

Heizungssteuerung der Kirche Rohrbach

Ausgangslage

Die Heizungssteuerung in der Kirche soll überarbeitet werden. Dazu soll die alte SPS-Steuerung mit einer modernen und schlanken KNX-Steuerung ersetzt werden.

Ziel muss sein, dass

- Heizen per Eventkalender möglich ist
- die Kirche nicht mehr als 1,5°C/h aufgeheizt wird
- die Maximaltemperatur der Bodenheizungen nicht überschritten wird
- das Anpassen der Soll-Werte (Event und Frostschutz) via SmartPhone möglich ist
- eine möglichst Energieeffiziente Lösung realisiert wird

Realisierung

Die Logik für diese Steuerung wurde mittels LUA-Scripts auf einer LogicMachine realisiert.

Es wurde zuerst eine Schaltuhr erstellt, in welcher Events definiert werden können. Die Schaltuhr steuert ein «Dummy-Kommunikationsobjekt» an, welches schlussendlich keine Funktion hat. Wichtig sind die Zeiten im Kalender.

Mittels LUA-Scripts werden die einzelnen Ereignisse aus dem Kalender ausgelesen und verglichen, ob das Event in der benötigten Vorlaufzeit (berechnet aus der Differenz zwischen Soll- und Ist-Temperatur der Kirche) liegen. Ist mind. ein Event innerhalb dieser Vorlaufzeit, wird ein entsprechendes «Flag» gesetzt.

Ein zweites Script prüft die Temperatur, welche vor einer Stunde in der Kirche geherrscht hat, mit der aktuellen Temperatur. Ist die Differenz kleiner als 1°C, wird überprüft, ob die Bodenfühler die maximale Temperatur noch nicht überschritten haben. Ist dies auch in Ordnung wird geprüft, ob ein Event, mit dem vorher gesetzten «Flag» Vorlauf für die Heizung benötigt. Wenn dies auch zutrifft, wird anhand der aktuellen Temperatur geprüft, ob die Heizung entsprechend entweder mit einer Stern- oder Dreiecksschaltung (Relais-Schaltung) angesteuert werden muss, damit entweder mit der vollen oder nur 1/3 der Leistung die Soll-Temperatur erreicht wird.

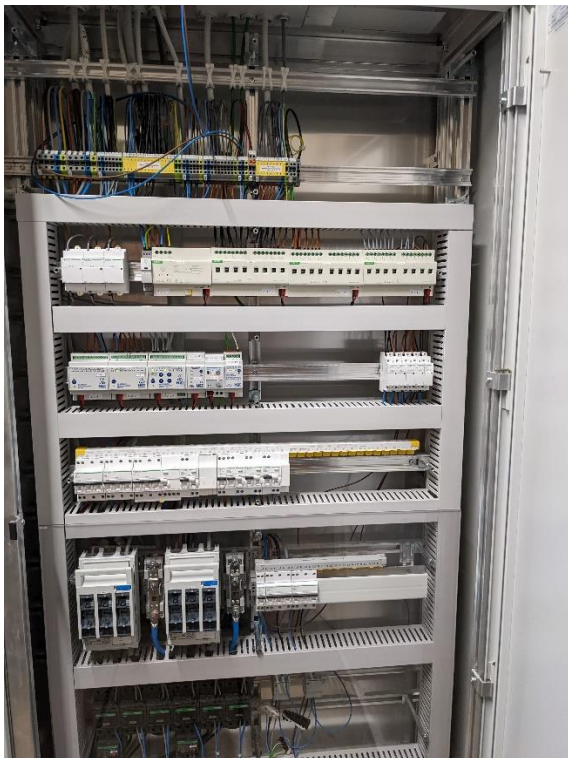
Beide Scripts laufen mehrmals in der Stunde in einem zyklischen Rhythmus, damit die Kirche mögliches Energieeffizient aufgeheizt werden kann.

Mit dem SmartPhone können die Soll-Werte für die Temperatur während eines Events, wie auch die Temperatur für den Frostschutz einfach verändert werden. Auch ist via SmartPhone die aktuelle Temperatur der Kirche sichtbar & Events können definiert werden.

Mittelbedarf

Gerät	Hersteller
Spannungsversorgung	Video-Star
Schaltaktoren	MDT technologies
LogicMachine	Embedded Systems SIA

Bilder



Status: aktiv
1. Januar - 31. Dezember [Bearbeiten]

Name	Ausführen am	Wert	[Ereignis hinzufügen]
Gottesdienst	09:30	aus	[Bearbeiten] [Löschen]
Konzert Jodler	18:15	aus	[Bearbeiten] [Löschen]
Jodler	19:30	aus	[Bearbeiten] [Löschen]

Heizungssteuerung

Ist-Temp

17°C

Soil-Wert Event

- 19°C +

Soil-Wert Standby

- 12°C +

```

if ((CurrentValue - ComparisonValue) < 1.5) then
-- solange die Bodentemperatur unter der definierten Max. Bodentemperatur ist, kann das Script fortfahren
grp.write('32/5/0', 1)
if (CurrentFloorTempChoir < MaxFloorTemperature) and (CurrentFloorTempShip < MaxFloorTemperature) then
-- Wenn ein Event ansteht und Wärmeanforderung besteht, wird die Komforttemperatur angesteuert
grp.write('32/5/1', 1)
if PreheatActive then
-- Hauptschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
grp.write('32/5/2', 1)
if (CurrentValue < UpperComfortSetpointValueMainContactor) and not (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 1)
elseif (CurrentValue > UpperComfortSetpointValueMainContactor) and (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 0)
end
-- Dreiecksschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
if (CurrentValue < UpperComfortSetpointValueDeltaContactor) and not (StatusDeltaContactor1 or StatusDeltaContactor2) then
os.sleep(Delay)
grp.write(DeltaContactor, 1)
elseif (CurrentValue > UpperComfortSetpointValueDeltaContactor) and (StatusDeltaContactor1 or StatusDeltaContactor2) then
grp.write(DeltaContactor, 0)
end
-- Sternschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
if (CurrentValue > LowerComfortSetpointValueStarContactor) and not (StatusStarContactor1 or StatusStarContactor2) then
os.sleep(Delay)
grp.write(StarContactor, 1)
elseif (CurrentValue < LowerComfortSetpointValueStarContactor) or (CurrentValue > UpperComfortSetpointValueStarContactor) and (StatusStarContactor1 or StatusStarContactor2) then
grp.write(StarContactor, 0)
end
-- Wenn kein Event ansteht, wird die Standbytemperatur angesteuert
elseif BKNight then
-- Hauptschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
grp.write('32/5/2', 0)
if (CurrentValue < UpperStandbySetpointValueMainContactor) and not (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 1)
elseif (CurrentValue > UpperStandbySetpointValueMainContactor) and (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 0)
end
-- Dreiecksschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
if (CurrentValue > UpperStandbySetpointValueDeltaContactor) and not (StatusDeltaContactor1 or StatusDeltaContactor2) then
os.sleep(Delay)
grp.write(DeltaContactor, 1)
elseif (CurrentValue > UpperStandbySetpointValueDeltaContactor) and (StatusDeltaContactor1 or StatusDeltaContactor2) then
grp.write(DeltaContactor, 0)
end
-- Sternschütze einschalten, wenn der aktuelle Wert unter dem Soll-Werte für die Schütze ist
if (CurrentValue > LowerStandbySetpointValueStarContactor) and (CurrentValue < UpperStandbySetpointValueStarContactor) and not (StatusStarContactor1 or StatusStarContactor2) then
os.sleep(Delay)
grp.write(StarContactor, 1)
elseif (CurrentValue < LowerStandbySetpointValueStarContactor) or (CurrentValue > UpperStandbySetpointValueStarContactor) and (StatusStarContactor1 or StatusStarContactor2) then
grp.write(StarContactor, 0)
end
else
grp.write('32/5/2', 0)
if (StatusStarContactor1 or StatusStarContactor2) then
grp.write(StarContactor, 0)
elseif (StatusDeltaContactor1 or StatusDeltaContactor2) then
grp.write(DeltaContactor, 0)
elseif (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 0)
end
end
else
grp.write('32/5/1', 0)
if (StatusStarContactor1 or StatusStarContactor2) then
grp.write(StarContactor, 0)
elseif (StatusDeltaContactor1 or StatusDeltaContactor2) then
grp.write(DeltaContactor, 0)
elseif (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 0)
end
end
end
else
grp.write('32/5/0', 0)
if (StatusStarContactor1 or StatusStarContactor2) then
grp.write(StarContactor, 0)
elseif (StatusDeltaContactor1 or StatusDeltaContactor2) then
grp.write(DeltaContactor, 0)
elseif (StatusMainContactor1 or StatusMainContactor2) then
grp.write(MainContactor, 0)
end
end
end
end
end

```

