

Informatik Labor

Smarthome - Heizungssteuerung

Inhaltsverzeichnis

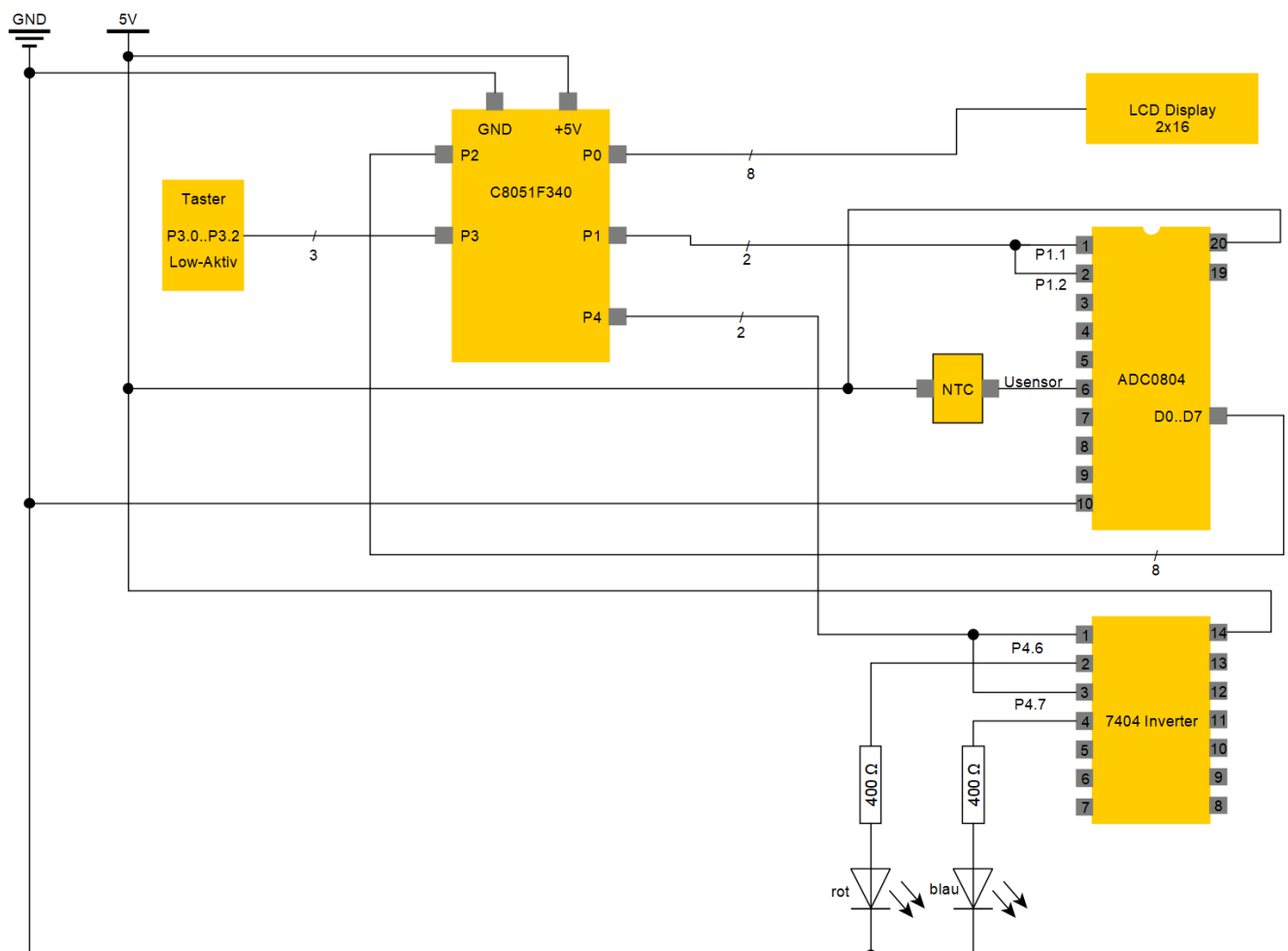
Projekt Heizungssystem	3
Beschreibung	3
Schaltplan.....	3
Bedienung	4
Aufbau	5
Programmablaufplan	6
Code	7
Variablen	7
Funktionen	7
Init()	7
Ausgabe()	7
ReadTemperatur()	8
Main()	8
Projektplanung	9

Projekt Heizungssystem

Beschreibung

Im Rahmen des Gruppenprojektes Smart-Home muss jede Kleingruppe ein unabhängiges Teilsystem entwickeln. In diesem Teilsystem geht es um eine automatisierte Heizungssteuerung, die manuell aktivieren/deaktivieren kann und eine Richttemperatur mit Hilfe von zwei Tastern einstellen kann. Der Status sowie die aktuelle Temperatur und die Richttemperatur wird auf einem LCD-Display angezeigt. Das System funktioniert mit einem NTC-Widerstand (Negative Temperature Coefficient Thermistor), der mit Hilfe des ADC0804 an den Mikrocontroller C8051F340 von Silicon Laboratories angeschlossen ist.

Schaltplan

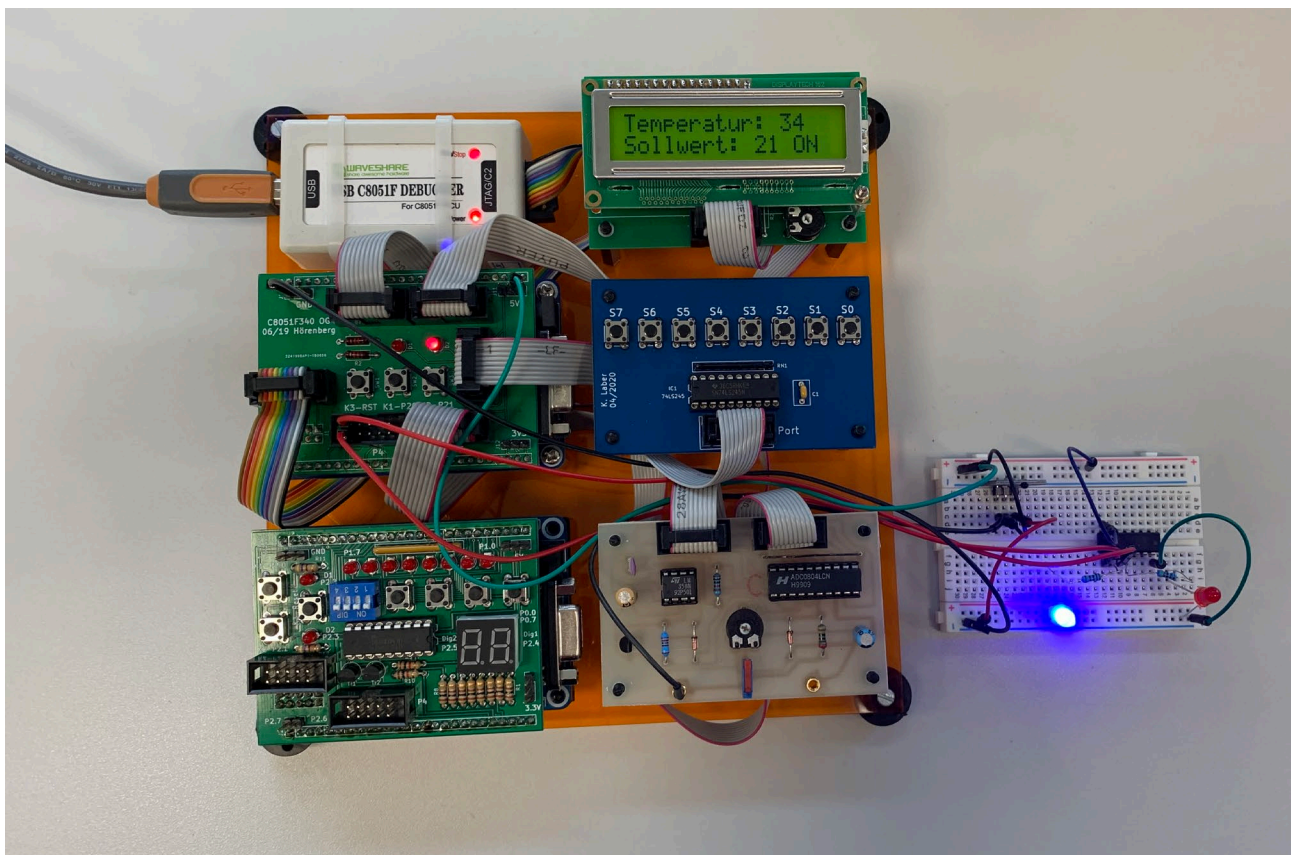
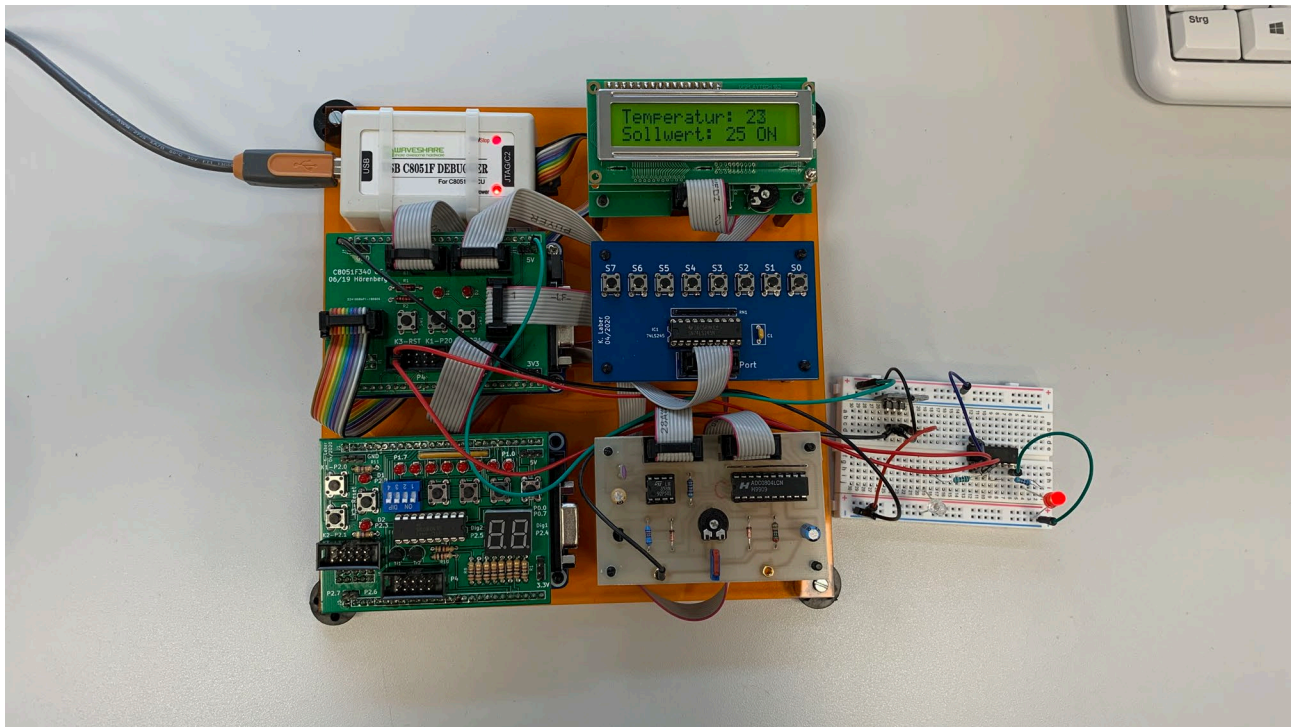


Bedienung

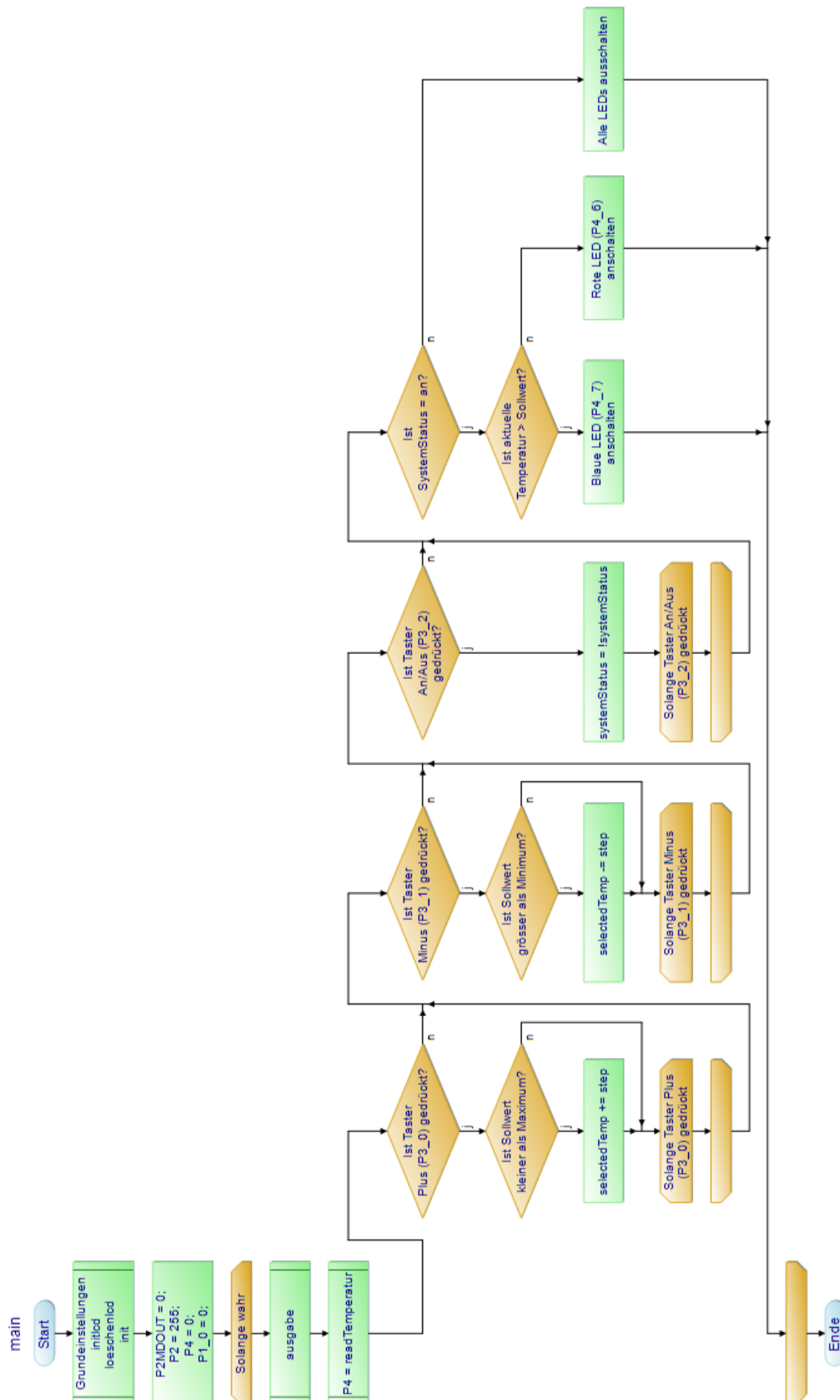
Zu Beginn wird der Sollwert mit Hilfe der zwei Taster (Plus und Minus) durch das Drücken des jeweiligen Tasters eingestellt. Der Sollwert lässt sich in 1°C Schritten verändern. Die aktuelle Temperatur sowie der derzeitige Sollwert werden auf dem LCD-Display angezeigt. Auch der Systemstatus ist dort vermerkt. Dieser lässt sich durch das Drücken des An/Aus-Tasters verändern. Die untere Grenze für den Sollwert beträgt 15°C und die obere Grenze beträgt 25°C. Nach dem Einschalten des Systems ist das System standartmässig aktiviert und auf einen Sollwert von 21°C eingestellt. Ist die aktuelle Temperatur grösser als der Sollwert, dann leuchtet die Blaue LED auf (Simulation für Kühlung). Ist die aktuelle Temperatur kleiner als der Sollwert, so leuchtet die Rote LED auf (Simulation für Heizung). Ist das System deaktiviert (Systemstatus = Off) so leuchtet keine der beiden LEDs. Es werden trotzdem die aktuelle Temperatur sowie der eingestellte Sollwert angezeigt. Auch während das System deaktiviert ist lässt sich der Sollwert mit Hilfe der beiden Taster (Plus und Minus) verändern.



Aufbau



Programmablaufplan



Code

Variablen

```

39 //-----
40 // Pin/Port Zuweisungen
41 //-----
42 unsigned char buf [20] ;           // Zwischenspeicher f. LCD-Text
43
44 unsigned char systemStatus = 1;    // Systemstatus: 1 -> an, 0 -> aus
45
46 int step = 1;                      // Schrittgrösse
47
48 int currentTemp = 21;               // Derzeitige Temperatur
49 int selectedTemp = 21;              // Ausgewählte Temperatur
50
51 int minTemp = 15;                  // Untere Temperaturgrenze
52 int maxTemp = 25;                  // Obere Temperaturgrenze
53
54 unsigned int i = 0;                // ReadTemperatur CounterVariable
55

```

Hier sind alle Variablen deklariert sowie deren Verwendungszweck im Programm angegeben.

Funktionen

```

// externe Funktionen in lcd.c; Diese Datei zum Projekt hinzufügen
extern void initlcd (void);           // Initialisierung LCD-Display an P0
extern void loeschenlcd (void);
extern void textlcd (unsigned char *text,unsigned char zeile);

void init(void)
void ausgabe(void)
unsigned int readTemperatur()

```

Das sind alle Methoden, die von der main()-Methode oder von sich selbst verwendet werden. Die drei Methoden initlcd(), loeschenlcd() sowie textlcd() sind für die Ansteuerung des LCD-Displays verantwortlich.

Init()

```

66 void init(void)
67 {
68     selectedTemp = 21;
69     systemStatus = 1;
70 }

```

Zu Beginn wird in der Init der standardmässige Sollwert eingestellt. Des weiteren wird festgelegt, dass das System aktiviert ist.

Ausgabe()

```

72 void ausgabe(void)
73 {
74     //LCD Display
75     sprintf (buf,"Temperatur: %d",currentTemp);
76     // Text mit Variable formatieren und in buf kopieren
77     // % Ausgabeformat folgt, d Dezimalzahl mit Vorzeichen, a auszugebende Variable
78     textlcd (buf,1);
79
80     if (systemStatus == 0)
81     {
82         sprintf (buf,"Sollwert: %d OFF",selectedTemp);
83         textlcd (buf,2);
84     }
85     else
86     {
87         sprintf (buf,"Sollwert: %d ON ",selectedTemp);
88         textlcd (buf,2);
89     }
90 }

```

Die Methode ausgabe() kümmert sich um die Ausgabe der Werte auf dem LCD-Display. In der ersten Zeile wird die aktuelle Temperatur ausgegeben. In der 2. Zeile wird der Sollwert und der Systemstatus (ON/OFF) ausgegeben.

ReadTemperatur()

```

93 unsigned int readTemperatur()
94 {
95     int i = 0;
96     P1_1 = 1;
97     P1_2 = 0;
98     P1_2 = 1;
99     for (i = 0; i < 5000; i++)
100     P1_1 = 0;
101
102
103     //Calculate Temperature From ADC Signal
104     //0b11100001: 4,72V (25°C) - 0b11101011: 4,96V (15°C)
105     currentTemp = 250 - P2;
106     //return currentTemp;
107
108     return P2;
109 }

```

Die Methode readTemperatur() liest die Temperatur vom ADC0804 aus. Zuerst stellen wir mit dem bit P1.1 ein, dass keine Daten ausgelesen werden. Danach wird die Wandlung durch einen kurzen Impuls an P1.2 gestartet und wieder beendet. Die Schleife dient als Zeitverzögerung, sodass das wandeln des analogen Signals problemlos erfolgen kann. Durch das Löschen des Bits P1.1 wird der ermittelte Wert zum Lesen freigegeben. Zum Schluss wird noch die aktuelle Temperatur berechnet, indem wir mit Hilfe der gemessenen Spannungen des NTCs und dem ADC-Signal die Temperatur berechnen.

Main()

```

115 void main(void)
116 {
117     Grundeinstellungen();           // Siehe Datei Einstellungen.h
118     initlcd ();                     // Initialisierung LCD-Display an P3
119     loeschenlcd ();                 // Anzeige l'schen
120     init();
121
122     P2MDOUT = 0;
123     P2 = 255;
124     P4 = 0;
125     P1_0 = 0;
126
127     while (1)
128     {
129         ausgabe();
130         P4 = readTemperatur();
131         P4 = P2;
132         if (!P3_0)                  //Plus
133         {
134             if (selectedTemp <= (maxTemp - step))
135                 selectedTemp += step;
136             while (!P3_0);           //Warten bis Taster losgelassen
137         }
138         if (!P3_1)                  //Minus
139         {
140             if (selectedTemp >= (minTemp + step))
141                 selectedTemp -= step;
142             while (!P3_1);           //Warten bis Taster losgelassen
143         }
144         // An / Aus
145         if (!P3_2)
146         {
147             systemStatus = !systemStatus;
148             while (!P3_2);           //Warten bis Taster losgelassen
149         }
150
151         if (systemStatus == 1)
152         {
153             P4 = 0b01111111;
154             if (currentTemp > selectedTemp)
155                 P4 = 0b01111111;
156             else if (currentTemp < selectedTemp)
157                 P4 = 0b10111111;
158         }
159         else
160             P4 = 255;
161     }
162 }

```

Die main()-Methode spiegelt das ganze Programm wieder. Zunächst wird die oben erklärte Methode ausgabe() aufgerufen. Danach werden die 3 Taster (+ / - / On/Off) überprüft und der Sollwert erhöht/verringert oder das System Aktiviert/Deaktiviert. In den Zeilen 151 – 161 werden die blaue und die rote LED angesteuert. Da der Port P4 nicht bitadressierbar ist, muss hier immer ein ganzes Byte geschrieben werden. Beachte, dass die LEDs durch einen Inverter angesteuert werden, sodass die 0 in dem jeweiligen Byte den Status aktiviert widerspiegelt.

Projektplanung

Teilschritt	Zu erledigen bis	Verantwortlicher
LCD-Display Ansteuerung	20.09.2021	Tim Schlachter
Taster (+ / - / On/Off)	04.10.2021	Anton Tran
Temperatursensor anschliessen	04.10.2021	Anton Tran
Ansteuerung ADC0804	18.10.2021	Tim Schlachter
Temperatursensor Analogspannung messen	08.11.2021	Anton Tran
Anschliessen LEDs	08.11.2021	Anton Tran
Temperatur digital auswerten mit ADC	22.11.2021	Tim Schlachter
Simulation (Heizung/Kühlung)	22.11.2021	Tim Schlachter
Dokumentation	20.12.2021	Tim Schlachter/Anton Tran