Team Members: Ted Paulsen and Daniel Machlab

Professor Beichel

Embedded Systems

Lab 2 Report

1. **Introduction**

The goal of this lab was to create a counter that could count up, count down, and reset. The number increment or decrement on the counter is to be activated by a pushbutton press of less than one second. Change of modes—increment or decrement mode—is to be activated by a pushbutton press between one and two seconds. A counter reset to zero is to be activated by a pushbutton press of more than two seconds.

1. **Schematic**

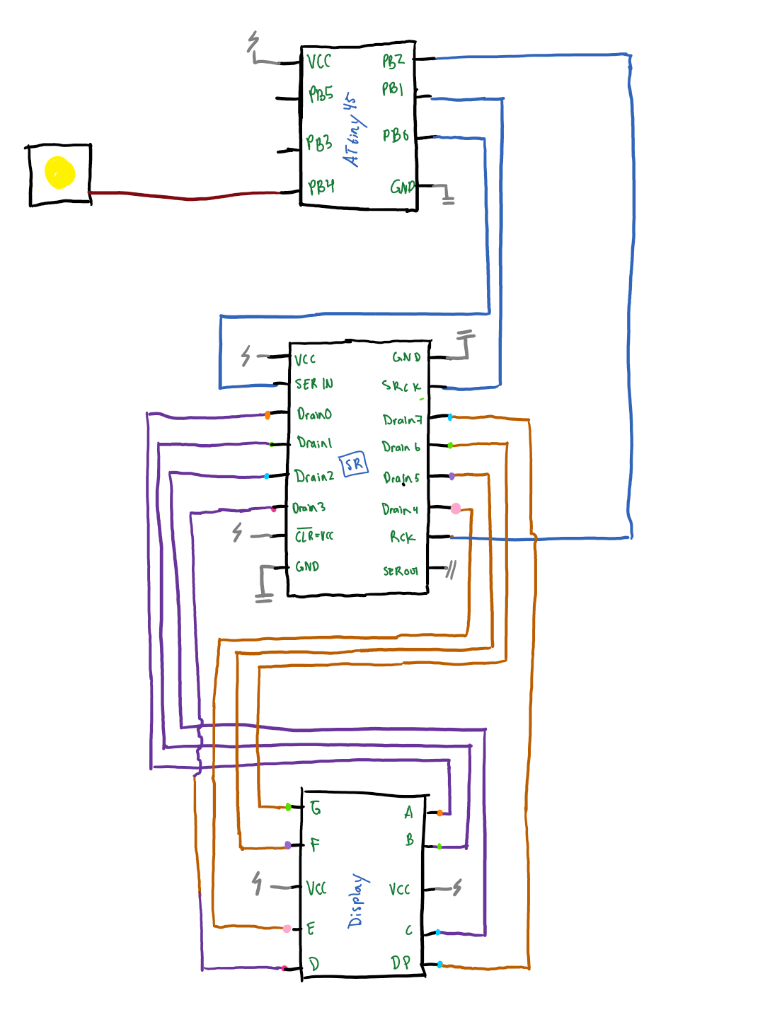


Figure 1: Circuit as implemented

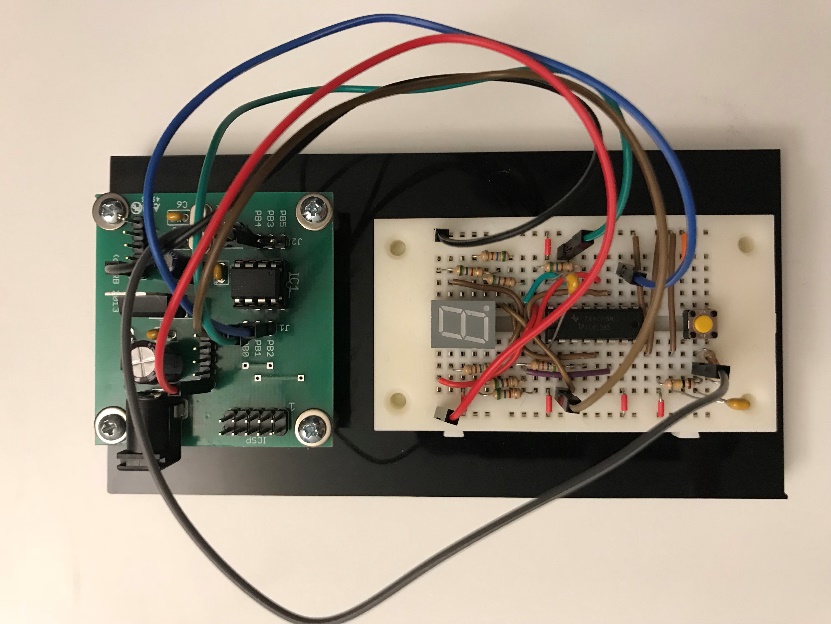
Figure 1 shows the circuit design as it was implemented. PB0, PB1, and PB2 are all configured as outputs for SER\_IN, SRCK, and RCK respectively. PB4 is configured as in input from the pushbutton. When the pushbutton is pressed, the voltage transitions to low.

1. **Discussion**

Our project consisted of two basic design phases: hardware and software. First, we worked on hardware. Using out class notes as reference we were able to successfully implement hardware debouncing, wire the correct shift register pins the ATtiny, and connect the display to the shift register (Fig. 1). Second, we began working on software—the more difficult part of the project because neither of us have experience with assembly.

We continuously iterated on our code to take it one step further each time. We began by displaying a single digit. After that we worked on wiring the push button. We confirmed that we could receive input from the push button using the oscilloscope. Our next iteration was to change between two numbers. Once we achieved that, we worked to count up to nine and wrap around. The reset functionality caused us a lot of difficulties. We solved this by implementing a listener loop to wait for button presses. A separate loop was then called to count until the button is released. The final iteration was to include the decrement mode and implement it as a third possible state chosen by the push button.

At a high level, our program works as follows. A main loop, which is running constantly, waits until the user presses the button. When the button is pressed, a timer subroutine is called. The timer subroutine counts until the button is released. When released, the program jumps to a logical routine to determine the action to be taken. If the value stored in register from the timer method is greater than 2 seconds, zero is displayed. If the value stored in the register from the timer method is between one and two seconds, the value in the register that stores the mode is toggled. To display the decimal point when in decrement mode, we computed the bitwise or of the register which stored the absolute value of the current number and 10000000. This had the effect of setting the decimal point on. Finally, if the value of the timer subroutine was less than one second, it called another subroutine, nextnum, which displayed (via a series of hard-coded comparisons) the appropriate next digit.



*Figure 2: Circuit wiring overview*

1. **Conclusion**

From this lab we have really learned the basics of assembly. At the beginning of the lab, we started with hardware. Initially we had no clue were to start with wiring our breadboard, but after a short time we were able to figure out what direction we needed to go in. Eventually we fully understood what was going on and finished the wiring. This same story happened with software. Initially were clueless, but slowly we progressed and learned more until the lab was finished. Now, we both have a firm understanding of both hardware and software used in the lab.