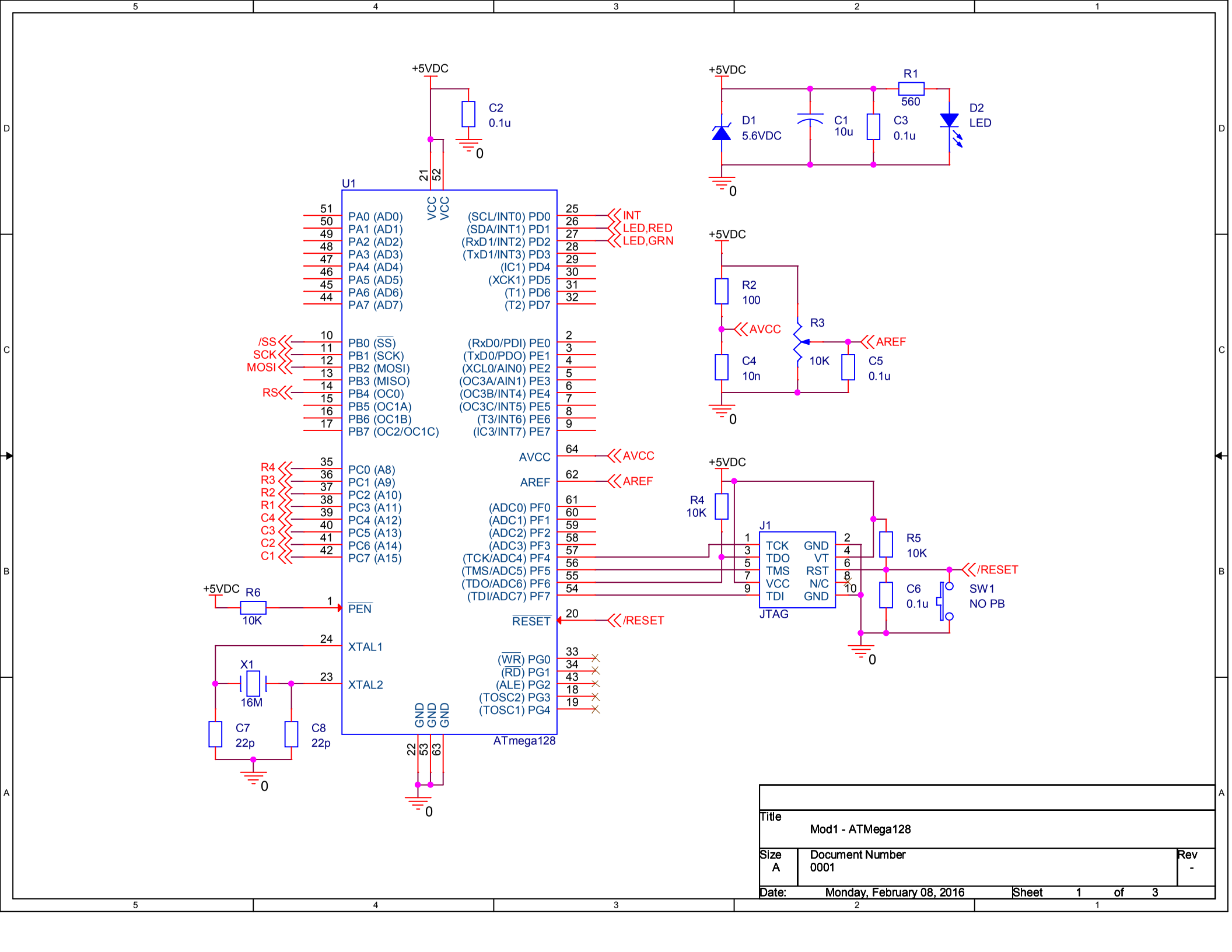
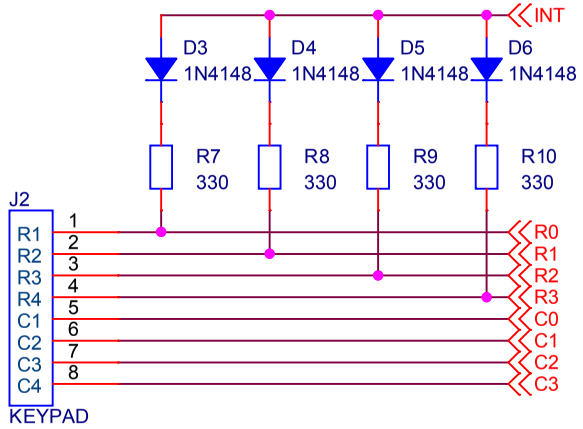
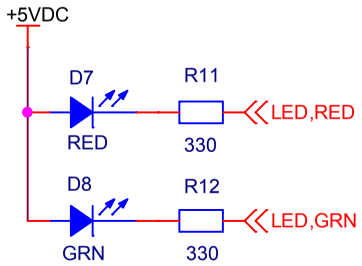
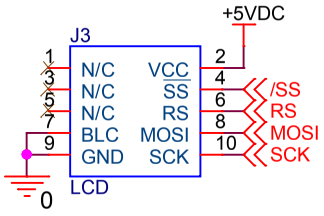
Experiment Name:	Mod 1 Keypad and LCD Display
Experiment Number:	#1
Lab Participant 1:	Ben Jarnagin, 107692807
Lab Section:	#1
Bench:	#3


+5VDC
C2
0.1u
0
U1
21
52
VCC
VCC
51
50
49
48
47
46
45
44
PA0 (AD0)
PA1 (AD1)
PA2 (AD2)
PA3 (AD3)
PA4 (AD4)
PA5 (AD5)
PA6 (AD6)
PA7 (AD7)
25
26
27
28
29
30
31
32
(SCL/INT0) PD0
(SDA/INT1) PD1
(RxD1/INT2) PD2
(TxD1/INT3) PD3
(IC1) PD4
(XCK1) PD5
(T1) PD6
(T2) PD7
INT
LED,RED
LED,GRN
/SS
SCK
MOSI
RS
10
11
12
13
14
15
16
17
PB0 (SS)
PB1 (SCK)
PB2 (MOSI)
PB3 (MISO)
PB4 (OC0)
PB5 (OC1A)
PB6 (OC1B)
PB7 (OC2/OC1C)
2
3
4
5
6
7
8
9
(RxD0/PDI) PE0
(TxD0/PDO) PE1
(XCL0/AIN0) PE2
(OC3A/AIN1) PE3
(OC3B/INT4) PE4
(OC3C/INT5) PE5
(T3/INT6) PE6
(IC3/INT7) PE7
64
AVCC
AVCC
62
AREF
AREF
R4
R3
R2
R1
C4
C3
C2
C1
35
36
37
38
39
40
41
42
PC0 (A8)
PC1 (A9)
PC2 (A10)
PC3 (A11)
PC4 (A12)
PC5 (A13)
PC6 (A14)
PC7 (A15)
61
60
59
58
57
56
55
54
(ADC0) PF0
(ADC1) PF1
(ADC2) PF2
(ADC3) PF3
(TCK/ADC4) PF4
(TMS/ADC5) PF5
(TDO/ADC6) PF6
(TDI/ADC7) PF7
+5VDC
R6
10K
1
PEN
RESET
20
/RESET
24
XTAL1
X1
16M
23
XTAL2
C7
22p
C8
22p
0
33
34
43
18
19
(WR) PG0
(RD) PG1
(ALE) PG2
(TOSC2) PG3
(TOSC1) PG4
GND
GND
GND
22
53
63
ATmega128
0
+5VDC
D1
5.6VDC
C1
10u
R1
560
C3
0.1u
D2
LED
0
+5VDC
R2
100
AVCC
R3
10K
AREF
C4
10n
C5
0.1u
0
+5VDC
R4
10K
J1
1
3
5
7
9
2
4
6
8
10
TCK
TDO
TMS
VCC
TDI
GND
VT
RST
N/C
GND
JTAG
R5
10K
C6
0.1u
/RESET
SW1
NO PB
0
Title
Mod1 - ATMega128
Size
A
Document Number
0001
Rev
-
Date:
Monday, February 08, 2016
Sheet
1
of
3





Title: Mod1 - Keypad Interface		
Size: A	Document Number: 0001	Rev: -
Date: Monday, February 08, 2016	Sheet 2 of 3	



Title: Mod1 - DOG LCD Interface		
Size: A	Document Number: 0001	Rev: -
Date: Monday, February 08, 2016	Sheet 3 of 3	

```
//***********************************************************************
//
// Name:         Module 1 - Keypad Project
// Filename:     keypad.c
// Author(s):    Timothy Caro
//               Peter Carcaterra
//
// Modified by: Ken Short (1/13/07)
//               Ben Jarnagin (PORTC substitution, delay resync to 16MHz, tbl)
//
// Description: Configures ports and interrupt. When keypad interrupt occurs,
//               key matirx is scanned and is encoded using a table lookup. The
//               green LED is flashed the number of times equal to the
//               keycode. If the keycode is 0, the red LED is flashed
//               once. The keypad is connected to PORTC while the LEDS are
//               connected to PORTD. See diagram in laboratory description.
//
//***********************************************************************

//Definitions

#define debug   //this can be uncommented to remove delays for simulation

#include <iom128.h>         //Atmega128 definitions
#include <intrinsics.h>     //Intrinsic functions.
#include <avr_macros.h>     //Useful macros.
/*
 * Port pin numbers for columns and rows of the keypad and for LEDs
 */
//PORT Pin Definitions.
#define COL1  7   //pin definitions for PortB
#define COL2  6
#define COL3  5
#define COL4  4
#define ROW1  3
#define ROW2  2
#define ROW3  1
#define ROW4  0

#define INT0  0   //pin definitions for PortD
#define LED1  1
#define LED2  2
/*
 * Function declarations.
 */
void flash_leds(char num);
void check_release(void);

/*
 * Lookup table declaration
 */
const char tbl[16] = {1, 2, 3, 15, 4, 5, 6, 14, 7, 8, 9, 13, 10, 0, 11, 12};
```

```
//*********************************************************************************
//Code

/*
 * Interrupt service routine
 */
#pragma vector=INT0_vect          //Declare Vector location.
__interrupt void ISR_INT0(void)   //Declare Interrupt Function
{
  char keycode;                   //Holds keycode. (UNSIGNED by default)

//Note: TESTBIT returns 0 if bit is not set and a non-zero number otherwise.

  if(!TESTBIT(PINC,ROW1))         //Find Row of pressed key.
    keycode = 0;
  else if(!TESTBIT(PINC,ROW2))
    keycode = 4;
  else if(!TESTBIT(PINC,ROW3))
    keycode = 8;
  else if(!TESTBIT(PINC,ROW4))
    keycode = 12;

  DDRC = 0x0F;                    //Reconfigure PORTC for Columns.
  PORTC = 0xF0;

#ifndef debug
  __delay_cycles(4096);            //Let PORTC settle.
#endif

  if(!TESTBIT(PINC,COL1))         //Find Column.
    keycode += 0;
  else if(!TESTBIT(PINC,COL2))
    keycode += 1;
  else if(!TESTBIT(PINC,COL3))
    keycode += 2;
  else if(!TESTBIT(PINC,COL4))
    keycode += 3;

  DDRC = 0xF0;                    //Reconfigure PORTC for Rows.
  PORTC = 0x0F;

  flash_leds(tbl[keycode]);       // Fix keycode and Flash LEDS.
  check_release();                //Wait for keypad release.
}

/*
 * Main entry point for program.
 */
int main(void)
{
  DDRD = 0xFE;                    //Initialize PORTD.
  PORTD = 0x07;
```

```
  DDRC = 0xF0;                    //Initialize PORTC.
  PORTC = 0x0F;

  MCUCR = 0x30;                   //Config interrupts and sleep mode.
  EIMSK = 0x01;                   //Enable interrupt masks.
  __enable_interrupt();           //Enable global interrupts.

  return 0;                       //Return
}

/*
 *  Flashes LEDS set number of times.
 */
void flash_leds(char num)
{
  if(num == 0)                    //If num is 0 flash LED1 (red LED) once.
  {
    CLEARBIT(PORTD,LED1);
#ifndef debug
    __delay_cycles(4000000);       //Delay (.25secs) / (1 / 16MHz) cycles.
#endif
    SETBIT(PORTD, LED1);
  }
  else                            //IF num is > 0 flash LED2 (green LED)
                                  //that many times.
  {
    for(char i = 0; i < num; i++)
    {
      CLEARBIT(PORTD,LED2);
#ifndef debug
      __delay_cycles(4000000);     //Delay (.25secs) / (1 / 16MHz) cycles.
#endif
      SETBIT(PORTD, LED2);
#ifndef debug
      __delay_cycles(4000000);     //Delay (.25secs) / (1 / 16MHz) cycles.
#endif
    }
  }
  return;
}

/*
 *  Check keypad is released and not bouncing.
 */
void check_release(void)
{
#ifndef debug
  while(!TESTBIT(PIND,INT0));     //Check that keypad key is released.

  __delay_cycles(800000);          //Delay (.05secs) / (1 / 16MHz) cycles.

  while(!TESTBIT(PIND,INT0));     //Check that key has stopped bouncing.
#endif
```

```
  return;
}
```