Mon Sep 24 14:42:38 2018

/\* Lab.asm - Show a room

\* Version 2.5.4.

\* Written By : Socheath Sok

\* ID # : 014470701

\* Date : September 25 2018

\* Lab Section : T/Th, 3 pm

\*/

.INCLUDE <m328pdef.inc>

.DSEG

000100 dir: .BYTE 1

.CSEG

.ORG 0X0000

RST\_VECT:

000000 c131 rjmp reset

.ORG 0x0100

.INCLUDE "spi\_shield.inc"

reset:

000132 ef0f ldi r16, low(RAMEND) // RAMEND address 0x08ff

000133 bf0d out SPL, r16 // stack Pointer Low SPL at i/o address 0x3d

000134 e008 ldi r16, high(RAMEND)

000135 bf0e out SPH, r16 // stack Pointer High SPH at i/o address 0x3e

000136 940e 0100 call InitShield

loop:

; SPI Software Wires

000138 940e 0116 call ReadSwitches //read switches into r6

00013a 2c86 mov spiLEDS, switch //wire switches to the 8 discrete LEDs

00013b 2d06 mov r16, switch //move switcch r7 to temporary register r16

00013c 7003 cbr r16, 0xFC //mask-out significant 6 bits

00013d 9300 0100 sts dir, r16 //save formatted value to SRAm variable dir

00013f fa67 bst switch, 7 //wire switch 7 to segment g (south wall)

000140 f876 bld spi7seg, seg\_g

000141 fa66 bst switch, 6 //wire switch 6 to segment f (west wall)

000142 f875 bld spi7seg, seg\_f

000143 fa65 bst switch, 5 //wire switch 5 to segment b (east wall)

000144 f871 bld spi7seg, seg\_b

000145 fa64 bst switch, 4 //wire switch 4 to segment a (north wall)

000146 f870 bld spi7seg, seg\_a

000147 940e 0121 call WriteDisplay //write r7 to the 7 segment display

000149 cfee rjmp loop

;Lab 2 Direction Finder

00014a 9100 0100 lds r16, dir //load direction into temporary register r16

00014c 2477 clr spi7SEG //start with all 7-segment off

; FACING NORTH

00014d 2f20 mov r18,r16 //load r16 into r18

00014e fb01 bst r16,1 //store bit 1 in T flag

00014f f910 bld r17,0 //loads it to r17

000150 2321 and r18,r17 //B = A\*B

000151 fb20 bst r18,0 //stores bit 0 in T flag

000152 f870 bld spi7seg,seg\_a //loads answer to seg\_a

; FACING SOUTH

000153 2f20 mov r18,r16

000154 fb01 bst r16,1 //stores bit 1 in T-flag

000155 f910 bld r17,0 //loads it to r17 bit 0

000156 9510 com r17 //inverts A

000157 9520 com r18 //inverts B

000158 2321 and r18,r17 //B = /A\*/B

000159 fb20 bst r18,0 //stores answer into t-bit

00015a f876 bld spi7seg,seg\_g //loads answer to seg\_g

; FACING WEST

00015b 2f20 mov r18,r16

00015c fb01 bst r16,1 //stores bit 1 in T-flag

00015d f910 bld r17,0 //loads it to r17

00015e 9520 com r18 //inverts B

00015f 2321 and r18,r17 //B = A\*/B

000160 fb20 bst r18,0 //stores answer into t-bit

000161 f875 bld spi7seg,seg\_f //loads answer to seg\_f

; FACING EAST

000162 2f20 mov r18,r16

000163 fb01 bst r16,1 //stores bit 1 in T-flag

000164 f910 bld r17,0 //loads it to r17

000165 9510 com r17 //inverts A

000166 2321 and r18,r17 //B = /A\*B

000167 fb20 bst r18,0 //stores answer into t-bit

000168 f871 bld spi7seg,seg\_b //loads answer to seg\_b

000169 940e 0121 call WriteDisplay

00016b cfcc rjmp loop

Assembly complete, 0 errors, 0 warnings