Austin Petersen, Ayden Dauenhauer

Prof. Wolfe

ELEN 120L Tuesday 2:15 p.m.

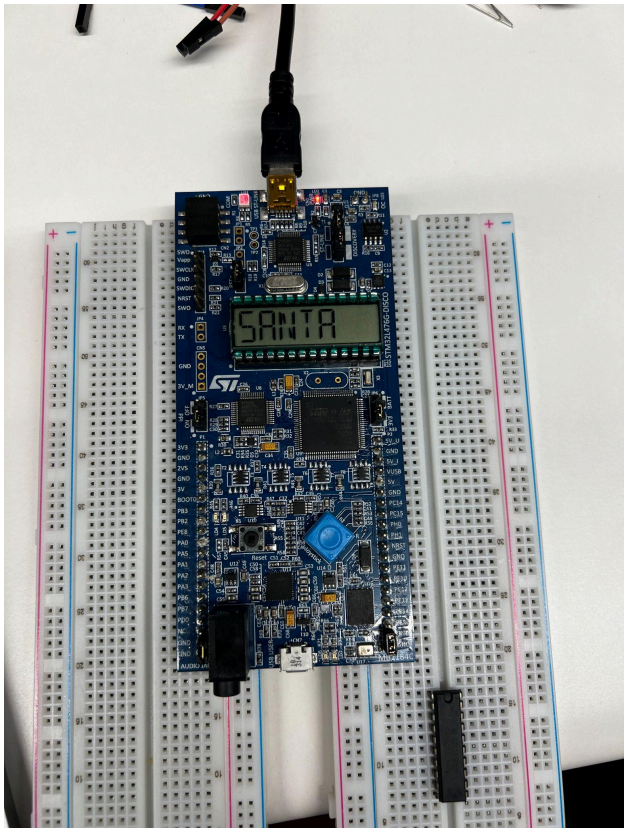
28 November 2023

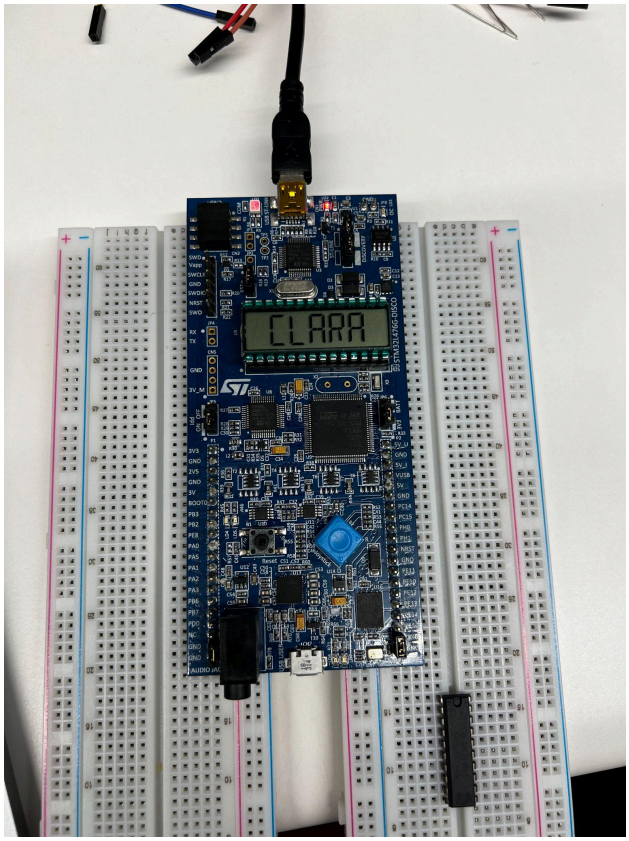
Lab 8 - Matrix I/O

Part 1:

```
30  ;*****************************************
31  ;   Put your code here to display Santa Clara
32  ;*****************************************
33              mov     r0, #83
34              bl      let2font
35              mov     r1, #0x01
36              bl      lcd_draw
37
38              mov     r0, #65
39              bl      let2font
40              mov     r1, #0x02
41              bl      lcd_draw
42
43              mov     r0, #78
44              bl      let2font
45              mov     r1, #0x03
46              bl      lcd_draw
47
48              mov     r0, #84
49              bl      let2font
50              mov     r1, #0x04
51              bl      lcd_draw
52
53              mov     r0, #65
54              bl      let2font
55              mov     r1, #0x05
56              bl      lcd_draw
57
58              mov     r0, #0x000f000f
59  delay       sub     r0, #1
60              cmp     r0, #0
61              bne     delay
62
63              bl      lcd_clear
64
65              mov     r0, #67
66              bl      let2font
67              mov     r1, #0x02
68              bl      lcd_draw
69
70              mov     r0, #76
71              bl      let2font
72              mov     r1, #0x03
73              bl      lcd_draw
74
75              mov     r0, #65
76              bl      let2font
77              mov     r1, #0x04
78              bl      lcd_draw
79
80              mov     r0, #82
81              bl      let2font
82              mov     r1, #0x05
83              bl      lcd_draw
84
85              mov     r0, #65
86              bl      let2font
87              mov     r1, #0x06
88              bl      lcd_draw
89
90              mov     r0, #0x000f000f
91  delay2      sub     r0, #1
92              cmp     r0, #0
93              bne     delay2
```

```
783 let2font    PROC
784             EXPORT let2font
785             ;   r0 is an ascii letter a-z (0x41-0x5A or 0x61-7A)
786             ;   return font in r0
787             ;   convert lower to upper - return 0 for out of range
788
789 ;*****************************************
790 ;   Put your code here for this subroutine
791 ;*****************************************
792             push    {lr}
793             cmp     r0, #97
794             blt     continue
795             cmp     r0, #122
796             bgt     continue
797             sub     r0, #32
798 continue    sub     r0, #0x41
799             ldr     r1, =letfont
800             lsl     r0, #1
801             add     r1, r0
802             ldrh    r0, [r1]
803             pop     {lr}
804             bx      lr
```

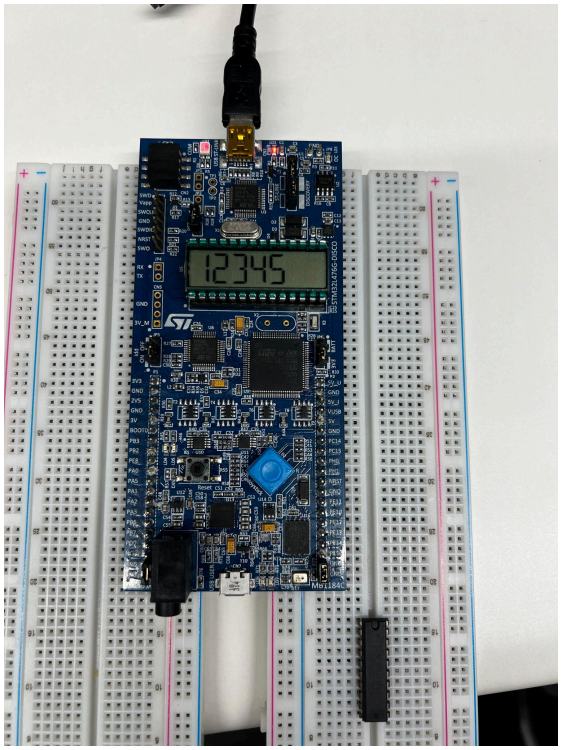

SANTA

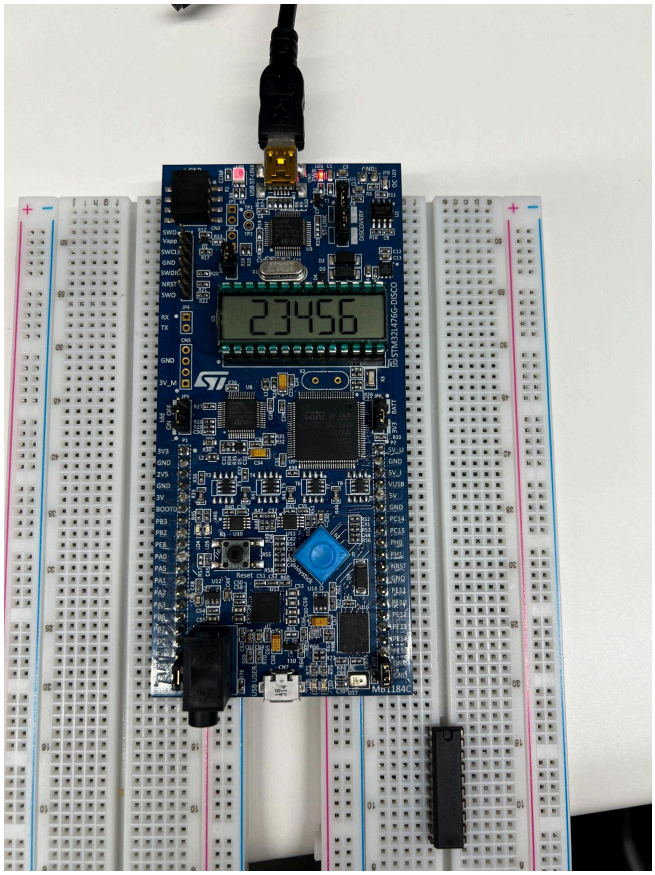

CLARA

Part 2:

```
30  ;*****************************************
31  ;    Put your code here to display Santa Clara
32  ;*****************************************
33              mov     r0, #49
34              bl      num2font
35              mov     r1, #0x01
36              bl      lcd_draw
37
38              mov     r0, #50
39              bl      num2font
40              mov     r1, #0x02
41              bl      lcd_draw
42
43              mov     r0, #51
44              bl      num2font
45              mov     r1, #0x03
46              bl      lcd_draw
47
48              mov     r0, #52
49              bl      num2font
50              mov     r1, #0x04
51              bl      lcd_draw
52
53              mov     r0, #53
54              bl      num2font
55              mov     r1, #0x05
56              bl      lcd_draw
57
58              mov     r0, #0x000f000f
59  delay       sub     r0, #1
60              cmp     r0, #0
61              bne     delay
62
63              bl      lcd_clear
64
65              mov     r0, #50
66              bl      num2font
67              mov     r1, #0x02
68              bl      lcd_draw
69
70              mov     r0, #51
71              bl      num2font
72              mov     r1, #0x03
73              bl      lcd_draw
74
75              mov     r0, #52
76              bl      num2font
77              mov     r1, #0x04
78              bl      lcd_draw
79
80              mov     r0, #53
81              bl      num2font
82              mov     r1, #0x05
83              bl      lcd_draw
84
85              mov     r0, #54
86              bl      num2font
87              mov     r1, #0x06
88              bl      lcd_draw
89
90              mov     r0, #0x000f000f
91  delay2      sub     r0, #1
92              cmp     r0, #0
93              bne     delay2
```

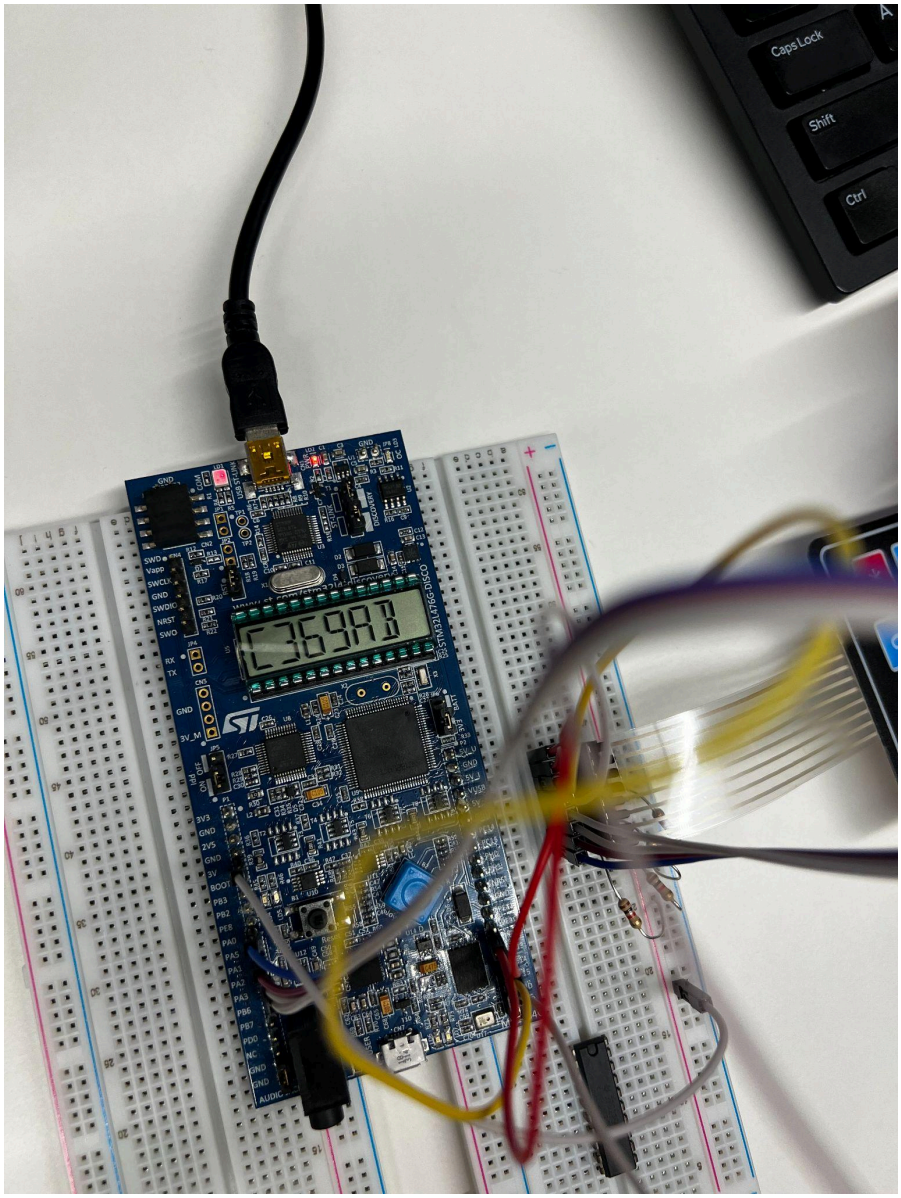
```
763 num2font    PROC
764             EXPORT num2font
765             ;   r0 is an ascii number 0-9 (0x30-0x39)
766             ;   return font in r0
767             ;   Only use last hex digit 0-9; zero out A-F
768
769 ;*************************************
770 ;   Put your code here for this subroutine
771 ;*************************************
772             push {lr}
773             sub     r0, #0x30
774             lsl     r0, #1
775             ldr     r1, =numfont
776             add     r1, r0
777             ldrh    r0, [r1]
778             pop {lr}
779             bx lr
780
781             ENDP
```


12345


23456

Part 3:

```
58  kpad_scan       PROC
59          EXPORT  kpad_scan
60          push    {lr}
61
62          cmp     r0, #0x07
63          moveq   r1, #3
64          beq     continue
65
66          cmp     r0, #0x0b
67          moveq   r1, #2
68          beq     continue
69
70          cmp     r0, #0x0d
71          moveq   r1, #1
72          beq     continue
73
74          cmp     r0, #0x0e
75          moveq   r1, #0
76          beq     continue
77
78
79
80  continue    push {r1}
81          mov r0, #0x07
82          bl      kpad_row_read
83          cmp     r0, #0xf
84          movne   r0, #0
85          bne done
86
87          mov     r0, #0x0b
88          bl      kpad_row_read
89          cmp     r0, #0xf
90          movne   r0, #1
91          bne done
92
93          mov     r0, #0x0d
94          bl      kpad_row_read
95          cmp     r0, #0xf
96          movne   r0, #2
97          bne done
98
99          mov     r0, #0x0e
100         bl      kpad_row_read
101         cmp     r0, #0xf
102         movne   r0, #3
103         bne done
104
105
106
107 done    pop     {r1, pc}
108         ENDP
```


Caps Lock
Shift
Ctrl