Arduino and STM32-Discovery

Xu FENG, Minyi GU, Tianming GUO

RICM 4, March - April, 2013
Outline

1 Context
   1.1 Arduino
   1.2 The Maple project
   1.3 STM32 microcontrollers
   1.4 Objective of the project
   1.5 Tools
      1.5.1 STM32F0-Discovery board
      1.5.2 Arduino UNO board
      1.5.3 OpenOCD
      1.5.4 ARM-NONE-EABI

2 Evaluation of the Libmaple

3 Evaluation of the Arduino lib

4 Compilation template

5 Adaption of Arduino IDE for STM32

6 Conclusion

Bibliography

Appendix A
1 Context

1.1 Arduino

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. Arduino boards are equipped with 2 types of microcontrollers: AVR and SAM. These microcontrollers are produced by Atmel. Compared to the AVR microcontrollers, SAM is an ARM-based solution, for example, the microcontroller on the latest Arduino DUE. The Arduino IDE is an environment to program the Arduino boards. Its 1.5.x+ versions which support the Arduino DUE board are compatible with Cortex-M.

1.2 The Maple project

The Maple project, which was launched by MIT, is based on Cortex-M3 microcontrollers which have a more powerful computing capability than the older Arduino ones. Maple is offered in an Arduino-compatible format, complete with Arduino pin layouts and programming environment. Maple IDE is based on an earlier version of the Arduino IDE. Maple is a great way to get started with an advanced 32 bit-processor that, until now, has principally lived in the commercial domain.

1.3 STM32 microcontrollers

STM32 is a family of integrated circuit microcontrollers made by ST Microelectronics, which is based on the 32-bit ARM Cortex-M4F, Cortex-M3 and Cortex-M0 cores.

<table>
<thead>
<tr>
<th>STM32 Series</th>
<th>CPU Core</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3, F4</td>
<td>ARM Cortex-M4F</td>
</tr>
<tr>
<td>F1, F2</td>
<td>ARM Cortex-M3</td>
</tr>
<tr>
<td>F0</td>
<td>ARM Cortex-M0</td>
</tr>
</tbody>
</table>

Fig 1.1 - Correspondence between STM32 and CPU Core
1.4 Objective of the project

The objective of this project is to simplify the use of STM32 boards by integrating them into the Arduino IDE. To do this, several steps are indispensable.
- Getting familiar with different boards, especially Arduino and STM32.
- Analysis and evaluation of the libmaple library.
- Analysis and evaluation of the Arduino library.
- Development of a compilation template for the STM32 boards.
- Adoption of Arduino IDE for STM32.

1.5 Tools

Several tools are available and necessary for this project.

1.5.1 STM32F0-Discovery board

The STM32F0 series is the first group of ARM Cortex-M0 chips in the STM32 family. The STM32F0-Discovery owns two microcontrollers. The bigger one is STM32F051 which is not connected to the USB port. The smaller one is STM32F103 which owns the USB port, but none of its IO pins is available. It is basically a built-in programming circuit for the former.

1.5.2 Arduino UNO board

The Arduino UNO board is equipped with an ATmega328 microcontroller. This microcontroller comes preburned with a bootloader that allows to upload new code to it without the use of an external hardware programmer.

1.5.3 OpenOCD

OpenOCD, Open On-Chip Debugger, aims to provide debugging, in-system programming and boundary-scan testing for embedded target devices. It supports the STM32 boards.

1.5.4 ARM-NONE-EABI (Sourcery CodeBench Lite Edition for ARM processors)
Sourcery CodeBench is a complete development environment for embedded C/C++ development on ARM, Power, ColdFire, and other architectures. ARM-NONE-EABI is one of the releases that supports ARM processors. ARM-NONE-EABI includes:

- GNU C and C++ compilers
- GNU assembler and linker
- C and C++ runtime libraries
- GNU debugger

2 Evaluation of the Libmaple

Libmaple is the library of the Maple project, developed for the STM32 line of ARM Cortex-M3 microcontrollers. Libmaple is split into two layers, the lower-level layer written in pure C, called “proper” Libmaple (in the libmaple/ directory of the source repository); the higher-level layer written in C++, and the Wiring-style C++ application programming interfaces written on top of the former, called Wirish (in wirish/), the interfaces of which are largely compatible with the AVR libraries written for the Arduino and Wiring development boards.

However, the Libmaple bootloader is designed for the single-chip boards, which is not the case for the STM32F0-Discovery board. The bigger processor STM32F051 which does not have any bootloader at all on the STM32F0-Discovery board is not directly connected to the USB port. It needs to be reprogrammed externally. The small processor STM32F103 does not really have a bootloader either. Its own code never changes and it just uploads programs to the bigger one.

A Libmaple branch which was created by Anton has improved the peripheric drivers and the files in the folder Wirish, and has resolved the problem so that the modified version of Libmaple can work with STM32VL-Discovery which is a board owning two chips. Unfortunately, Anton has not changed the fact that Libmaple only supports Cortex-M3 and does not work with STM32F0. Hence, the libmaple of Anton can not be adapted directly.

3 Evaluation of the Arduino lib
The Arduino lib has two important parts, AVR and SAM. AVR is the library for AVR microcontrollers and SAM for ARM-based microcontrollers. A majority of Arduino boards use AVR microcontrollers. However, the latest products like the Arduino DUE board use ARM Cortex-M3 microcontrollers. Thus, there is a sub-folder CMSIS (Cortex Microcontroller Software Interface Standard) in the Arduino SAM library folder, which contains supports for the Cortex-M CPU cores. CMSIS provides a single standard across all Cortex-Mx processor series vendors, and it enables code re-use and code sharing across software projects.

The folder CMSIS is the same as the one in the STM32F0 template library. But the device peripheric drivers of Arduino lib are for the Arduino boards, which are not totally compatible with the STM32F0-Discovery board.

Fig 3.1 below shows a comparison between the peripherics of Arduino, Maple and STM32F0-Discovery boards. The signification of the abbreviations are detailed in Annexe A.

![Fig 3.1 - Comparison between the peripherics of Arduino, Maple and ST boards](image)

4 Compilation template
A compilation template is a basic structure using a wiring-based language (syntax + libraries) to compile the source code and program the boards. It consists basically of device initialization files, libraries, peripheric drivers, main programs and Makefiles. A simple folder structure is shown in the Fig 4.1.

![Folder structure of a STM32 compilation template](image)

**Fig 4.1 - General structure of a STM32 compilation template**

Device initialization files are usually written in assembler language, libraries and peripheric drivers in C or C++. In this project, the core libraries are from CMSIS, and the drivers are for the peripherics of the STM32 boards.

## 5 Adaption of Arduino IDE for STM32

![Main interface of Arduino IDE](image)

**Fig 5.1 - Main interface of Arduino IDE**

Arduino IDE, the IDE for the Arduino boards, is written in Java, using Ant to compile and to run it. Once a new library folder is added under the hardware folder of Arduino IDE, new items which shows the boards’ information described in the specification files are immediately added in the menus. Fig 5.2 below shows the change of the menu after adding the STM32F0-Discovery board in the hardware folder (Fig 5.3).

![Fig 5.2 - The STM32F0-Discovery board is added in Arduino IDE](image)

![Fig 5.3 - A new folder “st” is added under hardware](image)
6 Conclusion

The Libmaple works with STM32F1 and STM32F2 series which are ARM Cortex-M3 microcontrollers. The SAM part of Arduino library is compatible with all ARM Cortex-M microcontrollers but the peripheral drivers work only with the Arduino SAM boards. The Arduino IDE has a built-in editor to edit the program code, but it does not support the ST boards.

To propose a solution, an IDE for the STM32 boards can be an adaption of the Arduino IDE by adding a modified version of Libmaple to support STM32F1 and STM32F2, OpenOCD to be able to debug and upload programs to STM32 boards and the peripheral drivers for STM32F0.
Bibliography

1. Arduino
   http://www.arduino.cc/
2. STM32
3. A discussion on STM32VL-Discovery in Leaflabs Forum
4. Sourcery CodeBench
   http://www.mentor.com/solutions/arm
5. LeafLabs Documentation
   http://leaflabs.com/docs/
6. Libmaple branch created by Anton
   https://github.com/anton19286/libmaple
7. Libmaple-master
   https://github.com/leaflabs/libmaple
8. OpenOCD
   http://openocd.sourceforge.net/
Appendix A

Signification of the peripheric abbreviations:

adc := Analog to Digital Convertor
cec := Consumer Electronics Control
comp := comparators
crc := CRC computation unit peripheral
dac := Digital to Analog Convertor
dbgmcu := Debug MCU peripheral
dma := Direct Memory Access controller
exti := External Interruption
flash := Flash management
gpio := General Purpose Input Output
i2c := Inter-Integrated circuit
iwdg := Independent watchdog
misc := add-on to CMSIS functions
pwr := Power Controller
rcc := Reset and clock control
rtc := Real-Time Clock
spi := Serial peripheral interface
usart := Universal synchronous asynchronous receiver transmitter
wwdg := Window watchdog