on Eclipse Theia [4]. This IDE supports inclusion of the mBed OS itself in a build and any additional libraries required to construct a software product for the ARM based boards. These libraries include:

- 'Tiny interactions'
- Communication mechanisms
- Hardware Abstraction Layer(s)

ARM supplies a tutorial on installing the mBed Theia environment on MS Windows 10[5]. We will examine Mac and Windows on Intel processor machines. For example, I develop embedded software on a MacBook Pro and also demonstrate with a Mac Min running MS Windows with bootcamp. This work also seeks to produce Raspberry Pi and iPad tools to augment the STM Micro-controllers.

2.1 ST Microelectronics Libraries

ST Microelectronics started as a French company. It operates out of many nations. In addition to producing the micro-controller boards themselves, they also produce support libaries for their boards[6]. For example, ST Microelectronics is not the first manufacturer to produce ARM based digital signal processor (DSP)[7]. DSP engines provide a similar acceleration to vector / array math

[1] https://os.mbed.com
[2] https://www.youtube.com/watch?v=HsTtzqL-GP8
[3] https://os.mbed.com/studio/
[4] https://youtu.be/NLkQzx6rrnU
https://www.youtube.com/channel/UCXu7pV552EkR99mCzjwGvTQ/featured
[5] https://os.mbed.com/teams/ST-Americas-mbed-Team/wiki/Getting-Started-with-mbed-and-the-STM32F
[6] http://www.emcu.it/STM32F4xx/STM32F4-Library/STM32F4-Library.html
[7] https://www.st.com/content/st_com/en/products/embedded-software/mcu-mpu-embedded-software/stm32-embedded-software/stm32-standard-peripheral-libraries/stsw-stm32065.html

operations that a graphics processor units (GPU) provides to two and three dimensional arrays. In some cases, GPU systems provide the DSP and general purpose accelerated capabilities all in one facility.

2.2 Hardware Abstraction Layer(s)

ST Microelectroincs and mbedOS supplies Hardware Abstraction Layer(s) to access key MCU capabilites. The mbedOS is not the only real time operating system (RTOS) for micro-controllers. ARM's mbedOS does support many implementations of its chip set [8] and builds into a tidy size for each set it supports.

The mBed website carries white paper references for each of the boards it supports. For example, this piece focuses on the Nucleo-F413ZH[9]. This platform and others can be found at [10]. Alternative cards include:

- Discovery F413H [11]
- Nucleo L4R5ZI-P [12]
- Nucleo WB55RG [13]

Some of the main library sources from STM are provided on the STM32f4 Discovery web site[14] forum.

2.3 It is about the Libraries, Shhsss.

For any task associated with a micro-controller, ARM based micro-controllers require the libraries. ARM micro-controllers require the frameworks of the RTOS itself, in this case mBedOS.

Developers refactor their code as they get into development. These libraries serve as building blocks. This product explores basic library creation to benefit fellow developers and makers of products.

As we create libraries and programs, it behoves us to consider license programs and how to market our productions. We want the fruits of our labor to bring us profit. It might if we can attract people to use it and honor our contributions. It certainly won't if we hoard it or never make it.

So we explore creating programs, importing existing libraries, and creating new libraries. Documentation of such products contributes to fellow makers utilizing our work to make things and solving problems. We make money from establishing those connections and making the software that provide things their connection to real life.

[8] https://os.mbed.com/questions/53876/CMSIS-vs-STM32CUBEHAL-vs-MBED/
[9] https://os.mbed.com/platforms/ST-Nucleo-F413ZH/
[10] https://os.mbed.com/platforms/
[11] https://os.mbed.com/platforms/ST-Discovery-F413H/
[12] https://os.mbed.com/platforms/ST-Nucleo-L4R5ZI-P/
[13] https://os.mbed.com/platforms/ST-Nucleo-WB55RG/
[14] http://stm32f4-discovery.net/2014/05/all-stm32f429-libraries-at-one-place/

2.3.1 Create a Repository

We present software creation from a independent producers point of view. Many corporations and government entities insist on their own repository tools. For these entities, we can comprehend the need of such restrictions. They have assests to protect and the finances to provide on property protection.

2.3.2 Creating A Program

The mBed Suite documentation provides an excellent how-to guide on creating a program [15]. This process works out in a very straight forward way.

2.3.3 Importing A Program

2.3.4 Importing Libraries

[16]

2.3.5 Creating Libraries

3 Basic Hard Tools

3.1 Work on Timer

In cases where we ask the process to wait, we should ensure that the I/O pipe empties. Otherwise, we have no idea when this process takes place. It can occur in the wrong times.

There are third party USB drivers such as [17].

Also, the timer's have issue with floating point numbers.

Lastly, the STDIO has trouble with printing floats.

3.2 Flushing The Standard Input Output

3.3 Blinky

This is a basic hello world type program. A programmer can build this blinky world program for the on board LED or a circuit example.

3.4 Eight Bit Shift Register

Eight bit Shifter IC 74HC595

[15] https://os.mbed.com/docs/mbed-studio/current/create-import/index.html

[16] https://os.mbed.com/docs/mbed-studio/current/manage-libraries/index.htmlimporting-a-library

[17] https://os.mbed.com/users/gte1/code/STM32_USBDevice/

4 Analog to Digital Conversion

4.1 Simple Analog In - Voltmeter

The terms simple and analog rare work together in practice. This work shows examples to highlight how the analog to digital converter (ADC) works, distinction when an optical isolator, and safety mechanisms to protect your micro-controller from damage.

I learned to look up the pin name and its location on the board. There pin names that don't necessarily correspond to the label next pin itself. The A0 pin on the CN9 pin set is actually PA_3. It is easy to make the mistake in calling this pin PA_0. This distinction can lead to massive mistakes and cause to operate the board in a wrong manner.

To demonstrate simple ADC in, I use a simple voltage divider circuit with three resistors. Voltage is proportial to resistance. Therefore, we can measure voltage across a resistor and expect its value. That proportion should change if we can the resistance of one of the other resistors.

Therefore, I establish this measurement across a constant 330 ω resistor. Then, I do similar across a 330 ω potentiometer. We see the voltage go from V_{in}. This helps establish a baseline for comparing the ADC readings.

I built the circuit with parts acquired from [18]. The fun kit [19] provides many of the resistors I need. The UNO Project starter [20] provides some addition parts that I can use later. Also the UNO Project starter provides a Arduino UNO that I can use for comparison. Lastly, I used some competitor products to provide the potentiometers and inductors that I need for future examples.

Also, it is important to note a fix on stdio. Mbed.os by default turns off floating point parameters in stdio/UART[21]. There is a parameter in platform/mbed_app.json. By default, a new mbed project has the value for 'minimum-printf-enable-floating-point' set to false. In order to print out floating point, this value needs to be set to true.

With this, we can see in the example that the voltage read from the potenimeter is pretty close to what a common multimeter reads. There are adjustment factors that we apply to give a rough calibration. For production, it would help to obtain a more calibrated set of values.

We see a similar case with thermister based thermometers. I set up a simple 330ω resistor in series with a thermister out of the fun kit. Here we can measure the voltage across the thermister. We then can measure ratios and offsets for

[18] http://www.rexqualis.com/products/

[19] http://www.rexqualis.com/product/electronics-component-fun-kit-w-power-supply-module-male-to-female-jumper-wire-830-tie-points-breadboard-precision-potentiometer-resistor-for-arduino-raspberry-pi-stm32/

[20] http://www.rexqualis.com/product/uno-project-super-starter-kit-for-arduino-w-uno-r3-development-board-detailed-tutorial/

[21] https://github.com/ARMmbed/mbed-os/blob/master/platform/source/minimal-printf/README.md

temperature.

$$T = \frac{(mV - 225.0)}{10.0} \tag{1}$$

4.2 Change Flashing Rate with Potentiometer

4.3 Thermostat

4.4 Light Meter

4.5 Thermo-resistor -Taking the Temperature

This is an example of an Analog in problem

4.6 Sound Level Meter

5 Analog Out

5.1 Fixed Voltage

5.2 Sawtooth

5.3 Sine Wave Generator

5.4 Combining the Capabilites of Each Wave Generator Type

5.5 Buzzer Fun

5.6 Pulse Width Modulated Buzzers

6 Optical Isolation

I remember a lesson from my embedded computing class back in the Spring of 1998. Professors Darrel Vines, Michael Parton, and Mike Giesselman told us over and over about the importance of optical isolation. These professors worked in pulsed power, and I wish I had the wisdom back then to realize just how important this subject is.

Pulsed Power delivers a large amount of current in a very short space and time. The amount of current involved can easily melt a microcontroller and destroy it. Yet, these controllers are essential in regulating such power elements to fulfill their purpose.

The fun kit comes with a 4N35 white paper can be found at[22]. The inputs on pins 1 and 2 can be thought of as in and out for a light emitting diode (LED). The light from the LED feeds a photo-transistor that can handle much higher currents.

[22]https://www.digchip.com/datasheets/parts/datasheet/161/4N35-pdf.php

7 Transistor Circuits

PN2222
Transistor

8 Ethernet Web Server Libraries

[23]

9 Digital Signal Processing

[24]

10 Secure Digital Block Devices

11 Home Made RADAR

12 Registering the Serial Number

[23] https://github.com/khoih-prog/EthernetWebServer_STM32
[24] http://www.emcu.it/STM32F4xx/STM32F4-Library/STM32F4-Library.html