

FeSCADA & ADAM 6017

Introduction

An application was done to show the possibilities of FeSCADA. By the end of this paper the reader will learn how FeSCADA can be used with Advantech ADAM 6017 Ethernet IO module.

1. Description
2. Hardware
3. ADAM 6017 setup
4. ADAM 6017 Modbus registers
5. FeMODBUS communication setup
6. FeSCADA project
7. Special application
8. Conclusions

1) Description

A heating and cooling application was developed with the aid of Advantech ADAM 6017 Ethernet IO module and a FeSCADA PID controller. Two RTD sensors and one thermocouple sensor are used to read temperatures. A 25W, 10 Ω resistor is used as the heating element. Two Solid State Relays (SSR) are used to control the heating power (from a 24 VDC power supply) and a fan (cooling power).

The example is common for other types of Ethernet IOs, because it is an industrial practice for many producers to use Modbus TCP communication.

2) Hardware

The hardware is composed of one Advantech ADAM 6017 Ethernet IO module, with the following characteristics:

- input voltage: 10-30 VDC
- 8 analog inputs, 16-bit resolution, 100 samples/second
ranges - voltage: from ± 0.15 V up to ± 10 V
- current: ± 20 mA, 0~20 mA, 4~20 mA
- 2 digital outputs, sink, 0...30 VDC
- communication protocols: Modbus/TCP, TCP/IP, UDP, HTTP, DHCP, MQTT, SNMP, ASCII command



Two RTD sensors Pt100, together with two temperature transmitters: 0...250°C to 4-20mA and 0...500°C to 4-20mA.



One K type thermocouple wire.



Two Solid State Relays (SSR) 100VDC, 20A.



One NTE 25W, 10Ω resistor, the heating element.



Cooling fan, 24VDC, 0.16A.



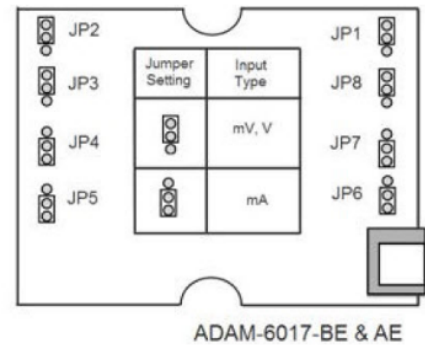
Hardware prices.

Name	Unit price	Qty	Price	Description
ADAM 6017	\$380.00	1	\$380.00	8 channels analog IO module
uxcell Temperature Transmitter	\$12.00	2	\$24.00	Pt100 signal converter 0...250 °C to 4-20 mA 0...500 °C to 4-20 mA
TWTADE PT100	\$13.00	2	\$26.00	Pt100, 3 wire sensors -50...200 °C
Thermocouple	\$10.00	1	\$10.00	K thermocouple wire
D1D20	\$72.00	2	\$144.00	Solid State Relay 100VDC, 20A

Power supply 24VDC	\$40.00	1	\$40.00	120W, 24VDC, 5A
Nidec fan, 24DVC	\$13.00	1	\$13.00	0.16A inverter cooling fan
TOTAL = \$637.00				

3) ADAM 6017 setup

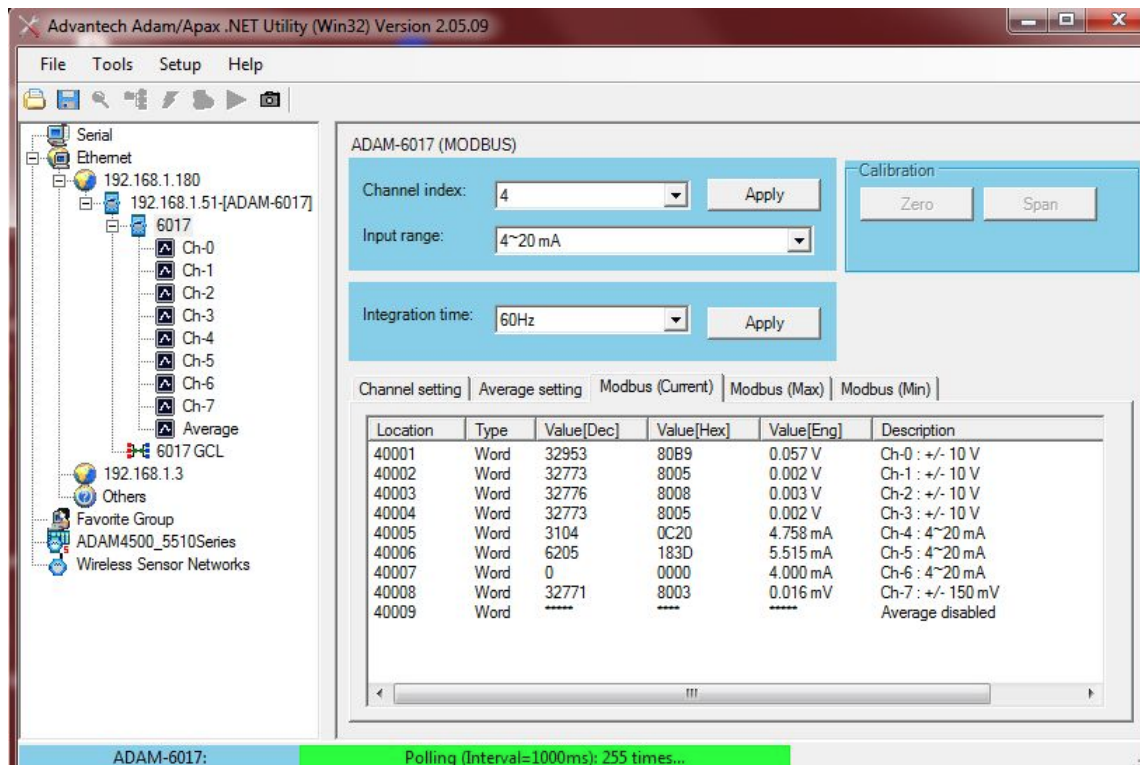
ADAM 6017 has to be setup for the type and range of our temperature transmitters and for the thermocouple. By default, the module is setup to read $\pm 10V$. To change a channel to measure current a jumper has to be moved, for that particular channel.



Because it has 16 bits resolution, the whole measuring range is converted to:

- a) 0...65535, for positive ranges: 0~0.5V, 0~1V, 0~5V, 0~10V, 0~20mA, 4~20mA
- b) -32768 ... 32767 for \pm ranges: $\pm 0.15V$, $\pm 0.5V$, $\pm 1V$, $\pm 5V$, $\pm 10V$, $\pm 20 mA$

To setup ADAM 6017 for the IP address and the measuring ranges, the software ADAM.NET is used. We set the IP address to 192.168.1.51. See below



4) ADAM 6017 Modbus registers

The following table is showing the Modbus register numbers for the IO module: channels data and digital outputs.

Parameter	Modbus register number	Description
Data channels 0...7 (read only)	0...7	Holding registers , 16 bits, with the values from the 8 channels
Digital outputs 0,1 (read/write)	16, 17	Coils addresses for the two digital outputs

5) FeMODBUS communication setup

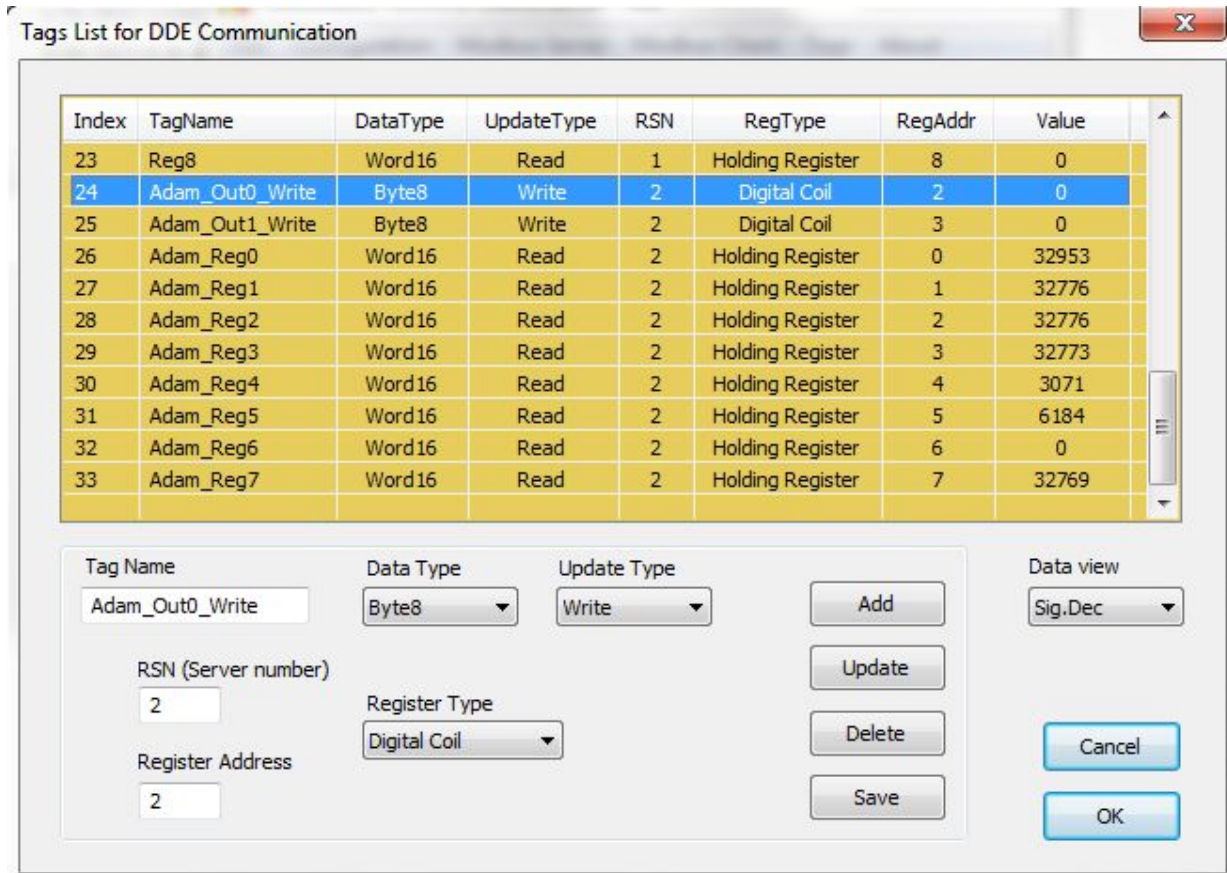
FeMODBUS software is used to connect to ADAM 6017, which is a Modbus server on port 502. We send 2 requests: one is to read the first 10 holding registers every 751 ms, the second is to write 2 coils, every 10 ms. We put two times the coils writing, to make sure the ADAM will get the command. The fast coil writing is essential for a good PID control.

The screenshot displays the 'Servers & Requests' Lists for this Client software interface. It features several key components:

- Servers List:** A table with columns: ServerName, ServerIP, Server..., LocalIP, LocalPort, RSN, Active, Connec..., Error. It lists two servers: 'Remote_Ard...' and 'Advantech'.
- Requests List:** A table with columns: RN, Func..., Uni..., RegAddr, RegNo, Offset, Active, Cyclical, CycleTime, Error, StatusBits. It lists three requests.
- Holding Registers:** A table with columns: Index, Value. It shows values for indices 0, 1, 2, and 3.
- Input Registers:** A table with columns: Index, Value. It shows values for indices 0, 1, 2, and 3.
- Digital I/O:** Sections for '16 Digital Inputs (Read Only)' and '16 Digital Coils (Set/Reset)' with checkboxes for each bit.
- Configuration Panels:** 'TCP/IP Connection' (My IP Address, My Port, Server IP Address, Server Port, Server Name) and 'Modbus Request Parameters' (Function Code, Unit Address, Remote Register Address, How Many Registers?, Local Register Address, Cyclical? Yes/No, Cycle Time [ms], Active).

After setting up all the requests we defined tags which we linked with different registers on the local computer. The tags are used for DDE communication between

FeMODBUS and FeSCADA. All the tags are assigned to the Remote Server Number 2 (RSN 2).



6) FeSCADA project

The first step in a FeSCADA project is to define the DDE communication channels and the tags. In the picture below one can see that we defined one DDE channel as channel number 1: DDE_Application = "MB" and DDE_Topic = "TAGS".

Every tag has an internal name used in FeSCADA and a DDE Name for communication with the MB DDE server. All the tags defined in FeMODBUS will have a correspondent tag in FeSCADA. We kept the name the same with 3 exceptions:

FeMODBUS	FeSCADA	Scaled						
Adam_Reg4	Adam_Reg4_RTD2	<input checked="" type="checkbox"/> Scaled <table border="1" style="display: inline-table; margin-left: 10px;"> <tr> <td>Max Eng Value</td> <td>Offset Value</td> <td>Max Raw Value</td> </tr> <tr> <td>500</td> <td>0</td> <td>65535</td> </tr> </table>	Max Eng Value	Offset Value	Max Raw Value	500	0	65535
Max Eng Value	Offset Value	Max Raw Value						
500	0	65535						
Adam_Reg5	Adam_Reg5_RTD1	<input checked="" type="checkbox"/> Scaled <table border="1" style="display: inline-table; margin-left: 10px;"> <tr> <td>Max Eng Value</td> <td>Offset Value</td> <td>Max Raw Value</td> </tr> <tr> <td>250</td> <td>0</td> <td>65535</td> </tr> </table>	Max Eng Value	Offset Value	Max Raw Value	250	0	65535
Max Eng Value	Offset Value	Max Raw Value						
250	0	65535						

Adam_Reg7	Adam_Reg7_TC	<input checked="" type="checkbox"/> Scaled <table border="1" style="display: inline-table; margin-left: 20px;"> <tr> <td>Max Eng Value</td> <td>3658</td> <td>Offset Value</td> <td>-32507</td> <td>Max Raw Value</td> <td>38458</td> </tr> </table>	Max Eng Value	3658	Offset Value	-32507	Max Raw Value	38458
Max Eng Value	3658	Offset Value	-32507	Max Raw Value	38458			

The scaling for type K thermocouple is based on its sensitivity of approximately 41 $\mu\text{V}/^\circ\text{C}$ and 150mV maximum range, for 32767.

The thermocouple has to be calibrated at 0 $^\circ\text{C}$ (water with ice) and it will be acceptable in the range 0 ... 100 $^\circ\text{C}$. For calibration we changed the *Offset Value* and *Max Raw Value*.

Observation: The thermocouple value is not as precise as the RTD's.

The screenshot shows the 'Tags Setup' window with the following data:

No	Tag Name	DDE Name	DDE...	Data Type	Update T...	Value
44	Adam_Out0_Write	Adam_Out0_Write	1	Integer	Write	0
45	Adam_Out1_Write	Adam_Out1_Write	1	Integer	Write	0
46	Adam_Reg0	Adam_Reg0	1	Integer	Read	0.057680
47	Adam_Reg1	Adam_Reg1	1	Integer	Read	0.001526
48	Adam_Reg2	Adam_Reg2	1	Integer	Read	0.002441
49	Adam_Reg3	Adam_Reg3	1	Integer	Read	0.001526
50	Adam_Reg4_RTD2	Adam_Reg4	1	Integer	Read	26.024261
51	Adam_Reg5_RTD1	Adam_Reg5	1	Integer	Read	52.483406
52	Adam_Reg6	Adam_Reg6	1	Integer	Read	-10.000305
53	Adam_Reg7_TC	Adam_Reg7	1	Integer	Read	55.928650
54	Counter	Reg0	1	Integer	Read/Write	485
55	Input0	Inp0	1	Integer	Read	0
56	Input1	Inp1	1	Integer	Read	0

No	DDE Ap...	DDE Topic	Con...
1	MB	TAGS	Yes
2	FESDDE	TAGS	No
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			

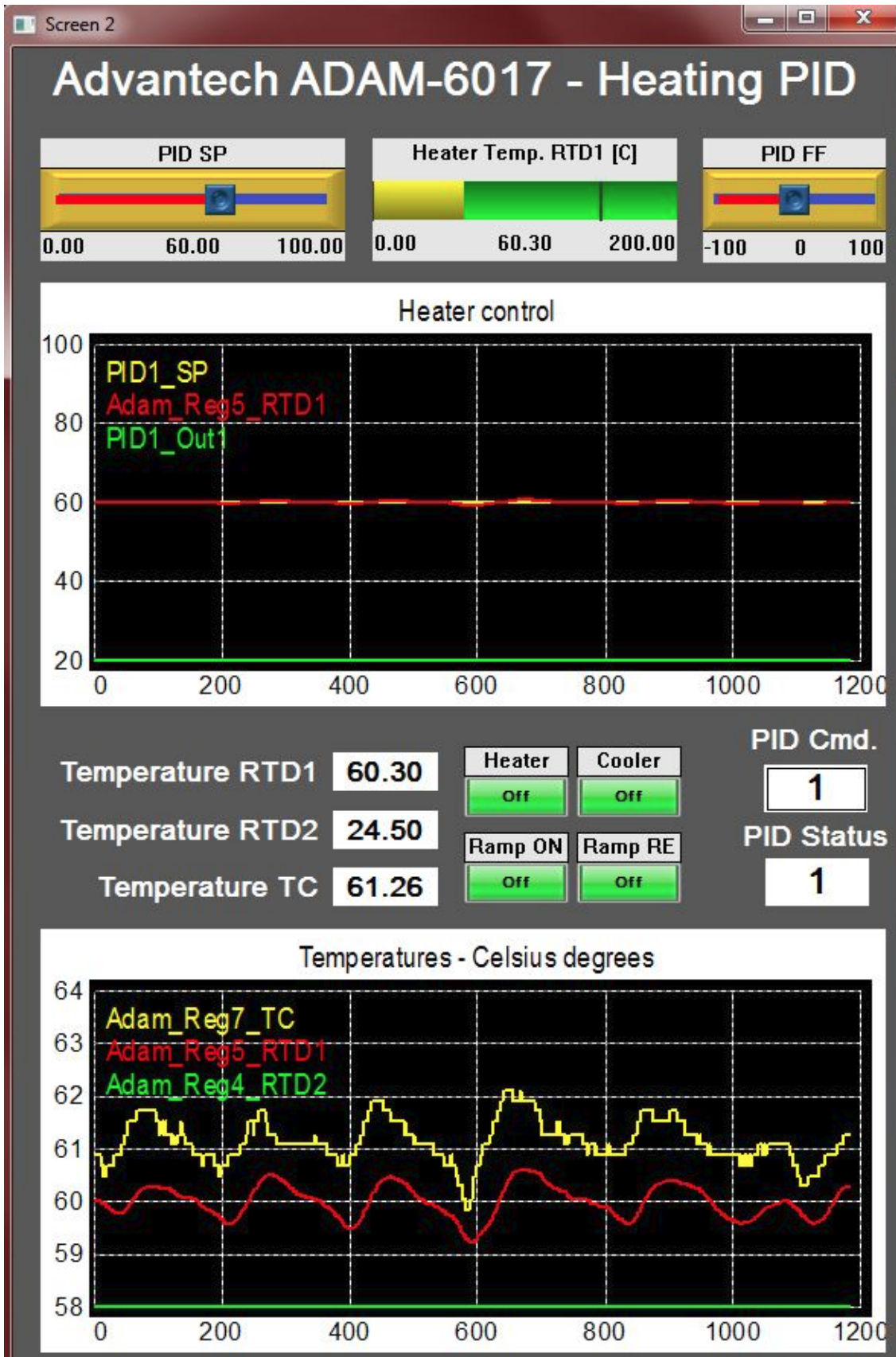
Configuration for Adam_Reg7_TC:

- Tag Name: Adam_Reg7_TC
- Data Type: Integer
- Update Type: Read
- DDE Name: Adam_Reg7
- DDE Channel: 1
- Initial Value: 0
- Scaling: Scaled

Max Eng Value	3658	Offset Value	-32507	Max Raw Value	38458
---------------	------	--------------	--------	---------------	-------

Formula: $\text{SCALED_VALUE} = \text{Max_Eng_Value} \times (\text{RAW_VALUE} + \text{Offset_Value}) / \text{Max_Raw_Value}$

Now we can build a screen (window) to show the data, and to put some indicators and buttons. In the picture below one can see a snapshot of this screen.



In FeSCADA we defined some more memory tags and a PID: *PID1_SP*, *PID1_CMD*, *PID1_STATUS*, etc. (see below).

The screenshot shows the 'PIDs Setup' window. At the top, there is a 'PIDs List' table:

No	Type	Control	Kp	Ki	Kd	In Min	In Max	Out Min	Out Max	Bias	PWM	PWM ...	SP Tag
0	PID	E=SP-PV	5	0.04	20	-100	100	-100	100	0.00	Yes	2.00	PID1_SP
1	PID	E=SP-PV	18	5.14	15.75	-100	100	-100	100	0.00	No	3.00	PID2_SP

Below the table are several configuration sections:

- Buttons:** Add, Update, Delete.
- Scaling:** The Input/Output Tags scaling is: Max = +100% and Min = -100% of PID values. Only if (OUT1 Tag <> OUT2 Tag) then: OUT1 = [0...Max] for PID output [0... +100%], OUT2 = [0...Min] for PID output [0... -100%].
- Type:** PID (dropdown).
- Control:** E=SP-PV (dropdown).
- Input Signals:** Filter [sec] = 1.000, 0...10 sec.
- Use PWM:** Yes (dropdown), PWM Cycle [sec] = 2.00.
- Parameters:**
 - Kp - Gain: 5
 - Ki: 0.04, Ti [sec]: 125.00
 - Kd: 20, Td [sec]: 4.00
 - Bias [%]: 0.00
- Application Tags List:**
 - Input SP TagName: PID1_SP
 - Input PV TagName: Adam_Reg5_RTD1
 - Input Bias TagName: PID1_Out1
 - + Output OUT1 TagName: Adam_Out0_Write
 - Output OUT2 TagName: Adam_Out1_Write
 - Command TagName: PID1_CMD
 - Status TagName: PID1_STATUS
- Output [%]:** Proportional [%] = 1.41, Integral [%] = 8.44, Derivative [%] = 0.41, Derivative2 [%] = 0.00.

When *PID1_CMD* has the value 1(one) the PID is active and it will compute the outputs to maintain the process value as close as possible to the set point value, *PID1_SP*. The set point value can be changed by the user. We read the process temperature from *Adam_Reg5_RTD1*.

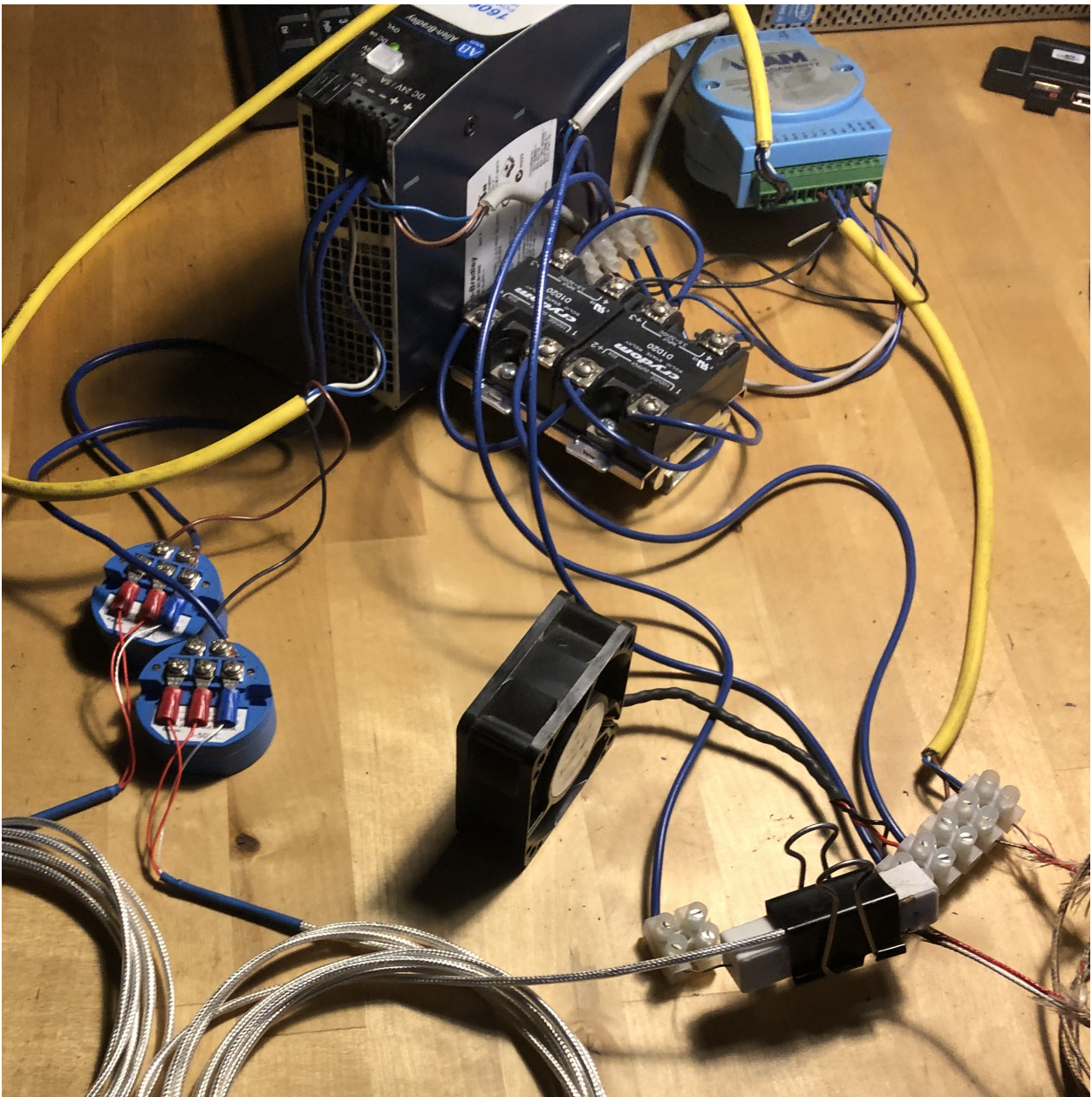
The PID outputs are using pulse width modulation (PWM). The cycle time is 2 seconds.

If the PID output is positive then the heating output (OUT1) is used in the proportion: 0...100% → 0... 2 seconds.

If the PID output is negative then the cooling output (OUT2) is used in the proportion: 0... -100% → 0...2 seconds.

OUT1 → *Adam_Out0_Write*, which is used for SSR1, current (heating),

OUT2 → *Adam_Out1_Write*, which is used for SSR2, fan (cooling).



7) Special application

We defined a ramp with the memory tags: *Ramp_CMD*, *Ramp_STATUS*, *Ramp_OUT*.

Initial value is for Time ramps.
Time is in seconds.

Initial Value: 25.000
Initial Derivative: 0.00
Final Derivative: 0.00

Initial and final derivatives are for spline interpolation in Variable ramps.

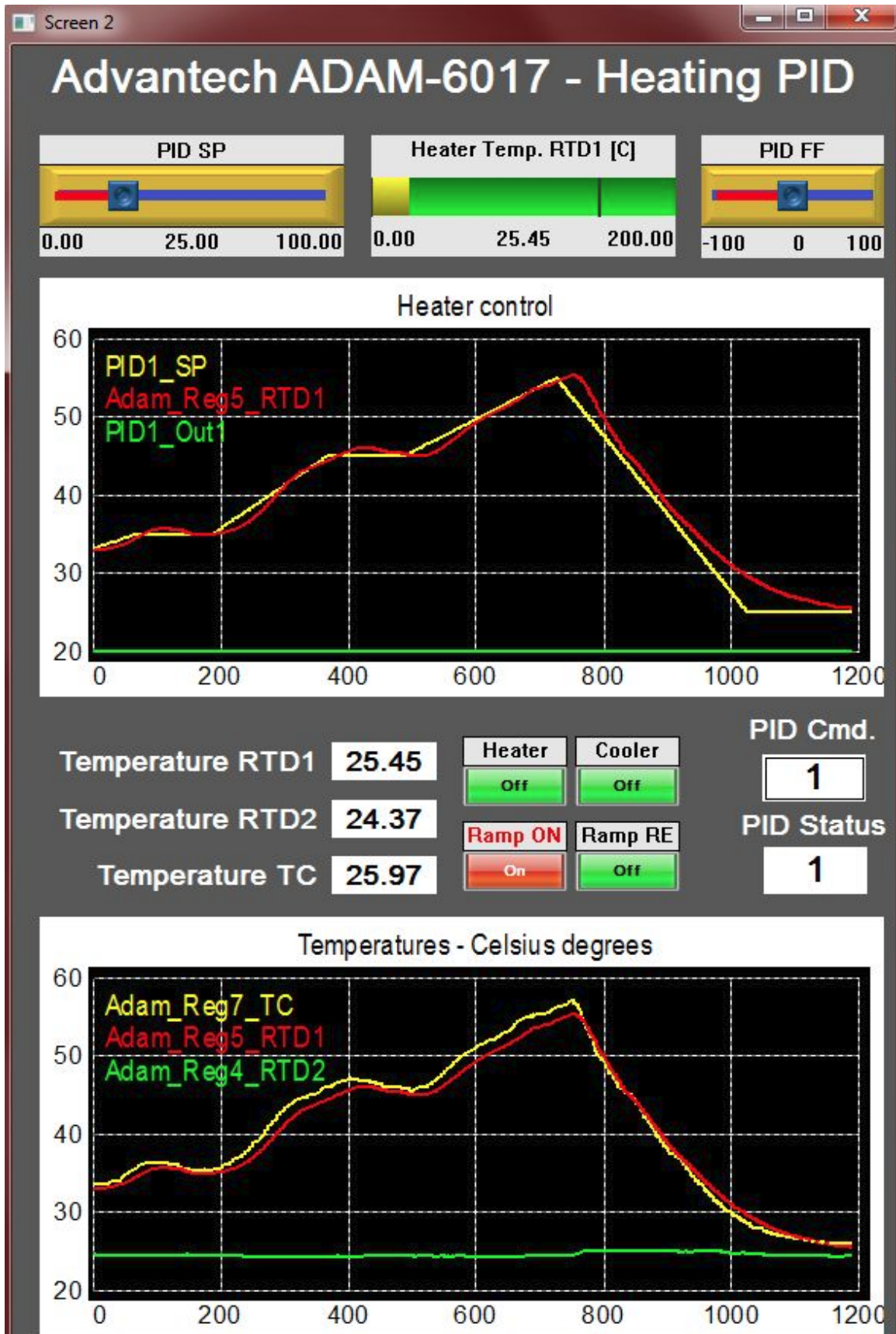
Step	Time	ValueOut
1	120	30
2	180	35
3	120	35
4	180	45
5	120	45
6	240	55
7	300	25

In the same time we defined a logic program that will copy the value of the tag *Ramp_Out* in the PID set point tag, *PID1_SP*, if the ramp is running (if *Ramp_CMD* is 1).

No	Name	Type	Cyclic	Trigger M...
0	Logic1	Timer	Yes	Different
1	Thermist...	Timer	Yes	Different
2	Buzzer	Timer	Yes	Equal
3	Remote	Timer	No	Equal
4	c2	Timer	Yes	Equal
5	PID	Timer	Yes	Equal

```

if Ramp_CMD == 1
then PID1_SP = Ramp_OUT ;
    
```



8) Conclusions

The application has shown an example of using FeSCADA and FeMODBUS to monitor 3 temperature sensors and to control a heating and cooling system with the aid of ADAM 6017 Ethernet IO module from Advantech. The communication protocol is Modbus TCP.