```c
#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>

int larm_status; //0 = off, 1 = on, 2 = utlöst
int pincode[4] = {4, 5, 6, 7};
int pressed_keys[4] = {0, 0, 0, 0};

int main(void) {

	disp_start();
	alarmOff();
	USART_Init();
	sei();
	INT_Init();

	while(1) {
		while (larm_status == 2) {
			_delay_ms(200);
			redLED_ON();
			_delay_ms(200);
			redLED_OFF();
		}
    }
}

void change_pin(char port, char pin, char val) { //port = A-D, pin = 0-7, val = 1/0
	char temp = 1 << pin; //Skapar en temp-variabel med 1 på samma plats som pin
	if (port == 'A') {
		temp &= PORTA; //Multiplicerar temp med PORTA
		if (temp && !val) {
			PORTA ^= temp; //Ändrar från 1 -> 0
		}
		if (temp == 0 && val) {
			temp = 1 << pin; //Återger temp på ursprungliga formen
			PORTA ^= temp; //Ändrar från 0 -> 1
		}
	} else if (port == 'B') {
		temp &= PORTB; //Multiplicerar temp med PORTB
		if (temp && !val) {
			PORTB ^= temp; //Ändrar från 1 -> 0
		}
		if (temp == 0 && val) {
			temp = 1 << pin; //Återger temp på ursprungliga formen
			PORTB ^= temp; //Ändrar från 0 -> 1
		}
	} else if (port == 'C') {
		temp &= PORTC; //Multiplicerar temp med PORTC
		if (temp && !val) {
			PORTC ^= temp; //Ändrar från 1 -> 0
		}
		if (temp == 0 && val) {
			temp = 1 << pin; //Återger temp på ursprungliga formen
			PORTC ^= temp; //Ändrar från 0 -> 1
		}
	} else if (port == 'D') {
		temp &= PORTD; //Multiplicerar temp med PORTD
		if (temp && !val) {
			PORTD ^= temp; //Ändrar från 1 -> 0
		}
		if (temp == 0 && val) {
```

```
                        temp = 1 << pin; //Återger temp på ursprungliga formen
                        PORTD ^= temp; //Ändrar från 0 -> 1
                }
        }
}

//Interrupt

INT_Init() {
        MCUCR = 0x0F;
        GICR = 0xE0;
        MCUCSR = 0x40;
}

ISR(INT0_vect) {
        pin_entry(keyset_pressed());
}

ISR(INT1_vect) { //Övre
        if (larm_status == 1) {
                alarmActivated();
        }
}

ISR(INT2_vect) { //Undre
        if (larm_status == 1) {
                alarmActivated();
        }
}

//Display

void disp_start() {
        DDRA = 0xFF; //Öppnar PORTA
        DDRB = 0x03; //Öppnar PORTB0-1, stänger resten
        DDRC = 0x00; //Stänger PORTC
        DDRD = 0xE2; //PD1, PD5-7 öppna, resten stängda
        PORTD = 0x20; //E hög
        PORTA = 0x38; //Function set
        toggle_E();
        PORTA = 0x0F; //Display ON
        toggle_E();
}

void write_cmd(char ch) {
        _delay_ms(5);
        change_pin('D', 7, 0); //RS låg
        PORTD = 0x20; //E hög
        PORTA = ch;
        toggle_E();
}

void write_data(char ch) {
        _delay_ms(5);
        change_pin('D', 7, 1); //RS hög
        change_pin('D', 5, 1); //E hög
        PORTA = ch;
        toggle_E();
}

void write_data_nbr(int nbr) {
        _delay_ms(5);
```

```
	change_pin('D', 7, 1); //RS hög
	change_pin('D', 5, 1); //E hög
	PORTA = nbr;
	toggle_E();
}

void toggle_E() {
	change_pin('D', 5, 0); //E låg
	change_pin('D', 5, 1); //E hög
}

void print_larmat() {
	write_cmd(0x01);
	write_data('L');
	write_data('A');
	write_data('R');
	write_data('M');
	write_data('A');
	write_data('T');
}

void print_avlarmat() {
	write_cmd(0x01);
	write_data('A');
	write_data('V');
	write_data('L');
	write_data('A');
	write_data('R');
	write_data('M');
	write_data('A');
	write_data('T');
}

void print_activated() {
	write_cmd(0x01);
	write_data('L');
	write_data('A');
	write_data('R');
	write_data('M');
	write_data(' ');
	write_data('A');
	write_data('K');
	write_data('T');
	write_data('I');
	write_data('V');
	write_data('E');
	write_data('R');
	write_data('A');
	write_data('T');
}

//Knappsats

int keyset_pressed() {
	char val = PINC;
	val &= 0b11000011;
	if (val == 0xC0) {
		return 0;
	}
	if (val == 0x40) {
		return 1;
	}
```

```
	if (val == 0x80) {
		return 2;
	}
	if (val == 0x00) {
		return 3;
	}
	if (val == 0xC2) {
		return 4;
	}
	if (val == 0x42) {
		return 5;
	}
	if (val == 0x82) {
		return 6;
	}
	if (val == 0x02) {
		return 7;
	}
	if (val == 0xC1) {
		return 8;
	}
	if (val == 0x41) {
		return 9;
	}
	return 10;
}

void pin_entry(int n) {
	if (n == 10) {
		return;
	}
	write_data_nbr(n);
	for (int i = 0; i < 3; i++) {
		pressed_keys[i] = pressed_keys[i+1];
	}
	pressed_keys[3] = n;
	if (pressed_keys[0] == pincode[0] && pressed_keys[1] == pincode[1] &&
pressed_keys[2] == pincode[2] && pressed_keys[3] == pincode[3]) {
		correct_pin();
	}
}

void correct_pin() {
	if (larm_status == 0) {
		print_larmat();
		alarmOn();
	} else {
		print_avlarmat();
		alarmOff();
	}
}

//LED

void greenLED_ON() {
	change_pin('B', 1, 1);
}

void redLED_ON() {
	change_pin('B', 0, 1);
}
```

```
void greenLED_OFF() {
	change_pin('B', 1, 0);
}

void redLED_OFF() {
	change_pin('B', 0, 0);
}

void alarmOn() {
	greenLED_OFF();
	redLED_ON();
	larm_status = 1;
	print_larmat();
}

void alarmOff() {
	redLED_OFF();
	greenLED_ON();
	larm_status = 0;
	print_avlarmat();
}

void alarmActivated() {
	larm_status = 2;
	USART_Transmit();
	print_activated();
}

//Serial

void USART_Init() {
	UBRRH = 0b00000000;
	UBRRL = 0b00110011; //9600 bits/sec (51)
	UCSRB = 0b00001000; //Transmitter ON
}

void USART_Transmit() {
	 while ( !(UCSRA & (1<<UDRE)) ) //Transmit buffer (UDR) not ready
		;
	UDR = 'A';
}
```