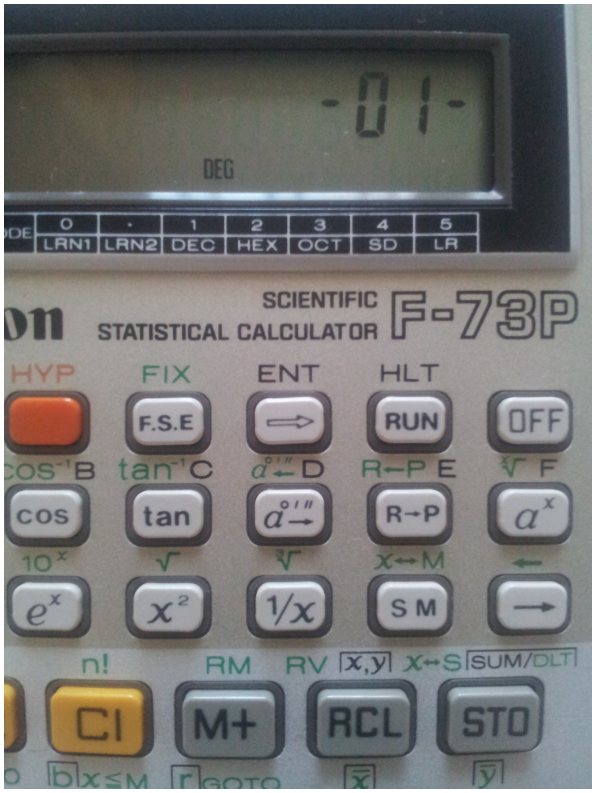
The MC-II Electronic Calculator

An Embedded Systems Project



By Danny Milutinovic
Melbourne, Australia
November 6, 2018

Contents

Preface

This book is about the MC-II calculator - a programmable, reverse polish notation graphing calculator and embedded systems development platform. Part I describes how to build the MC-II using a low cost development board, graphical liquid crystal display and matrix keypad. Part II focuses on the algorithms used to implement the mathematical functions, and uses diagrams and flowcharts to help explain each. Part III considers some of the MC-II's advanced mathematical applications and Part IV discusses how it can be used as a development platform for embedded systems projects, as well as how to build a hand held version of the MC-II.

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 assembly language programs referred to in this book are available at `https://github.com/DanielMilutinovic/MC-II`.

Part I

Getting Started

Chapter 1

Hardware

The following is a list of the hardware components required to build the MC-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.

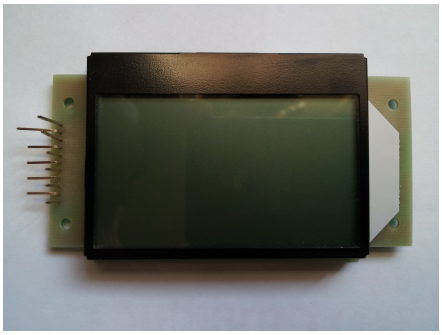


Figure 1.2: ST7565 GLCD

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/MC-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.

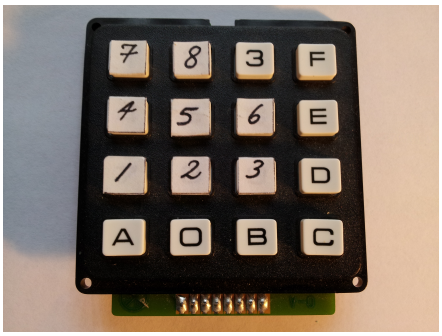


Figure 1.4: 4x4 matrix keypad by EOZ, with some key labels attached

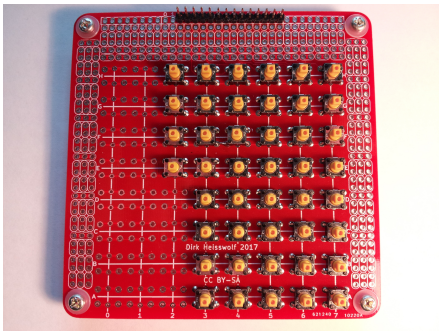


Figure 1.5: Custom 8x6 matrix keypad

Chapter 2

Software development tools

The 9S12XEP100 microcontroller on the DEVKIT-S12XE development board is programmed using the CodeWarrior development software 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, thereby saving the Flash memory on the device from repeated writes.

Chapter 3

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/MC-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 3.1).

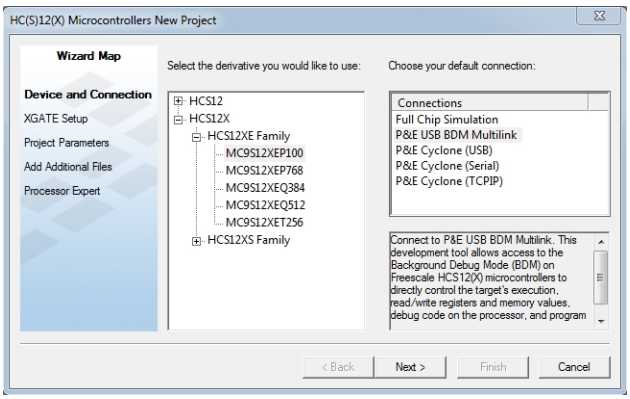


Figure 3.1: Selecting the microcontroller and connection

Select "Single Core(HCS 12X)" then press "Next" (Figure 3.2).

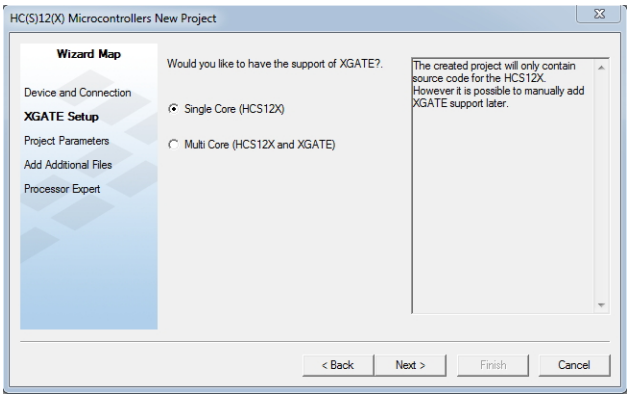


Figure 3.2: Selecting the core

Select "Absolute assembly" and enter the location and project name ("LED" in this example), then select "Finish" (Figure 3.3).

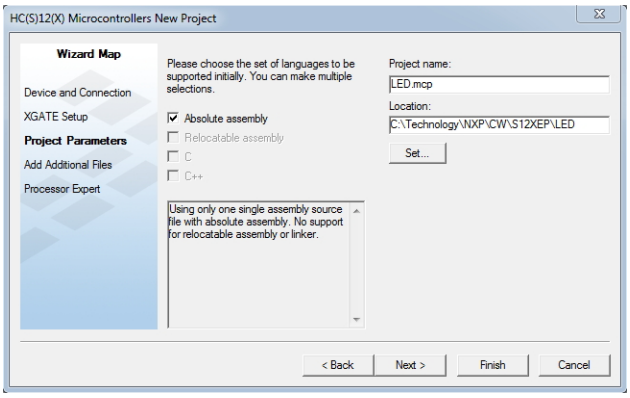


Figure 3.3: Choosing the language, name and location

Delete all of the pre-filled assembly code that now appears in the right-hand window. Go to `https://github.com/DanielMilutinovic/MC-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 3.4).

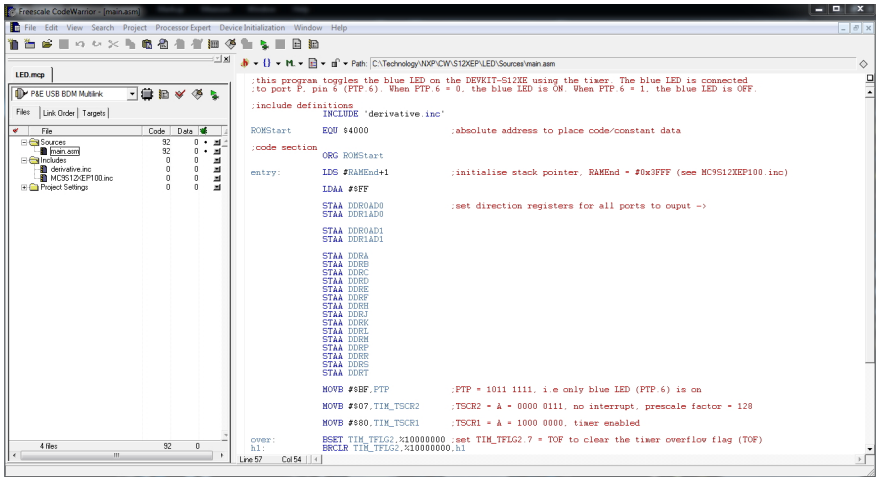


Figure 3.4: Assembling the file

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.

Chapter 4

Interfacing to the ST7565 GLCD

WARNING! Ensure the jumper at J13 on the DEVKIT-S12XE development board is set to 3.3V. Otherwise the ST7565 GLCD will be permanently damaged.

Figure 4.1 shows the wiring diagram for interfacing the DEVKIT-S12XE development board to the ST7565 GLCD (without a backlight). Refer to the DEVKIT-S12XE Quick Start Guide for the location of the pins on the development board and use the male-female jumper cables to connect the DEVKIT-S12XE to the ST7565 GLCD.

The display is divided into four sections:

1. The status line
2. The stack
3. The command line
4. The menu

4.1 Using simulated SPI

We will display the status line using simulated SPI, also referred to as "bit banging". Create a new project, copy and paste the code from GLCD1.asm, assemble the file, download and run. The display should appear as in Figure 4.2.

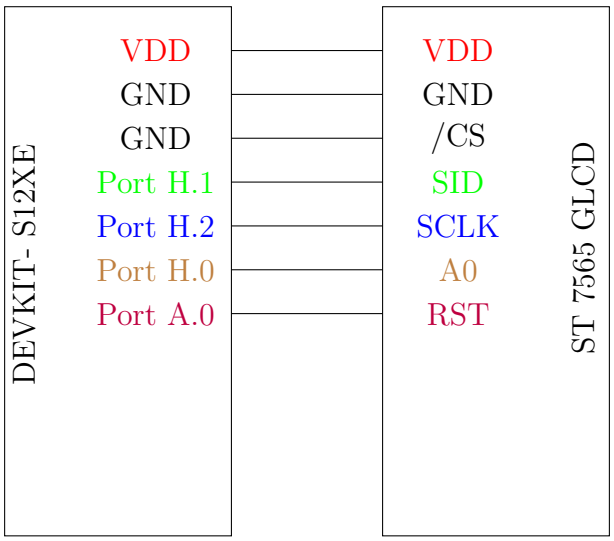


Figure 4.1: Interfacing the DEVKIT-S12XE to the ST7565 GLCD

Figure 4.2: The status line

4.2 Using SPI

We will know use SPI to display the status line, stack level labels and menu. Create a new project, copy and paste the code from GLCD2.asm, assemble the file, download and run. The display should appear as in Figure 4.3.

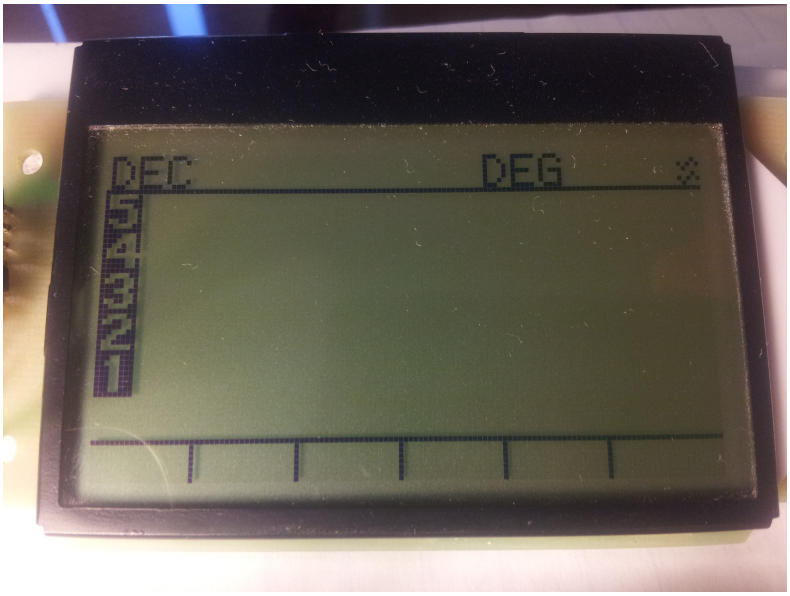


Figure 4.3: The status line, stack level labels and menu (currently empty)

4.3 Using interrupts

Finally, we add the blinking cursor on the command line by including an interrupt in GLCD3.asm (Figure 4.4).

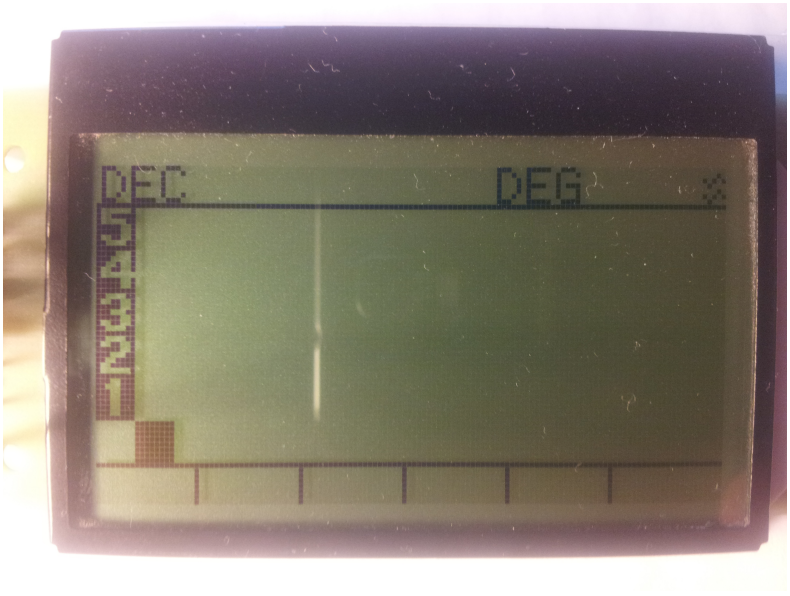


Figure 4.4: The completed display

In the next chapter we will complete our development system by interfacing to a matrix keypad.

Chapter 5

Interfacing to a matrix keypad

5.1 4x4 matrix keypad

Figure 5.1 shows the functions on the 4x4 matrix keypad.

Figure 5.2 shows the wiring diagram for interfacing the DEVKIT-S12XE development board to the 4x4 matrix keypad. Refer to the DEVKIT-S12XE Quick Start Guide for the location of the pins on the development board and use the male-female jumper cables to connect the DEVKIT-S12XE to the 4x4 matrix keypad.

5.2 8x6 matrix keypad

Figure 5.3 shows the functions on the 8x6 matrix keypad.

Figure 5.4 shows the wiring diagram for interfacing the DEVKIT-S12XE development board to the 8x6 matrix keypad. Refer to the DEVKIT-S12XE Quick Start Guide for the location of the pins on the development board and use the male-female jumper cables to connect the DEVKIT-S12XE to the 8x6 matrix keypad.

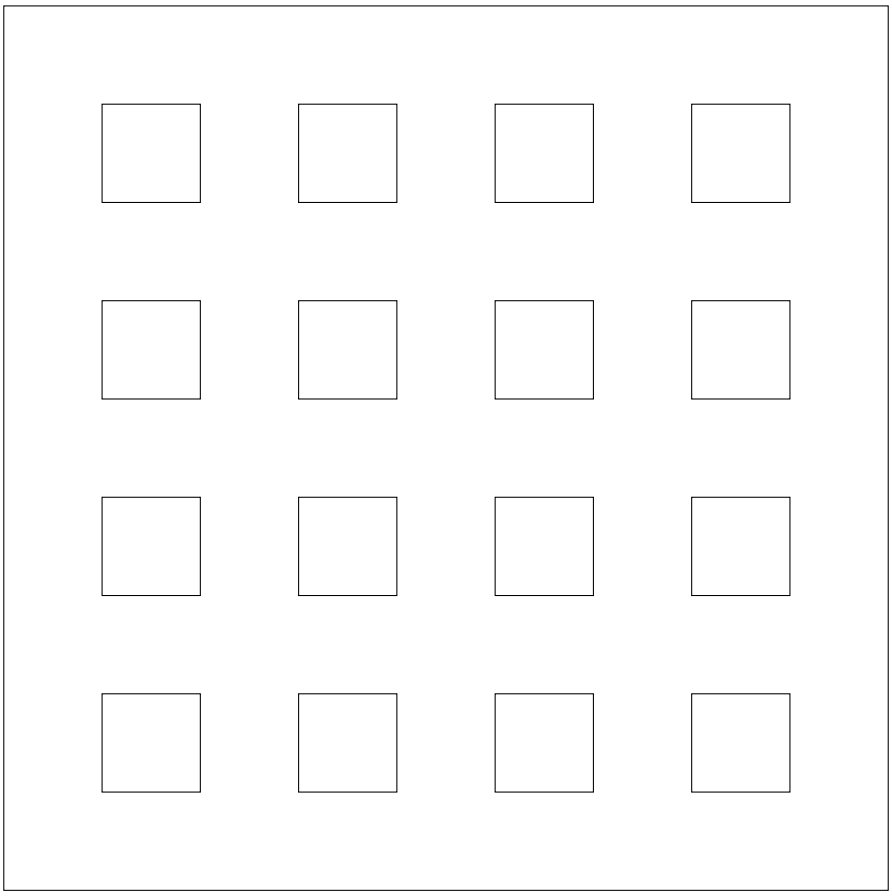


Figure 5.1: The 4x4 matrix keypad functions

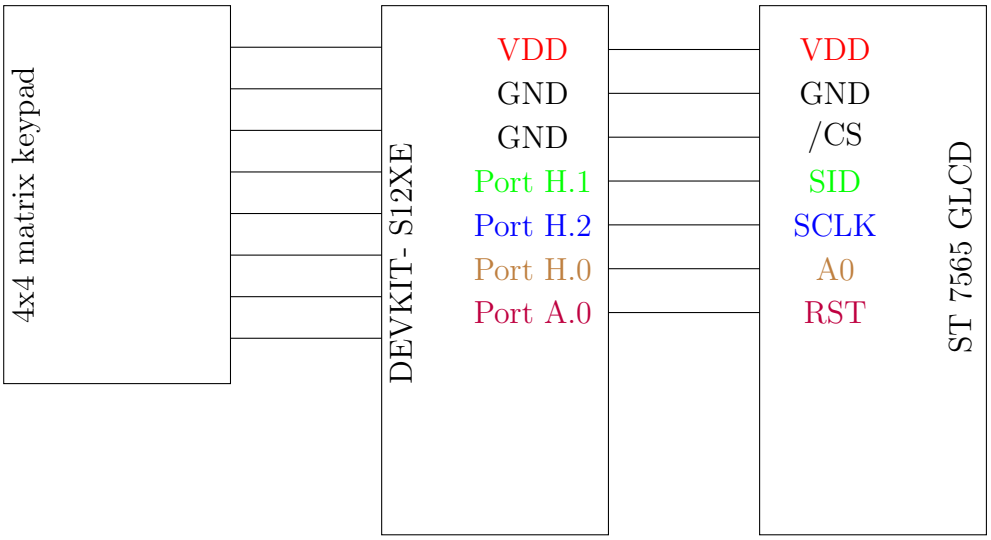


Figure 5.2: Interfacing the DEVKIT-S12XE to the 4x4 matrix keypad

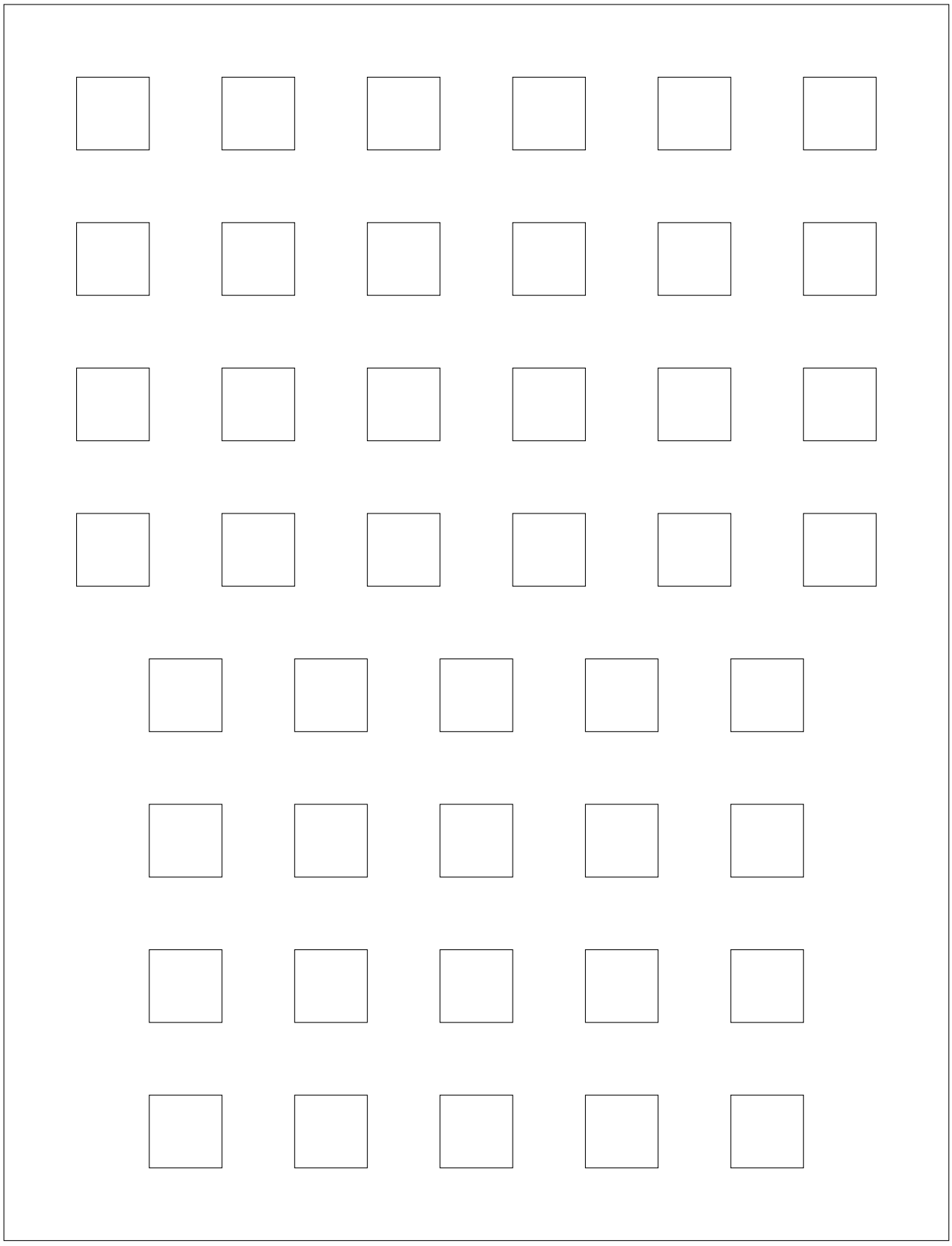


Figure 5.3: The 8x6 matrix keypad functions

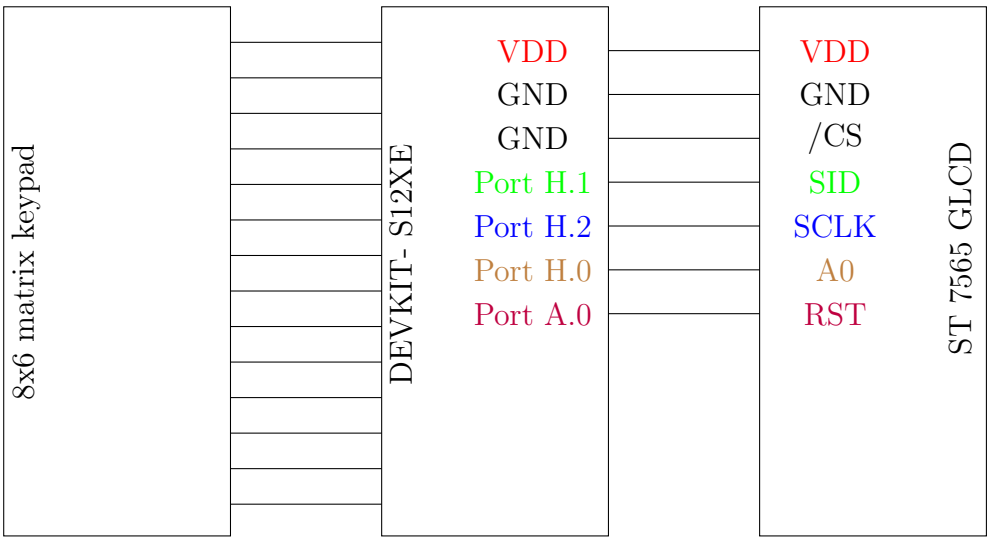


Figure 5.4: Interfacing the DEVKIT-S12XE to the 8x6 matrix keypad

Part II

Algorithms

Chapter 6

The command line and stack

6.1 Entering values on the command line

Note: from this point onward it is assumed that the DEVKIT-S12XE is interfaced to the 8x6 matrix keypad. Modifications to the code will need to be made if it is interfaced to the 4x4 matrix keypad.

6.2 Pushing values onto the stack

Chapter 7

The stack operations

7.1 Enter

7.2 Delete and Clear

7.3 Pick

7.4 Rotate and Unrotate

7.5 Over

Chapter 8

The arithmetic functions

In this chapter we will look at the algorithms used to implement the basic arithmetic functions.

8.1 Negation ($\pm$)

8.2 Addition ($+$)

8.3 Subtraction ($-$)

8.4 Multiplication ($\times$)

There are three common algorithms used to implement multiplication: shift and add, long multiplication using the assembly instruction emul and a CORDIC method.

8.5 Division ($\div$)

As with multiplication, there are three common algorithms used to implement division: shift and subtract, long division using the assembly instruction ediv and a CORDIC method.

Chapter 9

Memory

9.1 Storing and recalling values

9.2 Viewing the memory contents of stack level 1

It is convenient to be able to view the memory contents of stack level 1 when greater accuracy is required as well as for debugging purposes.

9.3 Viewing execution time

In order to compare the efficiency of different algorithms the microcontroller's timer may be used to obtain an approximation to the number of machine cycles required to execute a particular function.

Chapter 10

The scientific functions

10.1 Square (x^2)

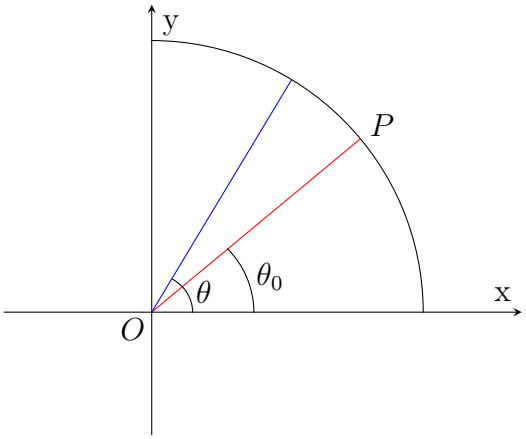
10.2 Factorial ($x!$)

10.3 Square root ($\sqrt{x}$)

10.4 Circular functions ($\sin\theta$, $\cos\theta$ and $\tan\theta$)

Firstly we look at the CORDIC algorithm for $\tan\theta$. Suppose we want to find $\tan\theta$, where θ is in radians, and suppose

$$\tan\theta_0 = \frac{Y}{X}, \text{ where } X, Y \text{are known integers and } \theta_0 < \theta$$



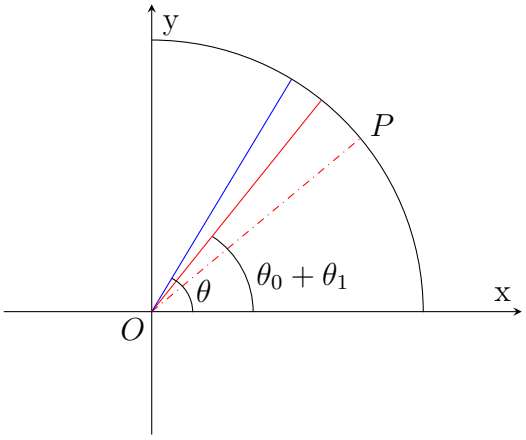
$\frac{Y}{X}$ is then an approximation to $\tan\theta$. To obtain a better approximation we rotate line OP

anticlockwise θ_1 radians, where θ_1 is chosen so that

$$\tan\theta_1 = 10^{-K}, \text{ for some } K = 0, 1, 2, ..., 13$$

and

$$\theta_0 + \theta_1 \leq \theta$$



Note that $\tan(\theta_0 + \theta_1)$ is a better approximation to $\tan\theta$.

We know use the trigonometric identity

$$\tan(\alpha + \beta) = \frac{\tan\alpha + \tan\beta}{1 - \tan\alpha\tan\beta}$$

to obtain

$$\tan(\theta_0 + \theta_1) = \frac{\tan\theta_0 + \tan\theta_1}{1 - \tan\theta_0\tan\theta_1} = \frac{\frac{Y}{X} + 10^{-K}}{1 - \frac{Y}{X}10^{-K}} = \frac{Y + 10^{-K}X}{X - 10^{-K}Y} = \frac{Y'}{X'}$$

Y' is obtained by shifting X K nibbles to the right and then adding Y. X' is obtained by shifting Y K nibbles to the right, negating the result and then adding X.
X' and Y' are copied to X and Y, respectively and the process repeated until

The following flowchart illustrates the algorithm

To find $\sin\theta$, $\tan\theta$ is found using the CORDIC algorithm and the result substituted into the trigonometric identity

$\sin\theta$

10.5 Inverse circular functions ($\sin^{-1} x$, $\cos^{-1} x$ and $\tan^{-1} x$)

10.6 Exponential functions e^x and 10^x

10.7 Logarithmic functions $\ln x$ and $\log_e x$

10.8 Exponential functions x^y and $\sqrt[y]{x}$

10.9 Random number generator

Chapter 11

Binary and hexadecimal numbers

Chapter 12

Complex numbers

Chapter 13

Rational numbers

Chapter 14

Statistical functions

Part III

Mathematical Applications

Chapter 15

Keystroke programming

Chapter 16

Equation Solver

Chapter 17

Plotting functions

Chapter 18

Matrices

Chapter 19

Triangle Solver

Part IV

Embedded Systems Applications

Chapter 20

Using the MC-II as a development platform

Chapter 21

Building a hand held MC-II