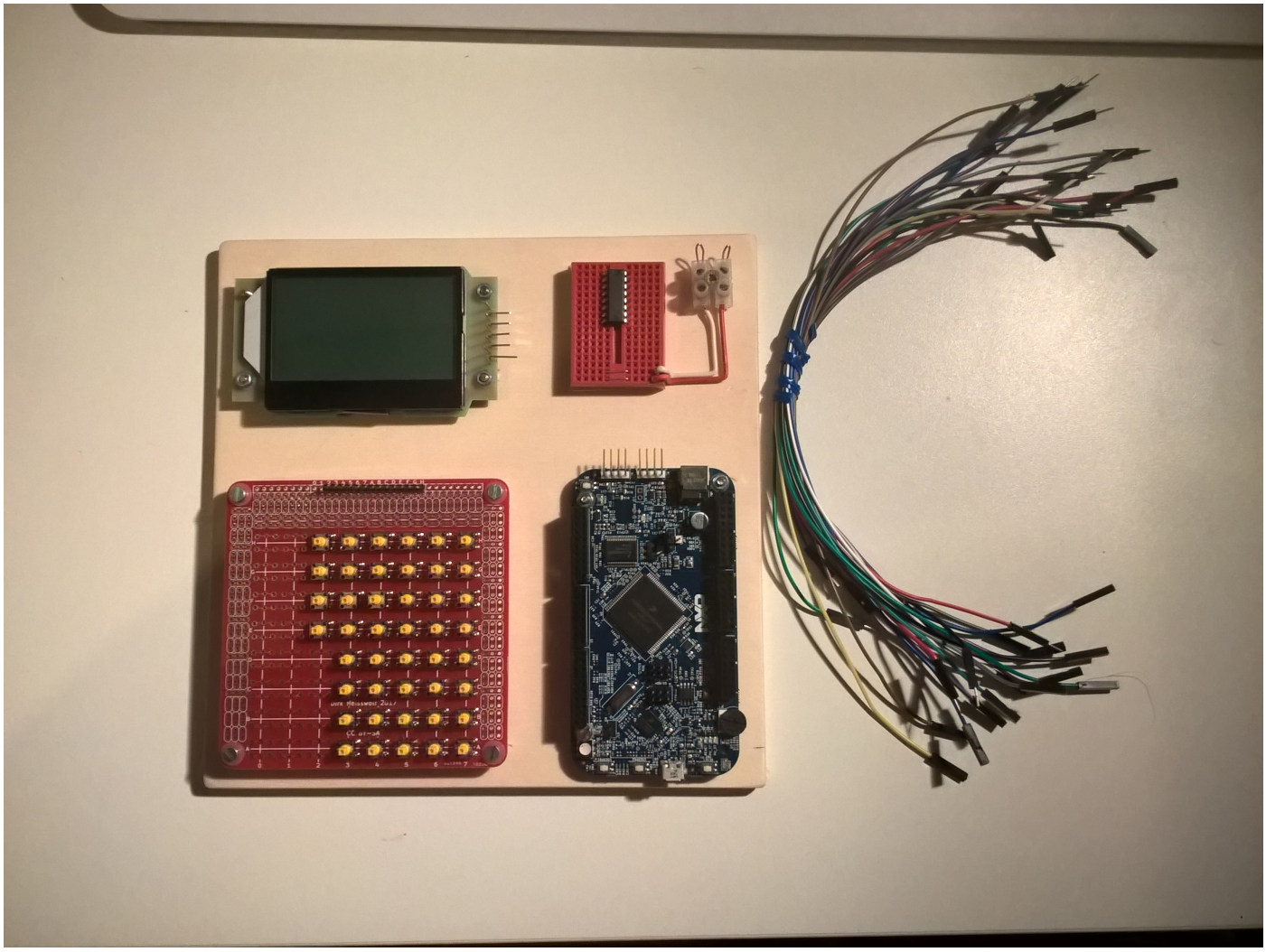
AC-II

A custom electronic calculator and embedded systems development platform



By Danny Milutinovic
Melbourne, Australia
August 8, 2019

Contents

Preface

This book is about the AC-II, a programmable, reverse polish notation scientific calculator and embedded systems development platform.

In 2017 I started writing firmware for the AriCalculator,a hand held calculator created by Dirk Heisswolf. The firmware was completed in 2018 and featured the standard scientific functions, as well as keystroke programming and a number of other features. I then ported the firmware to the NXP S12XEP100 microcontroller, the most powerful of the S12 series. I made a number of changes to the firmware and am rewriting it in mixed C/assembly in order to make it easier to maintain and port to other architectures.

Part I describes how to assemble the AC-II using a low cost development board and custom made shield, and how to upload firmware onto the AC-II. Part II focuses on the algorithms used to implement the mathematical functions, using diagrams and flowcharts to help explain each. Part III considers some of the AC-II's advanced mathematical applications and Part IV discusses how it can be used as a development platform for embedded systems projects.

Each chapter builds on the material covered in the previous one and the source code is developed in stages, making it easier to follow. The programs referred to in this book are available at `https://github.com/DanielMilutinovic/AC-II`.

Part I

Getting Started

Chapter 1

Hardware and Software Development Tools

In this chapter we will look at how to build the AC-II. The first and easiest way is to attach the AC-II shield (currently being designed) to the DEVKIT-S12XE.

The second way is to use separate components. This allows you to customise the keypad and use your preferred display but is more difficult. You will also need to work out how to build a hand held version - not a trivial task. The following is one list of hardware components that can be used to build the AC-II. They can be purchased from electronic components distributors such as Mouser, Digi-Key and element14:

1. The DEVKIT-S12XE development board by NXP (Figure 1.1). It features the 9S12XEP100 microcontroller and is provided with a USB cable to connect it to a computer.

Figure 1.1: DEVKIT-S12XE

2. An ST7565 128x64 graphical liquid crystal display (GLCD) (Figure 1.2). The model shown is available from Core Electronics and Adafruit. Note that eleven 0.1" (2.54mm) male header pins must be soldered to the board if using this particular model; however it is simple through-hole soldering.

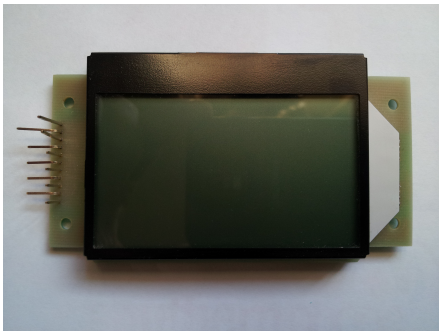


Figure 1.2: ST7565 GLCD

3. 0.1” (2.54mm) male-female jumper cables (Figure 1.3) to interface the DEVKIT-S12XE development board to the ST7565 GLCD and matrix keypad.

Figure 1.3: 0.1” male-female jumper cables

4. A matrix keypad. Standard 4x4 matrix keypads (Figure 1.4) are convenient but an 8x6 keypad (Figure 1.5) allows more functions to be added easily. Interfacing to each type of keypad is covered in Chapter 4. The 8x6 matrix keypad shown was designed by Dirk Heisswolf and manufactured by Seeed Studio. I have several 8x6 matrix keypad PCB’s left (contact me at `daniel_milutinovic@yahoo.com.au`) and the files are available at `https://github.com/DanielMilutinovic/AC-II` if you would like to order them (Seeed Studio has a minimum order of 10 PCB’s). The PCB’s can accommodate up to 64 keys (8 rows, 8 columns) but we will only use 6 columns. If making the 8x6 matrix keypad you will need to solder 0.1” (2.54mm) male header pins and 44 push buttons to the PCB. I used Omron 6x6 tactile switches, part no. B3F-1052, which work well.

Figure 1.4: 4x4 matrix keypad by EOZ, with some key labels attached

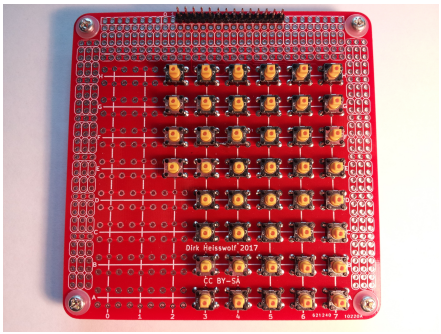


Figure 1.5: Custom 8x6 matrix keypad

The 9S12XEP100 microcontroller on the DEVKIT-S12XE development board is programmed using the CodeWarrior IDE by NXP. CodeWarrior may be downloaded free of charge and there are no code size limitations for assembly language files. It features a debugger and a convenient simulation mode, which enables code to be tested without downloading it to the actual microcontroller, which saves time and the Flash memory on the device from repeated writes.

Chapter 2

Blinking an LED

Install "CodeWarrior Development Studio for the S12(X) Version 5.2" onto your computer. Connect the DEVKIT-S12XE development board to your computer using the supplied USB cable, then connect a power supply to the development board (a 12V power supply is recommended but a 9V power supply works). The LED on the development board will turn on and change colour (the board comes with a pre-loaded example project - see the DEVKIT-S12XE Quick Start Guide for further information, which can be downloaded at
`https://github.com/DanielMilutinovic/AC-II`).

We will now download a new program to the board to blink the LED. Firstly create a new folder on your computer to save your projects to, then open CodeWarrior and choose "Create New Project". Select "MC9S12XEP100" and "P&E USB BDM Multilink". Then press "Next" (Figure 2.1).

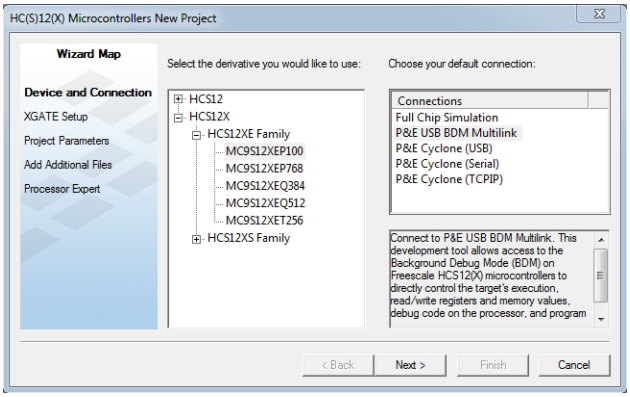


Figure 2.1: Selecting the microcontroller and connection

Select "Single Core(HCS 12X)" then press "Next" (Figure 2.2).

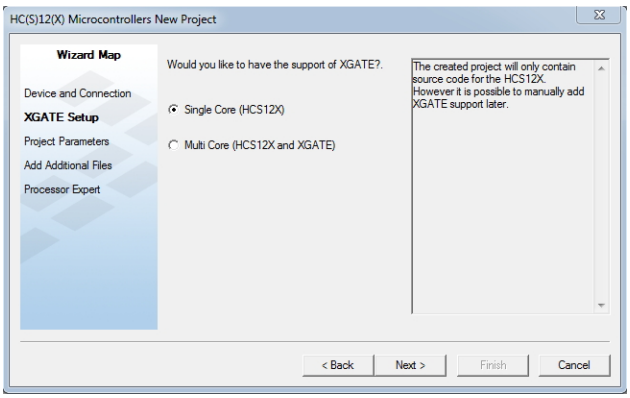


Figure 2.2: Selecting the core

Select "Absolute assembly" and enter the location and project name ("LED" in this example), then select "Finish" (Figure 2.3).

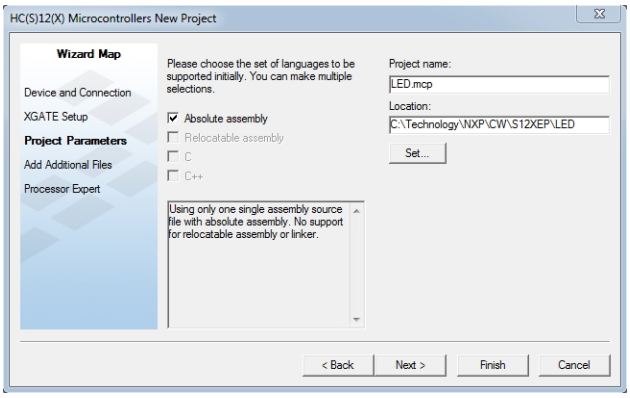


Figure 2.3: Choosing the language, name and location

Delete all of the pre-filled assembly code that now appears in the right-hand window. Go to `https://github.com/DanielMilutinovic/AC-II` and open the file LED.asm (an assembly language program to flash the LED). Copy all of the text that appears and paste it to where the pre-filled assembly code used to be. Then press "Make" in the tool bar to assemble the file (Figure 2.4).

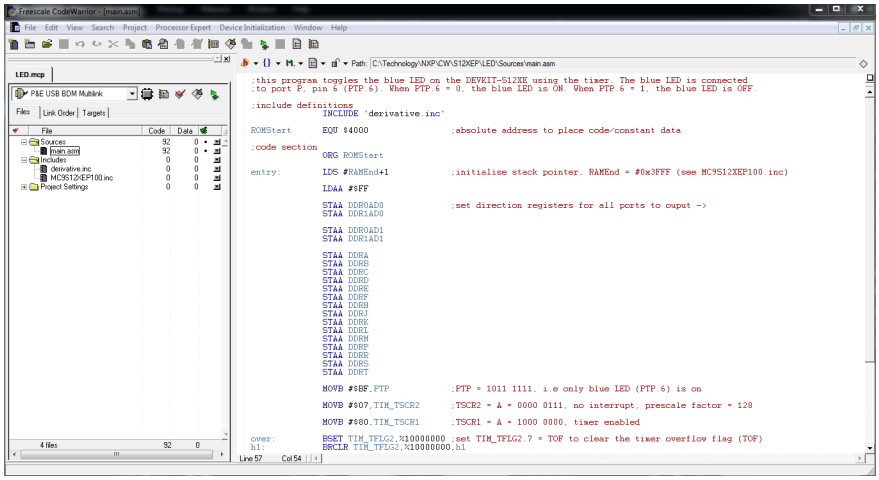


Figure 2.4: Assembling the file

Now press "Debug" in the tool bar to download the code to the microcontroller and then press "Start / Continue" in the new window to run the program. The blue LED on the board will start to blink. You are now ready to interface the DEVKIT-S12XE to the ST7565 GLCD.