Memory Interfacing

C8051 Microcontroller

Jeffrey Pistacchio

Christopher Pybus

Microprocessor Systems

14 November 2016

Introduction

The Magic 8 Ball has been a fun toy for many generations of children. This lab takes the idea of random answer to questions and applies the concept to a program on the C8051 Microprocessor. Furthermore this lab integrates the LCD and Keypad as input and output devices. Ultimately the program is designed to prompt the user to input the type of question (true/false, yes/no, day of the week, number) the user intends to ask and generate an applicable random answer. The entire decision flowchart can be seen below in Figure 1.

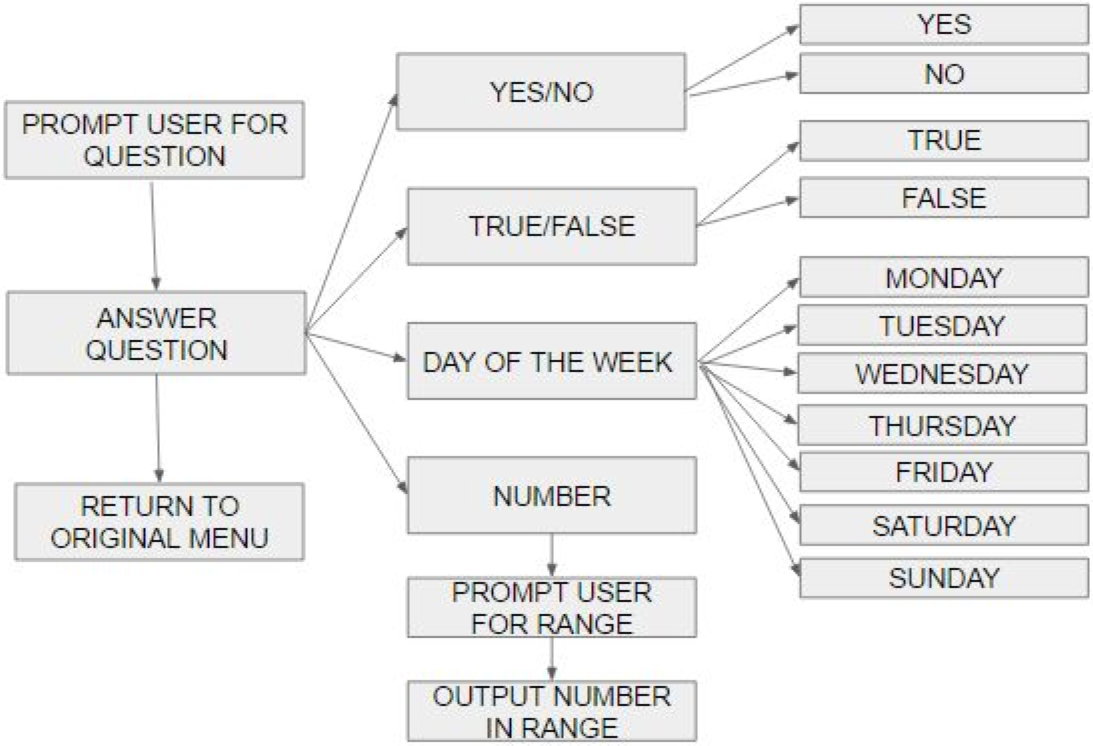


Figure 1: Decision Flow Chart

Methods and Procedures

The lab contained four parts. Part one involved creating the actual logic behind the entire program; making the user interface and the random number generator that would select answers for users' questions. Parts two and three were interfacing the LCD screen and the keypad, respectively. Part four was simply integrating all three previous parts.

Part 1

In part one, the team simply had to code the majority of the software side of the project. They had to come up with a logical flow of control for the program, so that the user could seamlessly use the magic 8 ball interface and get answers to their questions. All of this code was done using the ANSI terminal so that the team could be sure it all worked before attempting to interface it with the LCD screen and keypad.

Four functions were created to handle the four situations the user could ask for:

1) void YesNo(void)

   In this function, a random number is generated and then modulated by 2 in order to get a number that is either 0 or 1. If the number is 0, "yes" is printed. If the number is 1, "no" is printed.

2) void TF(void)

   This function uses the exact same logic as the yesno function.

3) void Day(void)

   In this function, a random number is generated and then modulated by 7 to get a number between 0 and 6 (inclusive). The function then goes through the 7 possible options and has a day of the week associated with each number. For example, if the number is 0, "monday" is printed. If the number is 1, "tuesday" is printed.

4) void range(void)

   This function handles user input as well as the random number portion of the code. It first prompts the user to enter a 3 digit number for the lower bound of the range. Each digit is taken in one at a time, converted to an integer, and then added to the previous digit. The upper bound for the range is handled in the same way. The final random number between the lower bound and upper bound is calculated by generating a random number and modulating it with the difference between the upper bound and lower bound, and then adding the lower bound.

All four of those functions are used from the main control function, which handles general control. The main function runs an infinite while loop that prompts the user for an input choice, and then calls the appropriate helper function above depending on the input. The while loop is broken if the user selects a number outside the available choices. All code can be seen in Appendix C.

Part 2

In part two, the team had to interface the LCD screen with the 8051 microprocessor. The process was relatively simple, as it only involved connecting ports on the LCD screen to ports on the crossbar. The pinout and schematic can be seen in Appendix A. The LCD screen was tested with the sample program provided to the team.

Part 3

In part three, the team had to interface the keypad with the 8051 microprocessor. This process was more difficult than part 2, as there was a good amount of logic involved. In the keypad, there is a mesh of eight wires - four that run along the rows and four that run along the columns. At each of the 16 intersections there is a button. Every time a button is pressed, the row and column wires are shorted and a signal is sent to the 8051 microprocessor.

The vertical wires (columns) are all help high to 5 volts with a 10k pull up resistor. The horizontal wires (rows) are all held low. The vertical wires are also all connected to an AND gate, and the output of that gate is connected to the INT0 interrupt on the 8051 microprocessor. When a button is pressed, it shorts the vertical wire to ground, which changes the output of the AND gate and sends a signal to INT0, telling the 8051 that a button has been pressed, but nothing else about which button has been pressed.-

When the 8051 receives the signal that a button has been pressed, it quickly sends pulses of power to each of the row wires and then analyzes the input from the column wires in order to determine exactly which button was pressed. The code for this part can be seen in Appendix D, and the schematic for the keypad can be found in Appendix B.

Part 4

For this part, all previous parts had to be integrated together so they would work together to provide the user a seamless experience. All of the changes were relatively simple code changes that directed output to the LCD screen instead of the ANSI terminal, and took input from the keypad instead of the keyboard. The difficulty here was that printing out to the LCD screen is much more than a prinf() statement like it is for the ANSI terminal. Also, the LCD screen has much less space for characters than the ANSI terminal, so the team had to change the wording they used in some places. The code for the integrated program can be found in Appendix E.

Results and Analysis

Each section of the lab performed as intended. The Magic 8 Ball followed the flowchart in Figure 1, took input from the keypad, and outputted answers to the LCD Screen. The results of the lab can be seen in Table 1. The results were obtained by running through the different options and observing the displayed result.

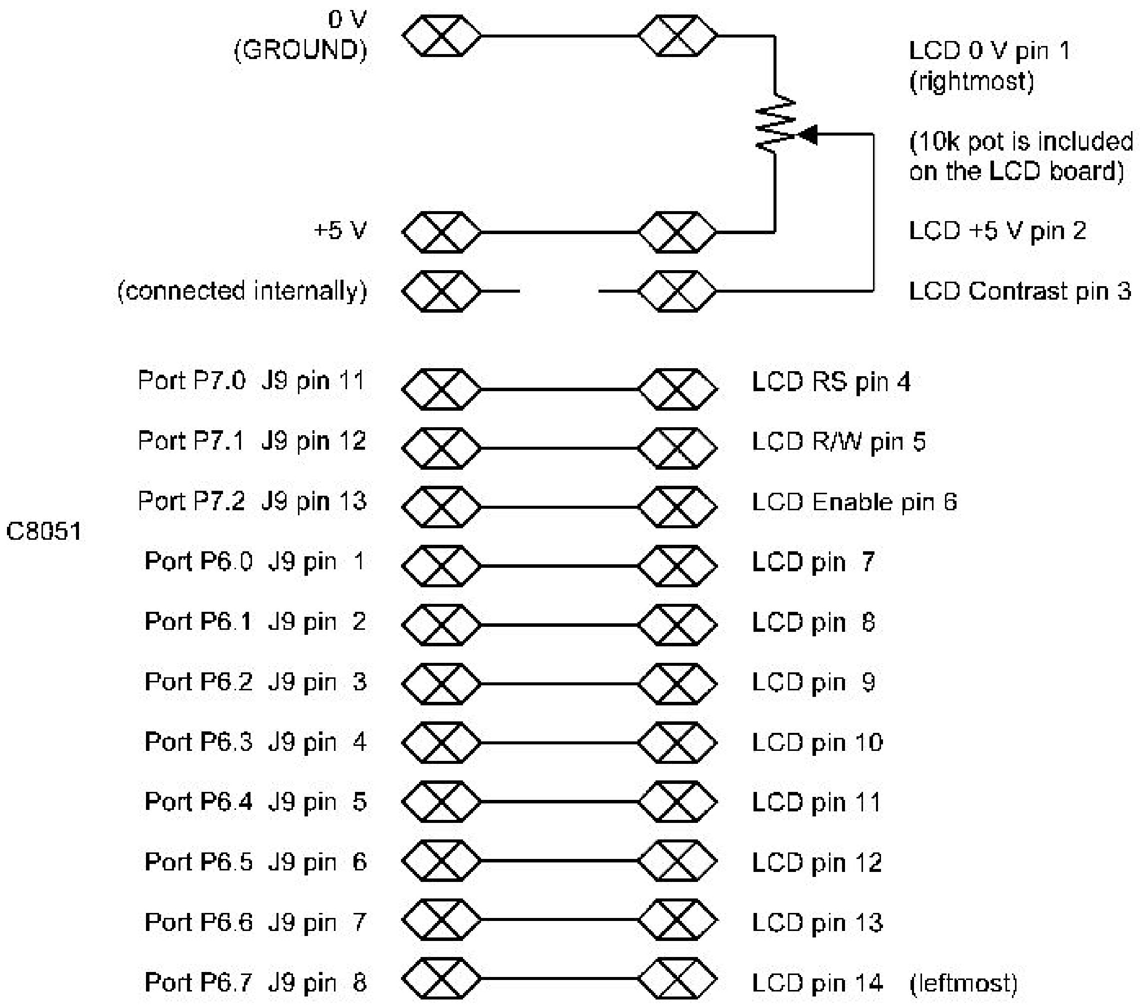
Table 1: Lab Results		
Question Type	**Expected Answer**	**Correct Output?**
Yes/No	Yes	Correct
	No	Correct
True/False	True	Correct
	False	Correct
Day of theWeek	Monday	Correct
	Tuesday	Correct
	Wednesday	Correct
	Thursday	Correct

	Friday	Correct
	Saturday	Correct
	Sunday	Correct
Range	# between upper and lower bound	Correct

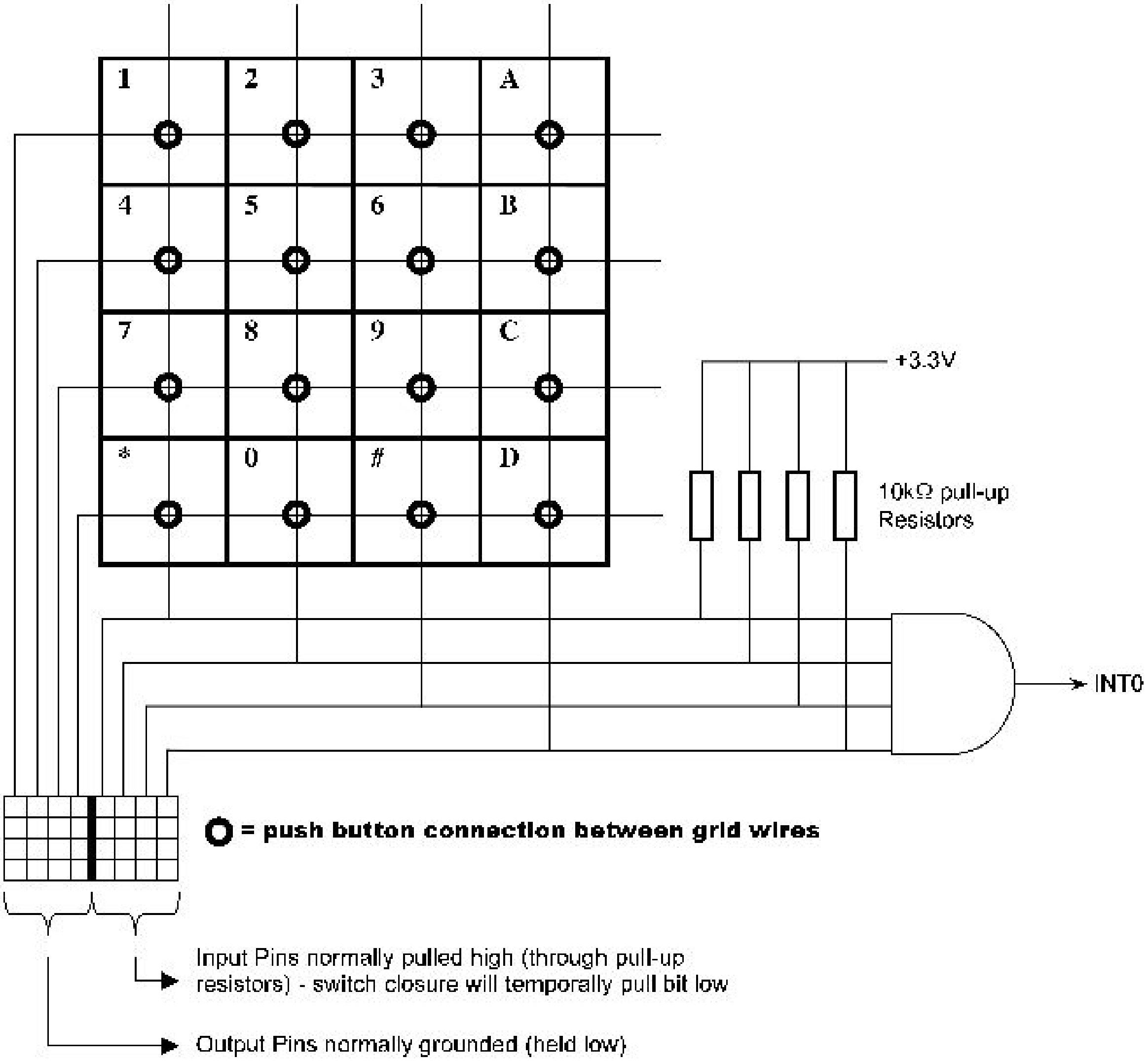
Conclusion

The team successfully accomplished all parts of Lab 6 by making several smaller programs that accomplished all the subgoals. Dividing the project into separate goals made the exercises significantly more approachable due to implementing the divide and conquer mindset. Each goal was simple enough that each group member could complete a goal by the time they came to class, and spend lab time debugging code and building hardware. Given more time the team would have liked to add enhancements, such as adding to the decision making flow chart or improving the randomness of the random number generator by using a continuous counter.

Appendix A - Schematic for LCD Screen



Appendix B - Schematic for Keypad


1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D
+3.3V
10kΩ pull-up Resistors
INT0
= push button connection between grid wires
Input Pins normally pulled high (through pull-up resistors) - switch closure will temporally pull bit low
Output Pins normally grounded (held low)