

main.s

```
1@ Ryan Bentz
2@ 372 Project 2 - Part 1
3@ This program initializes a New Haven LCD and displays my name.
4@ It uses the I2C protocol in a polling configuration to
5@ interact with the display.
6
7.data
8.align 2
9@ define stack sizes
10@-----
11LED_STATUS:      .word 0x00
12STACK1:          .rept 1024
13                  .word 0x00
14                  .endr
15STACK2:          .rept 1024
16                  .word 0x00
17                  .endr
18
19@ define word messages to send
20@-----
21MESSAGE:         .ascii "@RYAN BENTZ" @ ---- "@" symbol for the control/data
byte
22.align 4
23INIT_0: .byte 0x00, 0x38
24INIT_1: .byte 0x00, 0x39
25@INIT_2: .byte 0x00, 0x14, 0x78, 0x5E, 0x6D, 0x0C, 0x01, 0x06
26INIT_2: .byte 0x00, 0x14, 0x78, 0x5E
27INIT_3: .byte 0x00, 0x6D, 0x0C, 0x01, 0x06
28
29.text
30.global _start
31.global INT_DIRECTOR
32._start:
33
34@ Define the Register Addressed, Offsets, and write values to control the LEDs
35.equ     DELAY_1S, 0x0022DCD5
36.equ     DELAY_1MS,0x37C7B
37
38@ Initialize the stack frames
39@-----
40LDR R13, =STACK1          @ initialize stack one for supervisor mode
41ADD R13, R13, #0x1000     @ point stack pointer to top of stack
42CPS #0x12                 @ change to IRQ mode
43LDR R13, =STACK2          @ initialize stack for IRQ mode
44ADD R13, R13, #0x1000     @ point stack pointer to top of stack
45CPS #0x13                 @ change back to supervisor mode
46
47
48@ Initialize the peripheral clocks
49@-----
50@ initialize the clock to I2C1
51LDR R0, =0x44E00048       @ CMPEI.I2C1_CLKCTRL
52LDR R1, [R0]
53MOV R2, #0x02
54ORR R1, R1, R2
55STR R1, [R0]
56
57@ Delay 1s and wait for peripheral clocks
58@-----
```

main.s

```
59 LDR R0, =DELAY_1S
60 BL DELAY
61
62 @ Initialize the I2C
63 @-----
64 @ initialize the GPIO pins for the I2C functions
65 @ set the SCL pin
66 LDR R0, =0x44E1095C @ CONTROL.conf_spi0_cs0
67 LDR R1, [R0]
68 AND R1, R1, #0xFFFFFFFF @ mask bits [2:0]
69 ORR R1, R1, #0x02 @ enable mode 2
70 STR R1, [R0]
71
72 @ set the SDA pin
73 LDR R0, =0x44E10958 @ CONTROL.conf_spi0_d1
74 LDR R1, [R0]
75 AND R1, R1, #0xFFFFFFFF @ mask bits [2:0]
76 ORR R1, R1, #0x02 @ enable mode 2
77 STR R1, [R0]
78
79 @ set I2C ICLK prescaler
80 LDR R0, =0x4802A0B0 @ I2C_PSC: Clock prescaler register
81 LDR R1, [R0]
82 MOV R2, #0x4 @ 48 MHz clock / 4 = 12 MHz clock
83 ORR R1, R1, R2
84 STR R1, [R0]
85
86 @ set SCLL value
87 LDR R0, =0x4802A0B4 @ I2C_SCLL: Low time register
88 LDR R1, [R0]
89 MOV R2, #0x35
90 ORR R1, R1, R2
91 STR R1, [R0]
92
93 @ set SCLH value
94 LDR R0, =0x4802A0B8 @ I2C_SCLH: High time register
95 LDR R1, [R0]
96 MOV R2, #0x37
97 ORR R1, R1, R2
98 STR R1, [R0]
99
100 @ take the I2C module out of reset mode
101 LDR R0, =0x4802A0A4 @ I2C_CON: Control register
102 LDR R1, [R0]
103 MOV R2, #0x8000
104 ORR R1, R1, R2
105 STR R1, [R0]
106
107 @ configure I2C mode register without setting STT and STP
108 LDR R0, =0x4802A0A4 @ I2C_CON: Control register
109 LDR R1, [R0]
110 LDR R2, =0xFFFF0000
111 AND R1, R1, R2
112 ORR R1, R1, #0x8600 @ 7-bit address: 0x8600, 10-bit address: 0x8700
113 STR R1, [R0]
114
115 @ enable interrupt masks
116
117 @ configure the slave address
```

main.s

```
118 LDR R0, =0x4802A0AC          @ I2C_SA: Slave address register
119 LDR R1, [R0]
120 AND R1, R1, #0x0000
121 ORR R1, R1, #0x3C
122 STR R1, [R0]
123
124 @ clear the TX FIFO
125 LDR R0, =0x4802A094
126 LDR R1, [R0]
127 ORR R1, R1, #0x40
128 STR R1, [R0]
129
130 @ Delay 1s and wait for module
131 @-----
132 LDR R0, =DELAY_1S
133 BL DELAY
134
135 @ Initialize the display
136 @-----
137 INIT_DISPLAY:
138
139 LDR R0, =INIT_0
140 MOV R1, #0x02
141 BL I2C_TRANSMIT_PROC
142
143 LDR R0, =DELAY_1MS
144 BL DELAY
145
146 LDR R0, =INIT_1
147 MOV R1, #0x02
148 BL I2C_TRANSMIT_PROC
149
150 LDR R0, =DELAY_1MS
151 BL DELAY
152
153 LDR R0, =INIT_2
154 MOV R1, #0x04
155 BL I2C_TRANSMIT_PROC
156
157 LDR R0, =DELAY_1MS
158 BL DELAY
159
160 LDR R0, =INIT_3
161 MOV R1, #0x05
162 BL I2C_TRANSMIT_PROC
163
164 LDR R0, =DELAY_1MS
165 BL DELAY
166
167
168 @ transmit command to display "RYAN BENTZ"
169 LDR R0, =MESSAGE
170 MOV R1, #0xB
171 BL I2C_TRANSMIT_PROC
172
173 @ wait 1 ms
174 LDR R0, =DELAY_1MS
175 BL DELAY
176
```

main.s

```
177
178
179 @ MAIN LOOP
180 @-----
181 MAIN_LOOP:
182     NOP                @ do nothing and wait for interrupt
183     B MAIN_LOOP
184     B END
185
186
187 @-----
188 I2C_TRANSMIT_PROC:
189 @ R0 = Address of bytes to send
190 @ R1 = Number of bytes to write
191 @ R2 = Data Register Address
192 @ R3 = Value of data to write
193
194 STMFD R13!, {R0-R5, R14}    @ save the register states and link register
    location
195
196 @ check that the bus is ready
197 TX_WAIT:
198     LDR R4, =0x4802A024      @ I2C_IRQSTATUS_RAW
199     LDR R5, [R4]
200     MOV R6, #0x1000          @ Check Bit 12: Bus Busy Status
201     AND R5, R5, R6
202     CMP R5, #0x1000          @ If BB = 1, bus is occupied
203     BEQ TX_WAIT
204
205 @ initialize the counter register
206 LDR R4, =0x4802A098
207 STR R1, [R4]
208
209 @ write the transmit data to the FIFO
210 DATA_QUEUE:
211     LDR     R2, =0x4802A09C    @ I2C_DATA Register
212     LDRB    R3, [R0], #1      @ get the value of the data to send and
    post increment index
213     STRB    R3, [R2]          @ write the value to the FIFO
214     SUBS    R1, #1            @ decrement the number of bytes to send
215     BNE DATA_QUEUE
216
217 @ set module to master mode on every transfer
218 LDR R4, =0x4802A0A4          @ I2C_CON: Control register
219 LDR R5, [R4]
220 ORR R5, R5, #0x400          @ Master mode bit 10
221 STR R5, [R4]
222
223 @ configure the start/stop bits
224 LDR R4, =0x4802A0A4          @ I2C_CON: Control Register
225 LDR R5, [R4]
226 MOV R6, #0x03               @ Set the STT and STP bits to initiate transfer
227 ORR R5, R5, R6
228 @ begin transmitting data by setting the start/stop bits
229 STR R5, [R4]
230
231 LDMFD R13!, {R0-R5, PC}      @ return execution from the procedure
232
233
```

main.s

```
234 @-----
235 @ Delay Loop Subroutine
236 @ Handles the delay loop timing
237 DELAY:
238 STMFD R13!, {R4, R14}    @ save the register states and link register location
239 D_LOOP:
240     NOP
241     SUBS R0, #1
242     BNE D_LOOP
243 LDMFD R13!, {R4, PC}
244
245 END:
246 .END
```