

# Labor


## Grundlagen der Gebäudeautomation



BILDQUELLE: WAGO KONTAKTTECHNIK GMBH & CO. KG

## Versuch 2

### *Modularisierung von Programmen und Programmierung in CFC*

 <b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK	<p style="text-align: center;"><b>Labor</b></p> <p style="text-align: center;">Grundlagen der Gebäudeautomation</p> <p style="text-align: center;"><b>Versuch 2</b></p>	<p style="text-align: center;">Autor</p> <p style="text-align: center;"><b>Prof. Dr. Sven Rogalski</b></p>
--	---	--


## Lernziele

Die Versuche bieten die Möglichkeit zur Programmierung einfacher, modularer CoDeSys-Anwenderprogramme mit dem WAGO I/O-System 750. Hierbei sind neben den CoDeSys-Standardbausteinen auch eigene, zuvor erstellte Programmbausteine zu erstellen und einzubinden. In diesem Zusammenhang wird verstärkt mit der Programmiersprache Continuous Flow Chart (CFC) gearbeitet. Zudem sind bestehende Steuerungskonfigurationen anderer CoDeSys-Projekte wiederzuverwenden.

## Allgemeine Hinweise

- Arbeiten Sie die Aufgaben dieses Versuchs vollständig durch und lösen Sie diese bereits zuhause!
- Die in den Aufgaben zu erstellenden Schaltlogiken/SPS-Programme sind handschriftlich auf dem Papier vorzubereiten und werden nach Eingabe in den jeweiligen CoDeSys-Editor mit den Laborgeräten getestet! Dokumentieren Sie Ihre Lösungen individuell, d.h. **jedes Gruppenmitglied hat eine Lösung!**
- Jede der Aufgaben ist in einem neuen CoDeSys-Projekt zu bearbeiten.
- Bevor Sie mit einer neuen Aufgabe beginnen, lassen Sie die Programmfunktionalität der vorangegangenen durch das Laborpersonal abnehmen, um Folgefehler zu vermeiden!
- Sämtliche Ergebnisse Ihrer Arbeit sind in einem von Ihnen anzufertigenden Laborbericht zu dokumentieren! Dafür steht Ihnen im Labor auch ein Drucker zur Verfügung, über den Sie Ihre in CoDeSys erstellten Schaltlogiken ausdrucken und danach handschriftlich kommentieren können.
- Der **Laborbericht** wird bewertet und berücksichtigt neben den dokumentierten Ergebnissen auch die Qualität der Versuchsvorbereitung und Versuchsdurchführung!

**Achtung: Eine fehlende oder mangelhafte Vorbereitung Ihrerseits führt unweigerlich zum Ausschluss von diesem Laborversuch!**

 <b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK	<b>Labor</b> Grundlagen der Gebäudeautomation (GGA) <b>Versuch 2</b>	Autor <b>Prof. Dr. Sven Rogalski</b>
--	--	---

## Aufgabe 1: Erstellen eines Vorlagenprojekts

In dieser Aufgabe erstellen Sie ein neues CoDeSys-Projekt, in dem Sie vorbereitend für alle nachfolgenden Aufgaben die Kommunikationsparameter und den Klemmenbus für das auf dem LN-Schulungsboard befindliche WAGO-System definieren. Die dem Klemmenbus zuzuweisenden Systemparameter sind aus einer vorbereiteten Datei zu importieren, wobei die Parameterbezeichnungen der Eingänge von der *Temperatureingangsklemme (750-469)* durch eigene Bezeichner zu ersetzen sind.

- 1) Starten Sie CoDeSys und bestimmen Sie als Controller den Ihres LN-Schulungsboards (Controller-Typ: **750-880 - 750-880-040**). Wählen Sie im sich öffnenden Dialogfenster „Neuer Baustein“ die Programmiersprache CFC und speichern Sie Ihr Projekt unter dem Namen **"Vorlagen\_Projekt"**.
- 2) Überprüfen Sie unter *Kommunikationsparameter* die IP-Adresse Ihres Controllers, die wie folgt lauten muss: **70.70.10.xx** (vollständige IP-Adresse des Controllers steht auf dem LN-Schulungsboard).
- 3) Lesen Sie in CoDeSys die Hardware-Konfiguration Ihres LN-Schulungsboards aus (vgl. Punkt 2 im Dokument „Laboreinführung“) ein und legen Sie dann, für die beiden Eingänge der *Temperatureingangsklemme (750-469)*, zwei geeignete Namen als Systemparameter fest.
- 4) Da das Festlegen aller Systemparameter für die Ein- und Ausgänge der Klemmen zu zeitaufwändig ist, sollen Sie eine bereits vorbereitete *Hardwarekonfiguration* importieren.  
**Hinweis:** Das Vorgehen hierzu ist unter Punkt 2 im Dokument „Laboreinführung“ beschrieben.
  - Kopieren Sie hierzu die Datei "*Vorlage\_HardwareKonfig*" in Ihren Labor-Ordner.
  - Öffnen Sie die CSV-Datei und benennen Sie diese Eingänge der *Temperatureingangsklemme (750-469)* mit den zuvor durch Sie vergebenen Namen um!
  - Importieren Sie nun über die „*Hardware configuration*“ (innerhalb der Steuerungskonfiguration) die CSV-Datei in Ihr Projekt.
- 5) Überprüfen Sie das Ergebnis des Imports und zeigen Sie dies Ihrem Dozenten!
- 6) Exportieren Sie nun Ihre *Hardware-Konfiguration* als CSV-Datei in Ihren Labor-Ordner (ggf. in einen separaten Unterordner) und geben Sie der CSV-Datei einen geeigneten Namen!
- 7) Dokumentieren Sie Ihre Ergebnisse zu dieser Aufgabe im Laborbericht!

## Aufgabe 2: Funktion "Not XOR"

Es soll eine eigene Funktion "NOT XOR" erstellt werden, die **ausschließlich auf Basis der drei booleschen Grundoperationen** aufzubauen ist (andere in CoDeSys angebotene Standard-Operatoren sind nicht zulässig).

Diese Funktion soll die beiden Eingangsvariablen "xZustand1\_NXOR" und "xZustand2\_NXOR" haben sowie eine Ausgangsvariable, die den Namen der Funktion haben muss.

- Erzeugen Sie ein neues CoDeSys-Projekt (Menü: „Datei → Neu“) mit dem Namen "NOT\_XOR" und importieren Sie die zuvor exportierte Hardware-Konfiguration in ihr neues Projekt.
- Legen Sie neben dem Standardprogramm-Baustein "PLC\_PRG" eine neue Funktion namens "xFC\_NXOR" an und realisieren Sie dort die Schaltlogik des "NOT XOR" in der Programmiersprache **KOP**! Testen Sie die Schaltlogik Ihrer Funktion gemäß einer von Ihnen bereits zuhause erstellten Wertetabelle.
- Exportieren Sie Ihre neu erstellte Funktion "xFC\_NXOR" in einen separaten Ordner namens "**Eigene Bausteine**" innerhalb Ihres Labor-Ordners!
- Bilden Sie im Programmbaustein "PLC\_PRG" die in **Abbildung 1** dargestellte Schaltlogik in **Continuous Function Chart (CFC)** nach und verwenden Sie hierbei Ihre eigene Funktion "xFC\_NXOR". Beachten Sie, dass es sich bei den Variablen xS1 - xS4 um die bereits importierten Systemvariablen zum LN-Schulungsboard handelt, wohingegen sich die Bezeichnung „Lampe“ auf den Ausgang bezieht, über den die Lampe des LLT-Experimentiermodells geschaltet wird!

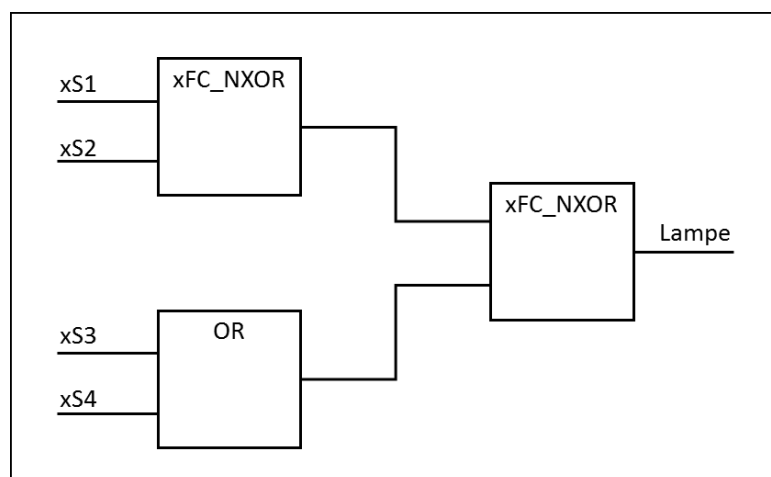



Abbildung 1: In CoDeSys umzusetzende Schaltlogik

- Testen Sie die Schaltlogik und zeigen Sie das Ergebnis Ihrem Dozenten!
- Welche Schaltzustände führen zu einem TRUE-Signal (Lampe leuchtet)?
- Wie viel Instanzen ihrer Funktion "xFC\_NXOR" wurden im Baustein "PLC\_PRG" erzeugt und wovon hängt die Anzahl der Aufrufe dieser Funktion ab?
- Dokumentieren Sie Ihre Ergebnisse zu dieser Aufgabe im Laborbericht!

 <b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK	<b>Labor</b> Grundlagen der Gebäudeautomation (GGA) <b>Versuch 2</b>	Autor <b>Prof. Dr. Sven Rogalski</b>
--	--	---

### Aufgabe 3: Funktionsblock zur Temperaturanzeige

Es soll ein eigener Funktionsblock erzeugt werden, der eine Umrechnung des von der *Temperatureingangsklemme (750-469)* gelieferten Rohwertes in Grad Celsius und Grad Fahrenheit ermöglicht.

Dieser Funktionsblock muss die Eingangsvariable "*iTemp\_Roh*" haben und über die Ausgangsvariablen "*rGrad\_Celsius*" und "*rGrad\_Fahrenheit*" die umgerechneten Temperaturwerte **auf eine Kommastelle genau** (z.B. 22,3°C) ausgeben!

- Erzeugen Sie ein neues CoDeSys-Projekt mit dem Namen "*Temp\_Konvert*" indem Sie das Projekt "*Vorlagen\_Projekt*" als **neue Vorlage** öffnen (Menüpunkt "*Neu aus Vorlage*")!
- Legen Sie in Ihrem Projekt einen neuen Funktionsblock namens "*FB\_Temp*" in der Programmiersprache **Continuous Function Chart (CFC)** an und realisieren Sie dort eine Schaltlogik zur Ermittlung des Temperaturwertes in °C und °F aus dem über die von der *Temperatureingangsklemme (750-469)* gelieferten Rohwert.  
**Hinweis:** Der Temperaturwert in °C ist ein Zehntel des Rohwertes (siehe auch Punkt 3 im Dokument „Laboreinführung“.
- Fügen Sie Ihren Funktionsblock in "*PLC\_PRG*" als **FUP**-Programm ein und verschalten Sie die vom Funktionsblock angebotenen Ein- und Ausgänge sinnvoll. Verbinden Sie danach das LLT-Experimentiermodell richtig mit dem LN-Schulungsboard und laden Sie Ihr Programm auf Ihren Wago-Controller. Testen Sie die Programmfunktionalität und zeigen Sie das Ergebnis Ihrem Dozenten!
- Exportieren Sie Ihren neu erstellten Funktionsblock "*FB\_Temp*" in Ihren Ordner "**Eigene Bausteine**"!
- Wie viel Instanzen ihres Funktionsblocks "*FB\_Temp*" wurden in Ihrem Projekt erzeugt?
- Dokumentieren Sie alle Ihre Ergebnisse zu dieser Aufgabe in Ihrem Laborbericht!

## Aufgabe 4: Funktionsblock zur Spannungsauswertung

Auf dem LN-Schulungsboard befinden sich zwei Potenziometer, die mit der *2-Kanal Analogeingangsklemme (750-477)* verbunden sind. In dieser Aufgabe ist ein Funktionsblock zu erstellen, der die vom Potenziometer gelieferten Spannungswerte (0-10V) in einen dezimalen Zahlenwert umrechnet und diesen auf eine Kommastelle genau ausgibt (z.B. 3,2V). Zusätzlich soll es für den Funktionsblock möglich sein, Grenzwerte festzulegen, die bei Über- bzw. Unterschreitung eines vorgegebenen Dezimalwertes ein Warn- bzw. Alarmsignal auslösen.

Aus diesem Grund ist vorgeschrieben, dass der Funktionsblock die in nachstehender Tabelle aufgeführten **Ein- und Ausgangsvariablen** beinhaltet:

Eingangsvariablen	
Variablenname	Erklärung
<i>wSpannungswert</i>	Wert der vom Potenziometer kommenden Spannung
<i>iAlarm_oben</i>	bei Überschreiten dieses Vorgabewerts wird ein Alarmsignal ausgelöst
<i>iAlarm_unten</i>	bei Unterschreiten dieses Vorgabewerts wird ein Alarmsignal ausgelöst
<i>iWarnung_oben</i>	bei Überschreiten dieses Vorgabewerts wird ein Warnsignal ausgelöst
<i>iWarnung_unten</i>	bei Unterschreiten dieses Vorgabewerts wird ein Warnsignal ausgelöst
Ausgangsvariablen	
<i>rWert_Dez</i>	Ausgabe des in eine Dezimalzahl umgewandelten Spannungswerts, gerundet auf eine Kommastelle
<i>xAlarm</i>	Ausgabe eines booleschen Signals, das bei Überschreiten/Unterschreiten des Vorgabewertes (bezogen auf Alarm) auf TRUE gesetzt wird
<i>xWarnung</i>	Ausgabe eines booleschen Signals, das bei Überschreiten/Unterschreiten des Vorgabewertes (bezogen auf Warnung) auf TRUE gesetzt wird

Tabelle 1: Ein- und Ausgangsvariablen des Funktionsblocks "FB\_VoltConvert"

- Erzeugen Sie ein neues CoDeSys-Projekt mit dem Namen "VoltConvert", das die gleiche Hardwarekonfiguration beinhaltet, wie das Projekt "Vorlagen\_Projekt".
- Legen Sie im Projekt einen neuen Funktionsblock namens "FB\_VoltConvert" in einer Programmiersprache ihrer Wahl an und realisieren Sie dort eine Schaltlogik, die einen Spannungswert 0-10V in einen REAL-Wert bis **auf eine Kommastelle genau** umwandelt.

**Hinweis:** Die Busklemme, an die das Potenziometer angeschlossen ist, liefert einen Spannungswert in Millivolt (mV) (siehe auch Punkt 3 im Dokument „Laboreinführung“).

**Achtung:** Die Verwendung des Bausteins „LIMITALARM“ ist hierbei nicht erlaubt!


- c) Entwickeln Sie, unter Verwendung Ihres neuen Funktionsblock, in "*PLC\_PRG*" eine Steuerung als **CFC**-Programm und verbinden Sie Ihr LN-Schulungsboard mit dem LLT-Experimentiermodell, so dass die Lampe auf dem LLT-Experimentiermodell von einem der beiden Potenziometer wie folgt angesteuert wird:

Aktion	Spannungswert
Lampe ist aus	Spannungswert befindet sich innerhalb der Warngrenzen von <i>iWarnung_oben</i> und <i>iWarnung_unten</i>
Lampe blinkt mit Frequenz von 1 Sekunde	Spannungswert befindet sich außerhalb der Warngrenzen von <i>iWarnung_oben</i> und <i>iWarnung_unten</i> , jedoch noch innerhalb der Alarmgrenzen <i>iAlarm_oben</i> und <i>iAlarm_unten</i>
Lampe blinkt mit Frequenz von 0,5 Sekunden	Spannungswert befindet sich außerhalb der Alarmgrenzen von <i>iAlarm_oben</i> und <i>iAlarm_unten</i>

*Tabelle 2: Aktionen auf dem LLT-Experimentiermodell in Abhängigkeit vom Spannungswert*

**Achtung:** Beachten Sie vor der Verwendung des Potenziometers auf das Setzen einer entsprechenden Kabelbrücke auf dem LN-Schulungsboard!

- d) Laden Sie Ihr Programm auf Ihren Wago-Controller und testen Sie die Programmfunktionalität. Zeigen Sie das Ergebnis Ihrem Dozenten!
- e) Exportieren Sie Ihren neu erstellten Funktionsblock "*FB\_VoltConvert*" in Ihren Ordner "**Eigene Bausteine**"!
- f) Ist die von Ihnen entworfene Logik des Funktionsblocks auch als Funktion realisierbar? Begründen Sie Ihre Antwort!
- g) Nennen Sie ein Beispiel aus dem Gebäudebereich, für das ein solcher Funktionsblock relevant sein könnte. Erklären Sie dabei die sinnvolle Verwendung der implementierten Ein- und Ausgangsvariablen!
- h) Dokumentieren Sie Ihre Ergebnisse im Laborbericht!

 <b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK	<b>Labor</b> Grundlagen der Gebäudeautomation (GGA) <b>Versuch 2</b>	Autor <b>Prof. Dr. Sven Rogalski</b>
--	--	---

## Aufgabe 5: CO<sub>2</sub>-Überwachung in einer Tiefgarage

Um den natürlichen CO<sub>2</sub>-Gehalt in einer Tiefgarage aufrecht zu erhalten, soll ein CO<sub>2</sub>-Raumluft-Sensor simuliert werden. Er überwacht die Güte der Raumluft, indem er deren Kohlendioxid-Gehalt misst. Dieser Wert ist jedoch stark von der Größe der Tiefgarage abhängig, weshalb meist mehrere CO<sub>2</sub>-Raumluft-Sensoren zu platzieren sind. Da man bei Parkhäusern davon ausgeht, dass sich Personen nicht länger als 15 Minuten darin aufhalten, gilt der Kurzzeitwert für CO<sub>2</sub> Grenzwerte. Dieser darf 10.000ppm (parts per million - 1ppm = 0,0001%) nicht übersteigen.

In diesem Übungsbeispiel sollen die beiden Potenziometer auf Ihrem LN-Schulungsboard jeweils einen CO<sub>2</sub>-Sensor simulieren, bei denen eine Spannung von 0 Volt = 0ppm entspricht und 10 Volt = 10.000ppm. Sollte bei einem oder beiden der zwei Sensoren (Potenziometer) der CO<sub>2</sub>-Gehalt größer 7.000ppm sein, wird der Lüfter des LLT-Experimentiermodell eingeschaltet, um zu simulieren, dass die CO<sub>2</sub>-Konzentration durch Frischluftzufuhr in der Tiefgarage gesenkt wird. Steigt die CO<sub>2</sub>-Konzentration dennoch weiter an und erreicht den oberen Grenzwert von 9.000ppm, soll zusätzlich die Lampe auf dem LLT-Experimentiermodell eingeschaltet werden, um einen Alarm in der Tiefgarage zu signalisieren. Ist die CO<sub>2</sub>-Konzentration unter den 7.000ppm sind Lüfter und Lampe ausgeschaltet.

- Erstellen Sie ein neues Projekt namens "CO<sub>2</sub>-Control" und schreiben Sie ein CFC-Programm, das bei Erreichen der definierten Grenzwerte die entsprechenden Aktionen auslöst. Verwenden Sie dazu Ihre als Vorlage definierte Hardwarekonfiguration „*Vorlage\_HardwareKonfig.csv*“ und Ihren in Aufgabe 4 erzeugten Funktionsblocks "*FB\_VoltConvert*", der aus Ihrem Ordner "Eigene Bausteine" zu importieren ist.

**Hinweis:** Der aktuelle CO<sub>2</sub>-Wert soll in einer eigenen Variablen als Prozentwert angezeigt werden.

- Verbinden Sie Ihr LLT-Experimentiermodell mit dem LN-Schulungsboard entsprechend der in Ihrem Programm definierten Ein- und Ausgänge. Testen Sie danach Ihre Programmfunktionalität und zeigen Sie diese Ihrem Dozenten.
- Dokumentieren Sie die Ergebnisse in Ihrem Laborbericht!



## Aufgabe 6: Steuerung einer Heizungsanlage

In dieser Aufgabe, die sich in zwei Teilbereiche untergliedert, wird exemplarisch am LLT-Experimentiermodell die Steuerung einer Heizungsanlage simuliert. Folgender Steuerungsablauf ist dabei zugrunde zu legen:


Sofern die Heizungsanlage eingeschaltet ( $S1 = \text{TRUE}$ ) und der NOTAUS-Schalter nicht betätigt wurde ( $S2 = \text{TRUE}$ ), ist bei Unterschreitung eines zuvor definierten Temperaturwertes (*Heizen\_An*) der elektrothermische Stellantrieb einer Flächenheizung zu öffnen. Nach einer zeitlichen Verzögerung von 5 Sekunden schaltet sich die Umwälzpumpe der Heizungsanlage ein. Sobald ein zweiter zuvor festgelegter Temperaturwert (*Heizen\_Aus*) überschritten wird, muss der elektrothermische Stellantrieb wieder schließen und gleichzeitig die Umwälzpumpe abschalten. Gleiches passiert auch bei Betätigung des NOTAUS-Schalters ( $S2 = \text{FALSE}$ ) oder beim Ausschalten der Heizungsanlage ( $S1 = \text{FALSE}$ ) im laufenden Betrieb. Hierbei werden Stellantrieb und Umwälzpumpe spannungslos geschaltet.

Nachstehende Tabelle listet die Komponenten die Labor-Komponenten auf mit dem Verweis darauf, welches Element in der Heizungsanlage sie repräsentiert/simuliert:

Labor-Komponente	Element in der Heizungsanlage
Potenziometer (LN-Schulungsboard)	Raum-Temperatursensor
Lüfter (LLT-Experimentiermodell)	Motor der Umwälzpumpe
Lampe (LLT-Experimentiermodell)	Schaltzeituhr für Nachtabsenkung
elektrothermischer Stellantrieb	elektrothermischer Stellantrieb

*Tabelle 3: Bedeutung der Labor-Komponenten*

Den **elektrothermischen Stellantrieb** finden Sie bei den Ihnen zur Verfügung gestellten Labormitteln. Dieser wird entweder mit 0V oder mit 24V Spannung versorgt. Im spannungslosen Zustand schließt der Stellantrieb (Sicherheitsstellung = zu)! Auf dem Eckventil des thermischen Stellantriebs ist zudem ein Mikrometer befestigt, anhand dessen Sie die Veränderung der Ventilstellung besser nachvollziehen können.

 <b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK	<b>Labor</b> Grundlagen der Gebäudeautomation (GGA) <b>Versuch 2</b>	Autor <b>Prof. Dr. Sven Rogalski</b>
--	--	---

### Aufgabe 6.1: Steuerung ohne Nachtabsenkung

- Erstellen Sie ein neues Projekt namens "*Hzg-Control*", importieren Sie Ihre als Vorlage definierte Hardwarekonfiguration „*Vorlage\_HardwareKonfig.csv*“.
- Schreiben Sie nun ein Programm in der Programmiersprache Ihrer Wahl, durch das der oben beschriebene Steuerungsablauf realisiert wird.
  - Ermitteln Sie für Ihren Modellversuch geeignete Temperaturgrenzwerte zum automatischen Ein- und Ausschalten der simulierten Heizungsanlage.
  - Überlegen Sie, inwieweit Sie die beiden von Ihnen erstellten Funktionsblöcke "*FB\_Temp*" und "*FB\_VoltConvert*" in Ihre Aufgabenlösung mit einbeziehen können.
  - Das Einschalten der Heizungsanlage soll durch den Schalter "*S1*" auf dem LN-Schulungsboard erfolgen, der *NOTAUS*-Schalter hingegen wird durch den Schalter "*S2*" repräsentiert.
- Verbinden Sie die erforderlichen Labor-Komponenten mit Ihrem LN-Schulungsboard entsprechend der in Ihrem Programm definierten Ein- und Ausgänge. Testen Sie danach Ihre Programmfunktionalität und zeigen Sie diese Ihrem Dozenten.
- Dokumentieren Sie die Ergebnisse in Ihrem Laborbericht!


### Aufgabe 6.2: Steuerung mit Nachtabsenkung

Aufbauend auf Aufgabe 6.1 ist das Projekt zusätzlich um eine Nachtabsenkung zu erweitern. Abhängig von der Tag- und Nachtzeit, ist hierbei eine Unterscheidung zwischen verschiedenen Temperaturgrenzwerten für das Ein- und Ausschalten der Heizungsanlage vorzunehmen. Diese Unterscheidung soll mit Hilfe der Lampe des LLT- Experimentiermodells erfolgen. Ist die Lampe eingeschaltet, signalisiert dies den Tagbetrieb der Heizungsanlage. Im ausgeschalteten Zustand der Lampe wird dagegen der Nachtbetrieb aktiviert.

- Beenden Sie Ihr Projekt "*Hzg-Control*" und öffnen Sie dieses als "*Neu aus Vorlage*". Speichern Sie es danach unter einem neuen, geeigneten Namen.
- Erweitern Sie Ihr in Aufgabe 6.1 erstelltes Programm um die Funktionalität der Unterscheidung zwischen Tag- und bei Nachtbetrieb!
  - Bestimmen Sie für Ihren Modellversuch reale Temperaturgrenzwerte für den Tag- und Nachtbetrieb der simulierten Heizungsanlage!

**Hinweis:** Da die Roh-Temperaturwerte diesmal über Potenziometer geliefert werden, müssen Sie eine sinnvolle Umrechnung für den ankommenden Spannungswertes festlegen.

  - Das Ein-/Ausschalten der Lampe auf dem LLT-Experimentiermodell (zum Signalisieren des Tag- bzw. Nachtbetriebs) soll über den "*Schalter S9*" am LN-Schulungsboard erfolgen!

 <p><b>h_da</b> HOCHSCHULE DARMSTADT UNIVERSITY OF APPLIED SCIENCES <b>fbeit</b> FACHBEREICH ELEKTROTECHNIK UND INFORMATIONSTECHNIK</p>	<p style="text-align: center;"><b>Labor</b> Grundlagen der Gebäudeautomation (GGA) <b>Versuch 2</b></p>	<p style="text-align: center;">Autor <b>Prof. Dr. Sven Rogalski</b></p>
--	---	---

- c) Überprüfen Sie die Verbindung Ihres LLT-Experimentiermodells mit dem LN-Schulungsboard. Testen Sie danach Ihre Programmfunktionalität und zeigen Sie diese Ihrem Dozenten.
- d) Erklären Sie, welche Bedeutung die "Sicherheitsstellung" von Ventilen hat!
- e) Worin besteht der Unterschied zwischen einem elektrothermischen und einem elektromotorischen Stellantrieb?
- f) Was bewirkt eine Nachtabenkung und in welchen Temperaturbereichen wird eine Heizung am Tage und in der Nacht bei Wohnhäusern betrieben?
- g) Deklarieren Sie eine Ihrer verwendeten Variablen (aus den Funktionsblöcken oder dem Programm), die Ihnen sinnvoll erscheint, einmal als *REATAIN*, einmal als *PERSISTENT* und einmal als *RETAIN PERSISTENT*. Testen Sie das Verhalten dieser Variablen bei:
  - RESET (Warmstart)
  - RESET (Kaltstart)
  - RESET (Ursprung)

Was fällt Ihnen auf und welche Bedeutung haben die drei verschiedenen RESET-Typen?
- h) Dokumentieren Sie Ihre Ergebnisse im Laborbericht!

### Aufgabe 6.3: Steuerung mit Nachtabenkung und NOTAUS-Signalisierung

Aufbauend auf Aufgabe 6.2 sollen Sie das Projekt dahingehend erweitern, dass im Falle des Betätigens des *NOTAUS*-Schalters die Lampe auf dem LLT-Experimentiermodell in einer von Ihnen festgelegten Frequenz zu blinken anfängt.

- a) Erstellen Sie hierzu ein neues Projekt unter einem geeigneten Namen.
- b) Erläutern Sie, zu welchem Problem die zusätzliche Beschaltung der Lampe auf dem LLT-Experimentiermodell durch den NOTAUS-Schalter führen könnte!
- c) Dokumentieren Sie Ihre Ergebnisse im Laborbericht!