Memo to: Mr. Cory Mettler
From: Julian Rechsteiner, Bennett Christenson
Date: April 13, 2021
Regarding: EELE 465, Project 6 – Peltier Device Control

Summary:

The purpose of this project was to control the temperature of a plant using a thermoelectric device and monitor the temperatures of the plant and the ambient air temperatures using temperature sensors and a Liquid Crystal Display (LCD).

Setup:

The keypad was connected to the master device using ports 1.0-1.7, while the I2C clock and data lines utilized ports 4.7 and 4.6 respectively, Figure 1. Two pull-up resistors were added to these lines to achieve a consistent logic high. These lines were then connected to ports 1.3 and 1.2 respectively to each slave device. The LM19 temperature sensor's Vout was connected to the Master via port 5.0 and was powered with the MSP 3.3V supply and grounded using the MSP ground. The RTC's data and clock pins were connected to the data and clock lines from the Master device and was powered via 3.3v supply. The LM92 temperature sensor's pins 1 and 2 were connected to the data and clock lines of the master, pins 6 and 7 were grounded to achieve a slave address of 0x48 and was powered via 3.3v and grounded using the master's ground and 3.3v supply. Slave device 1 interfaced with the LED bar through ports 1.0, 1.1, and 1.4-1.7, along with ports 2.7 and 2.6. A resistor of 101-ohm sat in the middle to reduce the current flow. Slave device 2 interfaced with the LCD display through ports 1.7-1.4 as data ports, 2.6 as the RS port, and 2.7 as the enable port. The LCD was powered by a 3.3-volt supply through a potentiometer output to the display power and the logic was powered using just the 3.3-volt supply. The Peltier thermoelectric device was controlled using port 5.2 to toggle the cooling function and port 5.1 to toggle the heating function. These ports were connected to a transistor gate each to control opening and closing of each transistor. A flyback diode was also connected to the drain to prevent any damage to any devices do to current. The Peltier Device was powered using a 12 V supply through a wall outlet.

Step 1 – LM19 Data Collection:

LM19 data collection was implemented to occur every half a second, Figure 7, via a timer set to overflow every .167 seconds, Figure 2. Whenever this timer triggered 3 or 6 times, LM19 data would be collected using the ADC module. The data collected would be stored in an array to be averaged and converted later, Memo 5.

Step 2 – LM92 Data Collection:

LM92 data collection was implemented to occur on the same timing as the LM19, every half of a second. Whenever this timing would occur, data would be collected from the LM92 via I2C, Figure 3. The LM92 would send two bytes of data indicating the temperature. The lowest three bits of the data were not used in the temperature, so they were removed, and the overall number was shifted to compensate for it. The data would then be multiplied by .0625 in order to acquire the temperature in degrees Celsius. This data was then stored in an array to be averaged by the moving average later.

Step 3 – RTC Seconds Tracking:

Real Time Clock interfacing would occur every second. This was done by utilizing the timer and only gathering data once it triggered 6 times (this corresponded to activating once a second, Figure 6). Once this occurred, a subroutine would be activated and data from the RTC would be retrieved. This was done by changing the master's I2C module to receive data from the clock, Figure 4. The minutes and seconds would both be collected.

Once the data was collected, the minutes and seconds would be converted to integers and the minutes would be converted into seconds and added to the seconds that were collected. The result was the total number of seconds of operation.

Finally, the seconds were converted into individual ASCII characters and stored into an array to send to the LCD to be displayed.

Step 4 – Moving Average:

The moving average was programmed to be calculated only once the array with the LM19 data was populated with the correct number of data points. For example, if six data points were selected via the keypad, the moving average would only be calculated once there were six ADC values in the array.

Step 5 – Peltier Control:

The Peltier Control was implemented by dedicating two ports to connect to the gates of the two transistor switches to activate heating or cooling modes. When the user pressed the A button on the Keypad, the cooling port would be turned off and the heating port would turn on, which would close the heating switch, activating heating mode on the thermoelectric device, Figure 5. When B would be pressed, the heating port would turn off and the cooling port would turn on, closing the cooling switch and activating cooling mode. When the D button would be pressed, both ports would turn off and both switches would open, putting the device into a sort of standby mode.

Step 6 – LCD Display:

The LCD was controlled by sending ASCII characters to a slave device, which interfaced with the LCD, Memo 5. The LCD displayed the sample number that was selected by the user, the temperature read by the LM19 in Celsius, the mode of operation selected by the user, the number of seconds of use, and the temperature of the Plant in Celsius.

Step 7 – LED Display:

The LED display was controlled by sending the mode of operation to a slave device that interfaced with the LEDs, Memo 4. Patterns were added to create different scrolling effects when heating or cooling the Plant. A third alternating pattern was also added to indicate that the Plant was neither being actively heated or cooled. Each pattern would change states every third of a second, Figure 8.

Conclusion:

The focus of this project was to control the temperature of a plant using a thermoelectric device and temperature sensors. A major part of this project was unfortunately unable to be completed due to time constraints. This part being the C function for automatic heating and cooling of the Peltier device. However, heating and cooling were separately implemented successfully as were the temperature and time data collection and display.

Lessons Learned:

- Project Management often requires making sacrifices for the sake of deadline
- Often making a single timer and creating conditions within it is simpler than making more
- Getting the right sequential order for function calls matters when timing is a factor

Appendix A - Figures:

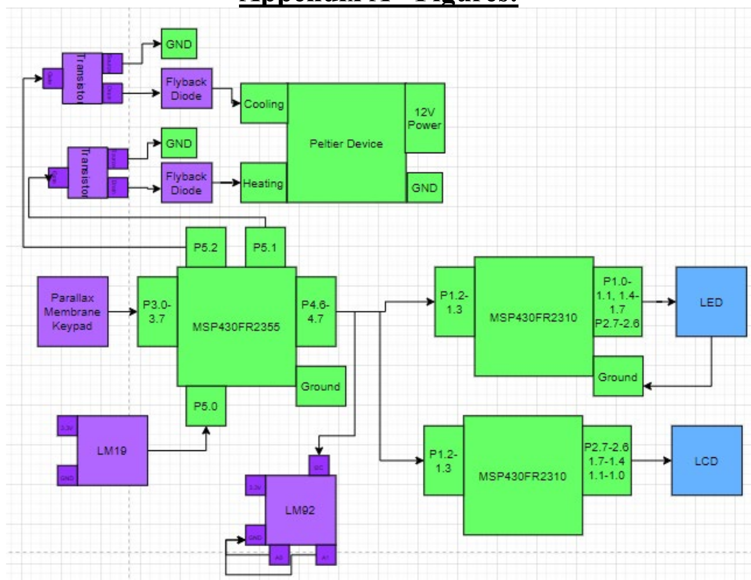


Figure 1: Circuit Diagram

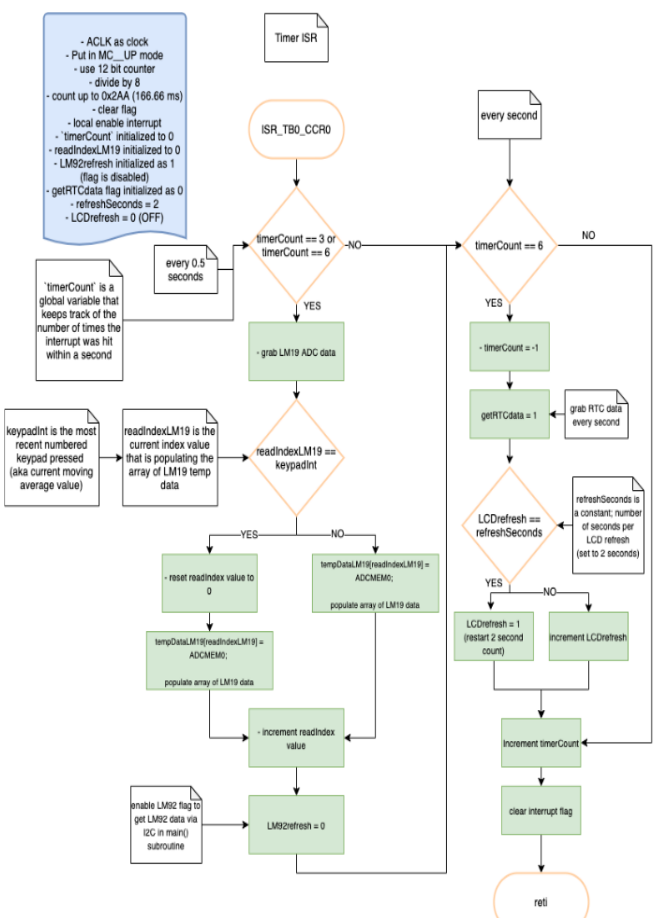


Figure 2: Timer ISR Flowchart

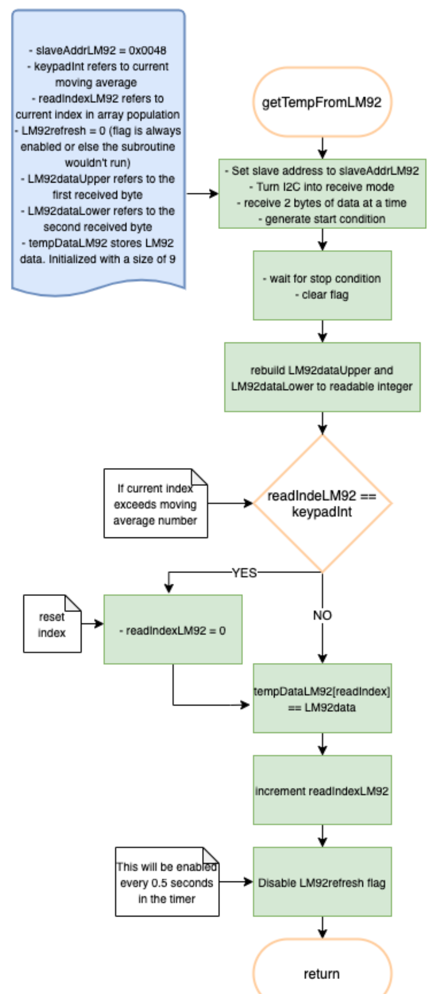


Figure 3: LM92 Communication

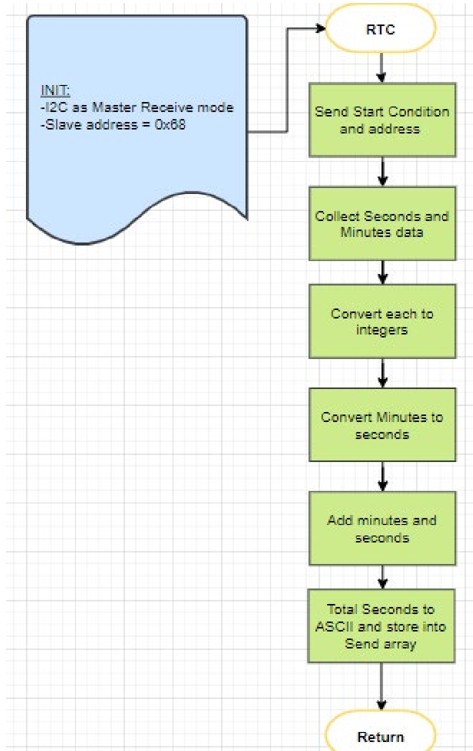


Figure 4: RTC Communication Flowchart

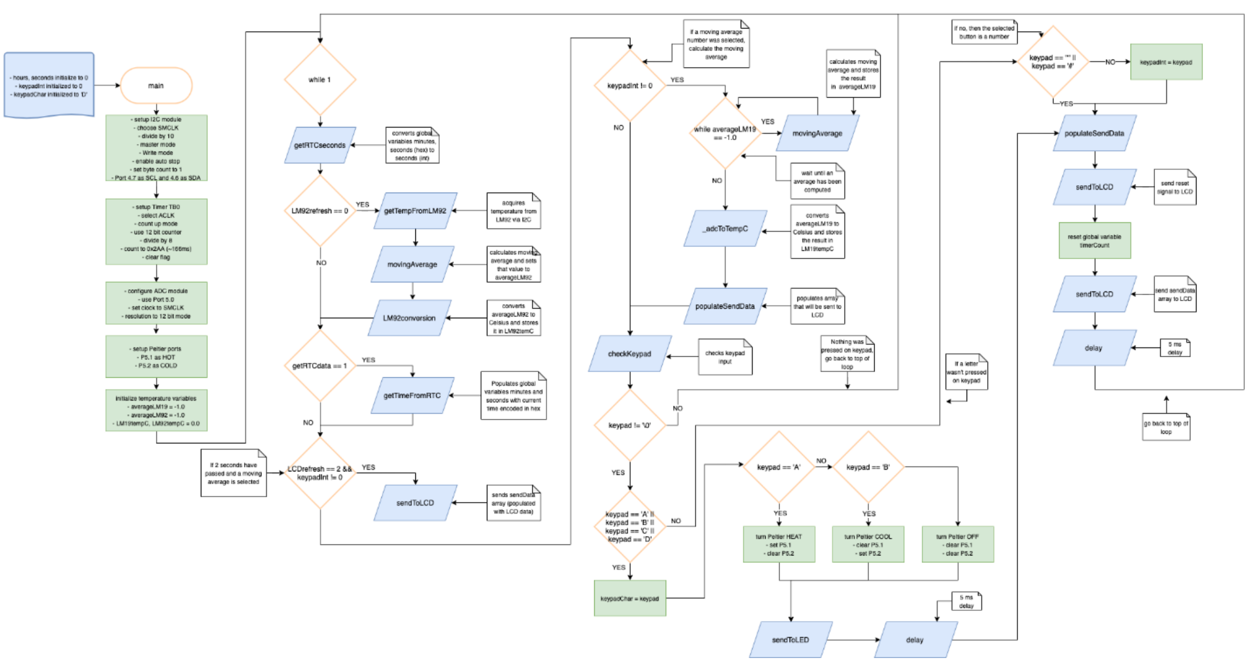


Figure 5: Master Flowchart

Appendix B – Timing Evidence

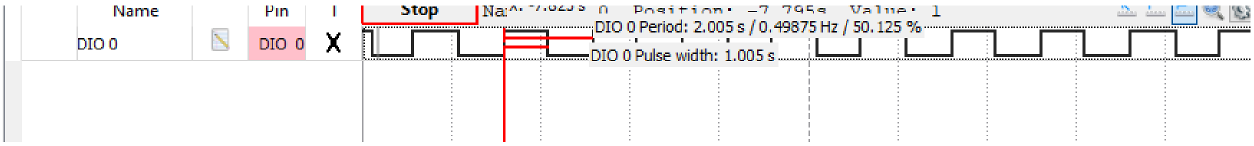


Figure 6: 1 Second RTC Timing Evidence

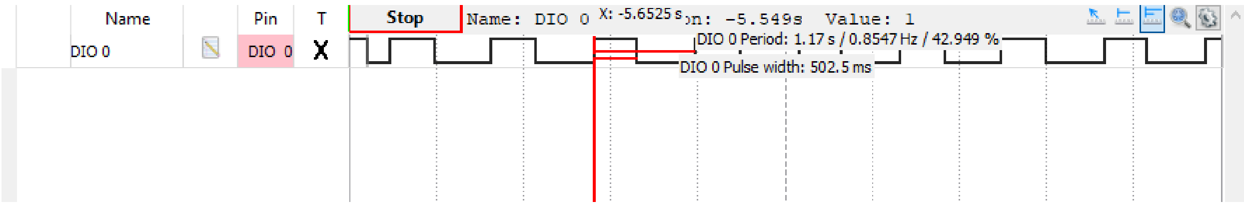


Figure 7: .5 Seconds Temperature Timing Evidence

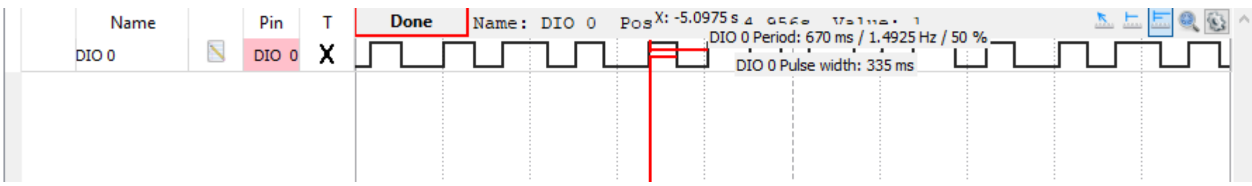
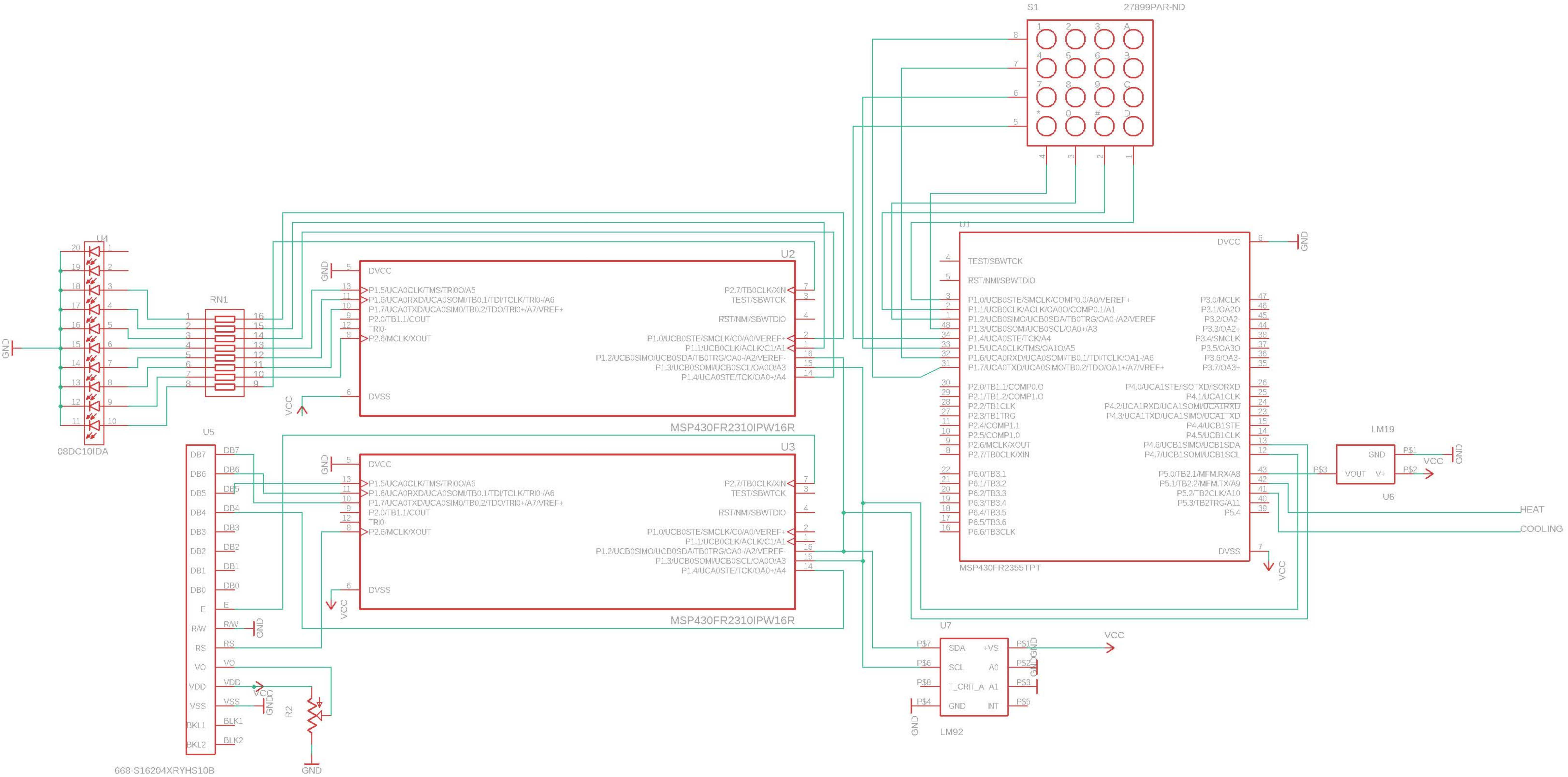


Figure 8: .333 Seconds LED Scrolling Timing Evidence



S1
27899PAR-ND
1
2
3
A
4
5
6
B
7
8
9
C
*
0
#
D
U4
08DC10IDA
RN1
U2
MSP430FR2310IPW16R
DVCC
P1.5/UCA0CLK/TMS/TRI0O/A5
P1.6/UCA0RXD/UCA0SOMI/TB0.1/TDI/TCLK/TRI0-/A6
P1.7/UCA0TXD/UCA0SIMO/TB0.2/TDO/TRI0+/A7/VREF+
P2.0/TB1.1/COUT
TRI0-
P2.6/MCLK/XOUT
DVSS
P2.7/TB0CLK/XIN
TEST/SBWTCK
RST/NMI/SBWTDIO
P1.0/UCB0STE/SMCLK/C0/A0/VEREF+
P1.1/UCB0CLK/ACLK/C1/A1
P1.2/UCB0SIMO/UCB0SDA/TB0TRG/OA0-/A2/VEREF-
P1.3/UCB0SOMI/UCB0SCL/OA0O/A3
P1.4/UCA0STE/TCK/OA0+/A4
U3
MSP430FR2310IPW16R
U1
MSP430FR2355TPT
TEST/SBWTCK
RST/NMI/SBWTDIO
P1.0/UCB0STE/SMCLK/COMP0.0/A0/VEREF+
P1.1/UCB0CLK/ACLK/OA0O/COMP0.1/A1
P1.2/UCB0SIMO/UCB0SDA/TB0TRG/OA0-/A2/VEREF
P1.3/UCB0SOMI/UCB0SCL/OA0+/A3
P1.4/UCA0STE/TCK/A4
P1.5/UCA0CLK/TMS/OA1O/A5
P1.6/UCA0RXD/UCA0SOMI/TB0.1/TDI/TCLK/OA1-/A6
P1.7/UCA0TXD/UCA0SIMO/TB0.2/TDO/OA1+/A7/VREF+
P2.0/TB1.1/COMP0.O
P2.1/TB1.2/COMP1.O
P2.2/TB1CLK
P2.3/TB1TRG
P2.4/COMP1.1
P2.5/COMP1.0
P2.6/MCLK/XOUT
P2.7/TB0CLK/XIN
P6.0/TB3.1
P6.1/TB3.2
P6.2/TB3.3
P6.3/TB3.4
P6.4/TB3.5
P6.5/TB3.6
P6.6/TB3CLK
DVCC
P3.0/MCLK
P3.1/OA2O
P3.2/OA2-
P3.3/OA2+
P3.4/SMCLK
P3.5/OA3O
P3.6/OA3-
P3.7/OA3+
P4.0/UCA1STE/ISOTXD/ISORXD
P4.1/UCA1CLK
P4.2/UCA1RXD/UCA1SOMI/UCA1RXD
P4.3/UCA1TXD/UCA1SIMO/UCA1TXD
P4.4/UCB1STE
P4.5/UCB1CLK
P4.6/UCB1SIMO/UCB1SDA
P4.7/UCB1SOMI/UCB1SCL
P5.0/TB2.1/MFM.RX/A8
P5.1/TB2.2/MFM.TX/A9
P5.2/TB2CLK/A10
P5.3/TB2TRG/A11
P5.4
DVSS
LM19
U6
GND
VOUT
V+
P$1
P$2
P$3
HEAT
COOLING
U5
668-S16204XRYHS10B
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
E
R/W
RS
VO
VDD
VSS
BKL1
BKL2
R2
U7
LM92
SDA
SCL
T_CRIT_A
GND
+VS
A0
A1
INT
P$7
P$6
P$8
P$4
P$1
P$2
P$3
P$5
VCC
GND