



## Programme für eine CCU (Homematic)

### **zur lastabhängigen Steuerung von Umwälzpumpen**

und

### **zur Anpassung der Wärmequellentemperatur**

an den Öffnungszustand der Thermostat-Ventile, also den Wärmebedarf

Am 27.10.2024 erhielt ich, [Helmut@Diedrichs.de](mailto:Helmut@Diedrichs.de), per E-Mail von einem Heizkalender-Anwender diesen Aufsatz mit Skripten. Sie wurden per KI erstellt und sind noch nicht erprobt. Bitte mit Vorsicht testen und gerne mich informiere, dass ich den Vorbehalt herausnehmen kann.

Hier sind die Programme und Systemvariablen zur lastabhängigen Steuerung von Umwälzpumpen und zur Anpassung der Wärmequellentemperatur. Die Programme berücksichtigen die Anzahl und den Öffnungsgrad der Thermostate, um den Wärmebedarf zu ermitteln und die Pumpen bzw. den Kessel entsprechend zu steuern.

### **Ausführung und Test**

1. **Turnus:** Diese Skripte werden idealerweise in Intervallen (z.B. alle 5 Minuten) ausgeführt, um eine dynamische Anpassung an die Heizanforderungen zu gewährleisten.  
Das geschieht nach den Regeln von Homematic
2. **Tipp für Tests:** Aktivieren Sie Debug-Ausgaben mit WriteLine() für Variablen, um die laufenden Werte zu beobachten und zu überprüfen.

Falls Anpassungen gewünscht sind, z. B. an den Temperatur- oder Bedarfsschwellen, können diese Werte einfach in den Skripten geändert werden.

Grüße, Alexander Klatt

Hier die Programme:

### **Systemvariablen**

Definieren Sie zunächst die Systemvariablen in der Homematic-CCU, um die Thermostate zu überwachen und Steuersignale zu speichern. Die Variablen beginnen alle mit „HK5-“ zur besseren Übersicht:

- **HK5-Anzahl-Thermostate:** Gesamtanzahl der Thermostate (Ganzzahl).
- **HK5-Thermostate-Offen:** Anzahl der offenen Thermostate (Ganzzahl).
- **HK5-Thermostate-Gesamtöffnung:** Summe der Öffnungsgrade aller Thermostate in Prozent (Ganzzahl).
- **HK5-Wärmebedarf:** Wärmebedarfsindex für Pumpen-/Kesselsteuerung (Ganzzahl).
- **HK5-Pumpenstufe:** Aktuelle Stufe der Umwälzpumpe (Ganzzahl).

### **Vier Skripte (als Programme in die CCU eingeben):**

Skript 1 und Skript 2 aktualisieren die Systemvariablen und berechnen den Wärmebedarfsindex anhand der offenen Thermostate und ihrer Öffnungsgrade. Skript 3 steuert die Pumpenstufe der Umwälzpumpe (Variable: HK5-Pumpenstufe). Skript 4: Steuerung der Wärmequellen-Temperatur.



## Skript 1: zur Ermittlung der Thermostate (Anzahl, Offen-Status, Gesamtöffnung)

```
!! javascript
! // Skript zur Ermittlung der Thermostat-Zustände und Öffnungsgrad
integer AnzahlThermostate = 0;
integer ThermostateOffen = 0;
integer GesamtÖffnung = 0;

! // Schleife durch alle Thermostate
foreach (string thermostatID, dom.GetObjectsByInterface("BidCos-RF").EnumIDs()) {
    var thermostat = dom.GetObject(thermostatID);

    ! // Prüfen, ob es sich um ein Thermostat handelt (abhängig vom Modell)
    if (thermostat.HssType() == "HM-CC-RT-DN" || thermostat.HssType() == "HmIP-eTRV-B") {

        ! // Anzahl der Thermostate erhöhen
        AnzahlThermostate = AnzahlThermostate + 1;

        ! // Öffnungsgrad auslesen (0-100%)
        integer Öffnungsgrad = thermostat.DPByHssDP("VALVE_STATE").Value();

        ! // ThermostateOffen inkrementieren, wenn der Öffnungsgrad > 0 ist
        if (Öffnungsgrad > 0) {
            ThermostateOffen = ThermostateOffen + 1;
            GesamtÖffnung = GesamtÖffnung + Öffnungsgrad;
        }
    }
}

! // Systemvariablen setzen
dom.GetObject("HK5-Anzahl-Thermostate").State(AnzahlThermostate);
dom.GetObject("HK5-Thermostate-Offen").State(ThermostateOffen);
dom.GetObject("HK5-Thermostate-Gesamtöffnung").State(GesamtÖffnung);
```

## Skript 2: zur Berechnung des Wärmebedarfsindex (HK5-Wärmebedarf)

Hier wird der Wärmebedarfsindex berechnet, basierend auf der Gesamtöffnung und der Anzahl der offenen Thermostate.

```
!! javascript
! // Skript zur Berechnung des Wärmebedarfsindex
integer GesamtÖffnung = dom.GetObject("HK5-Thermostate-Gesamtöffnung").Value();
integer AnzahlThermostate = dom.GetObject("HK5-Anzahl-Thermostate").Value();
integer Wärmebedarf = 0;

! // Berechnen des Wärmebedarfsindex
if (AnzahlThermostate > 0) {
    Wärmebedarf = (GesamtÖffnung * 100) / (AnzahlThermostate * 100);
}

! // Wärmebedarf auf Systemvariable setzen
dom.GetObject("HK5-Wärmebedarf").State(Wärmebedarf);
```



## Skript 3: Steuerung der Umwälzpumpe (HK5-Pumpenstufe)

Dieses Skript stellt die Pumpe in Abhängigkeit vom Wärmebedarf auf die entsprechenden Stufen ein.

```
!! javascript
! // Skript zur Steuerung der Umwälzpumpe je nach Wärmebedarf
integer Wärmebedarf = dom.GetObject("HK5-Wärmebedarf").Value();
integer Pumpenstufe = 0;

! // Stufenregelung je nach Wärmebedarf
if (Wärmebedarf >= 70) {
    Pumpenstufe = 3; ! // Hohe Stufe
} else if (Wärmebedarf >= 40) {
    Pumpenstufe = 2; ! // Mittlere Stufe
} else if (Wärmebedarf > 0) {
    Pumpenstufe = 1; ! // Niedrige Stufe
} else {
    Pumpenstufe = 0; ! // Pumpe aus
}

! // Pumpenstufe auf Systemvariable setzen
dom.GetObject("HK5-Pumpenstufe").State(Pumpenstufe);
```

## Skript 4: Steuerung der Wärmequellen-Temperatur

Das Skript passt die Temperatur der Wärmequelle je nach Anzahl der geöffneten Thermostate an.

!! javascript

```
! // Skript zur Anpassung der Wärmequellen-Temperatur
integer ThermostateOffen = dom.GetObject("HK5-Thermostate-Offen").Value();
integer Temperatur = 30; ! // Startwert der Wärmequellentemperatur

! // Wärmequellentemperatur anpassen
if (ThermostateOffen >= 10) {
    Temperatur = 65; ! // Viele Thermostate offen, höhere Temperatur
} else if (ThermostateOffen >= 5) {
    Temperatur = 55; ! // Mittelwert für moderate Last
} else if (ThermostateOffen > 0) {
    Temperatur = 45; ! // Wenige Thermostate offen, geringere Temperatur
} else {
    Temperatur = 30; ! // Standby-Temperatur
}

! // Setzen der Wärmequellentemperatur (Beispielausgabe)
dom.GetObject("HK5-Wärmequellentemperatur").State(Temperatur);
```