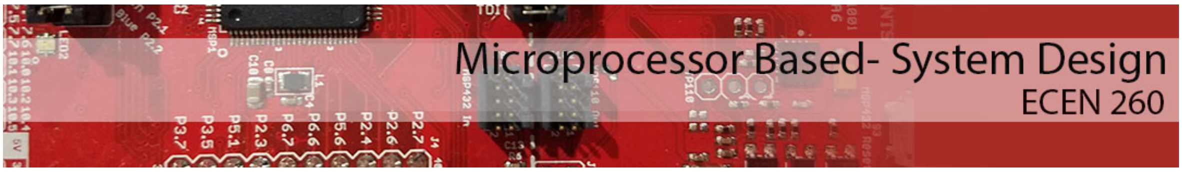

ECEN 260 - Final Project

Adjustable Binary Clock

Dane Artis

Instructor: Brother Allred
April 5, 2022

Contents

List of Tables

List of Figures

1 Lab Overview

1.1 Objectives

The objectives for this project are:

- Use the PCD8544 LCD display to show the time values in a binary format.
- Use the keypad to set or alter the time.
- Use an internal timer on the MSP432 to keep an accurate timing delay.

1.2 Parts

- TI MSP-EXP432P401R Launchpad
- Nokia PCD8544 GLCD Display
- proto board
- 4 X 4 matrix keypad
- 74C922 16-key encoder IC
- 1 uF capacitor
- 10 uF capacitor
- 14 male-female jumper cables
- 12 male-male jumper cables

1.3 Explanation

In this project, I set out to create a clock that would display the time in a binary format. I have seen binary clocks generally made using an array of LED's, however to simplify the system design, I wanted to use the Nokia PCD8544 LCD display. The Texas Instruments MSP432R microcontroller is the brain of the system and our software will be written in C. The MSP432 will interface with the display using the SPI interface. In addition, I will use a membrane keypad to enable altering the values of the clock.

I chose to separate the code for each respective function of the system into their own header files to help isolate the functionality and make the code easier to understand. There is an initialization function for each part that is called in main to set up the system when it is started and powered on. The timer and the keypad use interrupt handlers to control their behavior. See 3 for more information about the code.

1.3.1 Specifications

I wanted to allow multiple ways of changing the time, so I created a multi-modal interface for the keypad, the default mode allowing the user to directly assign a numerical value to each column of the display (using buttons to change which column is being assigned), and another mode that allows the user to just add one to each time category with a button for each respective category. There is an LED on the MSP432 that flashes when the system is put in setup mode.

The limitations of this system are found mainly in the way a user can interact with the device. The keypad is the main user interface, which is not fully reliable, so there is some limitation to interactivity. This dictates that the user must be deliberate with their interaction when pressing buttons, also requiring that the user be careful and aware of whether their input is registered and that the system is in the appropriate state and the values are correct.

1.3.2 Usage

Functions of the keypad:

- * - enter and exit set-up mode

In setup mode:

- D - switch between add and assign modes while in setup mode
- # - reset the time while in setup mode

Assign mode: Switch between the Tens and Ones place of each category.

- A - Hour
- B - Minute
- C - Second

Add mode: Add one (1) to the time category.

- A - Hour
- B - Minute
- C - Second

2 Schematics

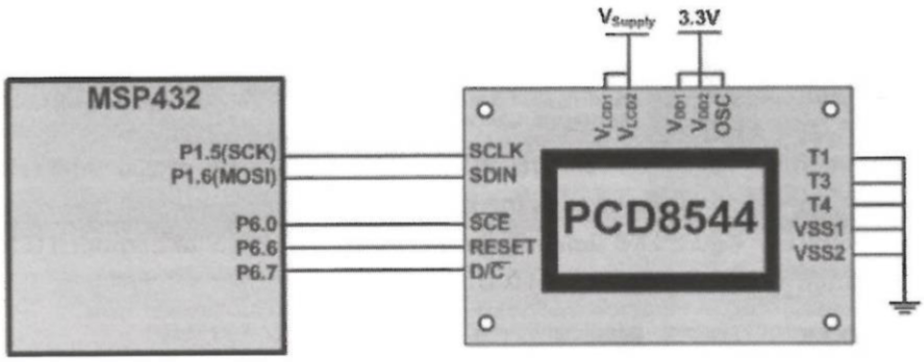

Figure 1: Wiring schematic for how to connect the MSP432 to the PCD8544 display.

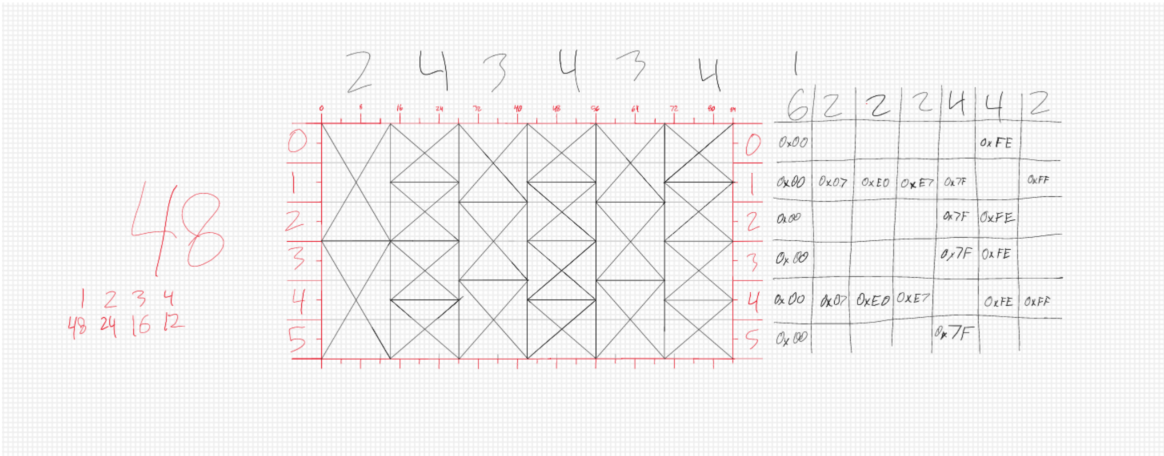

Figure 2: A drawing I made to help work out the logic for how the display should be formatted.

Pin Diagrams

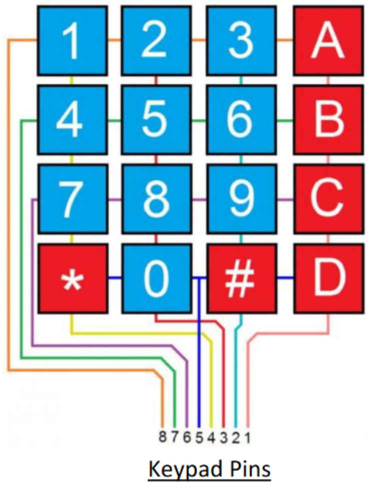

Figure 3: Pinout for the matrix membrane keypad.

Wiring Diagram

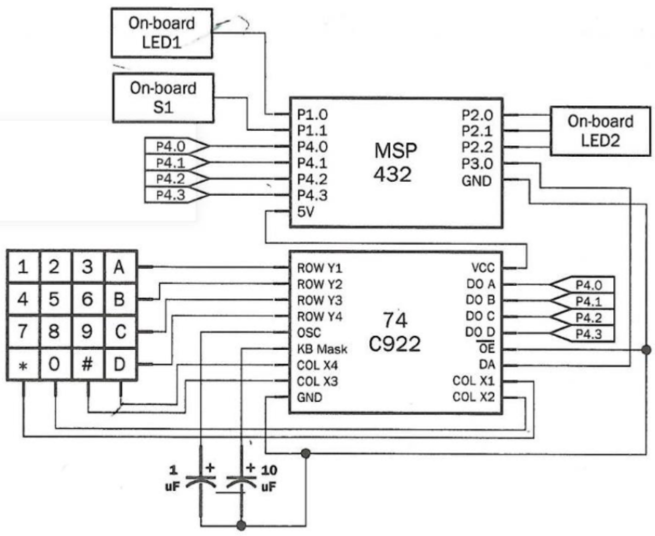

Figure 4: Wiring diagram for connecting the keypad to the MSP432 through the encoder IC.

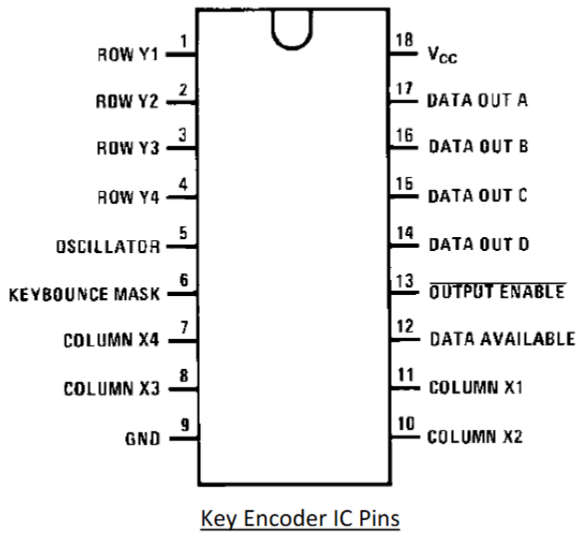

Figure 5: The IC pinout diagram for the keypad encoder.

3 Code

The code for the project can be found in the zip file. The code is broken up into header files for each part of the system. Below is a short description of each file in the directory.

3.1 Binary Time

binary_time.h Contains all of the logic for tracking the time using a decimal value for each column of the display in a size two array and converting the decimal value into an array of "binary" values. This means a individual 1 or 0 in each index of a size four array to simulate a 4-bit little-endian binary value to simplify the application of the time value on the display.

3.2 Display

display.h Contains the code required to control the display using the SPI interface (taken from the Lab 11 scaffolding code). There is also code that determines the logic behind displaying the correct binary figures on the display with converting from the "binary" array of time values to the correct byte values, broken up into the display banks.

3.3 Keypad

keypad.h Contains the code to decode the keypad input into usable values for the program. This code came from the scaffolding code in Lab 7.

3.4 Timer/Clock

timer.h The code in this file controls the internal Timer32 in the MSP432 (see Listing 7.7 in Programmable Microcontrollers [1]).

3.5 Port

port.h A driver file that contains functions for simplifying interfacing with the I/O ports and pins on the MSP432. This file was written by me and makes use of the msp driver files. This is used for the flashing light indicator for setup mode.

3.6 Main

main.c Initializes all parts of the system and contains the interrupt handlers for the timer and the port 3 IRQ for the keypad.

4 Test Plan and Test Results

I want to first test the clock display to make sure that all parts of the binary representations are drawn correctly on the display. I will try first drawing the maximum binary value that could be displayed on the screen, then iterate through the numbers to check to make sure each bit combination is displayed correctly. Next, I want to test the keypad, going through each mode and testing that all possible functions work correctly.

4.1 Expected and Observed Results

Table 1 shows the specific steps for the described scenarios, together with the expected and observed results of those steps.

	Test Plan Steps	**Expected Outcome**	**Observed Outcome**
1	Filled display value	Correct bit figures (2-4-3-4-3-4)	Correct bit figures (2-4-3-4-3-4)
2	2-bit value display	Correct binary combinations	Correct binary combinations
3	3-bit value display	Correct binary combinations	Correct binary combinations
4	4-bit value display	Correct binary combinations	Correct binary combinations
5	"*" = Setup mode w/LED	LED flashes, assign mode default	LED flashes, assign mode default
6	"#" = clears display in both modes	Resets display memory on push, only in setup mode	Resets display memory on push, only in setup mode
7	"A", "B", "C" switch time columns (Assign mode)	Hour default, tens place default, A - hour, B - minute, C - second	Hour default, tens place default, A - hour, B - minute, C - second
8	Number keys assign value correctly (Assign mode)	Correct value assigned to correct column	Correct value assigned to correct column
9	"D" switches between modes	Cycles between assign and add modes	Cycles between assign and add modes
10	"A", "B", "C" adds to value correctly (Add mode)	Adds only one per category, A - hour, B - minute, C - second	Adds only one per category, A - hour, B - minute, C - second

Table 1: Test cases and recorded results.

4.2 Video Demonstration

Explanation and Demonstration: `https://youtu.be/1WTI7uzjeys`

5 Discussion and Conclusion

In this project, I worked to combine multiple concepts from various labs from this semester to make a binary clock. I combined the concepts of the SPI communication protocol for using the PCD8544 display, the Timer32 and Matrix keypad using interrupts, and input/output with LED1 on the LaunchPad.

Something I had difficulty with in this project was getting things to display correctly on the display. Part of the difficulty was that I was misunderstanding what should have been displaying for certain time values. It took quite a bit of time working out what should be displayed for each part of the "LED" display value, and there was a point where I had the wrong byte value for one of the squares because I thought the the byte values should be mirrored, however it was the binary string that should be mirrored, meaning there needed to be a different byte value for that section of the display. Once that was resolved and I had simplified the overall logic of the display values, it was much easier to fix things and I was able to resolve the issues.

I really enjoyed working on a larger coding project like this because getting everything together and working was rewarding. This is a product that could actually be used as a clock and it would be accurate enough to keep time over an extended period. I feel that the scope of the project was appropriate and the result was satisfactory.

References

[1] C. Unsalan, H. Gurhan, and M. Yucel, *Programmable Microcontrollers: Applications on the MSP432 LaunchPad*. McGraw-Hill Education, 2017.