Kod

```
#define F_CPU 8000000UL
#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdbool.h>
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

#define white PA5
#define green PA6
#define red PA7

void setLed(char led, char state);
void setPin(char port, char pin, char state);
void printChar(char ch);
void setUpScreen();
void screenCommand(char cmd);

int tot_overflow=0;
int loop_overflow=0;
char hundredsOut;
char tensOut;
char singlesOut;
char buttonPressed;
int temperature;
int digit=0;

int mode = 9;
char maxTempTens;
char maxTempOnes;
char minTempTens;
char minTempOnes;


char checkButton() { //Kollar vilken knapp som trycks ner
            int button = checkRow();
            if (button == 11) {
                        return 0;
            }
            if (button == 12) {
                        return 1;
            }
            if (button == 13) {
                        return 2;
            }
            if (button == 14) {
                        return 3;
            }
            if (button == 21) {
                        return 4;
            }
            if (button == 22) {
                        return 5;
            }
            if (button == 23) {
                        return 6;
            }
            if (button == 24) {
            return 7;
            }
```

```
		if (button == 31) {
				return 8;
		}
		if (button == 32) {
				return 9;
		}
		if (button == 33) {
				return 'A';
		}
		if (button == 34) {
				return 'B';
		}
		if (button == 41) {
				return 'C';
		}
		if (button == 42) {
				return 'D';
		}
		if (button == 43) {
				return 'E';
		}
		if (button == 44) {
		return 'F';
		}
		return 20;
}
int checkRow() { //Kollar vilken rad
		DDRD = 0x0F; //Bestämmer att kolonner output och rader input
		PORTD = 0xF0; //Skicka ettor i alla kolonner
		int value = PIND & 0xF0;
		if(value == 0x70) {
		return checkCol(10);
		}
		if(value == 0xB0) {
		return checkCol(20);
		}
		if(value == 0xD0) {
				return checkCol(30);
		}
		if(value == 0xE0) {
				return checkCol(40);

		}
		return 0;

}

int checkCol(int x) { //Kollar vilken kolumn
		int val;
		setPin('D', PD0, 0);
		setPin('D', PD1, 0);
		setPin('D', PD2, 0);
		setPin('D', PD3, 1);
		val = PIND & 0xF0;
		if (val == 0xF0) {
				return (x + 1);
		}
		setPin('D', PD0, 0);
		setPin('D', PD1, 0);
		setPin('D', PD2, 1);
		setPin('D', PD3, 0);
		val = PIND & 0xF0;
```

```
                if (val == 0xF0) {
                                return (x + 2);
                }
                setPin('D', PD0, 0);
                setPin('D', PD1, 1);
                setPin('D', PD2, 0);
                setPin('D', PD3, 0);
                val = PIND & 0xF0;
                if (val == 0xF0) {
                                return (x + 3);
                }
                setPin('D', PD0, 1);
                setPin('D', PD1, 0);
                setPin('D', PD2, 0);
                setPin('D', PD3, 0);
                val = PIND & 0xF0;
                if (val == 0xF0) {
                                return (x + 4);
                }

}

char digitToChar(char ch){ //Säkerställer att ch är en siffra
                if(ch == 0) {
                                return '0';
                }
                if(ch == 1) {
                                return '1';
                }
                if(ch == 2) {
                                return '2';
                }
                if(ch == 3) {
                                return '3';
                }
                if(ch == 4) {
                                return '4';
                }
                if(ch == 5) {
                                return '5';
                }
                if(ch == 6) {
                                return '6';
                }
                if(ch == 7) {
                                return '7';
                }
                if(ch == 8) {
                                return '8';
                }
                if(ch == 9) {
                                return '9';
                }
                else {
                                return '  ';
                }

}
```

```
//Display
void setUpScreen() {
            screenCommand(0x38); //functional set
            screenCommand(0x0F); //display ON/OFF
            screenCommand(0x06); //Entry mode set

}
void printChar(char ch) {
            setPin('C', PC1, 0); // R/W = 0
            setPin('C', PC0, 1); // RS = 1
            setPin('C', PC6, 1); // E = 1
            PORTB = ch;
            setPin('C', PC6, 0); // E = 0
            setPin('C', PC6, 1); // E = 1
            return;
}

void screenCommand(char cmd) {
            //Power on
            setPin('C', PC0, 0); // R/W =0
            setPin('C', PC1, 0); //RS = 0
            setPin('C', PC6, 1); // E = 1
            PORTB = cmd;
            setPin('C', PC6, 0); // E = 0 , här läses allt av

}
void cursorHome() {
            screenCommand(0x02); //Display cursor home
}
void showStandard() {
            cursorHome();
            printChar('T');
            printChar('E');
            printChar('M');
            printChar('P');
            printChar('E');
            printChar('R');
            printChar('A');
            printChar('T');
            printChar('U');
            printChar('R');
            printChar(':');
            printActTemp();
            screenCommand(0xC0);
            printChar('M');
            printChar('I');
            printChar('N');
            printChar(':');
            printChar(digitToChar(minTempTens));
            printChar(digitToChar(minTempOnes));
            printChar(' ');
            printChar('M');
            printChar('A');
            printChar('X');
            printChar(':');
            printChar(digitToChar(maxTempTens));
            printChar(digitToChar(maxTempOnes));
}
```

```
void specifyIntervalMax() {
                        cursorHome();
                        printChar('A');
                        printChar('N');
                        printChar('G');
                        printChar('E');
                        printChar(' ');
                        printChar('M');
                        printChar('A');
                        printChar('X');
                        printChar(':');
}
void specifyIntervalMin() {
                        cursorHome();
                        printChar('A');
                        printChar('N');
                        printChar('G');
                        printChar('E');
                        printChar(' ');
                        printChar('M');
                        printChar('I');
                        printChar('N');
                        printChar(':');

}
void clearDisplay() {
            screenCommand(0x01);
}
void printLimit(int limit) {
            tensOut = limit/10;
            singlesOut = limit-tensOut*10;
            printChar(digitToChar(tensOut));
            printChar(digitToChar(singlesOut));
}

//Temperaturen
void startTempRead() {
            ADCSRA = 0xDC; //Starta termometern
            ADMUX =  0x24;
            SFIOR = 0x00;
}
int tempRead() {
            return ADCH; //Volt
}

void printActTemp() {
            temperature = tempRead()+tempRead()+tempRead()+tempRead();
            temperature = temperature >> 2;
            temperature = temperature - 126;
            tensOut = (temperature)/10;
            singlesOut = temperature-tensOut*10;
            printChar(digitToChar(tensOut));
            printChar(digitToChar(singlesOut));

}

//Lampor
void setLed(char led, char state) {
            setPin('A', led, state);

}
```

```
//Ändra pins
void setPin(char port, char pin, char state) {
            char set = 1 << pin;
            if(port == 'A') {
                        set &= PORTA;
                        if(set && !state) {
                                    PORTA ^= set; //set PIN
                        }
                        if(set == 0 && state){ // Clear PIN
                                    set = 1 << pin;
                                    PORTA ^= set;
                        }

            }
            else if(port == 'B'){
                        set &= PORTB;
                        if(set && !state){ //Set PIN
                        PORTB ^= set;
                        }
                        if(set == 0 && state){ //Clear PIN
                        set = 1 << pin;
                        PORTB ^= set;
                        }
            }
            else if(port == 'C'){
                        set &= PORTC;
                        if(set && !state){ //Set PIN
                        PORTC ^= set;
                        }
                        if(set == 0 && state){ //Clear PIN
                        set = 1 << pin;
                        PORTC ^= set;
                        }
            }
            else if(port == 'D'){
                        set &= PORTD;
                        if(set && !state){ //Set PIN
                        PORTD ^= set;
                        }
                        if(set == 0 && state){ //Clear PIN
                        set = 1 << pin;
                        PORTD ^= set;
                        }
            }

}
void timer0_init() {
            TCCR0 |= (1 << CS10);
            TCNT0 = 0;
            TIMSK |= (1 << TOIE0);
}
ISR(TIMER0_OVF_vect) {     //Interrupt för timer (och knapp-poll)
            buttonPressed = checkButton();
            tot_overflow++;
            if (tot_overflow >= 300) {
                        if (mode == 0) {
                                    if(loop_overflow >= 5) {
                                                clearDisplay();
                                                cursorHome();
                                                showStandard();
                                                loop_overflow=0;
                                    }
```

```
}
if (mode != 0) {
            if (buttonPressed == 0) {
                        printChar('0');
            } else if (buttonPressed == 1) {
                        printChar('1');
            } else if (buttonPressed == 2) {
                        printChar('2');
            } else if (buttonPressed == 3) {
                        printChar('3');
            } else if (buttonPressed == 4) {
                        printChar('4');
            } else if (buttonPressed == 5) {
                        printChar('5');
            } else if (buttonPressed == 6) {
                        printChar('6');
            } else if (buttonPressed == 7) {
                        printChar('7');
            } else if (buttonPressed == 8) {
                        printChar('8');
            } else if (buttonPressed == 9) {
                        printChar('9');
            }
}
if (mode == 1 && buttonPressed >= 0 && buttonPressed <= 9) {
            if (digit == 0) {
                        minTempTens = buttonPressed;
                        digit++;
            } else if (digit == 1) {
                        minTempOnes = buttonPressed;
                        digit = 0;
            }
}
if (mode == 2 && buttonPressed >= 0 && buttonPressed <= 9) {
            if (digit == 0) {
                        maxTempTens = buttonPressed;
                        digit++;
            } else if (digit == 1) {
                        maxTempOnes = buttonPressed;
                        digit = 0;
            }
}
if (buttonPressed == 'A') {

} else if (buttonPressed == 'B') {
            mode = 0;
            clearDisplay();
            cursorHome();
            showStandard();


} else if (buttonPressed == 'C') {
            mode = 1;
            clearDisplay();
            cursorHome();
            specifyIntervalMin();
} else if (buttonPressed == 'D') {
            mode = 2;
            clearDisplay();
            cursorHome();
            specifyIntervalMax();
}
```

```
                        tot_overflow = 0;

                        loop_overflow++;

            }

}

//Mainprogram

int main(void){
            setUpProcessor();
            setUpScreen();
            setUpTermo();
            cursorHome();
            timer0_init();

            sei();

            while(1) {
                        startTempRead();

                        if (mode == 0) {
                                    if (tensOut < minTempTens || (tensOut ==
                                    minTempTens && singlesOut < minTempOnes)) {
                                                setLed(green, 0);
                                                setLed(red, 0);
                                                setLed(white, 1);
                                    } else if (tensOut > maxTempTens || (tensOut ==
                                    maxTempTens && singlesOut > maxTempOnes)) {
                                                setLed(green, 0);
                                                setLed(white, 0);
                                                setLed(red, 1);
                                    } else {
                                                setLed(red, 0);
                                                setLed(white, 0);
                                                setLed(green, 1);
                                    }
                                    //Main, visa max och low och aktuell temperatur.
Lampor ska lysa korrekt.

                        }
                        if (mode == 1) {
                                    setLed(green, 0);
                                    setLed(red, 0);
                                    setLed(white, 0);
                                    //ange maxgränsen , lampor släckta
                        }
                        if (mode == 2) {
                                    setLed(green, 0);
                                    setLed(red, 0);
                                    setLed(white, 0);
                                    //ange mingränsen , lampor släckta
                        }
            }


}

void setUpProcessor() {
                        DDRA = 0xEF;
                        DDRC = 0xFF;
```

```
                        DDRB = 0xFF;

}
void setUpTermo() {
            maxTempTens = 9;
            maxTempOnes = 9;
            minTempTens = 0;
            minTempOnes = 9;
}
```