**Lab 2 Report**

Sage Marks, Matt Krueger

3360:0001 - Embedded Systems

Professor Beichel

University of Iowa, College of Engineering

## Introduction

The lab builds a hexadecimal up/down counter. It uses an ATmega328P microcontroller, an 8-bit shift register, a 7-segment LED display, and a pushbutton switch. The 7-segment display shows "0" at power on, and the counter increments by default. The pushbutton controls mode selection, increment/decrement, and reset. A pushbutton press for less than one second will increment or decrement the count based on your current mode. Pressing for one to two seconds will switch between increment and decrement modes. A press that exceeds two seconds will reset the count to 0 and you will once again be in increment mode.

## Schematic

A computer screen shot of a circuit board

AI-generated content may be incorrect.

Electrical circuit schematic created using KiCAD

**Materials List**

|  |  |  |
| --- | --- | --- |
| Hardware | Quantity | Description |
| Atmega 328P µC | 1 | Programmable µC |
| 74HC595 Shift Register | 1 | Storage of hex codes for 7-Segment display |
| 5161AS 7-Segment Display | 1 | Display current counter |
| Enable Low Push Button | 1 | Enables user interaction with 7-Segment display |
| 560Ω Resistor | 8 | Resist current into 7-Segment display LEDs |
| 10KΩ Resistor | 1 | Pull up resistor for push button |
| 100KΩ Resistor | 1 | Pull up resistor for push button |
| 0.1µF Capacitor | 2 | Decoupling capacitors for button and µC |

**Components**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | A screen shot of a computer program  AI-generated content may be incorrect. |
| 74HC595 | 5161AS | Push Button | Atmega 328P |
| #TODO | #TODO | #TODO | #TODO |
|  |  |  |  |

The design of this circuit successfully utilizes the ATmega328P and a 7-segment display with the use of a shift register. There are four I/O lines configured in the schematic, PB0 is configured as an output and as SER, PB1 as an output for RCLK, PB2 as an output or SRCLK, and PB3 as an input for a low signal from the button press. This is an active low push button meaning that when the push button is pressed the logic level goes to 0.

Our design utilizes a hardware-based debounce approach for the push button. This approach consists of a pull-up resistor (10K) to keep the node at a defined state when the button is not pressed and an RC low pass filter with a 0.1 microfarad capacitor and a 10K resistor. The low pass filter helps with possible oscillation that can occur when there is a button press, so the Arduino can recognize a press correctly and not increment/decrement when it is not supposed to.

## 3. Discussion

**Functionality**

As stated in the abstract, the 7-Segment display is controlled by user via interacting with a push button. There are three modes, and their descriptions are as follows:

i) Increment Mode

- Press button < 1 second: Increment count by 1

- When display shows "F": Next increment rolls over to "0"

ii) Decrement Mode

- Press button < 1 second: Decrement count by 1

- When display shows "0": Next decrement rolls over to "F"

- DP LED on to indicate decrement mode

iii) Reset

- Press and hold button ≥ 2 seconds: Reset to "0" and enter increment mode

#TODO – ADD IMAGES OF EACH MODE

## 4. Conclusion

## 5. Appendix A: Source Code

<https://github.com/mattnkrueger/embedded_systems_marks_krueger/tree/main/labs/lab2>

#TODO – flow diagram of code (use draw.io or sum)