Block Diagram

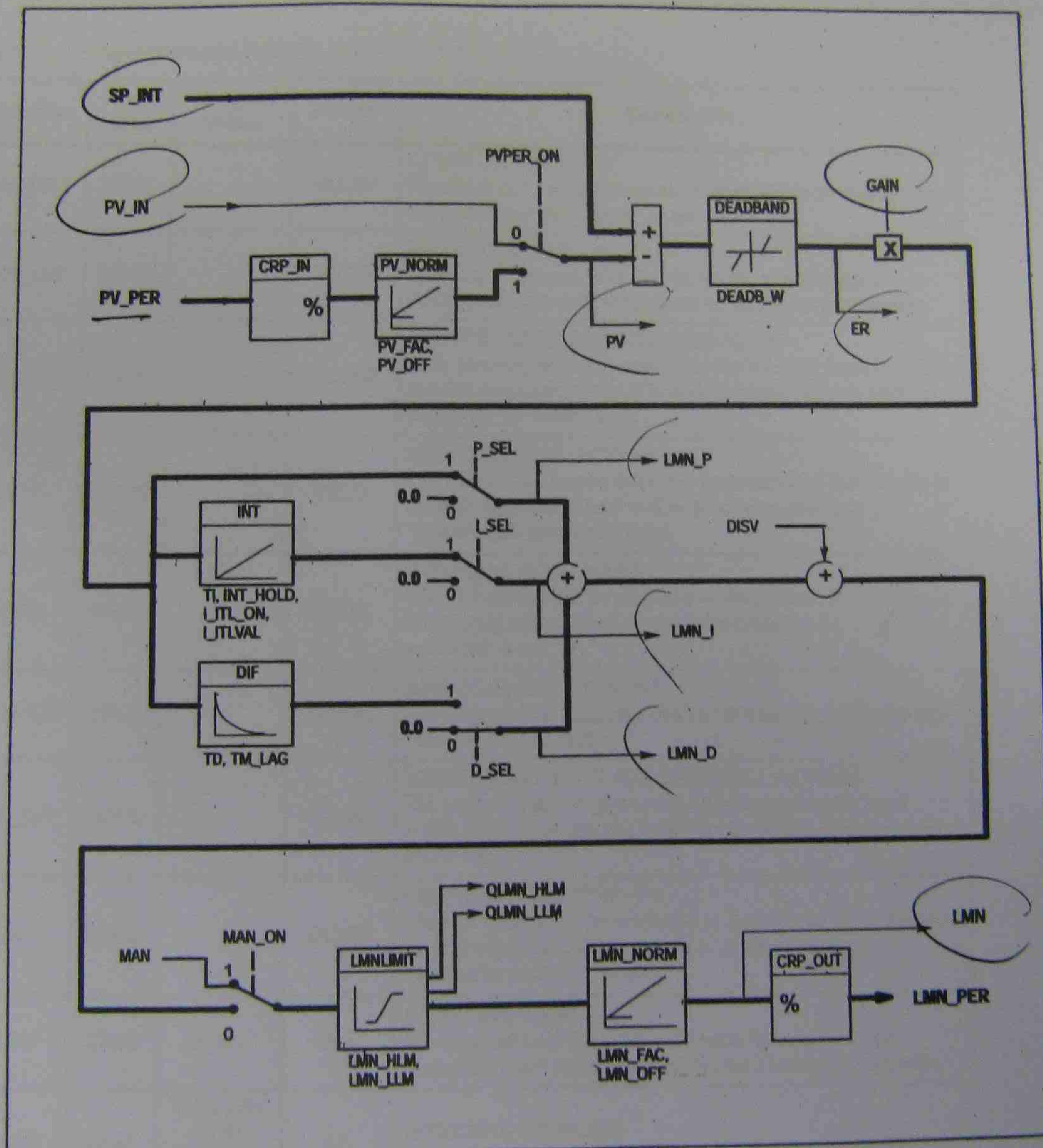


Figure 3-1 Block Diagram of CONT_C

~~David Holbeck~~ David Milbeck
SID: 31138392 2

Theory Research Assignment ***PLC System Applications***

1. In relation to Control System , define the following terms.

- Proportional Band (PB)
- Control Lag
- Dead Time
- Integral Time

2. Explain the terms Direct and Reverse Action , give an example of an application of each action

3. Explain the term "Offset" and it's causes. Explain how offset can be eliminated.

4. Explain the term Derivative Control , and Derivative Time.

- **In relation to PLC Communications answer the following.**

5. Define the following terms

- Fieldbus
- Protocol
- Baud Rate
- Address
- Master Slave
- Peer to Peer

Notes:

- Do your own work
- answer all the questions
- This is 10% of your final result
- Due Week 18 , marks will deducted for late submissions.
- Hand in this sheet with your answers

6. Outline the main differences between RS232 and RS485

7. To which Profibus layer would you connect a field device

8. What are the advantages of Profibus

9. What is the purpose of a "Token Ring" ?

10. Explain the difference between full and half duplex.

11. Could multiple devices be connected to an RS232 network ? Explain your answer.

12. What are the advantages of Fibre optic transmission.

PLC System Applications Theory Research Assignment

By David Holbeche SID: 311383922

1. Proportional Band(PB) (%) is the percentage change of the process variable required to give a 100% output change. It is the inverse of the gain. The %PB is the percentage of the entire process span which forms the proportional band around the setpoint. Above the proportional band, the output is fully off and below the proportional band the output is fully on. In other words the PB is the range of process variable over which the output will go from fully on, to fully off. Within the proportional band the output is switched on and off in the ratio of the measurement difference from the setpoint. If the process variable which is for example temperature, is further from the setpoint, the on and off times vary in proportion to this temperature difference. If the temperature is below the setpoint the output will be on longer, if the temperature is above the setpoint the output will be off for longer. In general the higher the PB, the slower, and the more stable the control is. The lower the PB is, the faster and less stable the control is.

The control lag of a system is the S-shaped response of a real world process: there is some delay between the change in final control element, and the actual response of the process variable.

Dead time is the delay from when the controller output signal is issued until when the measured process variable begins to respond to this output change.

The integral time of a control system is the time taken for the integral action to repeat the change in output due to the proportional action. The smaller the Ti(integral time), the greater the integral effect, or the steeper the ramp. The greater the Ti, the smaller the integral effect, and the slower the ramp.

2. In a direct acting controller, an increase in process variable will cause a decrease in the controller output. In other words, as the P.V increases the error will decrease, and the controller's proportional output will decrease. For a direct acting controller the gain Kp is positive. In a reverse acting controller, an increase in process variable will produce an increase in output, that is, the gain Kp is negative.
An example of a reverse-acting controller is controlling the flow of cooling water – if the temperature increases, the flow must be increased, to maintain the desired temperature (setpoint).
An example of a direct-acting controller is controlling the flow of steam for heating - if the temperature increases, the flow must be decreased, to maintain the desired temperature.

3.Offset is an inherent problem in proportional only control: there must be a certain error or "offset" for the controller to produce an output. This offset is the difference between the setpoint, and the process variable, once the output has settled. This offset is always present in proportional control as the output = Kp*error. The P.V will approach the setpoint, but will never reach it due to this offset. Offset is caused by a small Kp, where the system is very slow to respond to changes in the P.V. In an underdamped system, there is usually a permanent offset, when the controller output gradually decreases, as the error decreases and settles at a point below the setpoint. An increase in load will also cause an increase in the error, and an increase in offset. Offset can be removed by using Integral control as well as proportional control, or by increasing Kp.

4.Derivative (or Rate) control provides correction of the output which is proportional to the rate of change of the error. Derivative action slows the rate of change of the controller output and this effect is most noticeable close to the controller setpoint. Derivative control is used to reduce the magnitude of the overshoot produced by the Integral component and improve the combined controller-process stability. It is also used to speed up the response time to compensate for time lags in the control loop. As an example, a step change in P.V, or error, will produce a full power "kick" at the controller output to try and correct this error. If the error is constant, there will be no change in output. Derivative control is never used by itself, but is usually used in Proportional-Derivative control, or Proportional-Integral-Derivative control (PID). The reason for this is that Derivative control only knows the error is changing : it does not know what the setpoint is.

Derivative time(Td) is the amount of time that the derivative action apparently advances the output due to the proportional action. The higher the derivative effect, the higher the Td. The smaller the derivative effect, the smaller the Td.

5. Fieldbus is the name of a family of industrial computer network protocols used for real-time distributed control. It is a set of communication protocols that some hardware manufacturers use to make their field devices talk to other field devices. Fieldbus is a generic term that describes a new digital communications network that is being used in industry to replace the existing 4-20 mA analog signal standard.

A protocol is a standardized format or set of rules used for data exchange between computers, or in the case of a PLC, between PLCs and field devices. It is the definition of how data is arranged and coded for transmission on a PLC network.

Baud Rate is synonymous with symbols/second. It is the number of discrete conditions, or signal events per second in a serial communications channel. It is a measure of the speed of the communications channel. In RS232 and RS485 systems, this is the same as bits/s.

An address is a label which designates how the PLC and PLC software recognise a field device.

A master/slave configuration is a model for a communication protocol between PLCs. It consists of a master PLC, and one or more slave PLCs. The master initiates communication requests with the slaves, and the slaves respond to the master's requests for data. The direction of control is always from the master to the slave.

Peer to Peer is a communication model in which each party has the same capabilities, and either party can initiate a communication session. There is no master in a peer to peer network, and each PLC in the network is responsible for its own control site and only needs to be programmed for its own area of responsibility.

6. RS232 is an IEEE standard for serial communications that describes specific wiring connections, voltage levels, and other operating parameters for electronic data communications. There also are several other RS standards defined. RS232 and RS485 are both serial communication methods for computers and other devices. RS232 is an interface to connect one DTE (data terminal equipment) to one DCE (data communication equipment) . It is an old standard, and people soon looked for interfaces capable of one or more of the following 1. Connect DTEs directly without the need for a modem

2. Connect several DTEs in a network structure

3. Ability to communicate over longer distances

4. Ability to communicate at faster communication rates

RS485 is the most versatile communication standard in the standard series defined by the EIA(Electronic Industries Association), as it performs well on all four points listed above. RS485 is differential, whereas RS232 is not. RS485 operates in half-duplex mode, while RS232 can operate in half or full duplex. The maximum transmission distance of RS232 is 15m, whereas the maximum transmission distance for RS485 is 1200m.

7. Field devices should be connected to the physical Profibus layer.

8. Profibus encompasses several Industrial Bus Protocol Specifications, including Profibus-DP, Profibus-PA, Profibus-FMS, and PROFInet ad is one of the new generations of fieldbus I/O bus networks. Fieldbus is a generic term that describes a new digital communications network that is being used in industry to replace the existing 4-20 mA analog signal standard. The network is a digital , bi-directional, multidrop, serial-bus communication network used to link isolated field-devices, such as controllers, transducers, actuators and sensors.

The main advantage of a Fieldbus, or Profibus network is that it is a bus network and not a star network, which will reduce both configuration time and maintenance. It also improves performance, since Fieldbus networks perform at higher baud rates. This fieldbus approach produces significant cost savings in design,installation, and maintenance expenses over the old approach of point-to-point wiring.

Profibus has the following advantages over a conventional I/O system:

1. It is widely accepted with more than 1,100 member companies worldwide

2. Lower installation cost

3. point to point wiring is expensive

4. multiple bus transmission speeds and wiring length combinations are available.

9. Token ring is a LAN protocol which resides in the Data link layer of the OSI model. A token ring network is wired as a star, with hubs and arms which branch out to each station in the network. In a token ring network, a computer or PLC only has the right to transmit data if it holds the token(a special bit pattern). It then attaches the data and passes the token to the next computer in line. A transmitting station replaces the token with a frame which carries the information to be transferred. Data is sent from one machine or PLC to the next around the ring until it ends up back where it started.

10. A full Duplex system allows communication in both directions, simultaneously. Examples are mobile phones.

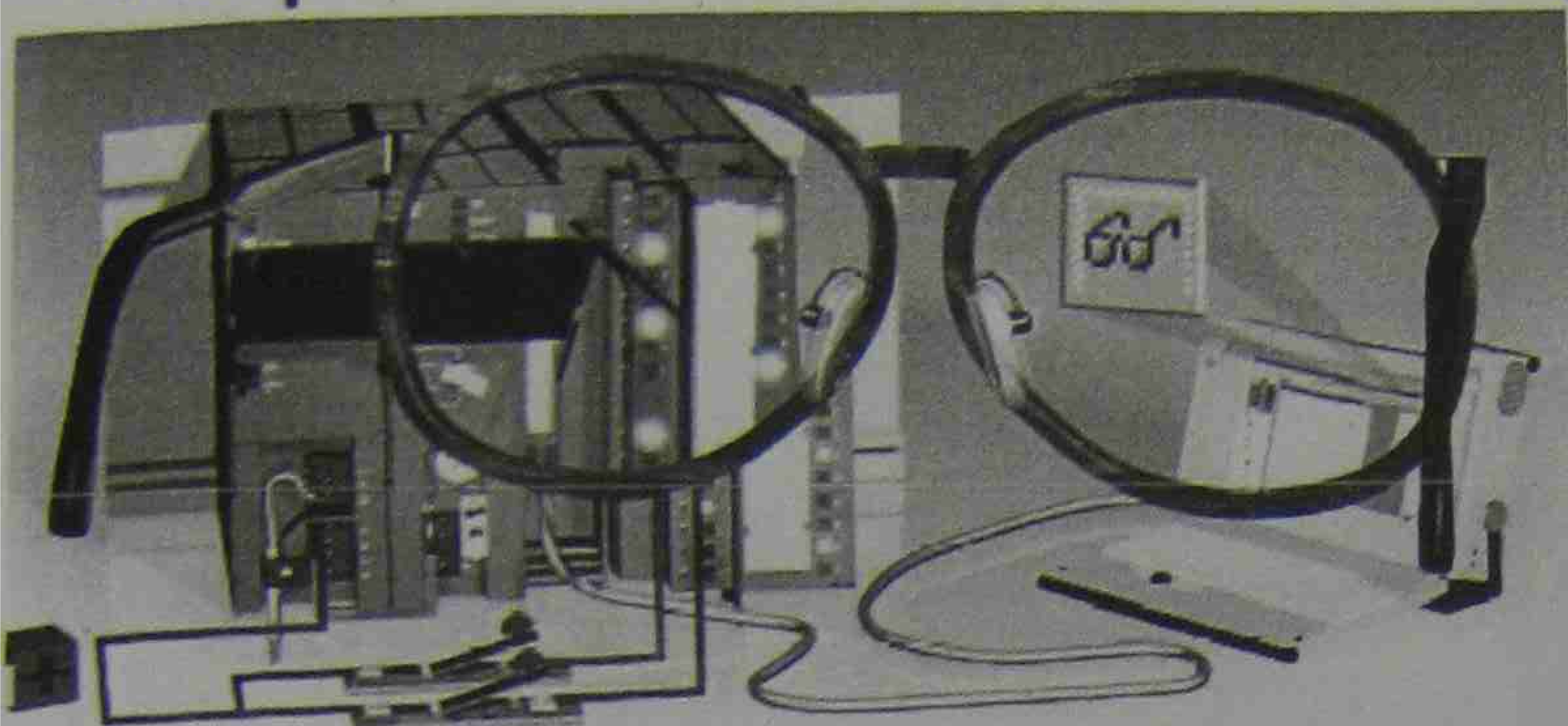
A half duplex system allows communication in both directions, but only one direction at a time (not simultaneously). An example of a half duplex system is a walkie-talkie in which one side must communicate then use the term "over", then the other side will communicate.

11. No, an RS232 network consists of two devices, the transmitting device , and the receiving device. These both consist of a DTE(data terminal equipment) and a DCE(data circuit terminating equipment). The network is a serial communications interface, and cannot support other network protocols like token ring, and peer to peer. It is simply a transmitting device and a receiving device.

12. The advantages of using fiber-optic transmission is that it is immune to electromagnetic Interference, has greater performance: greatly increased bandwidth and capacity, immunity to electrical noise, improved safety and electrical isolation (due to the fibers being non-conductive). Other advantages are reduced size and weight, environmental protection, and overall system economy. Fiber optic cables can be used over greater distances due to their low loss, high bandwidth properties.

PLC System Applications –Major Project
Temperature Control with a Step 7 PLC

~~SID: 3113839~~ SID: 3113839



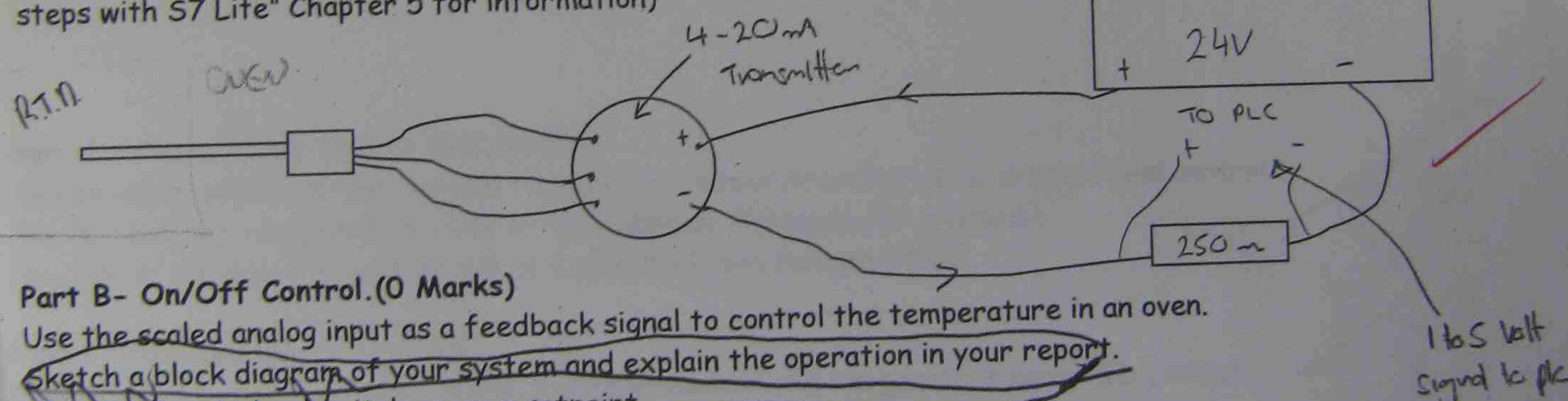
Aim: the aim of this major project is to assess student abilities to apply their knowledge to a new system. Building on previous knowledge the students will adapt this to a newer model PLC and software.

Procedure :

Examine the temperature control system that was used in a previous assignment.
Demonstrate that you are capable of connecting this system to your S7 313C plc.
Familiarize yourself with the S7 "lite" software , use the supplied PDF documents for reference material.

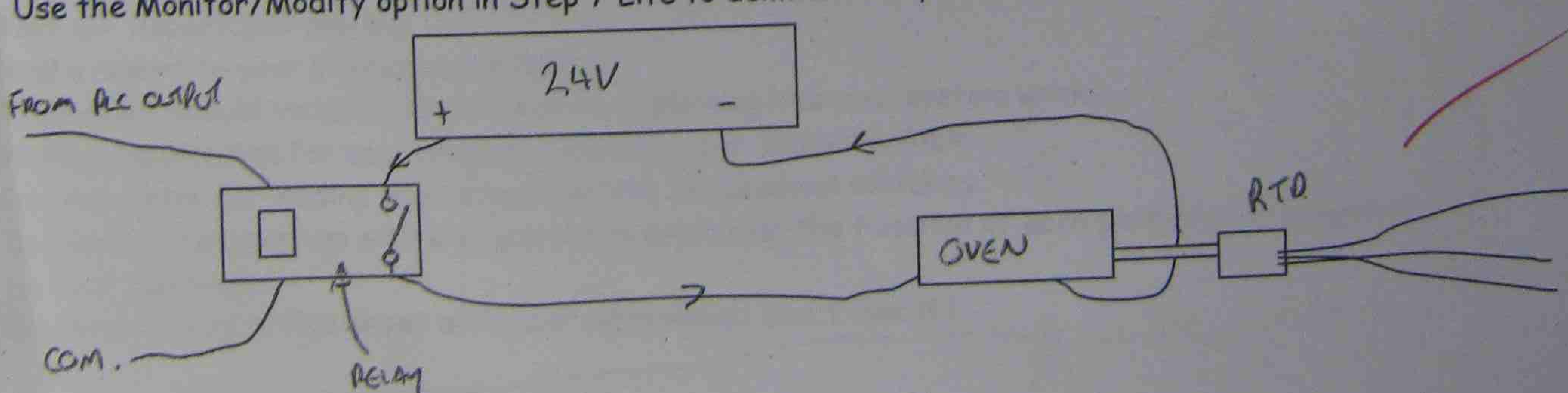
Part A – Analog Inputs.(0 Marks)

Connect the RTD input and scale this input to accurately display the temperature in a variable block or 7 segment display.
Create a full Symbolic table for the project and maintain the Symbolic table as your project develops. (See "First steps with S7 Lite" Chapter 5 for information)



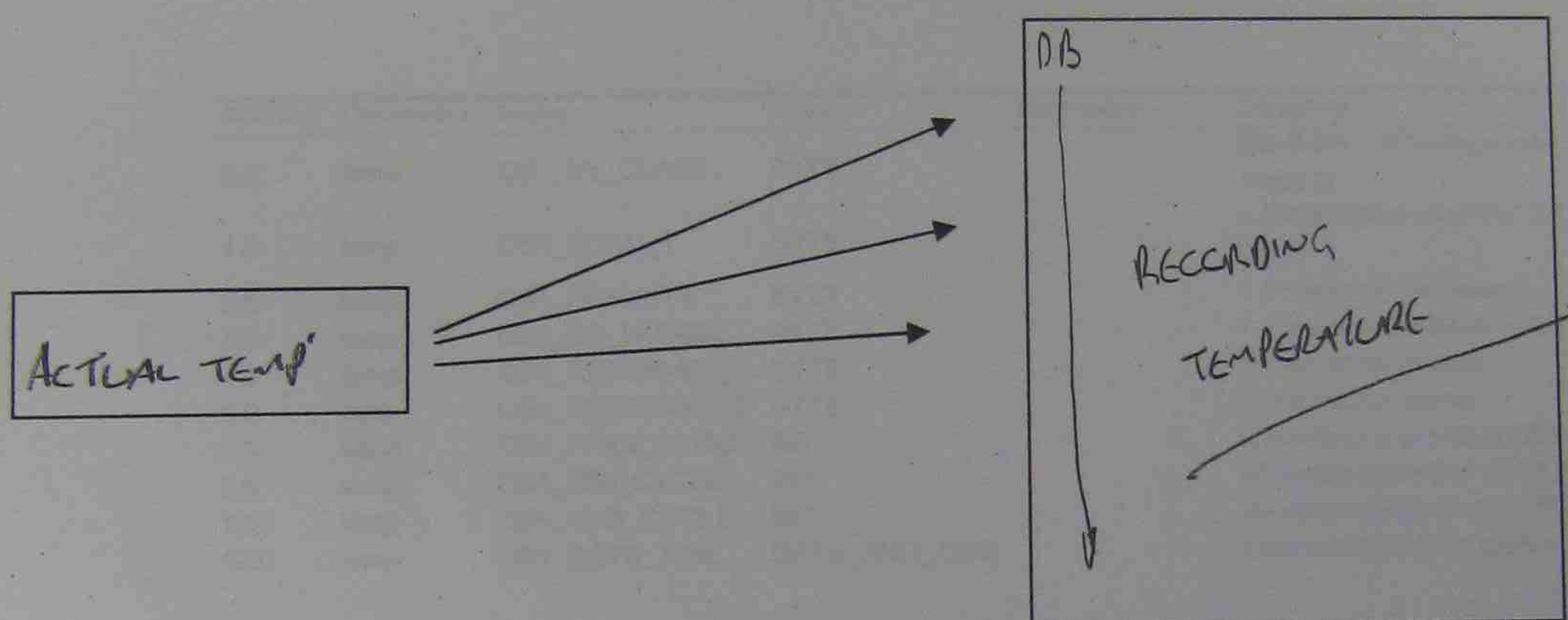
Part B- On/Off Control.(0 Marks)

Use the scaled analog input as a feedback signal to control the temperature in an oven.
Sketch a block diagram of your system and explain the operation in your report.
Use the Thumbwheel switch as your setpoint.
Use the Monitor/Modify option in Step 7 Lite to demonstrate your functional project)



Part C- Hystersis.(10 Marks)
Add some Hystersis to you your control system , the Hystersis temperature is set by the thumbwheel switch.
Make sure your symbolic table is up to date and use the Monitor/Modify option in S7 lite to demonstrate your program.

Part D- Data logging.(10 Marks)
Add a data logging system where the temperature can be recorded in a data block over a period of time.
Make the reading every 0.4 seconds for testing purposes.
(You may use a SCADA system to perform this function using trends as an alternative to using a data block in the PLC , it is your choice.)
Using the PLC to record the data must include the use of a "pointer" for full marks.
See the S7 manual for information on Address registers (LAR 1)



Part E - Proportional Control.(10 Marks)
Add an improvement to your system to make the output smoother using proportional control.
You may use the integrated blocks for this purpose. (Pulsegen for example)
See the S7 Manuals for information on Con-C FB41 and Pulsgen FB43)

Part F- PID control.(10 Marks)
Using the integrated special function blocks to control the temperature using PID control.

Part G- Report.(10 Marks)
Add a report to your printed out program.
The report should include a block diagram explaining how your system works.
Full Documentation for your program including your Symbolic Table.
Use one shots for loading your data from the thumbwheel switches.
Document your program with line comments explaining the function of each part of your program.
Do your own work.
You **MUST** hand in this sheet with your assignment! Don't lose it !

OB1:CYCL_EXC

ON/OFF Temp' Control + DATA RECORDING EXAMPLE

Cycle Execution

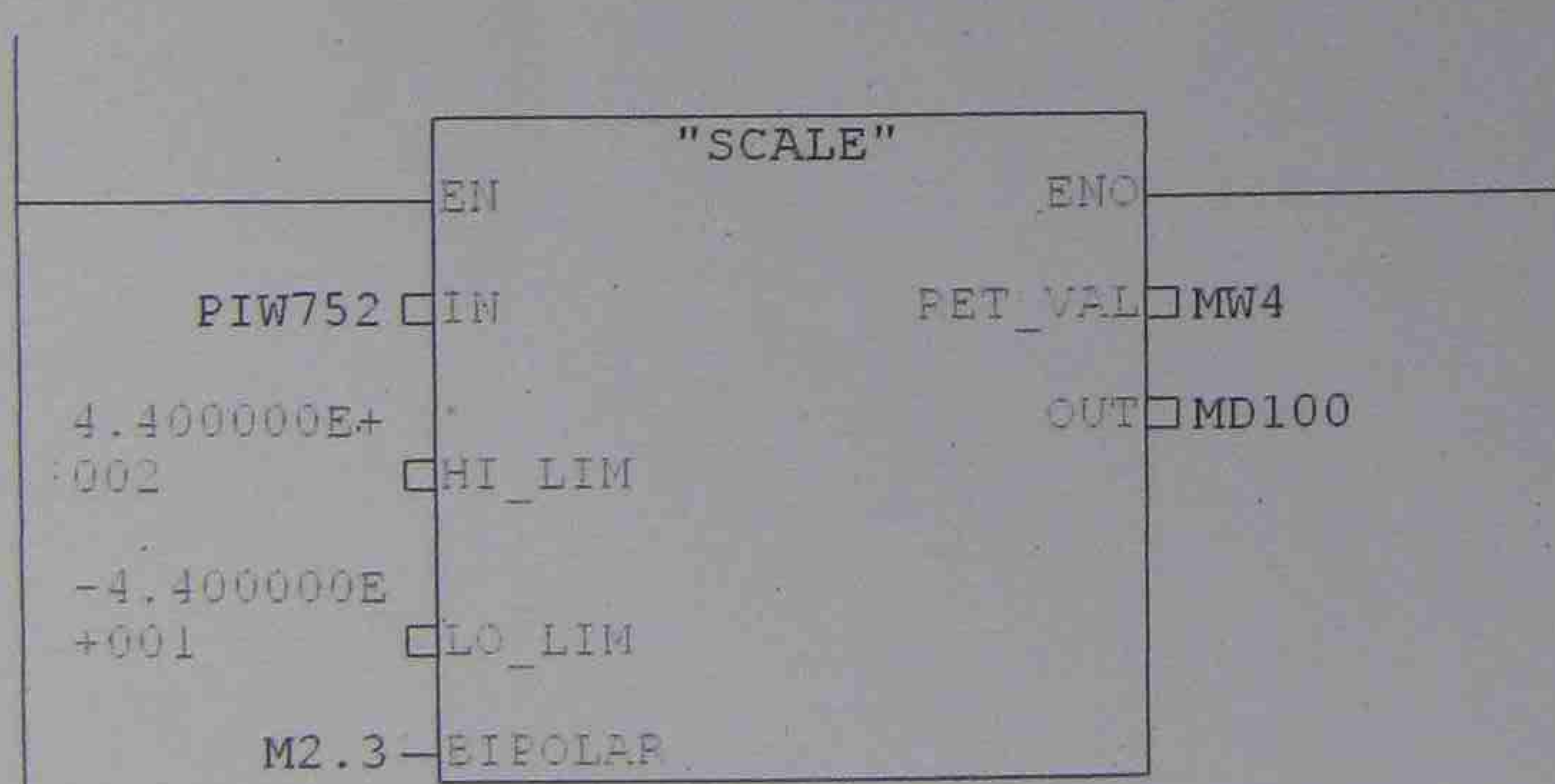
```
Name:                   Time stamp Code:        09/10/08
Author:                            Interface:   01/20/04
Family:                 Lengths    Block:       00422
Version:        0.0                Code:        00274
Code version: 2                    Data:        00028
```

Block: OB1 Temperature ON/OFF Control

Address	Declaration	Name	Type	Start value	Comment
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
1.0	temp	OB1_SCAN_1	BYTE		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
2.0	temp	OB1_PRIORITY	BYTE		1 (Priority of 1 is lowest)
3.0	temp	OB1_OB_NUMBR	BYTE		1 (Organization block 1, OB1)
4.0	temp	OB1_RESERVED_1	BYTE		Reserved for system
5.0	temp	OB1_RESERVED_2	BYTE		Reserved for system
6.0	temp	OB1_PREV_CYCLE	INT		Cycle time of previous OB1 scan (milliseconds)
8.0	temp	OB1_MIN_CYCLE	INT		Minimum cycle time of OB1 (milliseconds)
10.0	temp	OB1_MAX_CYCLE	INT		Maximum cycle time of OB1 (milliseconds)
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME		Date and time OB1 started

```
Network: 1      Analog Input - 4-20ma from RTD transmitter converted to a
1-5 vo
```

Scaled from -44 to 440 so that 1-5 Volts will correspond to a 0 to 198 degree temperature signal.



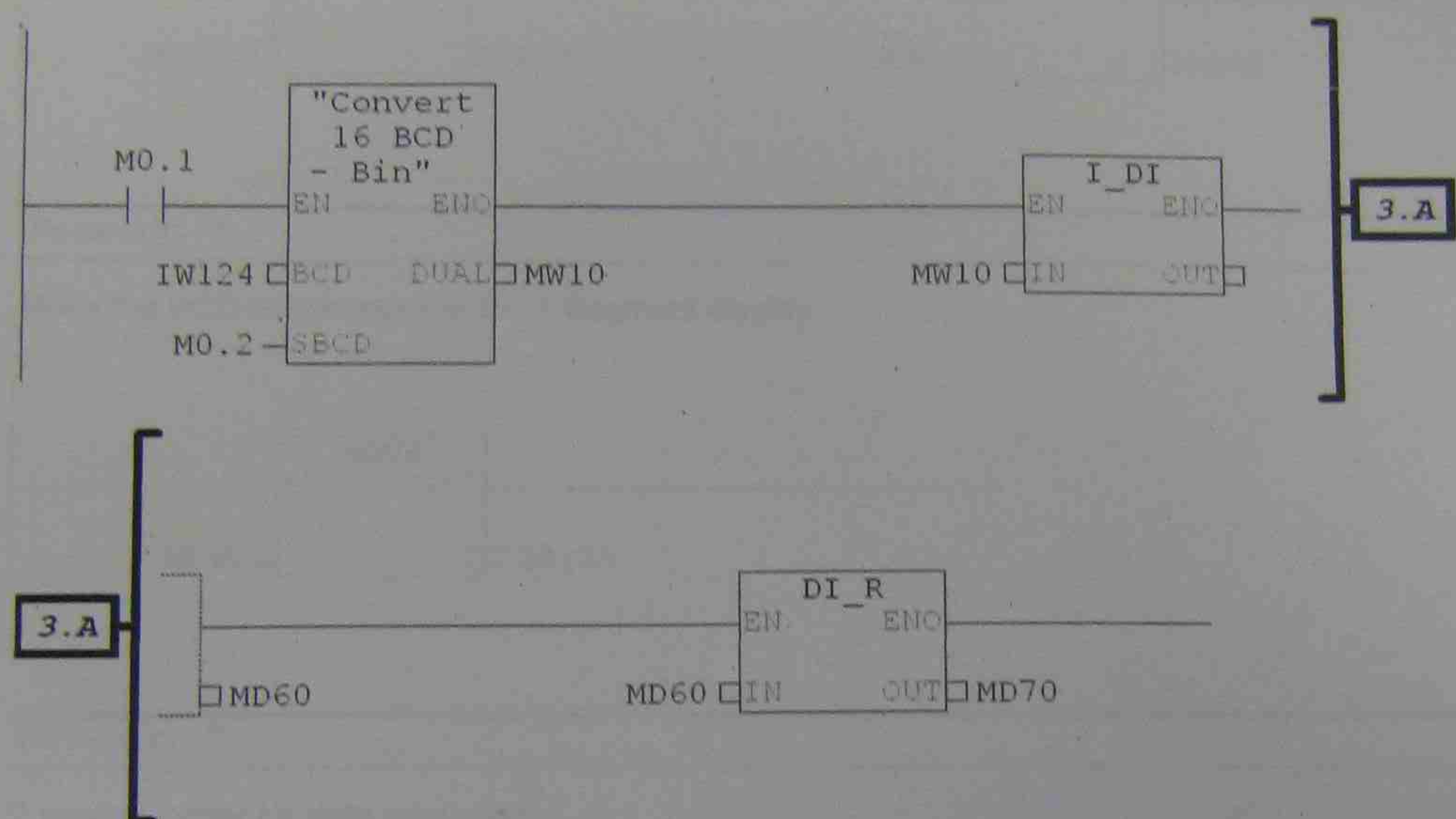
Network: 2

One Shot for setpoint loading

Network: 3 Load Setpoint to 32 bit number at MD20

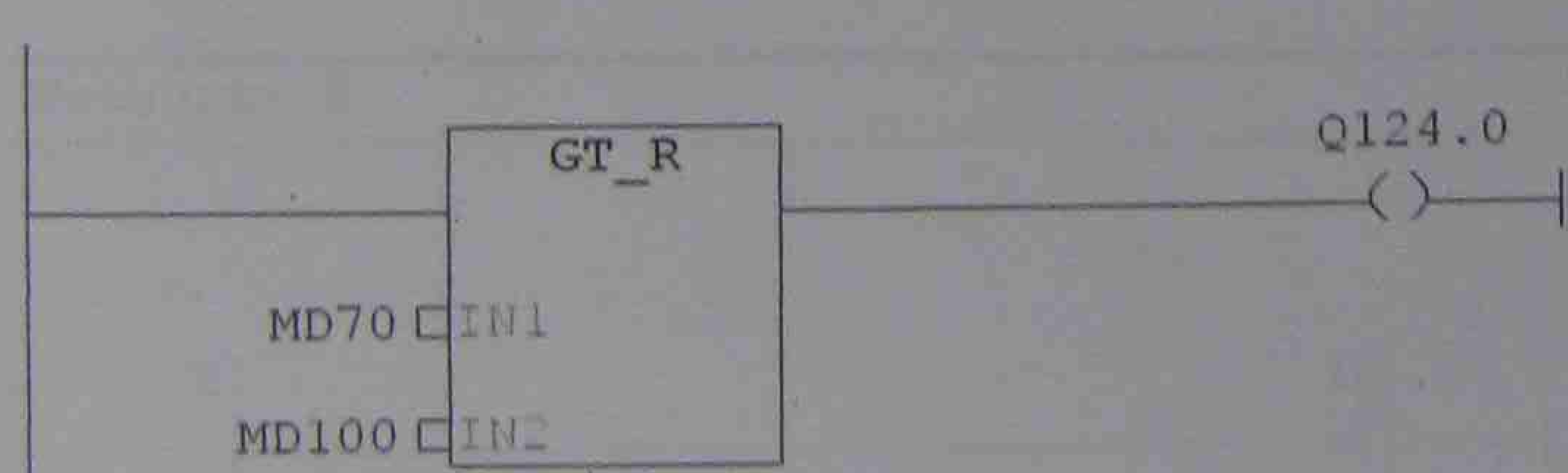
One shot
Convert IW124 from BCD into a Decminal 16 bit number
Then Convert the Integer into a 32 bit number (double Integer)
And then Convert into a real (floating point number)

We need the thumbwheel in the same format as the temperature for the comparison.



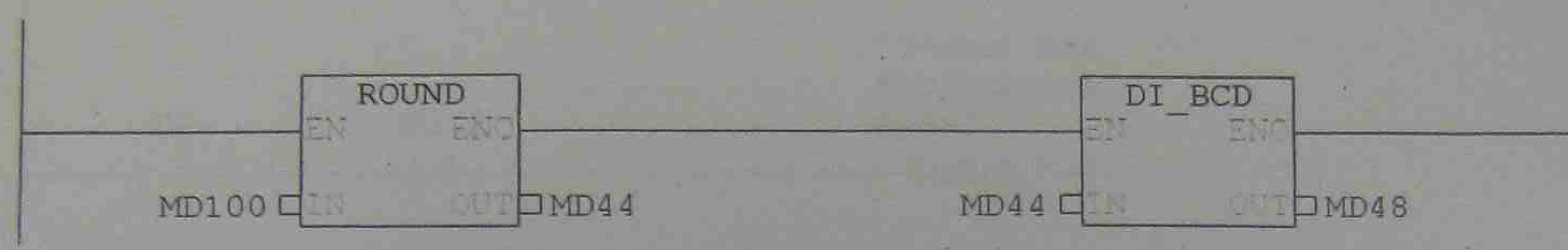
Network: 4

Compare setpoint to actual Temperature.
If the setpoint is greater then the ouput will turn on and the heater will be
on
in the oven.



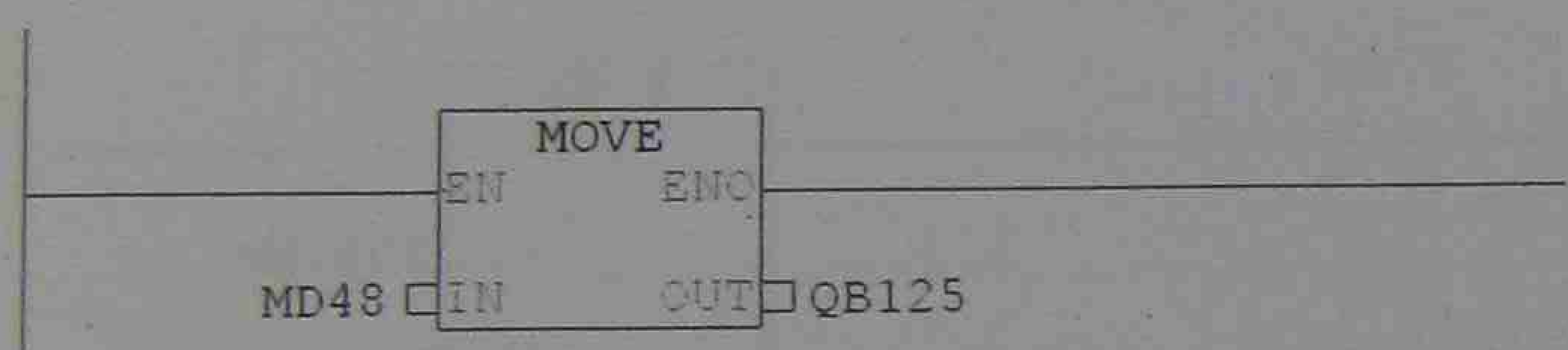
Network: 5

Get the temperature and round it off from a floating po9int number , then convert to a BCD number.



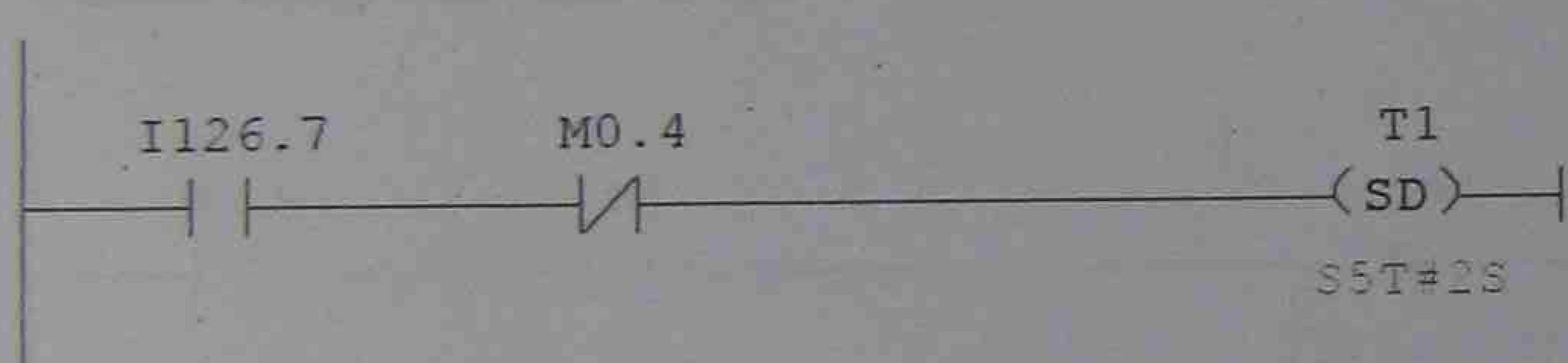
Network: 6

Move the BCD temperature to the 7 Segment display.

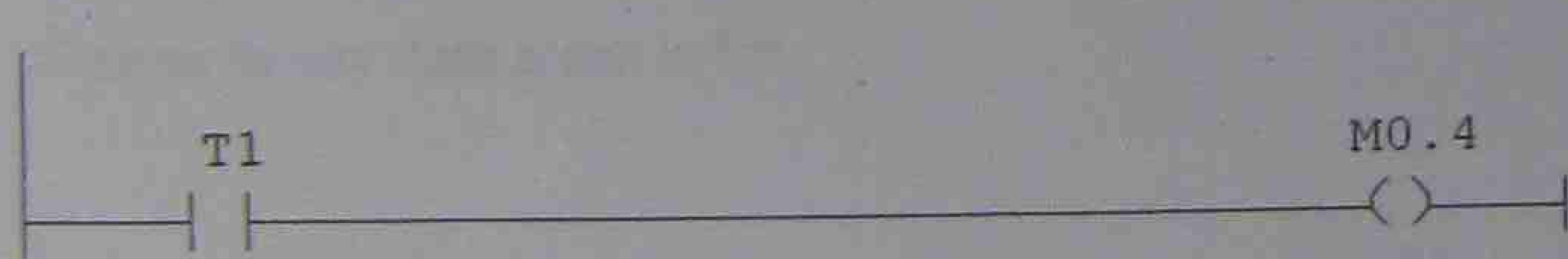


Network: 7

2 second cycler for data recording

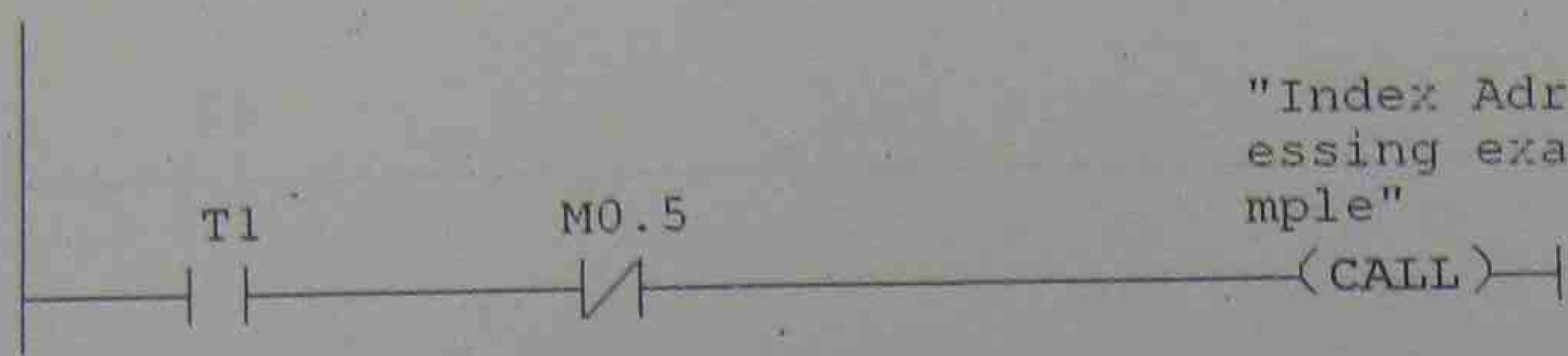


Network: 8



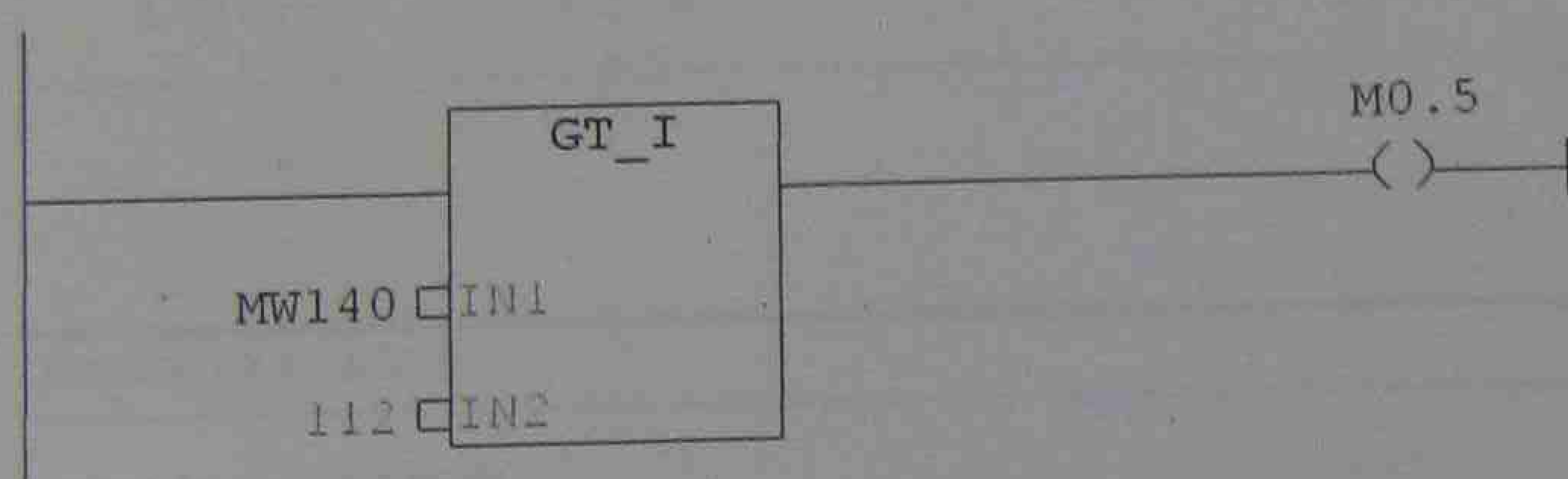
Network: 9

Jump to Indexed addressing block every 2 seconds



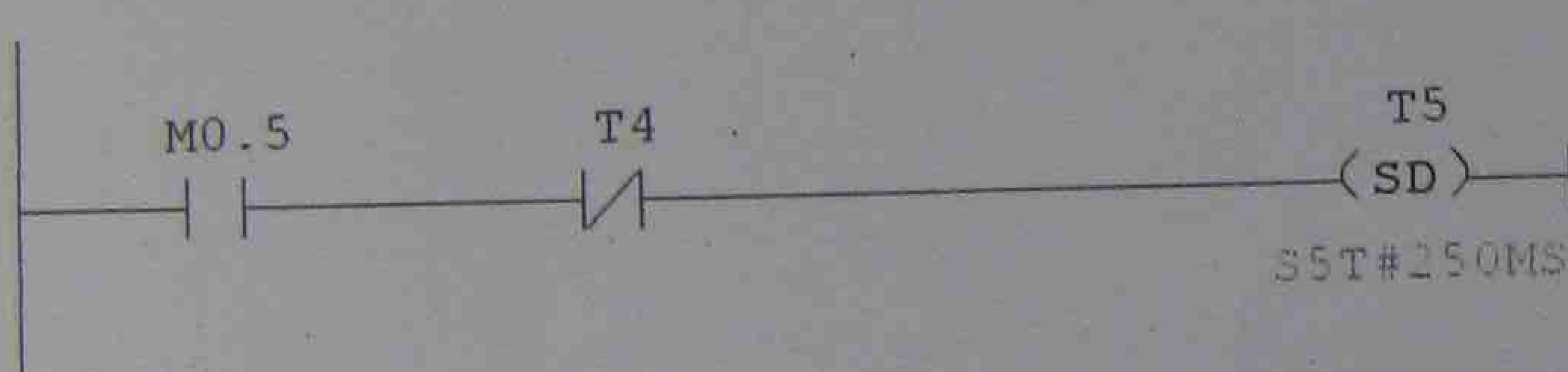
Network: 10

Comparator for when data block is full.



Network: 11

Flasher to say data block is full.

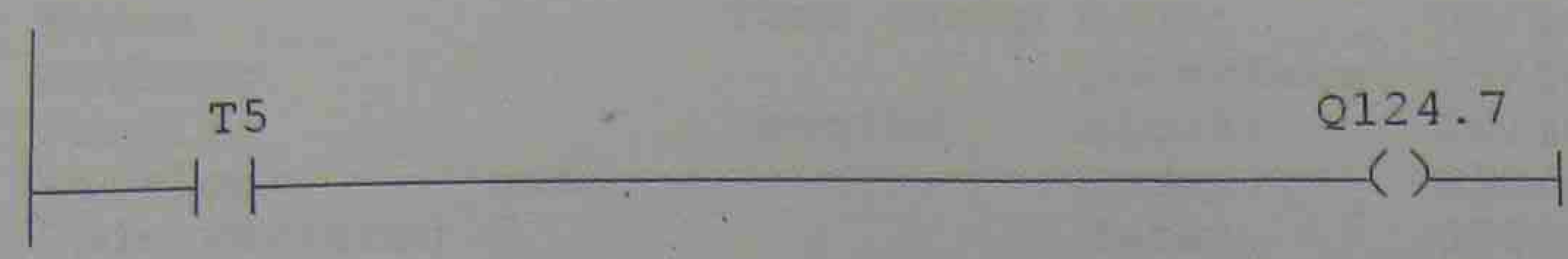


Network: 12

Flasher to say data block is full.

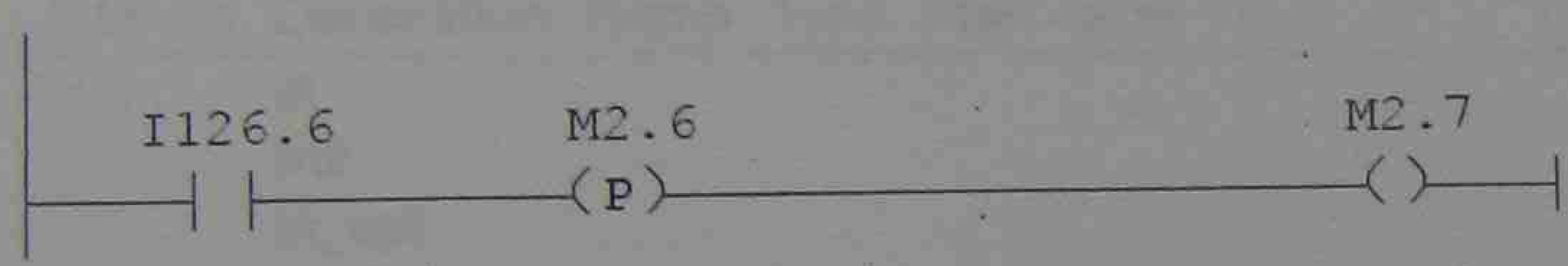
Network: 13

Flasher to say data block is full.



Network: 14

One shot for reset



Network: 15

Jump to reset block.

FC1:Index Adressing example

```
Name:                  Time stamp Code:       09/10/08
Author:                           Interface:  09/10/08
Family:                Lengths    Block:      00140
Version:       1.0                Code:       00046
Code version: 2                   Data:       00008
```

Block: FC1

Address	Declaration	Name	Type	Start value	Comment
	in				
	out				
	in_out				
	temp				

Network: 1

```
      OPN   "Data Recording"

      L     MW   140
      L     112
      >=I
      JC    here

      L     MW   140
      SLD   3
      LAR1

//load counter number into address reg      //load counter number
ister                                        into address register

      L     MD   ~~44~~ 100                    ~~//load rouned off temp~~

      T     DBW [AR1,P#0.0]

//load the temperature to the data wor      //load the temperature
d specified by the address register .        to the data word spec
                                            ified by the address r
                                            egister .

      L     MW   140
      L     4
      +I
      T     MW   140
```

```
// Increments the counter word by 4 so    // Increments the coun
 it will transfer the next available d    ter word by 4 so it wi
ata word.                                 ll transfer the next a
                                          vailable data word.


      SRU

here: L     0
      T     MW   140
```

FC2

```
Name:                  Time stamp Code:       09/10/08
Author:                           Interface:  09/10/08
Family:                Lengths    Block:      00218
Version:        1.0               Code:       00126
Code version: 2                   Data:       00008
```

Block: FC2 Reset

Address	Declaration	Name	Type	Start value	Comment
	in				
	out				
	in_out				
	temp				

Network: 1

```
      OPN   "Data Recording"

      L      0
      T     DBW    0
      T     DBW    4
      T     DBW    8
      T     DBW   12
      T     DBW   16
      T     DBW   20
      T     DBW   24
      T     DBW   28
      T     DBW   32
      T     DBW   36
      T     DBW   40
      T     DBW   44
      T     DBW   48
      T     DBW   52
      T     DBW   56
      T     DBW   60
      T     DBW   64
      T     DBW   68
      T     DBW   72
      T     DBW   76
      T     DBW   80
      T     DBW   84
      T     DBW   88
      T     DBW   92
      T     DBW   96
      T     DBW  100
      T     DBW  104
```

```
T      DBW   108
T      DBW   112
T      MW    200
```

FC81:Convert 16 BCD - Bin

Convert BCD to 16 bit binary

```
Name:          COD_B4      Time stamp Code:       01/14/03
Author:        AUT_1                  Interface:  01/14/03
Family:        S5_CNVRT    Lengths    Block:      00230
Version:       1.1                    Code:       00124
Code version: 2                       Data:       00014
Block protection
```

Block: FC81

Address	Declaration	Name	Type	Start value	Comment
0.0	in	BCD	WORD		
2.0	in	SBCD	BOOL		
4.0	out	DUAL	WORD		
	in_out				

FC105:SCALE

Scaling Values

```
Name:          SCALE      Time stamp Code:       01/14/03
Author:        SEA                   Interface:  01/14/03
Family:        CONVERT    Lengths    Block:      00336
Version:       2.1                   Code:       00208
Code version:  2                     Data:       00030
Block protection
```

Block: FC105 SCALING VALUES

Address	Declaration	Name	Type	Start value	Comment
0.0	in	IN	INT		input value to be scaled
2.0	in	HI_LIM	REAL		upper limit in engineering units
6.0	in	LO_LIM	REAL		lower limit in engineering units
10.0	in	BIPOLAR	BOOL		1=bipolar; 0=unipolar
12.0	out	RET_VAL	WORD		
14.0	out	OUT	REAL		result of the scale conversion
	in_out				

DB10:Data Recording

```
Name:                    Time stamp Code:       09/10/08
Author:                             Interface:  09/10/08
Family:                  Lengths    Block:      00298
Version:       1.0                  Code:       00140
Code version: 2                     Data:       00080
```

Block: DB10

REAL

Address	Declaration	Name	Type	Start value	Comment
0.0	stat		STRUCT		
+0.0	stat	Temperature	DWORD	DW#16#0	
+4.0	stat	Temperature1	DWORD	DW#16#0	
+8.0	stat	Temperature2	DWORD	DW#16#0	
+12.0	stat	Temperature3	DWORD	DW#16#0	
+16.0	stat	Temperature4	DWORD	DW#16#0	
+20.0	stat	Temperature5	DWORD	DW#16#0	
+24.0	stat	Temperature6	DWORD	DW#16#0	
+28.0	stat	Temperature7	DWORD	DW#16#0	
+32.0	stat	Temperature8	DWORD	DW#16#0	
+36.0	stat	Temperature9	DWORD	DW#16#0	
+40.0	stat	Temperature10	DWORD	DW#16#0	
+44.0	stat	Temperature11	DWORD	DW#16#0	
+48.0	stat	Temperature12	DWORD	DW#16#0	
+52.0	stat	Temperature13	DWORD	DW#16#0	
+56.0	stat	Temperature14	DWORD	DW#16#0	
+60.0	stat	Temperature15	DWORD	DW#16#0	
+64.0	stat	Temperature16	DWORD	DW#16#0	
+68.0	stat	Temperature17	DWORD	DW#16#0	
+72.0	stat	Temperature18	DWORD	DW#16#0	
+76.0	stat	Temperature19	DWORD	DW#16#0	
+80.0	stat	Temperature20	DWORD	DW#16#0	
+84.0	stat	Temperature21	DWORD	DW#16#0	
+88.0	stat	Temperature22	DWORD	DW#16#0	
+92.0	stat	Temperature23	DWORD	DW#16#0	
+96.0	stat	Temperature24	DWORD	DW#16#0	
+100.0	stat	Temperature25	DWORD	DW#16#0	
+104.0	stat	Temperature26	DWORD	DW#16#0	
+108.0	stat	Temperature27	DWORD	DW#16#0	
+112.0	stat	Temperature28	DWORD	DW#16#0	
+116.0	stat	Temperature29	DWORD	DW#16#0	
+120.0	stat	Temperature30	DWORD	DW#16#0	
+124.0	stat	Temperature31	DWORD	DW#16#0	
+128.0	stat	Temperature32	DWORD	DW#16#0	
+132.0	stat	Temperature33	DWORD	DW#16#0	
+136.0	stat	Temperature34	DWORD	DW#16#0	
=140.0	stat		END_STRUCT		

PLC System Applications Major Project Report

Temperature Control with a Step 7 PLC

By David Holbeche SID: 311383922

Aim:

The aim of this major project is to assess students' ability to apply their knowledge to a new system. Building on previous knowledge the students will adapt this to a newer model PLC and software.

The sub-aims of this major project are to set up and program four types of control systems using a step 7 PLC, to accurately control the temperature in an oven.

These are :

1. On/Off control
2. Hysteresis control
3. Proportional Control
4. PID control

These correspond to different parts of the major project which are

Part A : The aim of which is to learn to use an analog input to the PLC.

Part B : The aim of which is to set up an On/Off controller.

Part C : The aim of which is to set up a hysteresis On/Off controller.

Part D : The aim of which is to set up data logging for the system.

Part E: The aim of which is to implement a proportional controller.

Part F : The aim of which is to implement a full PID controller.

Procedure:

Part A - Analog Inputs

_We are using an RTD to monitor the temperature of the oven (The Feedback). This RTD is connected to a 4-20 mA current loop which is connected to a temperature transmitter. This is connected via a 250 Ohm precision resistor to the 24 V inputs on the PLC, and to the PLC analog input (PIW 752). This feeds the PLC's analog input with a 1 – 5 V signal.

The RTD input is to be scaled using the "SCALE" (FC105) function block, which is to be scaled from -44 to 440 so that 1-5 V at the analog input will correspond to a 0 to 198 degree temperature signal. The temperature is then to be displayed in BCD on the 7-segment display.

Part B – On/Off Control

The scaled analog input is to be used as a feedback signal to control the temperature in the oven. The thumbwheel switch is used to enter the setpoint.

The digital output of the PLC will provide 5 V to a relay when the output is "On", which will make the relay contacts close and provide 24 V to the oven. The RTD will monitor the temperature of the oven and provide feedback to the controller.

Part C – Hysteresis control

Some hysteresis is to be added to the control system. The Hysteresis temperature is set by the thumbwheel switch which is activated by a one shot.

Part D – Data logging

A data logging system is to be added in which the temperature can be recorded in a data block over a period of time. The reading should be taken every 0.4 seconds for testing purposes.

Part E – Proportional Control

An improvement is to be added to the system to make the output smoother using proportional control. The integrated function blocks Con-C (FB41) and Pulsegen (FB43) can be used to achieve this outcome.

Part F – PID Control

Using the integrated special function blocks the temperature is to be controlled using PID control.

Explanation of Operation:

Part A – Analog Inputs

We are using an RTD to monitor the temperature of the oven (The Feedback). This RTD is connected to a 4-20 mA current loop which is connected to a temperature transmitter. This is connected via a 250 Ohm precision resistor to the 24 V inputs on the PLC, and to the PLC analog input (PIW 752). This feeds the PLC's analog input with a 1 – 5 V signal.

The RTD input is scaled using the "SCALE" (FC105) function block, which is scaled from -44 to 440 so that 1-5 V at the analog input will correspond to a 0 to 198 degree temperature signal. Scaling is needed so that the 1-5 V signal at the PLC's input works with the PLC's 0 to 10 V analog input. The system is calibrated so that 4 mA corresponds to 1 V or 0 degrees Celsius, and 20 mA corresponds to 5 V or 198 degrees Celsius. **See the attached diagram on the operation of scaling.**

The temperature is then converted to BCD and displayed on the 7-segment display.

<u>Part B - On/Off Control</u>

An On/Off controller is the simplest form of temperature control device. The output of the On/Off controller is either On or Off, with no middle state. An On/Off controller will switch the output on or off when the temperature (Feedback) crosses the setpoint. For our temperature control purposes, the output to the oven is fully on (digital output) when the temperature is below the setpoint, and off when the temperature is above the setpoint. This is not an ideal form of temperature control since the temperature or process variable will be cycling continually , going from below the setpoint to above then below again.

For our purposes the setpoint is set by the left two digits of the thumbwheel switch via a one-shot.

Please see the attached Block Diagram.

<u>Part C – Hysteresis Control</u>

Hysteresis is added to the control system to improve the system's performance. On/Off control can cause problems when the temperature cycles quickly below and above the setpoint. This can damage contactors and alves, or cause contact chattering when there is noise at the output.

"Hysteresis" is achieved by adding an On/Off differential to the setpoint. This requires that the temperature exceed the setpoint by a certain amount before the output will turn off or on again. This differential prevents the output from "chattering" or making fast, continual switches when the cycling above and below the setpoint occurs very rapidly. The hysteresis differential is added and subtracted from the setpoint to create an *Upper-trip-point*, and *a lower-trip-point* , above and below the setpoint. Once the temperature reaches the *Upper-trip-point* the output will turn off, and it will not turn back on until the temperature drops to the *lower-trip point* again.

Hysteresis control is similar to On/Off control, as the output is either fully on or fully off : it improves the performance of On/Off control but must not be confused with continuous control.

For our purposes the hysteresis temperature is set by the right two digits of the thumbwheel switch via a one-shot.

Please see the attached diagram for the characteristics of hysteresis control.

Part D – Data logging

The actual temperature in the oven is recorded in data block 5 every 0.5 seconds. The data block stores 30 readings. When the 30 readings have been taken, an output at Q124.6 will flash. An input at I126.2 will reset the data block values to zero.

A 2 second cycler was used, so that the program will jump to an STL function block every 2 seconds. When 30 readings have been taken, a comparator will turn a flag on, and this flag will prevent the program from jumping to the STL function block when it is set. When this flag is set, a flasher will turn on, to indicate the data block is full. A one-shot will be activated when switch I126.6 is set to on and this one-shot will jump to a "reset" function block which resets the data block to zero.

The STL function block contains the program code which performs the data recording function. The memory word MW140 is used to keep track of the count. When the count reaches 112, the program jumps to "here", which contains the code to reset MW140 to zero. The counter number is loaded into the address register with the program code:

```
L       MW 140

SLD   3

LAR1
```

The analog input at MD 100 is loaded into the accumulator and transferred to a data block double word specified by the address register, with the *T DBD [AR1,P#0.0]* instruction. The counter specified by MW140 is loaded into the accumulator, incremented by 4 , then transferred back to MW140. This is so the program transfers the temperature reading to the next available data word. This is done with the code:

```
L       MW 140

L       4

+I

T       MW140
```

Part E – Proportional Control

Proportional Control is an improvement to the system to make the output smoother, and more stable (reduced oscillations around the setpoint). The Integral function blocks Con-C FB41 and Pulsegen FB43 are used for this purpose. These controller blocks implement a software controller which performs a full or partial PID function. The controller created consists of a series of subfunctions which can be activated or deactivated to suit the process being controlled.

Cont- C is used for controlling processes with continuous input and output variables. The FB implements a full PID controller with continuous manipulated variable output. For the purpose of this Project, only a few of the sub-functions are used to make the system less complicated, making problems easier to debug.

Both Cont-C and pulse-gen are located in OB35, a cyclic interrupt function block, with the Cont-C FB being called at a rate equal to the "cycle" time configured in Cont-C. That is the scan cycle I interrupted at a fixed rate and OB35 is executed, causing Cont-C and Pulse-gen to be updated and executed.

Please see the attached block diagram on the operation of Cont-C.

The input parameters of Cont-C are configured as follows:

P_SEL is connected to switch I126.7

I_SEL is connected to switch I126.6

D_SEL is connected to switch I126.5

These switches allow the system to be configured as a proportional only controller, or as a full PID controller.

For the proportional only controller, only I126.7 is switched on.

CYCLE is set to 10mS . This sampling time specifies the time between block calls.

SP_INT is set to MD174 and is the setpoint as a real number

PV_IN is set to MD100 and is the feedback as a real number.

GAIN is set to MD178 and is the gain as a real number.

LMN_HLM is set to 100.0 and is the MANIPULATED VALUE HIGH LIMIT : this is the upper limit of the scaled manipulated value.

LMN_LLM is set to 0.0 and is the MANIPULATED VALUE LOW LIMIT : this is the lower limit of the scaled manipulated value.

All other input parameters are assigned dummy flags initialised to "0" so they are not used.

The output parameters of Cont-C used are:

LMN is assigned to MD 200 and is the manipulated value, which is sent to the Pulse-gen input.

LMN_P is assigned to MD210 and is the proportional only output component.

LMN_I is assigned to MD214 and is the Integral only output component.

LMN_D is assigned to MD218 and is the derivative only component.

PV is assigned to MD222 and is the process variable, or feedback.

ER is assigned to MD226 and is the effective error of the system.

Please see the attached table outlining the input and output parameters of Cont-C and Pulsegen.

PulseGen, SFB43 is used to structure a PID controller with a pulse output. For our purpose, the output we use is controlled with ON/OFF control, rather than continuous control. Pulse-duration modulation is used to provide an amount of power to the output which is proportional to the error of the system. The PULSEGEN function transforms the input variable INV, by modulating the pulse duration into a pulse train with constant period, corresponding to the cycle time at which the input variable is updated, and must be assigned to PER_TM. The duration of a pulse per period is proportional to the input variable INV. The cycle assigned to PER_TM is not identical to the processing cycles of PULSEGEN. The PER_TM cycle is made up of several processing cycles of PULSEGEN. The number of PULSEGEN calls per PER_TM cycle is a measure of the accuracy of the pulse generation modulation. In other words, the period of the total cycle of the pulse(On, and Off time) , is made up of many PULSEGEN cycles, and the more PULSEGEN cycles per PER_TM, the higher the resolution and accuracy of the Pulse duration modulation. For example, with INV at 30% , and 10 PULSEGEN calls per PER_TM period, the output QPOS will be "one" for the first three calls of PULSEGEN, and "zero" for the last seven calls.

The LMN manipulated value output from Cont-C is connected to the INV input of Pulse-gen, and the pulsegen output is connected to our Final control element, the heater.

The input parameters of Pulse-gen are configured as follows:

INV is assigned to MD200 and is the input value, from 0 to 100%: this is the manipulated value from Cont-C.

PER_TM is set to 2s and is the period time of the pulse. This corresponds to the sampling time of the controller. The ratio between the sampling time of the pulse generator and the sampling time of the controller determines the accuracy of the pulse duration modulation.

RATIOFAC is set to 1.0. This is the default value and is not changed.

CYCLE is set to 100mS and is the SAMPLING TIME. The time between block calls must be constant, and the "sampling time" input specifies the time between block calls.

All other input parameters are assigned to dummy flags which are set to 0.

The output parameters of Pulse-gen are:

QPOS_P which is assigned to Q124.7, the oven output. It is also connected to a light output so the On time of the pulse can be observed. This is the OUTPUT POSITIVE PULSE, and is set to "one" when a pulse is to be output.

QNEG_P which is assigned to Q124.6, a light output . This is used for testing purposes so the Off time of the pulse can be observed.

Part F- PID Control

To implement a full PID controller, all three switches I126.7, I126.6 and I126.5 are switched on. Cont-C and Pulsegen function blocks are used, as with Proportional only control, except a few more input parameters are used:

TI is set to 10s. This is the reset time and determines the time response of the integrator. It is the time taken for the integral output to change by the same amount the proportional output did.10s is a reasonably large Ti: for our purposes this is helpful as it allows us to observe the change in output due to integral action more easily. A higher integral time means the integral action responds to the error more slowly, and the correction is slow. For a small Ti , the correction of the error is rapid, but overshoot may occur.

TD is set to 1s. This is the derivative time, and determines the time response of the derivate unit. The derivative time is the amount of time that the derivative action apparently advances the output due to the proportional action. In our case, the output due to proportional control only is advanced by 1s. A higher derivative time means a greater correction due to the derivative action. A smaller derivative time means a smaller correction, and a smaller derivative effect.

TM_LAG is set to 2s. This is the time lag of the derivative action.

The PID controller is the most effective form of control we have dealt with in this project.

Discussion of Results:

On/Off control is the simplest form of control. It is most suited to processes where precise accurate control is not needed, or slow moving processes such as temperature control. The output of the On/Off controller cycled around the setpoint, which is not an ideal output response, particularly when there is noise at the output which can lead to contact chattering.

To improve the controller's performance Hysteresis was added to the On/Off controller. This improved the performance of the controller and decreased the cycling frequency around the setpoint. Both of these forms of control do not provide a great deal of regulation of the process being controlled, as they do not react to the error in the system.

A further improvement is the addition of proportional control. This form of control provides a much more accurate output response and regulation of the process being controlled.

Please refer to the output trend for the proportional controller.

The output from the proportional controller does not oscillate around the setpoint, as it does with On/Off control.

The output trend showed a much more accurate response, with the output overshooting the setpoint one or two times, then settling below the setpoint with an error, or offset of 3.95 degrees. This is a reasonably accurate control of the temperature in the oven.

The final controller implemented is the PID controller which provides a correction to the inherent error in proportional only control. The output trend overshot the setpoint once or twice. For a change in setpoint, the integral action ramped up to correct the increase in error. Once the error became zero, the integral action stopped ramping. The output trend obtained was very accurate, with the output settling with virtually no offset, with an error of -0.26 degrees.

Please refer to the output trend for the PID controller.

Conclusion:

The PID controller implemented provides the most effective control over the process. It has three separate, but additive actions, proportional, Integral and derivative which work together to regulate the Process Variable. The output trend obtained is by far the best out of all the forms of controller implemented, with virtually zero error or offset, and the temperature being almost flat at the setpoint. This is the desired operation of a control system, as the residual error is almost zero, and the maximum value of the error is corrected by the integral action.

OB1:CYCL_EXC

Cycle Execution

```
Name:                 Time stamp Code:       23/06/09
Author:                          Interface:  20/01/04
Family:               Lengths    Block:      00520
Version:       0.0               Code:       00372
Code version: 2                  Data:       00028
```

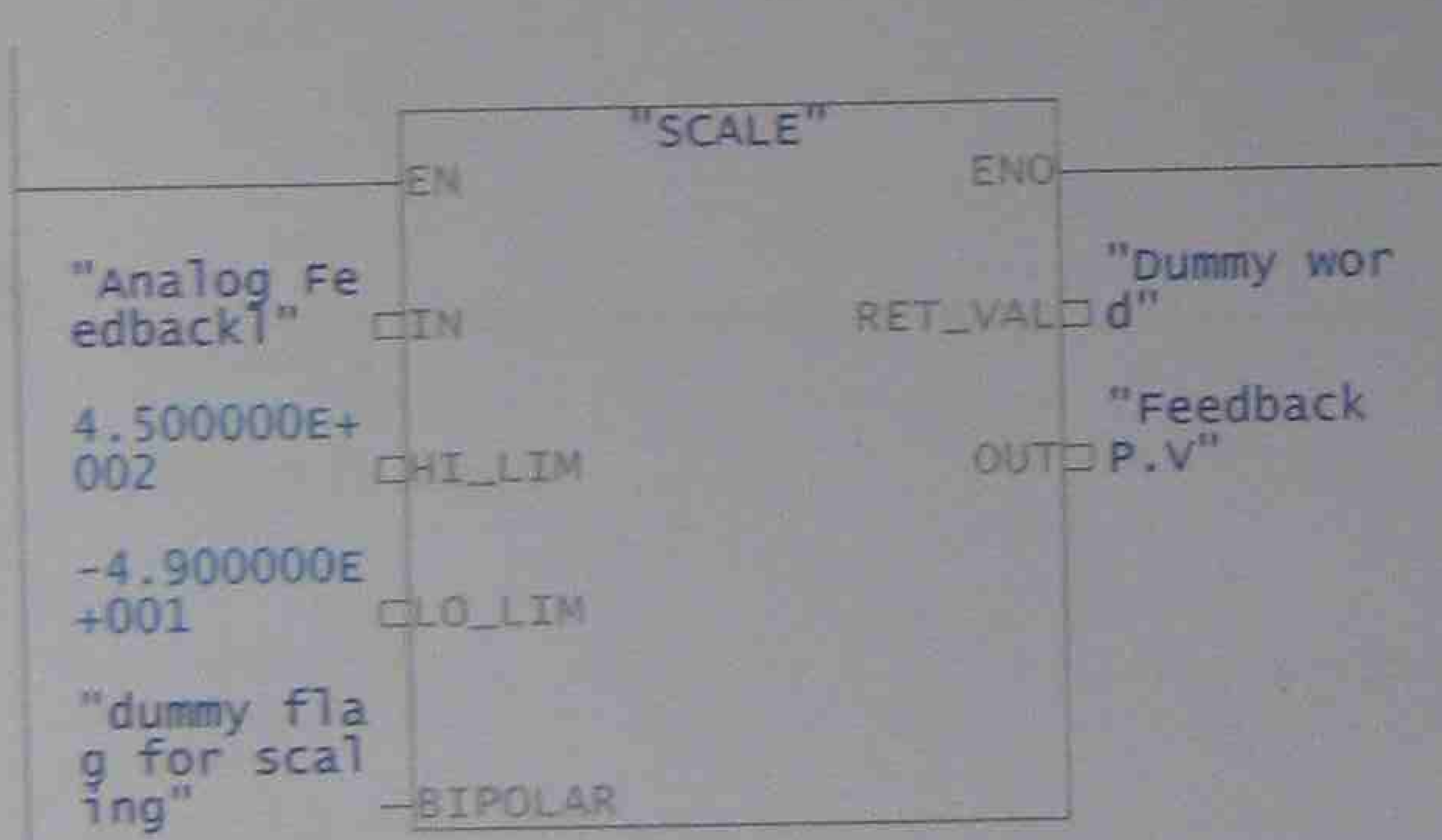
Block: OB1 "Main Program Sweep (Cycle)"

get analog input and scale it between -4.9 and 450, get setpoint and gain from thumbwheel switch and convert them into real numbers.

Address	Declaration	Name	Type	Start value	Comment
0.0	temp	OB1_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
1.0	temp	OB1_SCAN_1	BYTE		1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
2.0	temp	OB1_PRIORITY	BYTE		1 (Priority of 1 is lowest)
3.0	temp	OB1_OB_NUMBR	BYTE		1 (Organization block 1, OB1)
4.0	temp	OB1_RESERVED_1	BYTE		Reserved for system
5.0	temp	OB1_RESERVED_2	BYTE		Reserved for system
6.0	temp	OB1_PREV_CYCLE	INT		Cycle time of previous OB1 scan (milliseconds)
8.0	temp	OB1_MIN_CYCLE	INT		Minimum cycle time of OB1 (milliseconds)
10.0	temp	OB1_MAX_CYCLE	INT		Maximum cycle time of OB1 (milliseconds)
12.0	temp	OB1_DATE_TIME	DATE_AND_TIME		Date and time OB1 started

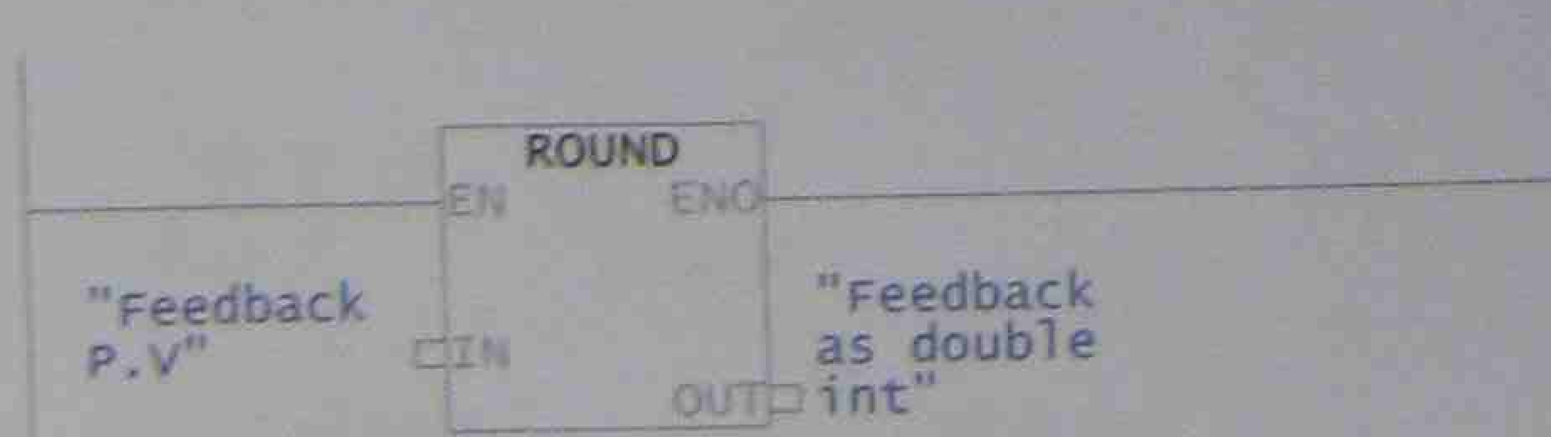
Network: 1

Scaling function block



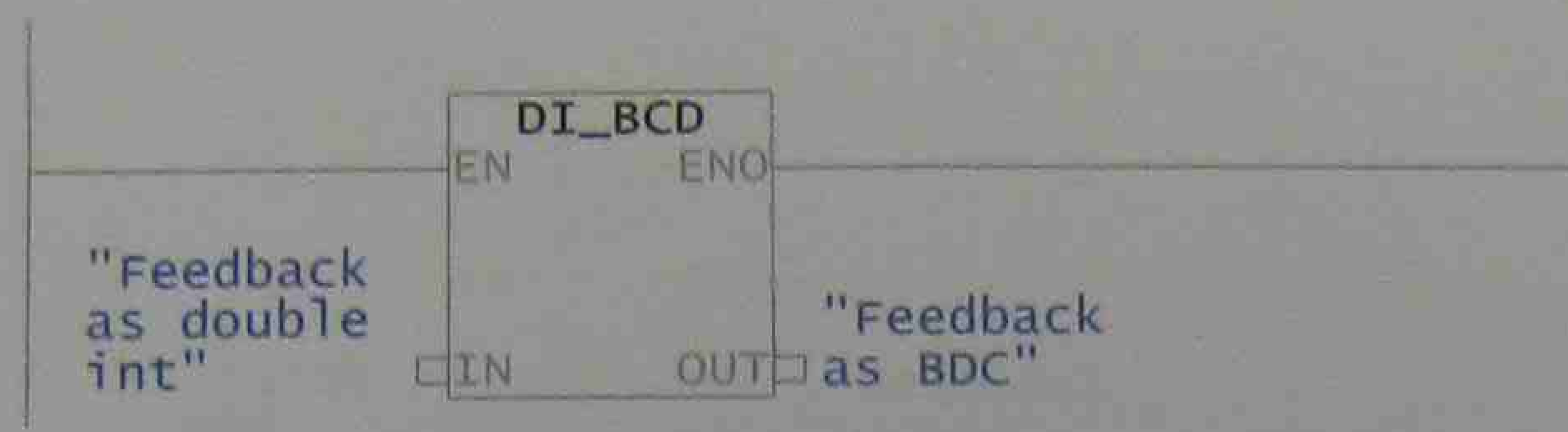
Network: 2

converts double floating point to double int



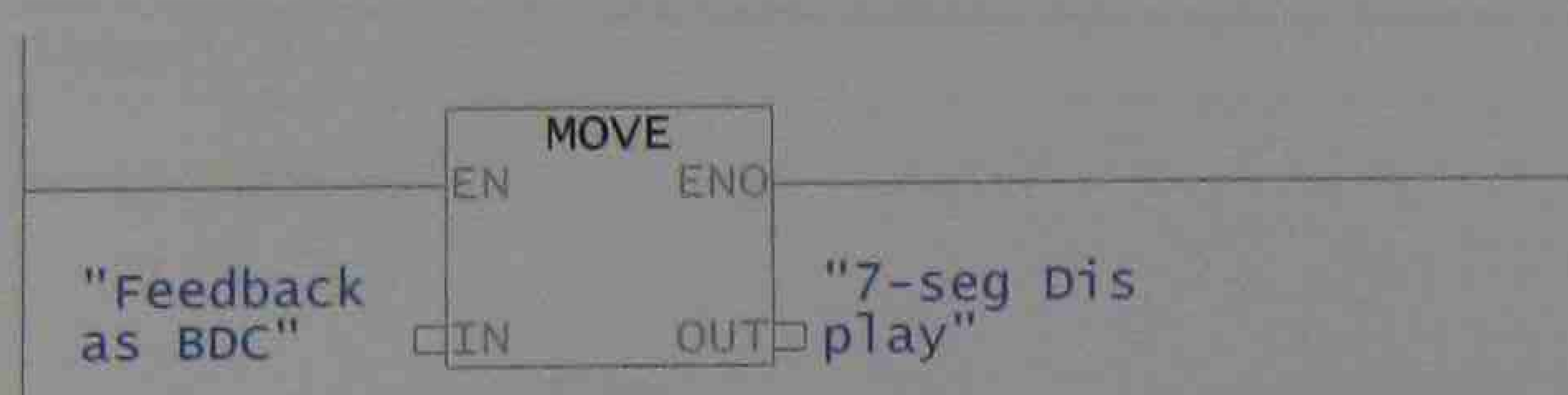
Network: 3

converts doubke integer to BCD



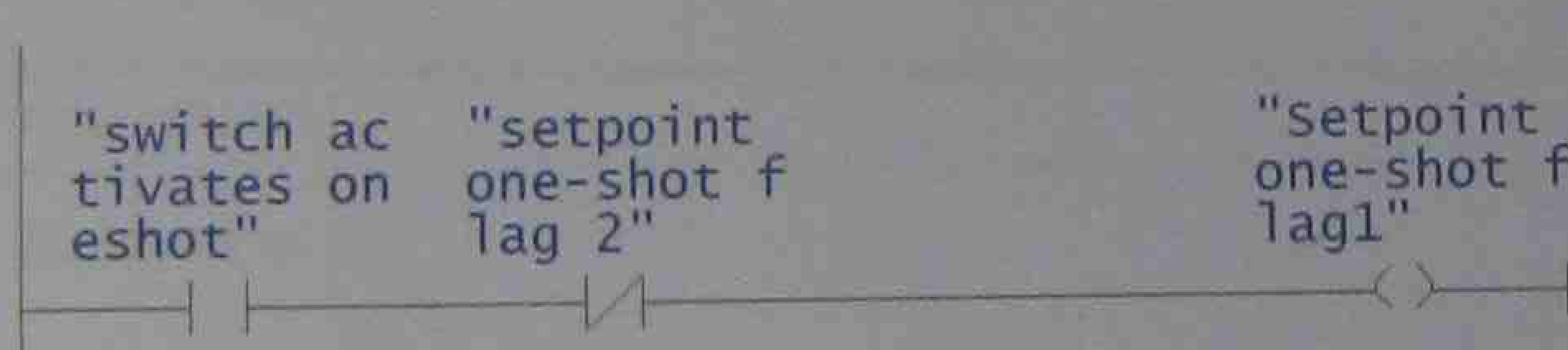
Network: 4

double BCD to 7-segment display



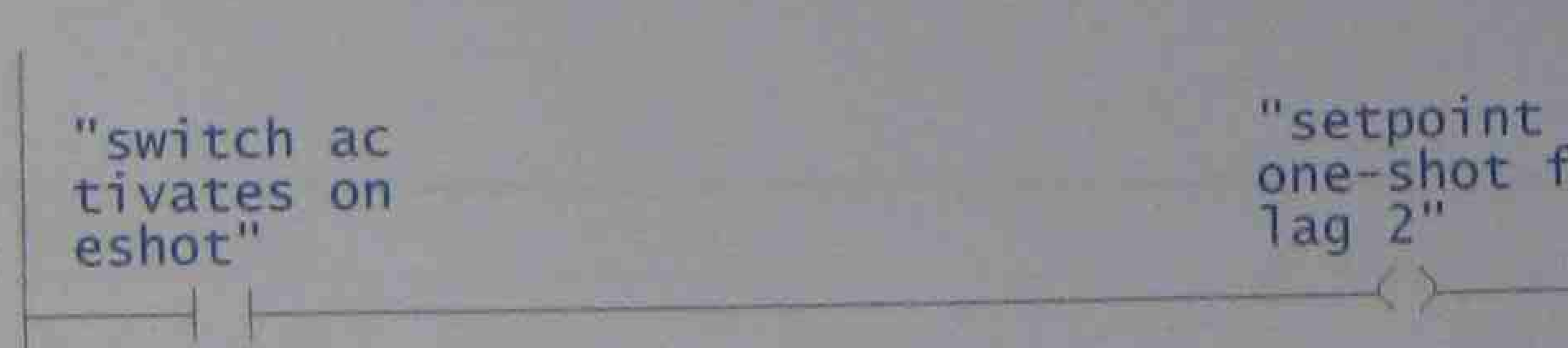
Network: 5

one shot for setpoint



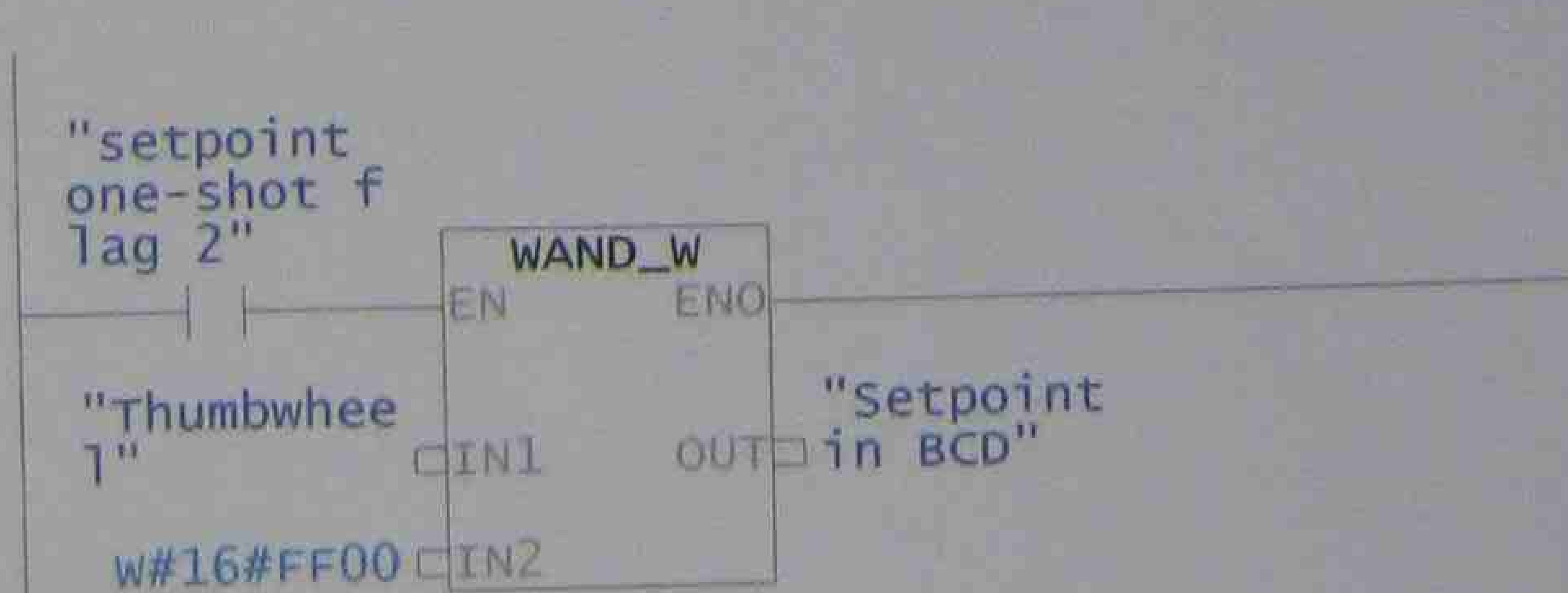
Network: 6

one shot for setpoint



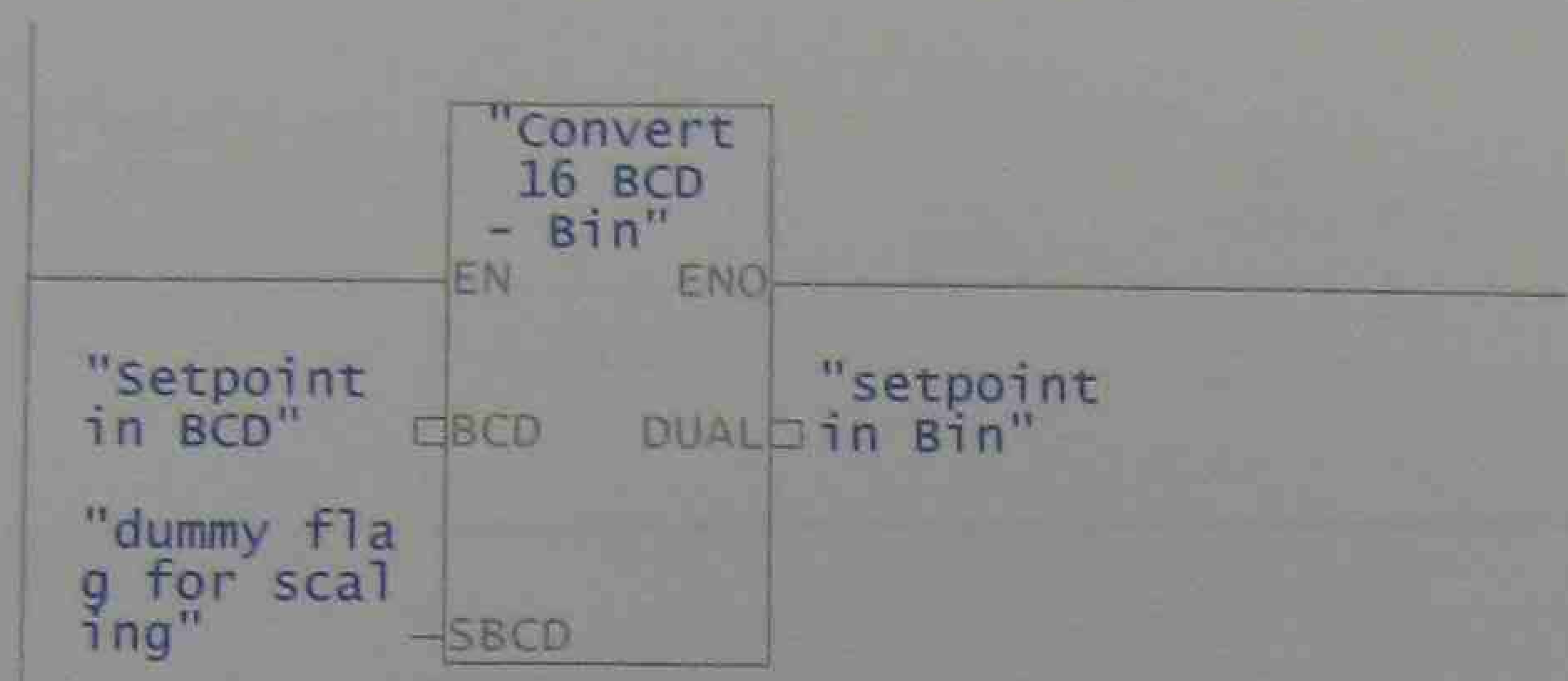
Network: 7

get thumbwheel data for setpoint



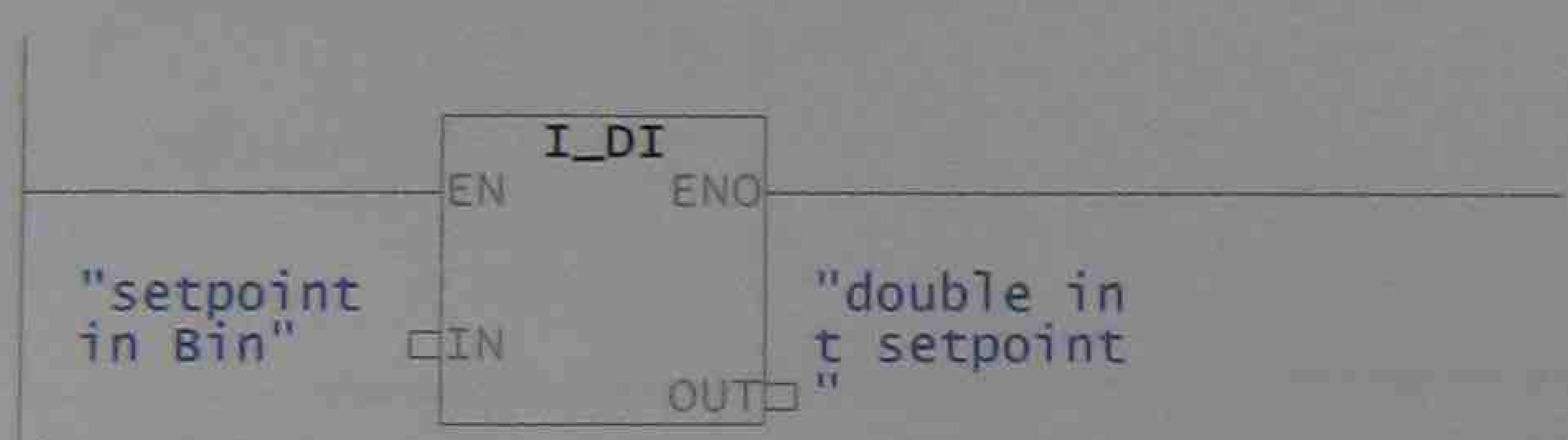
Network: 8

BCD - binary



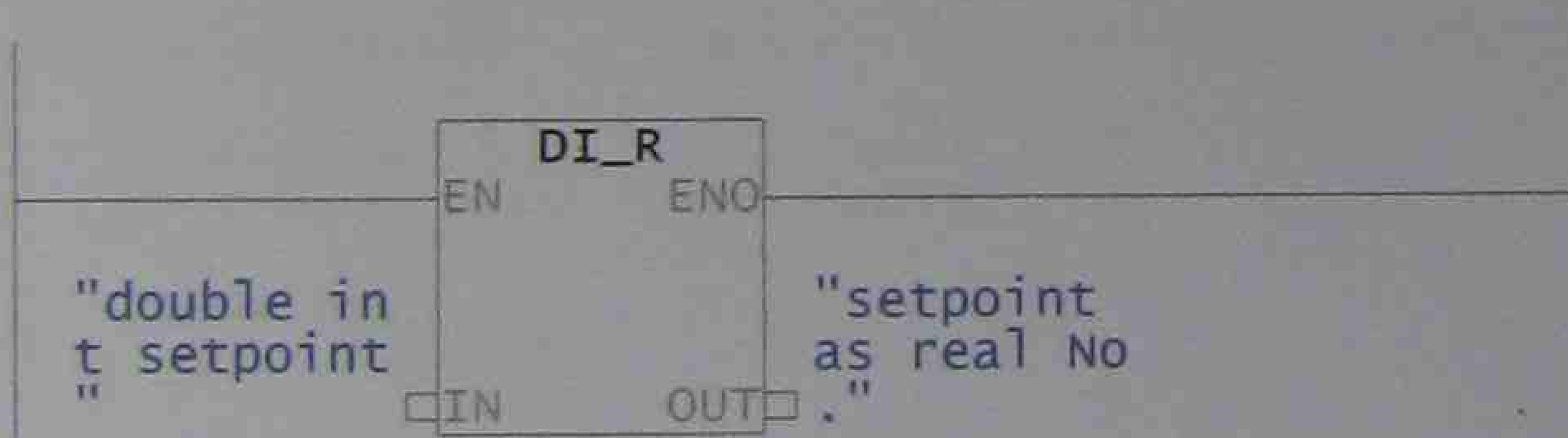
Network: 9

bin to double int



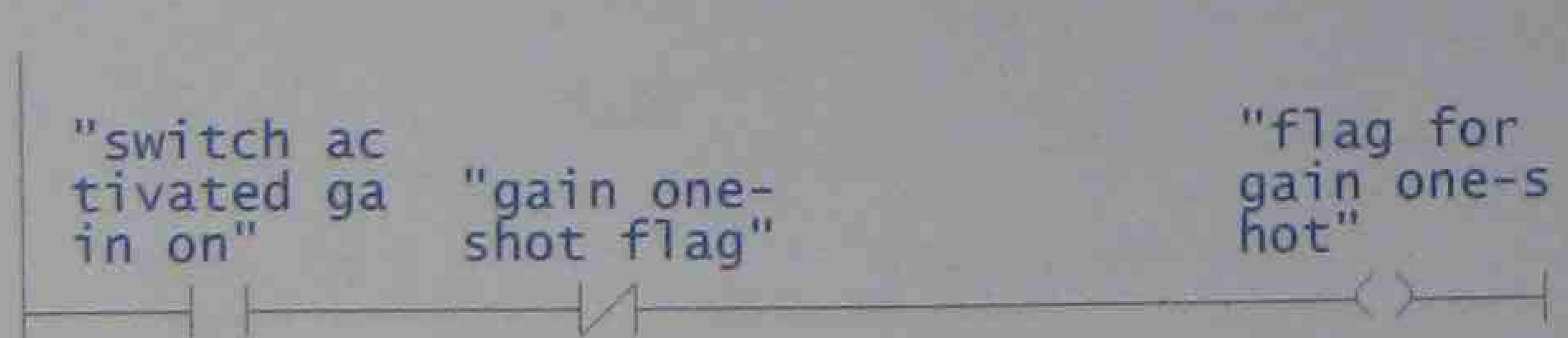
Network: 10

double int to real



Network: 11

gain one shot

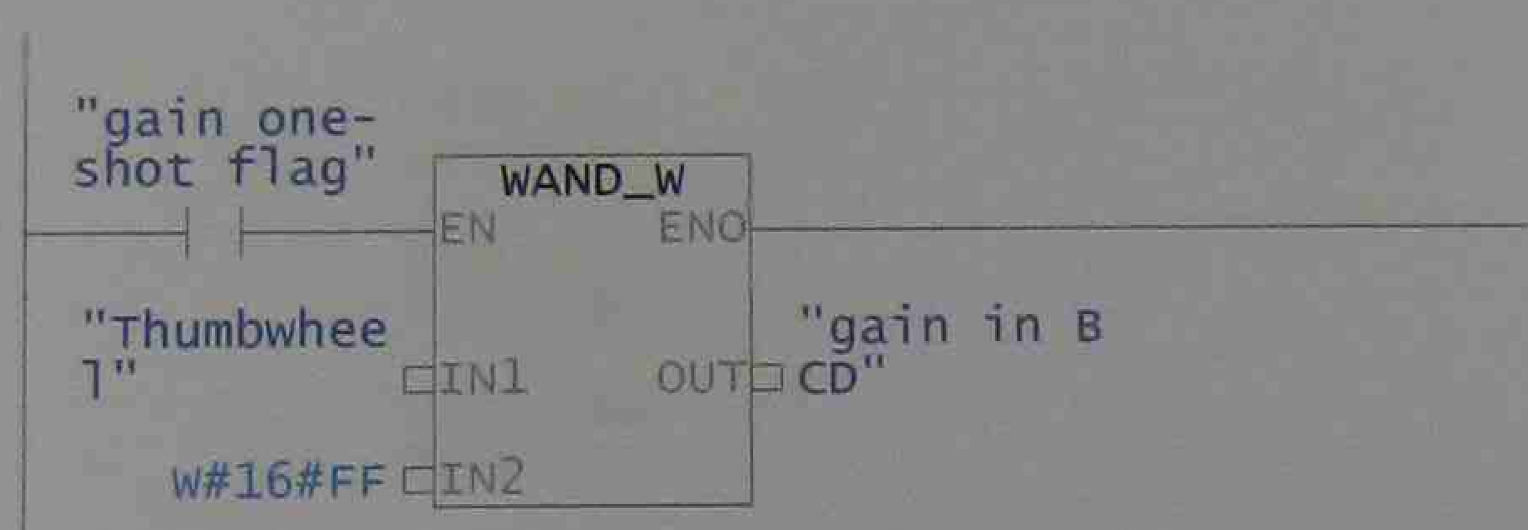


Network: 12

gain one shot

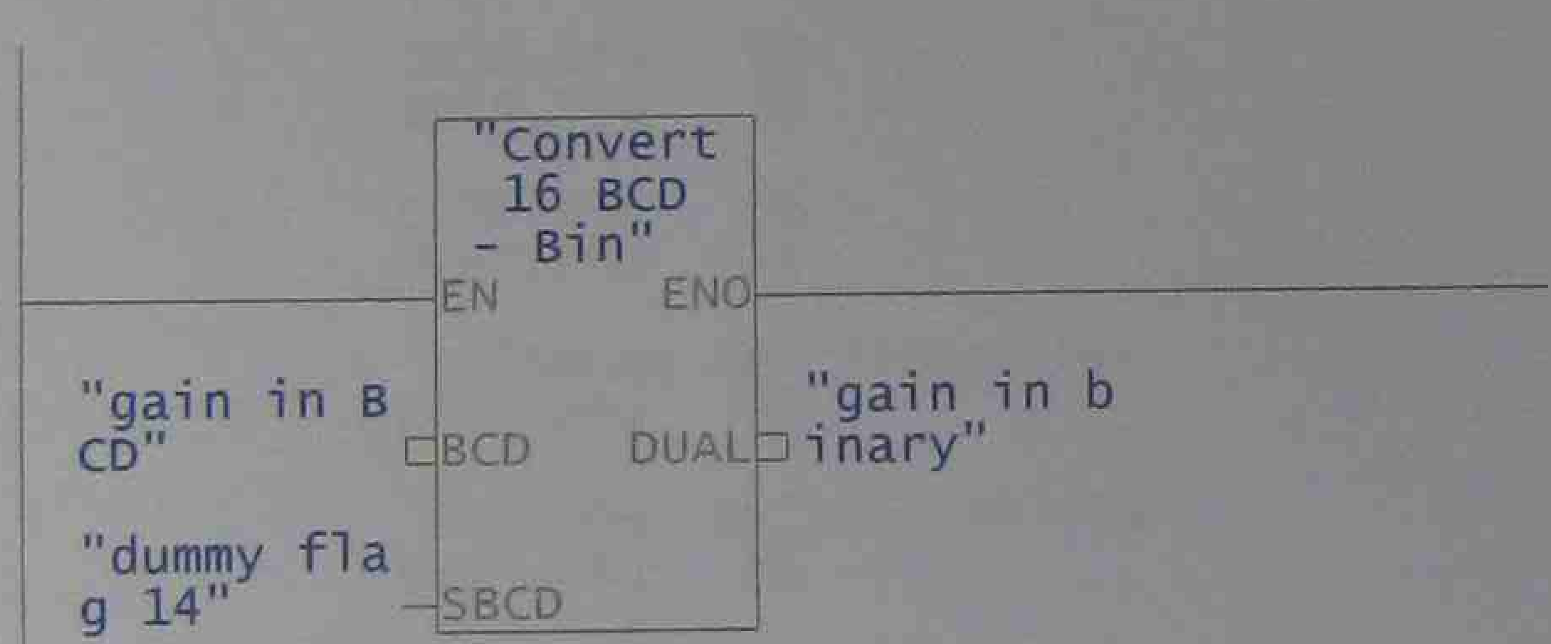
Network: 13

get thumbwheel data right two digits for gain



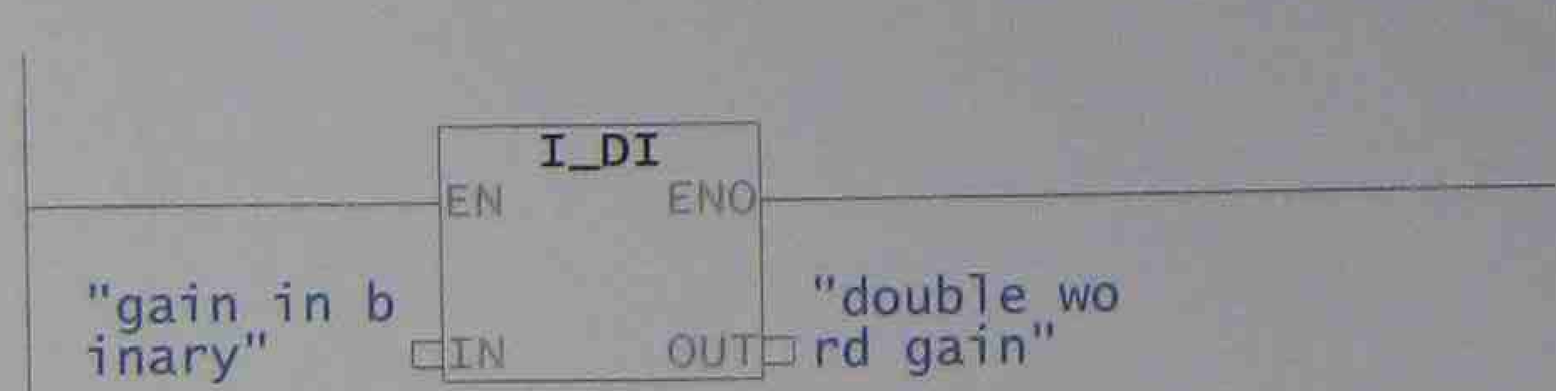
Network: 14

BCD to binary



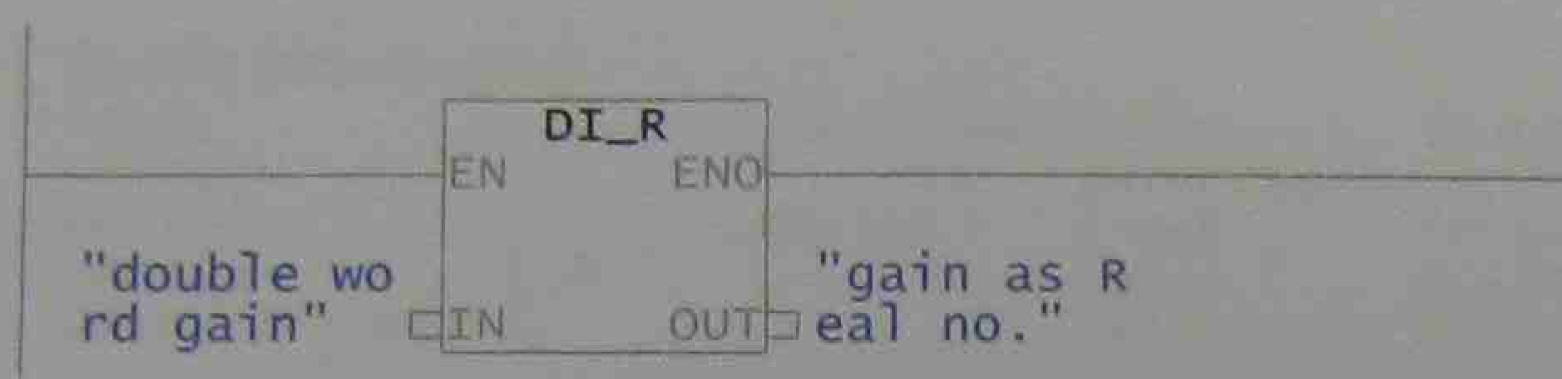
Network: 15

Binary to double word



Network: 16

double word to real for real gain



Network: 17

SCADA monitoring

```
L      DB30.DBW    6
T      DB60.DBW    0

L      DB30.DBW   10
T      DB60.DBW    4

L      DB30.DBW   20
T      DB60.DBW    8

L      DB30.DBW   72
T      DB60.DBW   20

L      DB30.DBW   80
T      DB60.DBW   20

L      DB30.DBW   84
T      DB60.DBW   24

L      DB30.DBW   88
T      DB60.DBW   28

L      DB30.DBW   96
T      DB60.DBW   16

L      "PulseGen DB".siZaehlPer
T      DB60.DBW   32

L      "PulseGen DB".siZaehlPTm
T      DB60.DBW   36
```

OB35:CYC_INT5

Cyclic Interrupt 5

```
Name:                    Time stamp Code:        23/06/09
Author:                             Interface:   20/01/04
Family:                  Lengths    Block:       00700
Version:       1.0                  Code:        00586
Code version: 2                     Data:        00028
```

Block: OB35 "Cyclic Interrupt"

Cont-C and pulsegen function blocks for Proportional and PID control using Pulse duration modukation

Address	Declaration	Name	Type	Start value	Comment
0.0	temp	OB35_EV_CLASS	BYTE		Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
1.0	temp	OB35_STRT_INF	BYTE		16#36 (OB 35 has started)
2.0	temp	OB35_PRIORITY	BYTE		11 (Priority of 1 is lowest)
3.0	temp	OB35_OB_NUMBR	BYTE		35 (Organization block 35, OB35)
4.0	temp	OB35_RESERVED_1	BYTE		Reserved for system
5.0	temp	OB35_RESERVED_2	BYTE		Reserved for system
6.0	temp	OB35_PHASE_OFFSET	WORD		Phase offset (msec)
8.0	temp	OB35_RESERVED_3	INT		Reserved for system
10.0	temp	OB35_EXC_FREQ	INT		Frequency of execution (msec)
12.0	temp	OB35_DATE_TIME	DATE_AND_TIME		Date and time OB35 started

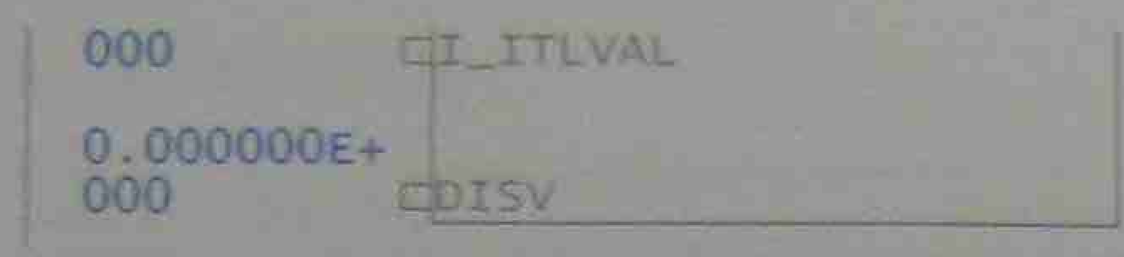
Network: 1

continuous control

"Cont-c DB"

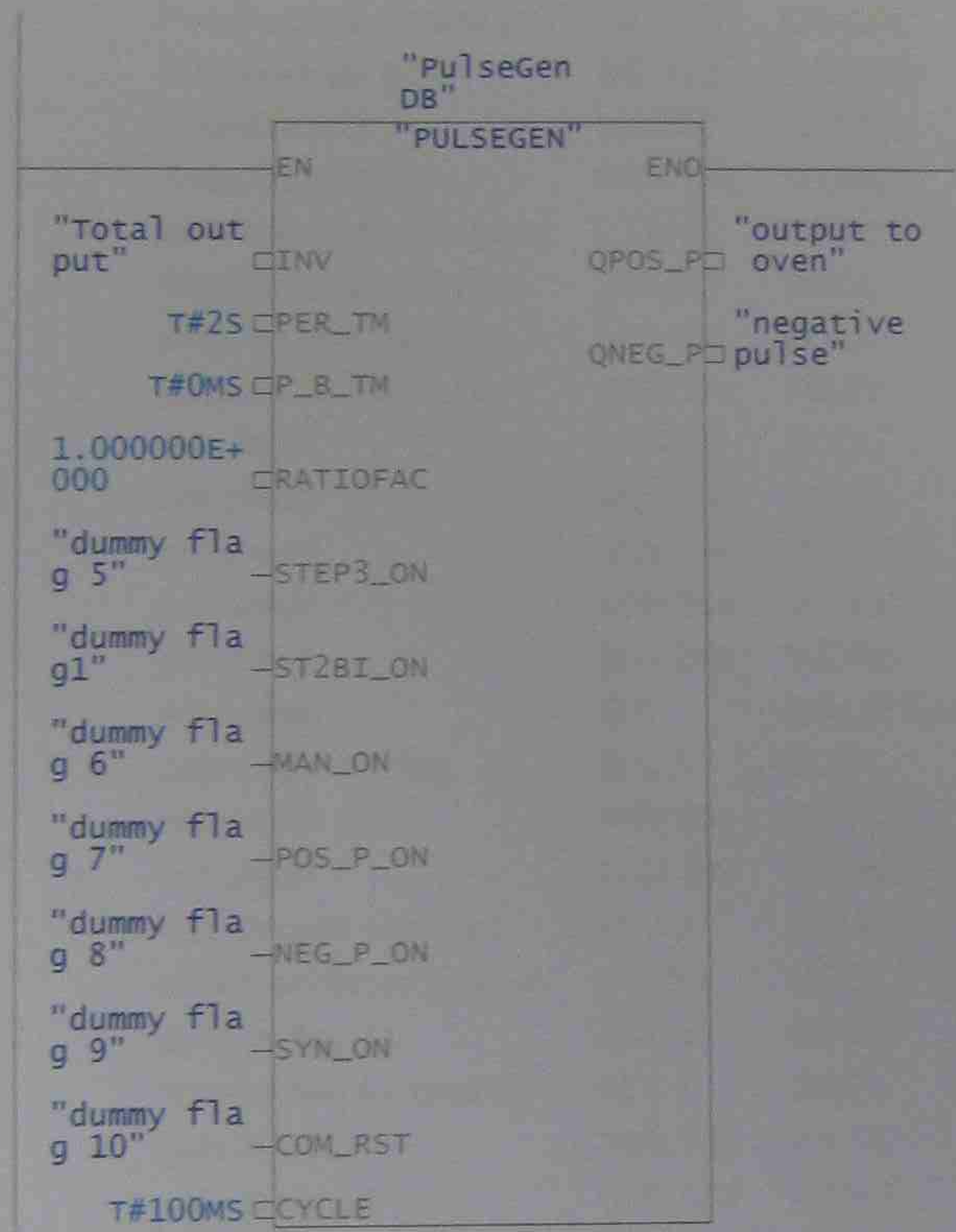
"CONT_C"

	EN	ENO	
"dummy flag 1"	COM_RST	LMN	"Total output"
"dummy flag 13"	MAN_ON	LMN_PER	"dummy word1"
"dummy flag 2"	PVPER_ON	QLMN_HLM	"dummy flag 11"
"P cont. switch"	P_SEL	QLMN_LLM	"dummy flag 12"
"I cont. switch"	I_SEL	LMN_P	"P component"
"dummy flag 3"	INT_HOLD	LMN_I	"I component"
"dummy flag 4"	I_ITL_ON	LMN_D	"D component"
"D cont. switch"	D_SEL	PV	"PV"
		ER	"Error"
T#10MS	CYCLE		
"setpoint as real No."	SP_INT		
"Feedback P.V"	PV_IN		
"dummy word2"	PV_PER		
0.000000E+000	MAN		
"gain as Real no."	GAIN		
T#10S	TI		
T#1S	TD		
T#2S	TM_LAG		
0.000000E+000	DEADB_W		
1.000000E+002	LMN_HLM		
0.000000E+000	LMN_LLM		
1.000000E+000	PV_FAC		
0.000000E+000	PV_OFF		
1.000000E+000	LMN_FAC		
0.000000E+000	LMN_OFF		
0.000000E+			



Network: 2

pulsegen



Symbol table

Status	Symbol	Address	Data Type	Comment
	CYCL_EXC	OB 1	OB 1	Cycle Execution
	SCALE	FC 105	FC 105	Scaling Values
	Dummy word	MW 40	WORD	
	Feedback P.V	MD 100	DWORD	
	Analog Feedbackl	PIW 752	WORD	
	Feedback as double int	MD 116	DWORD	
	Feedback as BDC	MD 122	DWORD	
	Thumbwheel	IW 124	WORD	
	Setpoint in BCD	MW 150	WORD	
	switch activates oneshot	I 126.0	BOOL	
	Setpoint one-shot flag1	M 0.0	BOOL	
	setpoint one-shot flag 2	M 0.1	BOOL	
	Convert 16 BCD - Bin	FC 81	FC 81	Convert BCD to 16 bit binary
	not used1	M 244.7	BOOL	
	not used2	MB 242	BYTE	
	not used3	MW 238	WORD	
	not used4	MW 244	WORD	
	not used5	MW 246	WORD	
	not used6	C 0	COUNTER	
	gain one-shot flag	M 0.3	BOOL	
	setpoint in Bin	MW 155	WORD	
?	not used7	MW 149	WORD	
	flag for gain one-shot	M 0.2	BOOL	
	switch activated gain on	I 126.1	BOOL	
	dummy flag 14	M 1.3	BOOL	
	dummy flag for scaling	M 1.1	BOOL	
	gain in BCD	MW 160	WORD	
	gain in binary	MW 164	WORD	
	dummy flag 13	M 7.1	BOOL	
	Cont-c DB	DB 30	FB 41	
	setpoint as real No.	MD 174	DWORD	
	7-seg Display	QB 125	BYTE	
	double int setpoint	MD 170	DWORD	
	double word gain	MD 168	DWORD	
	CONT_C	FB 41	FB 41	Continuous Control
	PULSEGEN	FB 43	FB 43	Pulse Generation
	CYC_INT5	OB 35	OB 35	Cyclic Interrupt 5
	dummy flag 1	M 7.0	BOOL	
	dummy flag 2	M 7.2	BOOL	
	dummy flag 3	M 7.4	BOOL	
	dummy flag 4	M 7.5	BOOL	
	Total output	MD 200	DWORD	
	PulseGen DB	DB 40	FB 43	
	dummy flag 5	M 8.7	BOOL	
	dummy flag 6	M 8.1	BOOL	
	dummy flag 7	M 8.2	BOOL	
	dummy flag 8	M 8.3	BOOL	

Status	Symbol	Address	Data Type	Comment
	dummy flag 9	M 8.4	BOOL	
	dummy flag 10	M 8.5	BOOL	
	I cont. switch	I 126.6	BOOL	
	dummy flag 11	M 9.0	BOOL	
	dummy flag 12	M 9.1	BOOL	
	DB for SCADA	DB 60	DB 60	
	dummy word2	MW 50	WORD	
	P component	MD 210	DWORD	
	I component	MD 214	DWORD	
	D component	MD 218	DWORD	
	PV	MD 222	DWORD	
	Error	MD 226	DWORD	
	dummy word1	MW 204	WORD	
	negative pulse	Q 124.6	BOOL	
	output to oven	Q 124.7	BOOL	
	dummy flag1	M 6.6	BOOL	
	gain as Real no.	MD 178	DWORD	
	D cont. switch	I 126.5	BOOL	
	P cont. switch	I 126.7	BOOL	

DB30:Cont-c DB

```
Name:                       Time stamp Code:       04/05/09
Author:        SIMATIC                 Interface:  10/12/02
Family:        ICONT        Lengths    Block:      00366
Version:       0.0                     Code:       00126
Code version:  2                       Data:       00162
```

Block: DB30

Address	Declaration	Name	Type	Start value	Comment
0.0	in	COM_RST	BOOL	FALSE	complete restart
0.1	in	MAN_ON	BOOL	TRUE	manual value on
0.2	in	PVPER_ON	BOOL	FALSE	process variable peripherie on
0.3	in	P_SEL	BOOL	TRUE	proportional action on
0.4	in	I_SEL	BOOL	TRUE	integral action on
0.5	in	INT_HOLD	BOOL	FALSE	integral action hold
0.6	in	I_ITL_ON	BOOL	FALSE	initialization of the integral action
0.7	in	D_SEL	BOOL	FALSE	derivative action on
2.0	in	CYCLE	TIME	T#1S	sample time
6.0	in	SP_INT	REAL	0.000000E+000	internal setpoint
10.0	in	PV_IN	REAL	0.000000E+000	process variable in
14.0	in	PV_PER	WORD	W#16#0	process variable peripherie
16.0	in	MAN	REAL	0.000000E+000	manual value
20.0	in	GAIN	REAL	2.000000E+000	proportional gain
24.0	in	TI	TIME	T#20S	reset time
28.0	in	TD	TIME	T#10S	derivative time
32.0	in	TM_LAG	TIME	T#2S	time lag of the derivative action
36.0	in	DEADB_W	REAL	0.000000E+000	dead band width
40.0	in	LMN_HLM	REAL	1.000000E+002	manipulated value high limit
44.0	in	LMN_LLM	REAL	0.000000E+000	manipulated value low limit
48.0	in	PV_FAC	REAL	1.000000E+000	process variable factor
52.0	in	PV_OFF	REAL	0.000000E+000	process variable offset
56.0	in	LMN_FAC	REAL	1.000000E+000	manipulated value factor
60.0	in	LMN_OFF	REAL	0.000000E+000	manipulated value offset
64.0	in	I_ITLVAL	REAL	0.000000E+000	initialization value of the integral action
68.0	in	DISV	REAL	0.000000E+000	disturbance variable
72.0	out	LMN	REAL	0.000000E+000	manipulated value
76.0	out	LMN_PER	WORD	W#16#0	manipulated value peripherie
78.0	out	QLMN_HLM	BOOL	FALSE	high limit of manipulated value reached
78.1	out	QLMN_LLM	BOOL	FALSE	low limit of manipulated value reached
80.0	out	LMN_P	REAL	0.000000E+000	proportionality component
84.0	out	LMN_I	REAL	0.000000E+000	integral component
88.0	out	LMN_D	REAL	0.000000E+000	derivative component
92.0	out	PV	REAL	0.000000E+000	process variable
96.0	out	ER	REAL	0.000000E+000	error signal
100.0	stat	sInvAlt	REAL	0.000000E+000	
104.0	stat	sIanteilAlt	REAL	0.000000E+000	
108.0	stat	sRestInt	REAL	0.000000E+000	
112.0	stat	sRestDif	REAL	0.000000E+000	
116.0	stat	sRueck	REAL	0.000000E+000	
120.0	stat	sLmn	REAL	0.000000E+000	
124.0	stat	sbArwHLmOn	BOOL	FALSE	

Address	Declaration	Name	Type	Start value	Comment
[illegible]	stat	[illegible]	BOOL	FALSE	
[illegible]	stat	[illegible]	BOOL	TRUE	

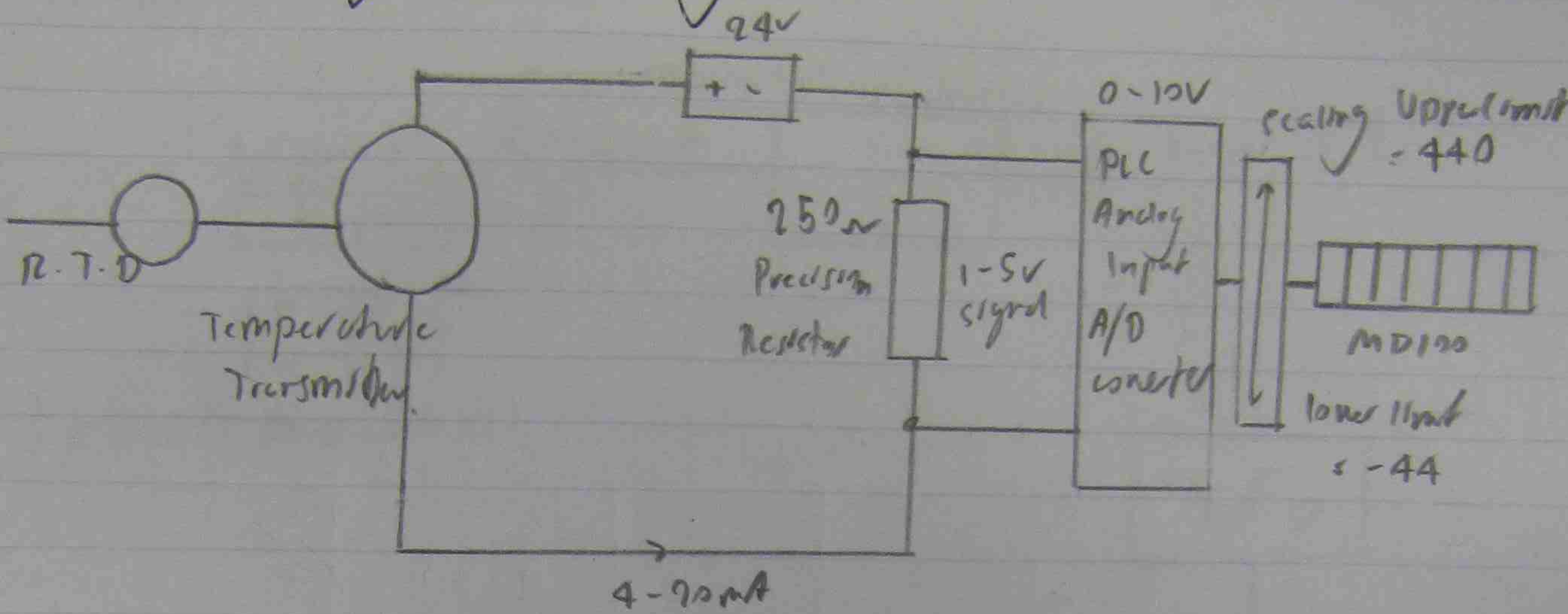
DB40:PulseGen DB

```
Name:                         Time stamp Code:       04/05/09
Author:        SIMATIC                   Interface:  10/12/02
Family:        ICONT          Lengths    Block:      00190
Version:       0.0                       Code:       00034
Code version:  2                         Data:       00078
```

Block: DB40

Address	Declaration	Name	Type	Start value	Comment
0.0	in	INV	REAL	0.000000E+000	input variable
4.0	in	PER_TM	TIME	T#1S	period time
8.0	in	P_B_TM	TIME	T#50MS	minimum pulse/break time
12.0	in	RATIOFAC	REAL	1.000000E+000	ratio factor
16.0	in	STEP3_ON	BOOL	TRUE	three step signal on
16.1	in	ST2BI_ON	BOOL	FALSE	two step signal for bipolar manipulated value on
16.2	in	MAN_ON	BOOL	FALSE	manual mode on
16.3	in	POS_P_ON	BOOL	FALSE	positiv pulse on
16.4	in	NEG_P_ON	BOOL	FALSE	negativ pulse on
16.5	in	SYN_ON	BOOL	TRUE	impuls output synchronisation on
16.6	in	COM_RST	BOOL	FALSE	complete restart
18.0	in	CYCLE	TIME	T#10MS	sample time
22.0	out	QPOS_P	BOOL	FALSE	positiv pulse on
22.1	out	QNEG_P	BOOL	FALSE	negativ pulse on
24.0	stat	sInvAlt	REAL	0.000000E+000	
28.0	stat	siZaehlPer	INT	0	
30.0	stat	siZaehlPTm	INT	0	
32.0	stat	sbPosP	BOOL	TRUE	

Diagram of Analog Input and Scaling



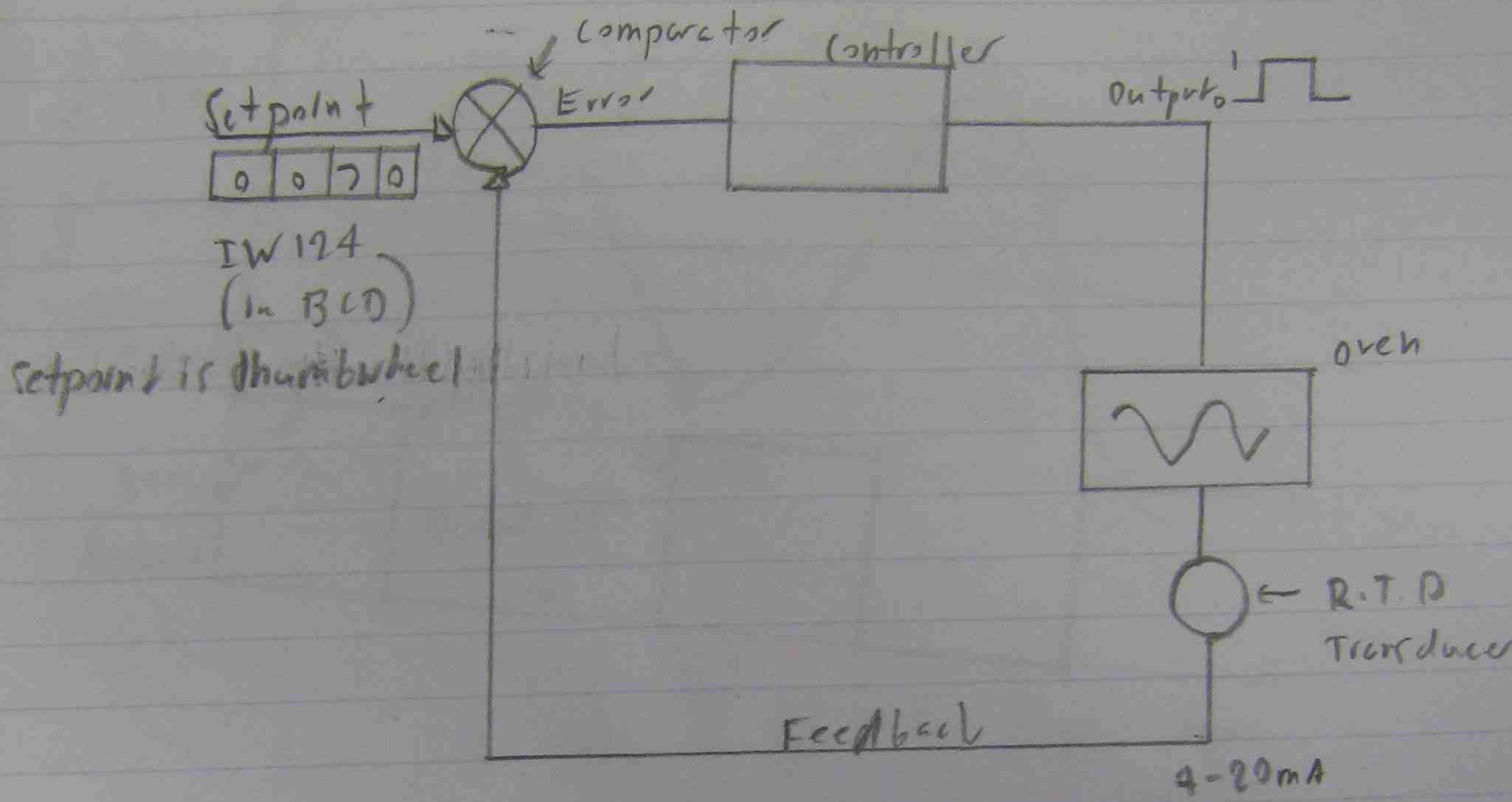
4mA = 0°C = 1V

20mA = 198°C = 5V

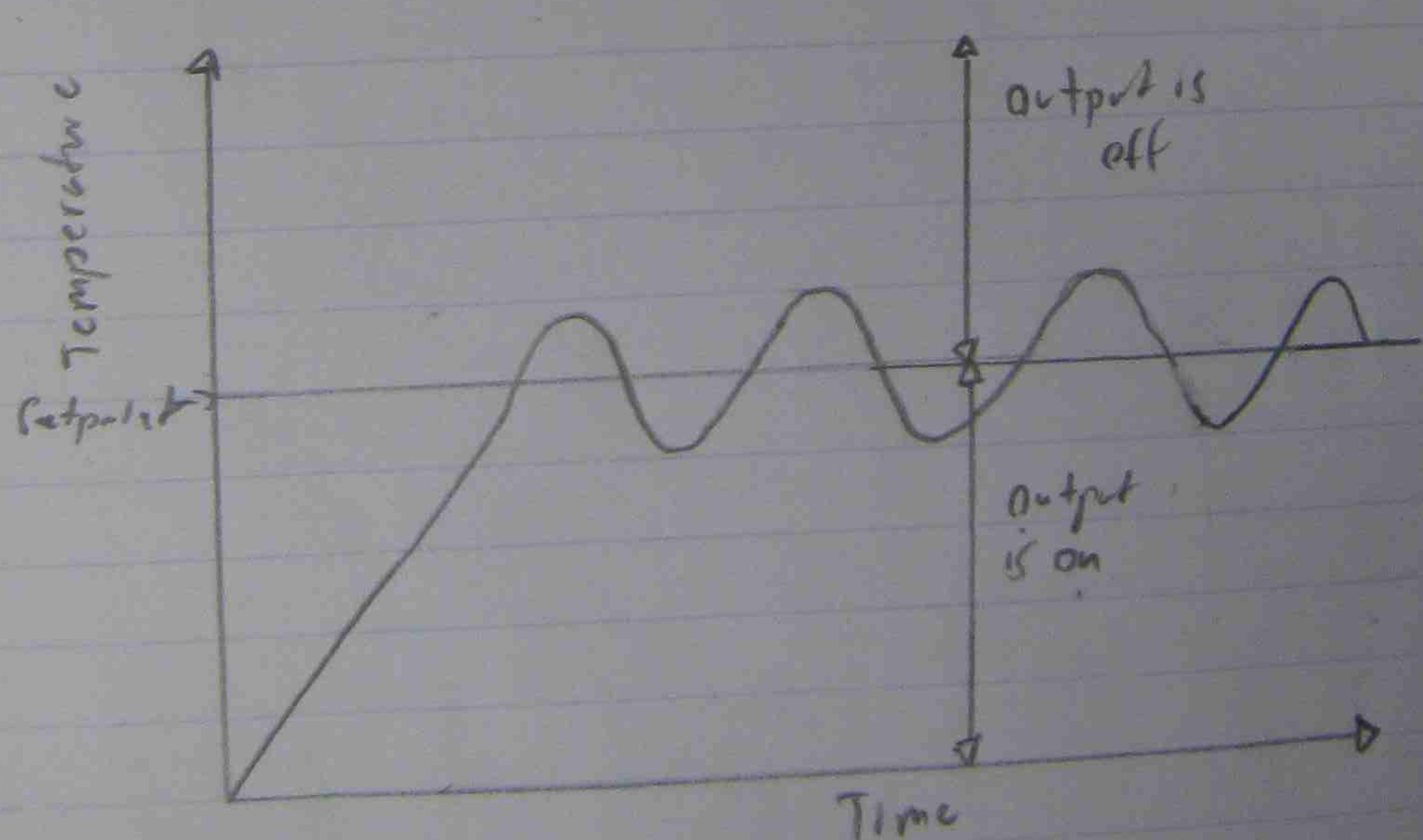
0V — 5V — 10V

-44.5 — 198 — +445

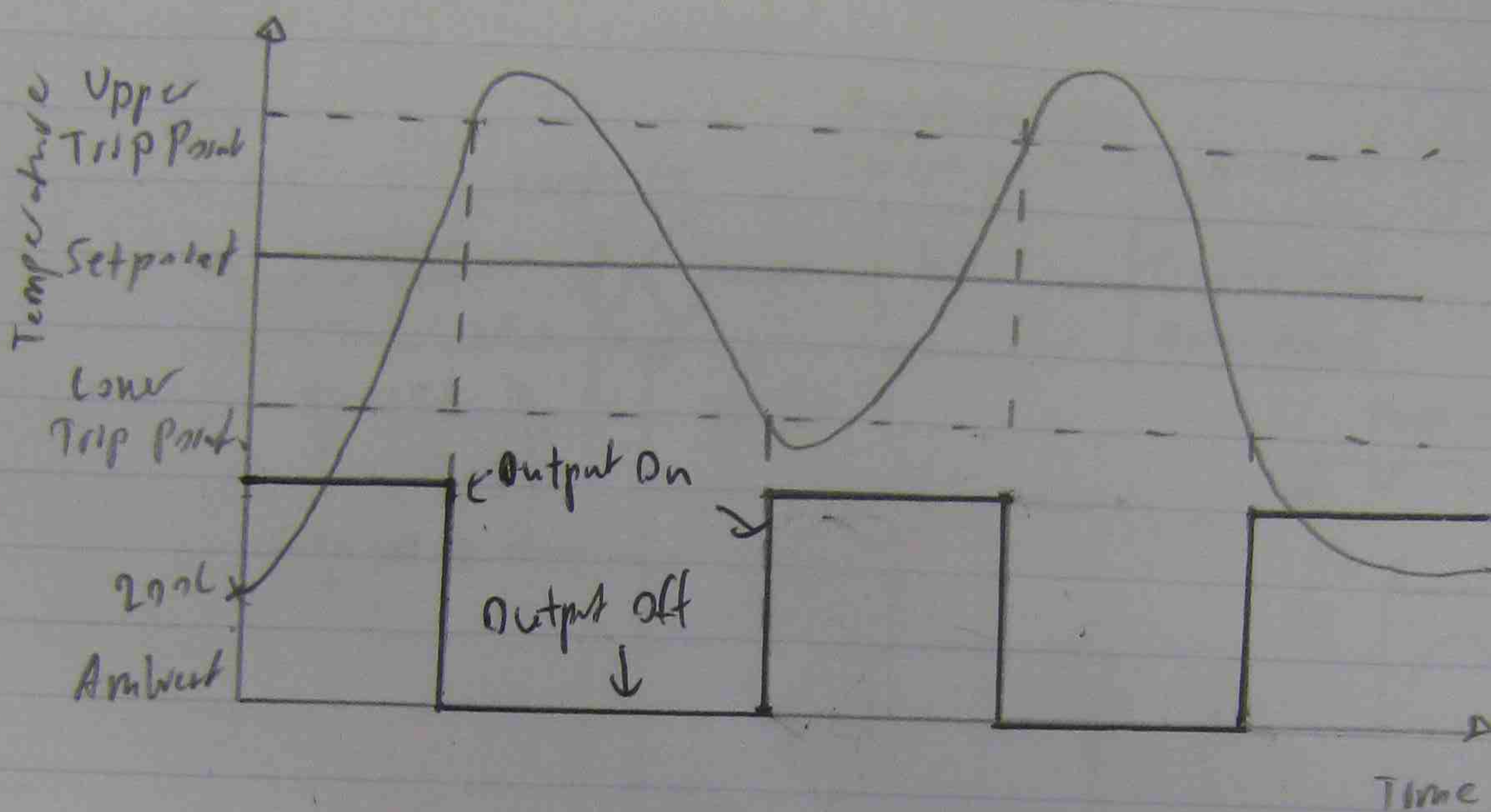
Block Diagram of On/Off Controller



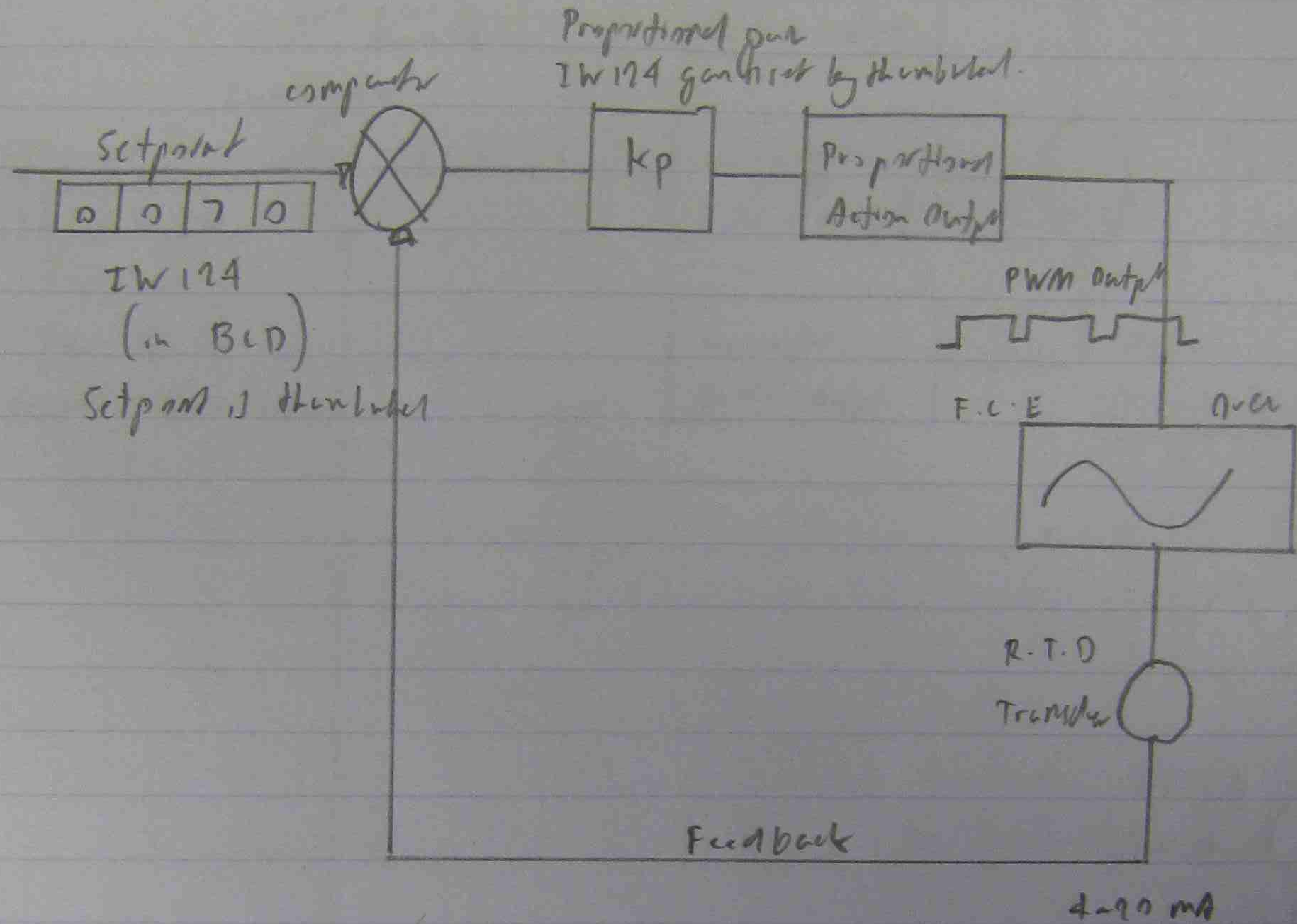
On/Off control Diagram



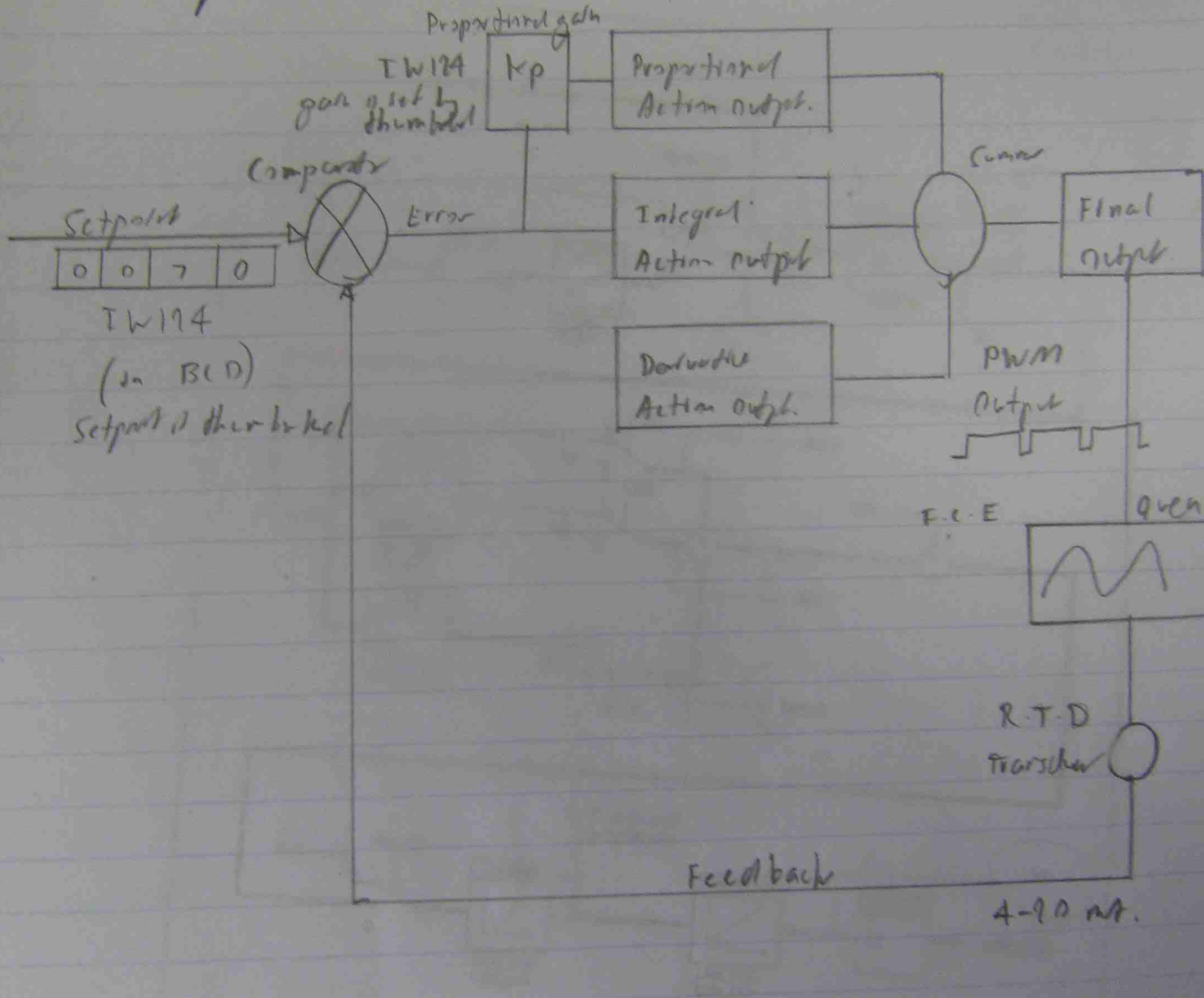
Hysteresis Control Diagram



Block Diagram of Proportional only Controller



Block Diagram of PID Controller



Output Parameters

Parameter	Data Type	Values	Default	Description
QPOS_P	BOOL		FALSE	OUTPUT POSITIVE PULSE The output parameter "output positive pulse" is set when a pulse is to be output. In three-step control, this is always the positive pulse. In two-step control, QNEG_P is always set inversely to QPOS_P.
QNEG_P	BOOL		FALSE	OUTPUT NEGATIVE PULSE The output parameter "output negative pulse" is set when a pulse is to be output. In three-step control, this is always the negative pulse. In two-step control, QNEG_P is always set inversely to QPOS_P.

See also:

Example of the PULSEGEN Block

Input Parameters Table 3-1 contains the description of the input parameters for FB41 "CONT_C".

Table 3-1 Input Parameters (INPUT) for FB 41 "CONT_C"

	Parameter	Data Type	Range of Values	Default	Description
	COM_RST	BOOL		FALSE	COMPLETE RESTART The block has a complete restart routine that is processed when the input "complete restart" is set.
	MAN_ON	BOOL		TRUE	MANUAL VALUE ON If the input "manual value on" is set, the control loop is interrupted. A manual value is set as the manipulated value.
	PVPER_ON	BOOL		FALSE	PROCESS VARIABLE PERIPHERAL ON If the process variable is read from the I/Os, the input PV_PER must be connected to the I/Os and the input "process variable peripheral on" must be set.
I126.7	P_SEL	BOOL		TRUE	PROPORTIONAL ACTION ON The PID actions can be activated or deactivated individually in the PID algorithm. The P action is on when the input "proportional action on" is set.
I126.6	I_SEL	BOOL		TRUE	INTEGRAL ACTION ON The PID actions can be activated or deactivated individually in the PID algorithm. The I action is on when the input "integral action on" is set.
	INT_HOLD	BOOL		FALSE	INTEGRAL ACTION HOLD The output of the integrator can be "frozen" by setting the input "integral action hold".
	I_ITL_ON	BOOL		FALSE	INITIALIZATION OF THE INTEGRAL ACTION The output of the integrator can be connected to the input I_ITL_VAL by setting the input "initialization of the integral action on".
I126.5	D_SEL	BOOL		FALSE	DERIVATIVE ACTION ON The PID actions can be activated or deactivated individually in the PID algorithm. The D action is on when the input "derivative action on" is set.
	CYCLE	TIME	>= 1ms	T#1s	SAMPLING TIME The time between the block calls must be constant. The "sampling time" input specifies the time between block calls.
	SP_INT	REAL	-100.0...100.0 (%) or phys. value 1)	0.0	INTERNAL SETPOINT The "internal setpoint" input is used to specify a setpoint.
	PV_IN	REAL	-100.0...100.0 (%) or phys. value 1)	0.0	PROCESS VARIABLE IN An initialization value can be set at the "process variable in" input or an external process variable in floating point format can be connected.
	PV_PER	WORD		W#16#0000	PROCESS VARIABLE PERIPHERAL The process variable in the I/O format is connected to the controller at the "process variable peripheral" input.

Table 3-1 Input Parameters (INPUT) for FB 41 "CONT_C", continued

Parameter	Data Type	Range of Values	Default	Description
MAN	REAL	-100.0...100.0 (%) or phys. value 2)	0.0	MANUAL VALUE The "manual value" input is used to set a manual value using the operator interface functions.
GAIN	REAL		2.0	PROPORTIONAL GAIN The "proportional value" input specifies the controller gain.
TI	TIME	>= CYCLE	T#20s 10s	RESET TIME The "reset time" input determines the time response of the integrator.
TD	TIME	>= CYCLE	T#10s 1s	DERIVATIVE TIME The "derivative time" input determines the time response of the derivative unit.
TM_LAG	TIME	>=CYCLE/2	T#2s	TIME LAG OF THE DERIVATIVE ACTION The algorithm of the D action includes a time lag that can be assigned at the "time lag of the derivative action" input.
DEADB_W	REAL	>= 0.0 (%) or phys. value 1)	0.0	DEAD BAND WIDTH A dead band is applied to the error. The "dead band width" input determines the size of the dead band.
LMN_HLM	REAL	LMN_LLM ...100.0 (%) or phys. value 2)	100.0	MANIPULATED VALUE HIGH LIMIT The manipulated value is always limited by an upper and lower limit. The "manipulated value high limit"input specifies the upper limit.
LMN_LLM	REAL	-100.0... LMN_HLM (%) or phys. value 2)	0.0	MANIPULATED VALUE LOW LIMIT The manipulated value is always limited by an upper and lower limit. The "manipulated value low limit"input specifies the lower limit.
PV_FAC	REAL		1.0	PROCESS VARIABLE FACTOR The "process variable factor" input is multiplied by the process variable. The input is used to adapt the process variable range.
PV_OFF	REAL		0.0	PROCESS VARIABLE OFFSET The "process variable offset" input is added to the process variable. The input is used to adapt the process variable range.
LMN_FAC	REAL		1.0	MANIPULATED VALUE FACTOR The "manipulated value factor" input is multiplied by the manipulated value. The input is used to adapt the manipulated value range.
LMN_OFF	REAL		0.0	MANIPULATED VALUE OFFSET The "manipulated value offset" is added to the manipulated value. The input is used to adapt the manipulated value range.

Table 3-1 Input Parameters (INPUT) for FB 41 "CONT_C", continued

Parameter	Data Type	Range of Values	Default	Description
I_ITLVAL	REAL	-100.0...100. 0 (%) or phys. value 2)	0.0	INITIALIZATION VALUE OF THE INTEGRAL ACTION The output of the integrator can be set at input I_ITL_ON. The initialization value is applied to the input "initialization value of the integral action".
DISV	REAL	-100.0...100. 0 (%) or phys. value 2)	0.0	DISTURBANCE VARIABLE For feedforward control, the disturbance variable is connected to input "disturbance variable".

1) Parameters in the setpoint and process variable branches with the same unit
2) Parameters in the manipulated value branch with the same unit

Output Parameters

Table 3-2 contains the description of the output parameters for FB41 "CONT_C".

Table 3-2 Output Parameters (OUTPUT) for FB 41 "CONT_C"

Parameter	Data Type	Range of Values	Default	Description
LMN	REAL		0.0	MANIPULATED VALUE The effective manipulated value is output in floating point format at the "manipulated value" output.
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERAL The manipulated value in the I/O format is connected to the controller at the "manipulated value peripheral" output.
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an upper and lower limit. The output "high limit of manipulated value reached" indicates that the upper limit has been exceeded.
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an upper and lower limit. The output "low limit of manipulated value reached" indicates that the lower limit has been exceeded.
LMN_P	REAL		0.0	PROPORTIONAL COMPONENT The "proportional component" output contains the proportional component of the manipulated variable.
LMN_I	REAL		0.0	INTEGRAL COMPONENT The "integral component" output contains the integral component of the manipulated value.
LMN_D	REAL		0.0	DERIVATIVE COMPONENT The "derivative component" output contains the derivative component of the manipulated value.

Table 3-1 Input Parameters (INPUT) for FB 41 "CONT_C", continued

Parameter	Data Type	Range of Values	Default	Description
I_ITLVAL	REAL	-100.0...100.0 (%) or phys. value 2)	0.0	INITIALIZATION VALUE OF THE INTEGRAL ACTION The output of the integrator can be set at input I_ITL_ON. The initialization value is applied to the input "initialization value of the integral action".
DISV	REAL	-100.0...100.0 (%) or phys. value 2)	0.0	DISTURBANCE VARIABLE For feedforward control, the disturbance variable is connected to input "disturbance variable".

1) Parameters in the setpoint and process variable branches with the same unit
2) Parameters in the manipulated value branch with the same unit

Output Parameters

Table 3-2 contains the description of the output parameters for FB41 "CONT_C".

Table 3-2 Output Parameters (OUTPUT) for FB 41 "CONT_C"

Parameter	Data Type	Range of Values	Default	Description
LMN	REAL		0.0	MANIPULATED VALUE The effective manipulated value is output in floating point format at the "manipulated value" output.
LMN_PER	WORD		W#16#0000	MANIPULATED VALUE PERIPHERAL The manipulated value in the I/O format is connected to the controller at the "manipulated value peripheral" output.
QLMN_HLM	BOOL		FALSE	HIGH LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an upper and lower limit. The output "high limit of manipulated value reached" indicates that the upper limit has been exceeded.
QLMN_LLM	BOOL		FALSE	LOW LIMIT OF MANIPULATED VALUE REACHED The manipulated value is always limited to an upper and lower limit. The output "low limit of manipulated value reached" indicates that the lower limit has been exceeded.
LMN_P	REAL		0.0	PROPORTIONAL COMPONENT The "proportional component" output contains the proportional component of the manipulated variable.
LMN_I	REAL		0.0	INTEGRAL COMPONENT The "integral component" output contains the integral component of the manipulated value.
LMN_D	REAL		0.0	DERIVATIVE COMPONENT The "derivative component" output contains the derivative component of the manipulated value.

Table 3-2 Output Parameters (OUTPUT) for FB 41 "CONT_C", continued

Parameter	Data Type	Range of Values	Default	Description
PV	REAL		0.0	PROCESS VARIABLE The effective process variable is output at the "process variable" output.
ER	REAL		0.0	ERROR SIGNAL The effective error is output at the "error signal" output.

All the signal outputs are set to 0.

Error Information

The error output parameter RET_VAL is not used.

Input Parameters

Parameter	Data Type	Range of Values	Default	Description
INV	REAL	-100.0...100.0 (%)	0.0	INPUT VARIABLE An analog manipulated value is connected to the input parameter "input variable."
PER_TM	TIME	>=20*CYCLE	T#1s	PERIOD TIME The constant period of pulse duration modulation is input with the "period time" input parameter. This corresponds to the sampling time of the controller. The ratio between the sampling time of the pulse generator and the sampling time of the controller determines the accuracy of the pulse duration modulation.
P_B_TM	TIME	>= CYCLE	T#0ms	MINIMUM PULSE/BREAK TIME A minimum pulse or minimum break time can be assigned at the input parameters "minimum pulse or minimum break time."
RATIOFAC	REAL	0.1 ...10.0	1.0	RATIO FACTOR The input parameter "ratio factor" can be used to change the ratio of the duration of negative to positive pulses. In a thermal process, this would, for example, allow different time constants for heating and cooling to be compensated (for example, in a process with electrical heating and water cooling).
STEP3_ON	BOOL		TRUE	THREE STEP CONTROL ON The "three-step control on" input parameter activates this mode. In three-step control, both output signals are active.
ST2BI_ON	BOOL		FALSE	TWO STEP CONTROL FOR BIPOLAR MANIPULATED VALUE RANGE ON With the input parameter "two-step control for bipolar manipulated value range on" you can select between the modes "two-step

				control for bipolar manipulated value" and "two-step control for unipolar manipulated value range." The parameter STEP3_ON = FALSE must be set.
MAN_ON	BOOL		FALSE	MANUAL MODE ON By setting the input parameter "manual mode on," the output signals can be set manually.
POS_P_ON	BOOL		FALSE	POSITIVE PULSE ON In the manual mode with three-step control, the output signal QPOS_P can be set at the input parameter "positive pulse on." In the manual mode with two-step control, QNEG_P is always set inversely to QPOS_P.
NEG_P_ON	BOOL		FALSE	NEGATIVE PULSE ON In the manual mode with three-step control, the output signal QNEG_P can be set at the input parameter "negative pulse on." In the manual mode with two-step control, QNEG_P is always set inversely to QPOS_P.
SYN_ON	BOOL		TRUE	SYNCHRONIZATION ON By setting the input parameter "synchronization on," it is possible to synchronize automatically with the block that updates the input variable INV. This ensures that a changing input variable is output as quickly as possible as a pulse.
COM_RST	BOOL		FALSE	COMPLETE RESTART The block has an initialization routine that is processed when the COM_RST input is set
CYCLE	TIME	>= 1ms	T#10ms	SAMPLING TIME The time between block calls must be constant. The "sampling time" input specifies the time between block calls.

Note

The values of the input parameters are not limited in the block. There is no parameter check.