Final Lab Report of EI209

Yijia Diao 518030910146

How to Test the Program

To run the project, just open "YijiaDiao_518030910146_Finallab.pdsprj", then build and run. You can press BT2 at any time to test.

The initial values are: Date: 0517, Time: 1959, down-count: 1000.

In the submission file, I place the source code file "YijiaDiao_518030910146_Finallab.ASM" and a test code file "YijiaDiao_518030910146_Finallab_test.ASM". Since 1 minute is too long to test, I change the time to 1s for the convinent of test.

The Main Idea of the Lab

Here we assume that the requirement 1,2,3 needs to display in Mode One, Mode Two, Mode Three.

1. **Time Count of the Program**

   - I choose **5ms** as the time period of interrupt. So I use Timer1's output as the input of U15:B. Timer1 will divide input OUT0 by 5.

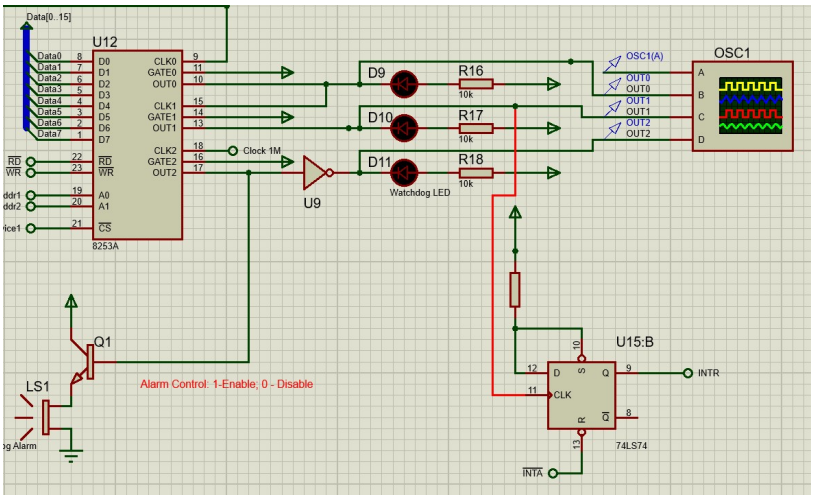


Figure 1: Circuit design of U15:B

   - From the base period of 5ms, the count number of 1s = 1000ms ÷ 5ms = 200, the count number of 1 min = 60 × 1000ms ÷ 5ms = 12000. I use variable "OneSecondCount" and "OneMinuteCount" to count the number of interrupt, and judge their value to decide whether it's 1s(1min) or not.
   - For the refreshment of LED digits, I choose to refresh the next digit every 5ms, and use variable "LEDDisplayCount" to do counting.
   - I choose to check the state of BT2 every main loop. I don't choose the method of checking per 0.1s since its respond is too slow.

2. **BT2 Switching Principle**

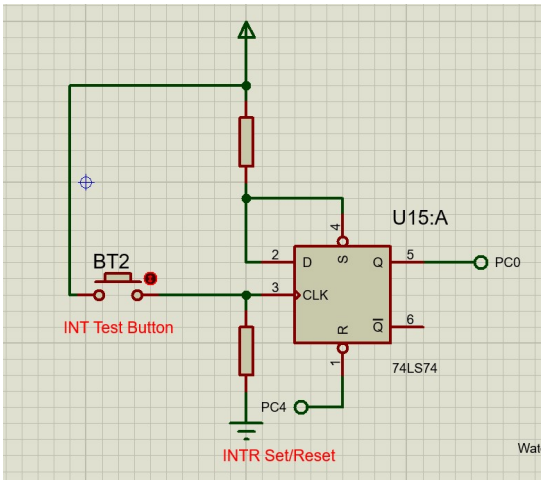


Figure 2: Circuit design of U15:A

I choose to connect Q pin to PC0, connect Reset pin to PC4.
Therefore, the initial state of PC4 must be 1, so that after pressing BT2, the content of this latch stay 1 until the program check the state of PC0. After program's check, to clear the content, we must set PC4 to 1. Finally we must to set PC4 to 0, in order to capture new push on BT2. You can refer to Fig.4 for its flow chart.
I use variable "ButtonPushTimes" to record the push times of BT2: Every time the program get the "push" signal (PC0 = 1), its value will be changed. Its value can be 0,1,2, and the program will choose which mode to display according to its value.

3. **Value Store, Calculate and Display Method**

- Store: There are 3 time values: "ModeOneData", "ModeTwoData", "ModeThreeData". I choose to use 4 bits to store a digit number, so a word for each time value.
- Calculate: Mode1 no need to change. Mode 2's calculate function is "ChangeMinute". Mode 3's calculate function is "CountDown". You can refer to the source code.
- Display: I use variable "DisplayIndex" to record the index of digit that will be refreshed. Every time we need to refresh the next digit, choose the relevant 4 bits of the time value, and output the index and 4-bit value to 8255 Chip.

4. **Flow Charts of the Program**

The first flow chart shows the modules of main program:

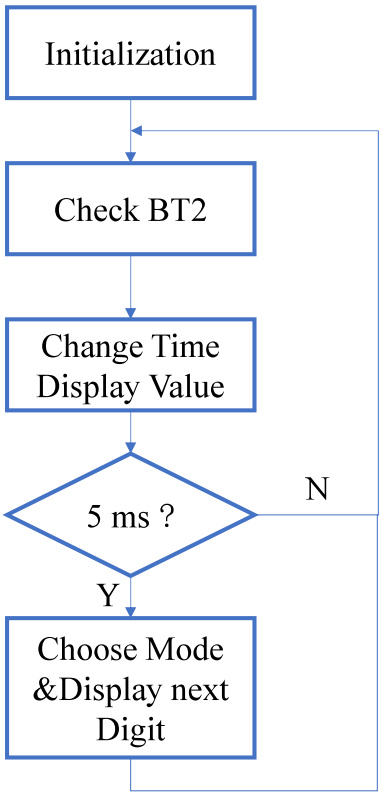
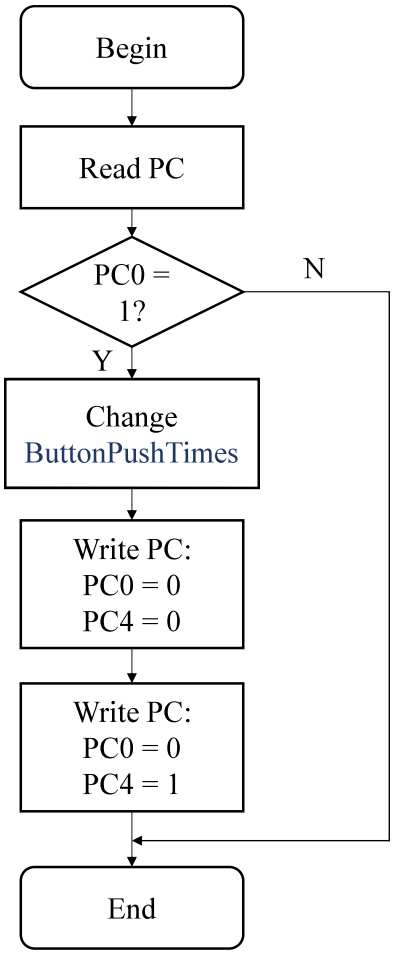


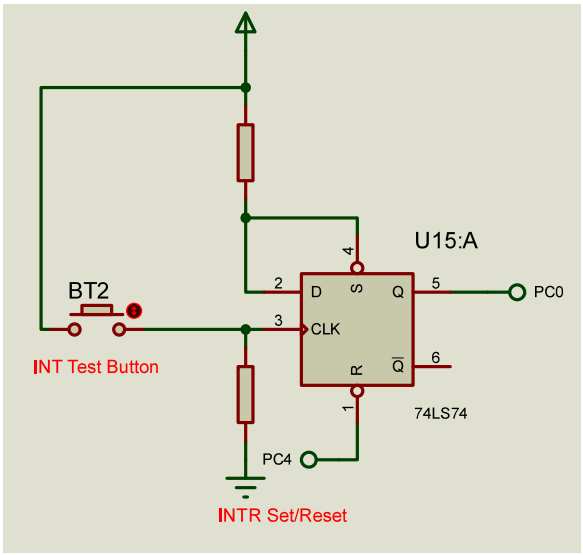
Figure 3: Modules of Main program

The judgment of "5ms" indicates that I refresh the next LED digit every 5ms.

Here to show the modules in flow charts below. They show the detailed frame of my code.

- **Check BT2 Module**
  The principle is explained in 2. The flow chart is:





Initial state:
BT2 not connected, PC4 = 1, PC0 = 0
Push BT2:
PC0 Changes to 1, maintain the state
Program Read PC0 = 1:
1. Reset PC0 to 0 through Reset Pin, so write PC4 = 0
2. Return Initial state.

Figure 4: Check BT2 Module

- **Change Time Display Module**

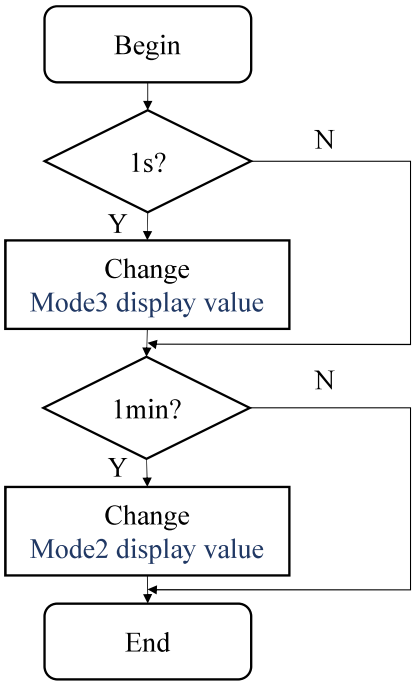


Figure 5: Change Time Display Module

- **Choose Mode & Display next digit Module**

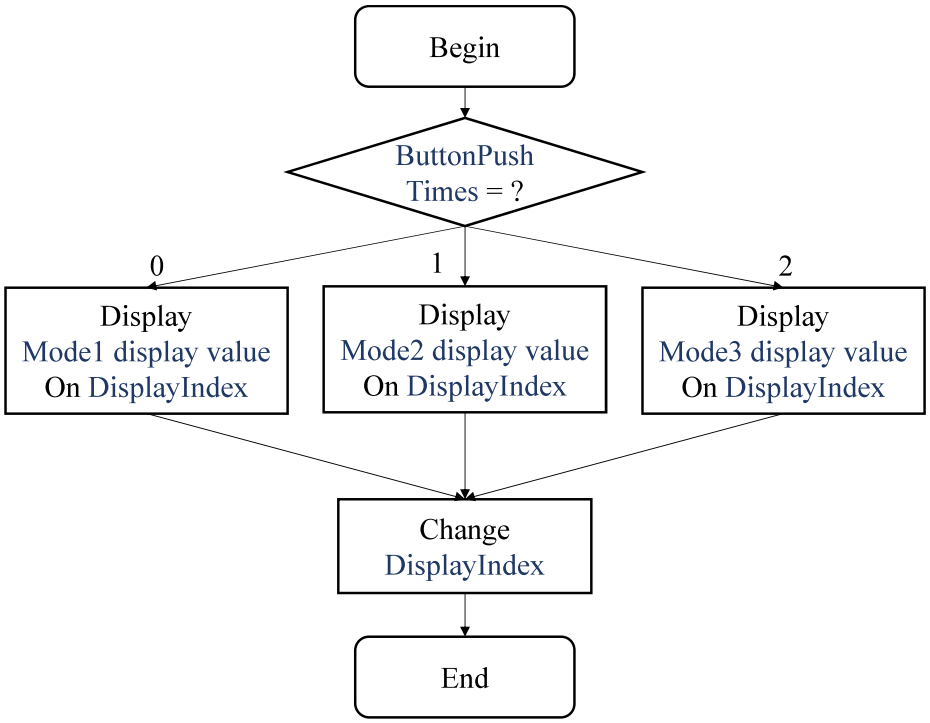


Figure 6: Choose Mode & Display next digit Module