//change to match target clock

//Note: default AVR CLKSEL is 1MHz internal RC

#define F\_CPU 8000000UL // 8 MHz

#include <avr/io.h>

// For AVR registers - change device in Project - Configuration Options

#include <util/delay.h>

// for delay routine

/\* Read Keypad on PortD and output binary result on PortB LEDs

Keypad has 7 pins - 3 columns and 4 rows. When a key is pressed, Column and

row goes low. To read it, set columns as inputs (bit 0..2) and wait for 0

on these bits. When column goes low, add to value (+0, +1 or +2) and then

set port to read rows. Read row and add 1, 4, 7 or A to value depending on

row Output value on LED - after inverting it (0 = LED on) output is

keypad.hex in default folder \*/

void LED\_init(void)

{

DDRB = 0xFF;

PORTB = 0xFF;

// set LEDs to output, 1 = LED off

}

void LED\_display(unsigned char value){

PORTB = ~value; // 0 = LED on

}

void Col\_init(void)

{

DDRD = 0xF8; // bit 0.2 are columns (0 is input)

PORTD = 0x07; //pullups on these bits

//very short delay

asm volatile ("nop");

asm volatile ("nop");

}

void Row\_init(void)

{

DDRD = 0x87; // bit 3..6 used for rows

PORTD = 0x78; //pullups on these bits

\_delay\_ms(1);

}

unsigned char Read\_key(void)

{

unsigned char value;

Col\_init(); //set up columns to read

value =0; // init value

// read columns first - 0 if key in that column pressed

if (!(PIND & 0x01))

{value = 2; }// Col2 = bit0 is low

else if (!(PIND & 0x02))

{value = 1;} // Col1 = bit1 is low

else if (!(PIND & 0x04))

{value = 0; } // Col0 = bit2 is low

Row\_init(); //set up rows to read

if (!(PIND & 0x08))

{value += 0x0A; } //row3 = bit 3 is low

else if (!(PIND & 0x10))

{ value += 0x07; } //row2 = bit 4 is low

else if (!(PIND & 0x20))

{ value += 0x04; } // row1 = bit 5 is low

else if (!(PIND & 0x40))

{ value += 0x01; } //row0 = bit 6 is low

\_delay\_ms(50); // switch debounce

return value; // value is sum of row and column

}

int main( void )

{ unsigned char led;

LED\_init(); // initialize LEDs

led = 0x00; // init variable

Col\_init(); // set keypad column read

while(1) // Eternal loop

{

if (!(PIND == 0x07)) {

// if a column is pressed bit 0,1 or 2 will go low active 0

led= Read\_key(); // read keypad

}

LED\_display(led); // display result

}

}