ENGI E1102 Departmental Project Report: Computer Science/Computer Engineering

Xiahui (Forrest) Huang, Andy Hadjigeorgiou, Siddharth Ramakrishnan, and Alex Kalicki

December 20, 2012

Abstract

Our goal for this project was to write a new firmware for a deprogrammed HP 20b calculator as well as to demonstrate and become experienced with embedded programming. To do so, we needed to get the calculator to recognize and respond to a button being pressed. The calculator was modified to provide access to the development ports, which allowed us to input our own code. Each lab began with several files, provided by Professor Edwards, which provided us the bare minimum on which to build out project. Some files were responsible for accessing and writing to the 128k flash memory, while others mapped the peripherals in the calculator and set up a C runtime environment. Building off of these files, the calculator was programmed to respond if a key was pressed. Following that, the calculator was programmed to respond with a symbol, determining the address of the key that was pressed. After finishing the project, we were able to produce a mostly functional calculator.

1. Introduction

The HP 20b calculator was released by Hewlett-Packard in 2008 as a financial calculator. While HP officially calls it a calculator, it is really just a small computer programmed for a specific purpose. The calculator boasts the standard functions, like addition and subtraction, as well as some more complicated ones that have specific applications in the financial industry. The calculator includes many pieces of hardware just like regular computers, and has built in firmware that it runs on in order to perform as advertised. In our project, one of the most important hardware features of the calculator was the serial port because it allowed us to access the calculator through a computer. This was important because the firmware on the calculator was wiped, and the purpose of our project was to reprogram the calculator to function properly.

2. User Guide

Our reprogrammed HP 20b calculator was built to function as an RPN system, which is described in detail later in the report. Since we built it as an RPN system, the calculator does not quite function the same as calculators most people are used to. To perform the function ‘1+3’ on the reprogrammed HP 20b, one would have to press ‘1 ⇒ INPUT ⇒ 3 ⇒ INPUT ⇒ +’. This

sequence would add 1 and 3 to display 4. More complex functions can be entered just like another calculator, but it just takes a few more key presses. to do '1+3x4' one would have to press '1 ⇒ INPUT ⇒ 3 ⇒ INPUT ⇒ 4 ⇒ INPUT ⇒ x ⇒ +'. Our calculator also includes some other capabilities. Pressing the negative button changes the inputted number to the opposite sign of what it already is. In addition, when one presses an operation button before entering a number the calculator will display the maximum integer C allows.

3. The Platform

3.1 - The Processor

The processor consists of many components, and is responsible for transferring memory to different locations. The processor functions by making decisions based on a set of instructions it is given, and its decisions change based upon the set of instructions it needs to act upon. The ARM7TDMI processor is the microprocessor that is present in the HP20B calculator. One particular component of this processor is the ALU microprocessor, short for the arithmetic/ logic unit. This microprocessor is responsible for the vital functions of the calculator, simple mathematical operations. These operations include addition, subtraction, multiplication, and division.

3.2 - The LCD Display

The LCD Display is a prevalent technology used in image display. Different from the LCD screen on the television, the one on the calculator is simplistic but still utilizes the same technology. The LCD consists of two plates of glass with a layer of liquid crystal in between. The polarizing films are laminated on the glass at 90 degree angles. Aligning the two polarizers at 90 degrees prevents transmission of light. However, due to the ability of liquid crystal to rotate polarized light, it is able to control the transmission of light effectively. This is done by the passage of AC voltage through the liquid crystal. The *lcd_put_char7* method was provided to us in order to interact directly with the LCD display using the language C. The method takes two parameters - first the character/number intended to be displayed and then the slot to place it. Using these two parameters, the *lcd_put_char7* method interacts with the processor and display the appropriate characters on the screen.

3.3 - The Keyboard

The keyboard on each HP 20b calculator is a grid of rubber buttons protruding from the device. Below the keys are two sheets of electrical contacts separated by a thin space so that they do not touch. When a key on the keyboard is pressed, the two electrical contact sheets are pressed together at the point underneath the key, allowing the wires from each sheet to connect at that

position. The processor is then able to determine which key was pressed from the column and row where contact was made and take appropriate action.

4. Software Architecture

Our software is split up into four different sections, and each builds off the previous ones to accomplish the final goal of making a working calculator. The goal of the first lab was to just get an integer to print on the LCD display. We ran this through the computer and did not have any hardware on the calculator functioning except the LCD display. In the next part, we were able to get the keyboard to function and print the key we pressed on the screen. We built upon this further, and in the third portion of the project we were able to press multiple keys, enter multiple digit numbers, and enter operations as well. We used the code we developed to make the keyboard work as well as the LCD display function to code the third portion of the lab. We were not able to finish the last portion of the lab, but it would have used all the elements we developed before, as well as a way to store what we entered in order to make a fully functioning calculator.

5. Software Details

Lab 1 - Getting Started: Hello World

The goal of Lab 1 was to get the group started programming in C for the HP 20b calculator by writing a simple method to display an integer on the LCD. The method (the majority of which is shown in Figure 1) simply takes the integer to display as an argument, shows it on the calculator screen, and exits. In order to display the full number, the method starts from the right hand of the screen and continually takes the modulus of the complete integer in order to strip the rightmost digit. It then adds a predefined constant in order to convert the digit into its ASCII equivalent, places it in the next slot on the calculator, and divides the integer by 10 to truncate it by one digit and prepare it for stripping the next digit to display. When the full integer has been displayed on the calculator, the method decides whether to display a negative symbol (based on the sign of the integer argument) and finishes by clearing the remainder of the screen.

```
while (slot >= 0)
{
    if (x > 0)
    {
        lcd_put_char7(x%10+ASCIIADD, slot);
        x /= 10;
    }
    else
    {
        if (negative)
        {
            lcd_put_char7('-', slot);
            negative = 0;
        }
        else
        {
            lcd_put_char7(' ', slot);
        }
    }
    slot--;
}
```

Figure 1

Lab 2 - Listening to the Keyboard

To fulfill the requirements for Lab 2, our group needed to write a method to return a code representing the key currently being pressed. In order to accomplish this task, we first defined a character array relating each keyboard key to the symbol it would return (Figure 2). The method would then scan through each column of the keyboard, setting it to low and checking each row to see if it was also set to low. If a given column and row combination is found to be set to low, the key at their intersection is currently pressed down and the code corresponding to the located key is returned. If both loops are completed without the method short-circuiting, no key has been pressed in that method cycle and -1 is returned. Whether or not a key has been found to be pressed, the column is set back to high so that the method will begin in its default state upon the next run. The *keyboard_key* method that our group wrote for Lab 2 is as follows in Figure 3.

```
const char keyArray[7][6] = {
    {'N',  'I', 'P', 'M', 'F', 'A'},
    {'C',  'R', 'V', 'B', '%', 'L'},
    {'i', '(', ')', '~', '<', 0},
    {'u', '7', '8', '9', '/',  0},
    {'d', '4', '5', '6', '*',  0},
    {'_',  '1', '2', '3', '-',  0},
    { 0,   '0', '.', '=', '+',  0}};
```

Figure 2

```
int keyboard_key ()
{
    int i;
    int j;

    for (j = 0; j < 7; j++)
    {
        keyboard_column_low(j);

        for (i = 0 ; i < 6 ; i++)
        {
            if (!keyboard_row_read(i))
            {
                keyboard_column_high(j);
                return keyArray[j][i];
            }
        }
        keyboard_column_high(j);
    }

    return -1;
}
```

Figure 3

Lab 3 - Entering and Displaying Numbers

Lab 3 combined the skills we learned and the methods we wrote in the previous two labs in order to store and display user input on the calculator LCD. Our goal was to write a method (Figure 4) that would allow the user to input any number followed by an operation and display both on the LCD. Each time through the loop, we check to see if a key is being pressed. If it is determined that a key is being pressed, then the method uses the *tempKey* variable to wait for the key to be released before continuing. If the key pressed is between zero and nine and we still have room on the display, then the method adds this digit to the previous number multiplied by ten, and stores the result into the data structure holding the number and the operation. If the key being pressed is not between zero and nine, we consider it an operation key and store its character representation in the data structure before ending the loop, displaying the number and operation on the screen, and terminating the method. If no key has been pressed before the operation key is pushed, we automatically stores the maximum value for the integer primitive type and store it in the structure.

```
while (!operationPressed)
{
    int keyPressed = keyboard_key();
    if (keyPressed != -1)
    {
        int tempKey = keyPressed;
        while (tempKey != -1)
        {
            tempKey = keyboard_key();
        }

        if (keyPressed >= '0' && keyPressed <= '9')
        {
            pressedKey = 1;
            if (slot_count < SLOTS)
            {
                result->number = result->number*10 + (keyPressed - '0');
                if (result->number != 0)
                    slot_count++;
            }
        }
        else
        {
            result->operation = (char)keyPressed;
            if (!pressedKey)
                result->number = INT_MAX;
            operationPressed = 1;
        }
    }

    lcd_print_int(result->number);
    lcd_put_char7(result->operation, 0);
}
```

Figure 4

Lab 4 - An RPN Calculator

Our group was not able to complete Lab 4, but the goal of this lab was to create code that built upon the other programs we created and allowed the calculator to fully function as an RPN calculator. This would have called for the implementation of a stack to store the numbers that have been entered along with the ability to perform functions on them as soon as an operation is pressed. Using the example back in the user guide, if one wanted to perform the function '1 + 3' he/she would have to press '1 ⇒ INPUT ⇒ 3 ⇒ INPUT ⇒ +'. This is because when '1 ⇒ INPUT' is pushed then the integer 1 is stored in a stack. Then when '3 ⇒ INPUT' is entered, 1 is moved up the stack, and 3 is stored in the bottom of the stack. Finally when '+' is pressed, the two numbers closest to the bottom of the stack will be added together, which in this case returns 4. More than two numbers can be entered in the stack. For example, if we had '1 ⇒ INPUT ⇒ 3 ⇒ INPUT ⇒ 3 ⇒ INPUT ⇒ +'. This would store 3 numbers in the stack, and when '+' is entered, 3 and 3 will be added to display 6.

6. Lessons Learned

Our group learned several important lessons through the completion of this course. Although the HP 20b calculators we used seem simple, the programming behind them is more complex than is readily visible to the outside world. Even the simple task of displaying digits on the LCD panel required its own method and more than a few lines of code. Another lesson we learned as part of the course is that there is never one set way of accomplishing a given task - through the code reviews the class presented we learned several methods of completing each lab. Although certain approaches were more efficient and seemingly "better" than others, none of the problems we tackled had a single solution and it was rewarding to expand our thinking through listening to the presentations of our peers.

If we were to provide advice to future students taking the course, we would suggest keeping an open mind about each problem and not getting frustrated if their group's approach to a lab needs to be revised or drastically altered in order to meet the design specifications. Constant editing and revisions only serve to improve the group's code in the long run, and sometimes taking a step back and thinking of your overall approach to a problem is necessary to understand what is going wrong. We'd also recommend that future students study basic programming techniques on their own if they are not familiar with computer science coming into the course. Although the course was billed as an introduction to computer science and computer engineering, students are quickly asked to write their own simple programs in the C language. This is not too difficult with prior exposure to some elementary programming, but could be somewhat daunting without previous experience.

7. Criticism of the Course

This course provides a good understanding of computer science and sets a solid foundation for students' programming skills; however, there are a few aspects we'd like the course to improve on as well. The Art of Engineering is an introductory course to various fields in engineering, giving students the option to choose which project section they would like to be in depending on their interest. In our Computer Science section, we are expected to have a basic understanding of programming in an object-oriented language in order to accomplish the tasks. While we have a few members that have had prior programming experience, we noticed some members who have no prior experience struggled with the basics of programming. In order to cope with such problem, we suggest that for the first two sections or so, we give every student a glance of the basic functionality of building programs in the C language and the extra resources such as tutorial websites in programming for those that need more in-depth information. This way, given the basic understanding, students who have no prior experience would accomplish the lab assignments at a greater speed.

We also recommend having more frequent communication among groups besides the group presentation for each lab. Since the pace at which the groups accomplish the labs can vary greatly, the group presentations sometimes may not provide the same level of learning experience if all of them were presented to groups who are on the same pace. Therefore, we recommend that sending one group member to one other group every week to talk about the assignment process of their own group and help with any questions the other group may have. This will not only give group members a better understanding of how other groups approach the tasks different, but also strengthen the coherence of the class.

Works Cited

ENGI E1112 Departmental Project Report: Computer Science/Computer Engineering. N.p., Nov. 2011. Web. 20 Dec. 2012. <http://www.cs.columbia.edu/~sedwards/classes/2012/gateway-fall/report.pdf>.

HP 20b Business Consultant Financial Calculator Manual. N.p., n.d. Web. 20 Dec. 2012. <http://h10010.www1.hp.com/wwpc/pscmisc/vac/us/product_pdfs/HP_20b_Online_Manual.pdf>.

The International Calculator Collector. N.p., 1997. Web. 20 Dec. 2012. <http://www.vintagecalculators.com/html/calculator_display_technology.html>.