```
/*
 * larm.c
 *
 * Created: 2015-03-27 09:38:35
 *  Author: digpi06
 */


#include <avr/io.h>
#include <util/delay.h>
#include <avr/interrupt.h>
#define F_CPU 8000000UL //set the frequency of the CPU unsigned long, 8 MHz

#define LEDGREEN PB0
#define LEDRED PB1
#define LEDYELLOW1 PB2
#define LEDYELLOW2 PB3
#define LEDYELLOW3 PB4
#define LEDYELLOW4 PB5
#define SENSOR PA0
#define BUZZER PB6
#define OUTPUT PD3

unsigned int status = 0;
unsigned int expectPin = 0;
unsigned int changingCode = 0;
unsigned int alarmTriggered = 0;
unsigned char pinCode[4] = {'1', '2', '3', '4'};
unsigned char pinCodeAttempt[5];
unsigned int codeOk;
unsigned char symbol;
unsigned int keyNbr = 0;

char readKeyPad(char code) {
	if (code == 0b00000000){
		return '3';
		}else if(code==0b00010000){
		return '7';
		}else if(code==0b00001000){
		return 'B';
		}else if(code==0b00011000){
		return 'F';
		}else if(code==0b00000100){
		return '2';
		}else if(code==0b00010100){
		return '6';
		}else if(code==0b00001100){
		return 'A';
		}else if(code==0b00011100){
		return 'E';
		}else if(code==0b00000010){
		return '1';
		}else if(code==0b00010010){
		return '5';
		}else if(code==0b00001010){
		return '9';
		}else if(code==0b00011010){
		return 'D';
		}else if(code==0b00000110){
		return '0';
		}else if(code==0b00010110){
		return '4';
		}else if(code==0b00001110){
		return '8';
```

```
		}else if (code==0b00011110){
		return 'C';
	}
}


int main(void) {

setUp();
enableKeypadInterrupt();

while(1){

if (status==1 && checkSensor()) {
	setLed(LEDRED,1);
	buzz();
} else if (status == 0) {
	setLed(LEDRED,0);

}


}

}
void buzz(){
	while(1){
	_delay_ms(1);
	setPin('B',BUZZER,1);
	_delay_ms(1);
	setPin('B',BUZZER,0);
	if (status==0) {
		break;
	}
	}
	}
void enableKeypadInterrupt() {
	GICR = (1 << INT0); //General Interrupt Control Register, says that we are enabling
INT0
	MCUCR = (1 << ISC01) | (1 << ISC00); // MCU Control and Status Register, register the
condition on which the interrupt will be fired = rising edge here
	//SREG = 0x80; //set till 10000010, i.e. that bit 7 är 1 och bit 1 är 1, bit 7 =
enable global interrupt
	GIFR = 0;
	sei(); //enable global interrupts, sätter bit 7 till 1 i SREG
}


 ISR(INT0_vect){
	 cli();

	 char code = PINA & 0b00011110;
	 char symbol = readKeyPad(code);

	 if(symbol=='A') { //activate alarm
		 if(status==0){
			 expectPin = 1;

		 }

	} else if(symbol=='D') { //deactivate alarm
				if(status==1){
					expectPin = 1;
				}
```

```
		}
		else if(symbol=='C'){ //change
		if(status==0){
			expectPin = 1;
			changingCode = 1;
		}

		 }
		 else if(symbol=='E'){
			resetCode();
		 }

		if (expectPin==1 && changingCode == 0){
			insertCode(symbol);
				if (keyNbr==5){
					isPinCorrect();
					if(codeOk==1 && status == 0){
						activate();
						} else if(codeOk == 1 && status==1){
							deactivate();
						}
				 resetCode();
					}
					}
	else if (expectPin==1 && changingCode==1) {
				insertCode(symbol);
					if(keyNbr==5) {
						isPinCorrect();
							if(codeOk==1){
								changingCode=2;
							}
						resetCode();
					}
	}
	if (changingCode==2){

						insertCode(symbol);
						if (keyNbr==5){
						changePin();
						setLed(LEDRED,1);
						_delay_ms(100);
						setLed(LEDRED,0);
						setLed(LEDGREEN,1); //larmet är ju avaktiverat
fortfarande
						resetCode();
						changingCode=0;
						}
					}


	sei();
 }


void resetCode() {
	keyNbr = 0;
	expectPin = 0;
	codeOk=0;
	setLed(LEDYELLOW1,0);
	setLed(LEDYELLOW2,0);
	setLed(LEDYELLOW3,0);
	setLed(LEDYELLOW4,0);
	pinCodeAttempt[0]=0;
	pinCodeAttempt[1]=0;
```

```
	pinCodeAttempt[2]=0;
	pinCodeAttempt[3]=0;
	pinCodeAttempt[4]=0;
}

void insertCode(char letter){
	pinCodeAttempt[keyNbr] = letter;
	ledCode(keyNbr);
	keyNbr++;
}


void ledCode(int i){
	if(i==1) {
		setLed(LEDYELLOW1,1);
	} else if(i==2){
		setLed(LEDYELLOW2,1);
	} else if(i==3){
		setLed(LEDYELLOW3,1);
	} else if(i==4){
		setLed(LEDYELLOW4,1);
		_delay_ms(1000); //så att alla 4 lyser tillsammans en sekund
	}
}

void changePin() {

	for(unsigned int i = 0; i<=3; i++) {
		pinCode[i] = pinCodeAttempt[i+1];
	}
}


void isPinCorrect() {
	if(pinCode[0]==pinCodeAttempt[1] && pinCode[1]==pinCodeAttempt[2] &&
pinCode[2]==pinCodeAttempt[3] && pinCode[3]==pinCodeAttempt[4]) {
		codeOk = 1;
	}
}

void activate() {
	if(status==0) {
		status = 1;
		setLed(LEDRED,1);
		setLed(LEDGREEN,0);
	}

}

void deactivate() {
	if(status==1) {
		status = 0;
		setLed(LEDGREEN,1);
		setLed(LEDRED,0);
	}
}


int checkSensor(){
	char set= PINA & 0b00000001;
	if(set == 0b00000001){
		return 1;
	} else {
		return 0;
```

```
	}
}

void blinkLed(char led) { //används inte i dagsläget men känns bra att ha om man vill lägga
till en blinkande lampa i senare skede
	setLed(led,1);
	_delay_ms(500);
	setLed(led,0);
	_delay_ms(500);
	setLed(led,1);
	_delay_ms(500);
	setLed(led,0);
	_delay_ms(500);
	setLed(led,1);
	_delay_ms(500);
	setLed(led,0);
	_delay_ms(500);
	setLed(led,1);
	_delay_ms(500);
	setLed(led,0);
}

void setLed(char led,char state){
	setPin('B',led,state);
}


void setPin(char port, char pin, char state){
	char set=1<<pin;
	if(port=='B'){
		set&=PORTB;
	if(set&&!state){
		PORTB^=set;
	}
	if(set==0&&state){
		set=1<<pin;
		PORTB^=set;
	}
} else if(port=='A'){
set&=PORTA;
if(set&&!state){
	PORTA^=set;
}
if(set==0&&state){
	set=1<<pin;
	PORTA^=set;
}
} else if(port=='C'){
set&=PORTC;
if(set&&!state){
	PORTC^=set;
}
if(set==0&&state){
	set=1<<pin;
	PORTC^=set;
}
}else if(port=='D'){
set&=PORTD;
if(set&&!state){
	PORTD^=set;
}
if(set==0&&state){
	set=1<<pin;
	PORTD^=set;
```

```
}
}
}

void setUp() {
	DDRA = 0b00000000;
	DDRB = 0b01111111;
	DDRC = 0b00000000;
	DDRD = 0b00001000;
	setPin('D', OUTPUT,0);
	setLed(LEDGREEN,1);
	codeOk=0;
}
```