Technical Note 191

Adaptive Programming for BAS setpoints

BACnet points to custom logic blocks

The ACH580 and ACH180 series of variable frequency drives contain data I/O points capable of being discovered over BACnet MS/TP as analog value objects. These can be assigned to drive parameters related to functions for HVACR applications. Drives already present in HVACR equipment used to adjust the speed of fans or pumps, can also be an extension of the building automation system (BAS) by providing end users, engineers, or technician flexibility in the system design or implementation. This technical note offers examples for using several different types of drive inputs and outputs along with Adaptive Programming to build custom programs or manage BAS setpoints similar to that of a building controller.

The system design in these examples will be using an ACH580 drive as the main controller for a rooftop air handling unit, see Figure 1. The drive will be wired for temperature sensor inputs and be programmed to control supply fan speed, cooling, heating and damper outputs as well as receiving setpoints communicated over BACnet MS/TP. To see more system details and programming of how an ACH580 drive can control a rooftop unit (RTU), refer to ABB Technical Note 137.

In a BAS, operators are provided adjustable setpoints to optimize their HVACR equipment for energy efficiency and occupant comfort. The drive is capable of receiving setpoints and writing the values to several parameters in different functions similar to BAS unit controllers. This will be done using the drive's Adaptive Programming, a custom programming tool. With this and additional programming referenced in Technical Note 137, an RTU unit controller could then be replaced by the drive saving hardware cost. Another benefit of using the drive as the main controller for an RTU is reducing the number of BACnet points required to be discovered by the BAS. System integrators may charge by the number of BACnet points they have to discover or program. By programming and managing more BACnet points within each drive, this has the potential to reduce costs in the field.

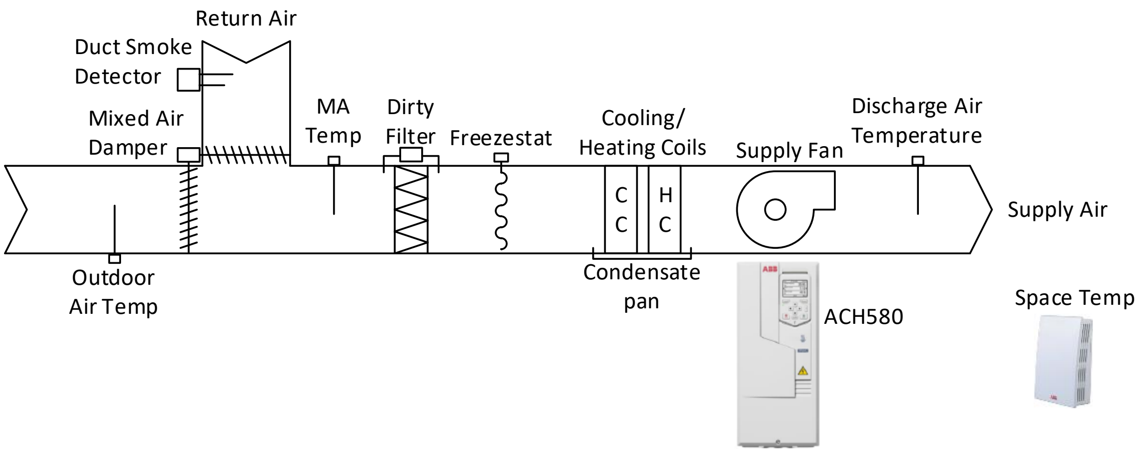

Figure 1: Flow diagram, single zone VAV, chilled and hot water coils

Getting started
To work with Adaptive Programming, you will need Drive Composer, entry or pro. This software is available online for download.

https://new.abb.com/drives/software-tools/drive-composer

To begin programming you will need to be connected to a powered drive. Using Drive Composer, connect your computer through either USB cable, BCBL-01 programming cable, or Bluetooth to the drives control panel. For steps to configure your drive for BACnet MS/TP see ABB Technical Note 044 or the drive's firmware manual. Once Drive Composer has discovered the drive, select the drive name from the navigation tree on the left and select Parameters, see Figure 2. Scroll down to parameter group 96 System and set Disable adaptive program (96.70) to No, see Figure 3. Next, again click on the drive name on the navigation tree to select and open the Adaptive Programming tab.

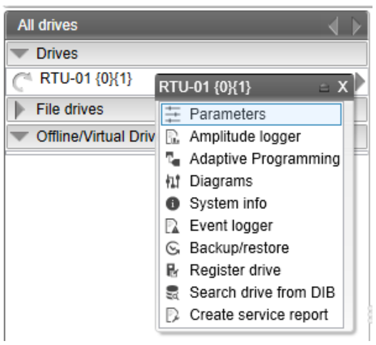

Figure 2: Drive Composer Options

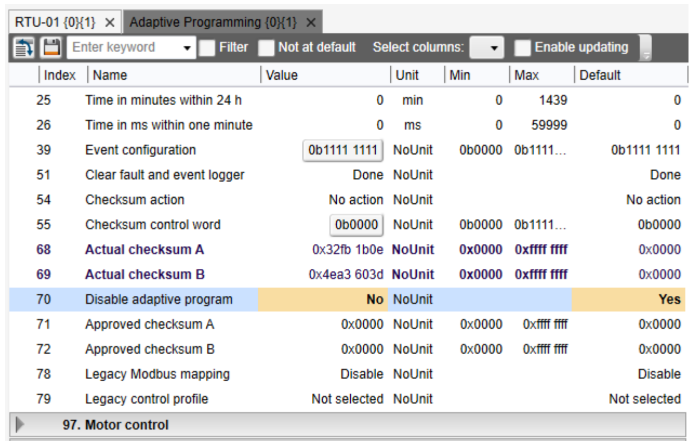

Figure 3: System parameters (group 96)

Basics of Adaptive Programming
Adaptive Programming is a graphical logic block programming tool available with the ACH580 and ACH180 drives. Here, a technician or end user can develop custom programs or sequences tailored to the specific needs of an application. For example, writing a space temperature setpoint and offset to the drive that is used to determine occupied heating and cooling setpoints for space temperature control and heat/cool mode changeover.

Adaptive Programming is laid out in sections with titles. The Base program will contain inputs, outputs and functional blocks. A program is created by connecting inputs to blocks, blocks to other blocks, and block to outputs. There are two types of programming areas, the Base program and Sequence program. The Base program, Figure 4, runs every 10 milliseconds once downloaded to the drive and set to run. This program is useful for continually monitoring inputs and controlling outputs that need to change often.

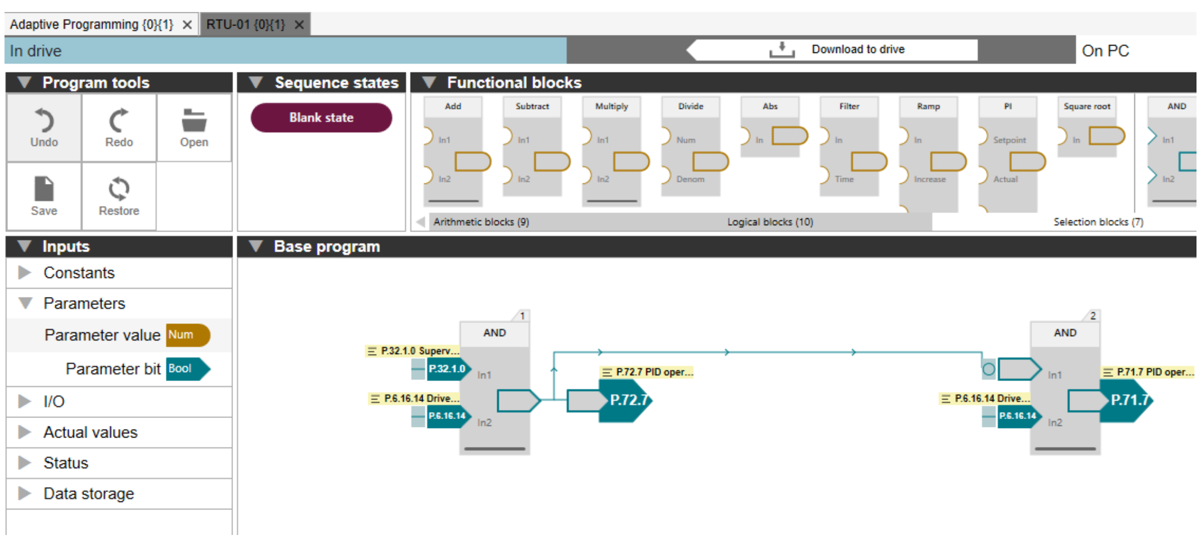

Figure 4: Adaptive Programming sections and Base program

A Sequence program contains State transitions that act like functions or sub-routines along with inputs, outputs and functional blocks. The first state S1 can include triggers to transition into another state, see Figure 5. In one or more of the other 4 state programs, a function can trigger a transition to another state or back to state S1. This can allow for single execution of a program that can be used for counters or limiting the number of times a function blocks reads or write as will be used in an example below.

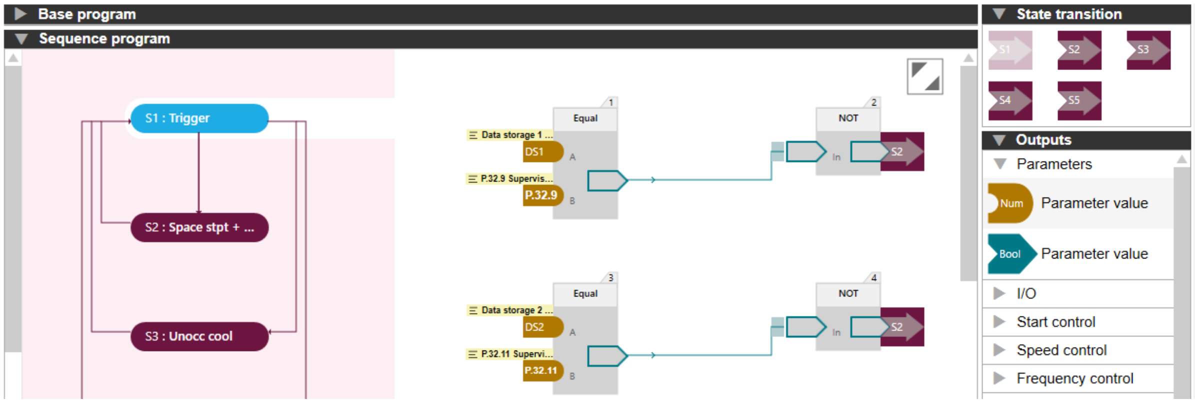

Figure 5: Sequence program with state transition flow diagram

Building automation programming examples
To aid in understanding how to apply Adaptive Programming with BACnet MS/TP to read data from and write data to the drive, below are some example applications of a drive controlling an RTU. For BACnet MS/TP, the drive has predefined analog, binary and multistate objects available to be discovered. However, not all drive parameters are mapped to BACnet objects. The embedded fieldbus Data-I/O points will be used for the following examples, which are BACnet objects AV120 through AV129.

1. Write to multiple drive parameters using one BACnet AV
2. Calculate separate drive setpoints from two BACnet AVs
3. Digital input toggle counter value to a BACnet AV
4. Control a drive analog output from different drive sources
5. Sequence program, control parameter changes on triggers

BACnet setpoints setup
To use the BACnet analog value objects, assign parameters to the Data I/O points in group 58 Embedded fieldbus. The following examples will use the parameters selected in Figure 6. Supervision 1 Low (32.9) for Data I/O 1 which is BACnet object AV120 will be used as the space temperature setpoint. Supervision 1 hysteresis (32.11) is set to be Data I/O 3 which is BACnet object AV122 and will be the space temperature offset or deadband. Depending on the parameter you choose to use, the size of the value may require one or two data I/O points. Parameters 32.9 and 32.11 are currently 32-bit words and require data I/O points 1-2 and 3-4 respectively. External PID 3 output min (73.36) will be for Data I/O 5 which is BACnet object AV124 and is used for the outdoor air damper minimum position. Data storage 1 (47.1) is set for Data I/O 6 which is BACnet object AV125 and will store a counter value.

Index	Name	Value	Unit	Min
47	AV21 & AV22 unit	**AO unit**	NoUnit	
101	Data I/O 1	**32.9[32]**	NoUnit	
102	Data I/O 2	**None**	NoUnit	
103	Data I/O 3	**32.11[32]**	NoUnit	
104	Data I/O 4	**None**	NoUnit	
105	Data I/O 5	**73.36[16]**	NoUnit	
106	Data I/O 6	**47.1[16]**	NoUnit	
107	Data I/O 7	None	NoUnit	
108	Data I/O 8	None	NoUnit	
109	Data I/O 9	None	NoUnit	
110	Data I/O 10	None	NoUnit	

Figure 6: Assign drive parameters to group 58 Data I/O for BACnet AVs

Write to multiple drive parameters using one BACnet AV

The first example is simply connecting the space temperature setpoint from BACnet AV120 for the Supervision 1 Low (32.9) parameter to also be written to the Process PID Set 1 and Set 2 Internal setpoint 1 (40.21, 41.21), see Figure 7. Now both the supervision function and process PID setpoints can be updated with one change from the BAS. Why do this here in Adaptive Programming instead of using three Data I/O points and three BACnet AVs? Since the drive is limited to only ten of these mappable analog values objects, we have the ability to use more of them if values can be written to multiple parameters. Also, the system integrator or controls technician now only has one BACnet AV to write a setpoint to instead of three.

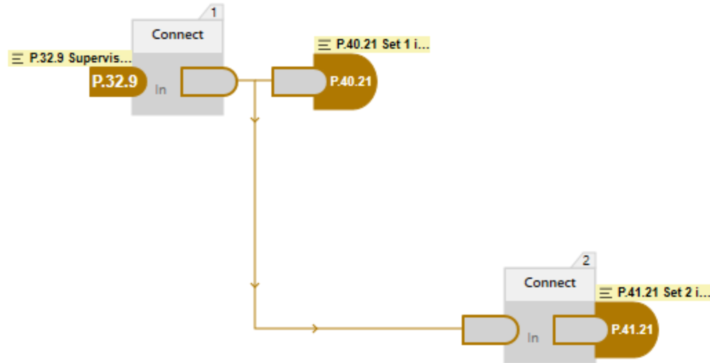

Figure 7: One BACnet AV written to three drive parameters

Calculate separate drive setpoints from two BACnet AVs

This example adds a space temperature offset from BACnet AV122 to the previous example and will be used to write separate space cooling and heating setpoints to the process PIDs, see Figure 8. Again, the space temperature offset is also used for Supervision 1 hysteresis (32.11). First, the value needs to be divided by two to become a plus and minus offset from the space temperature setpoint (32.9). This is done automatically as part of the supervision hysteresis function. Then the offset is added to the setpoint and written to the Process PID Set 1 internal setpoint 1 (40.21) for the cooling mode. Next, the offset is subtracted from the setpoint and written to the Process PID Set 2 internal setpoint 1 (41.21) for the heating mode. Now each process PID can be set to separate setpoints as well as the supervision function with just two BACnet AVs from the BAS.

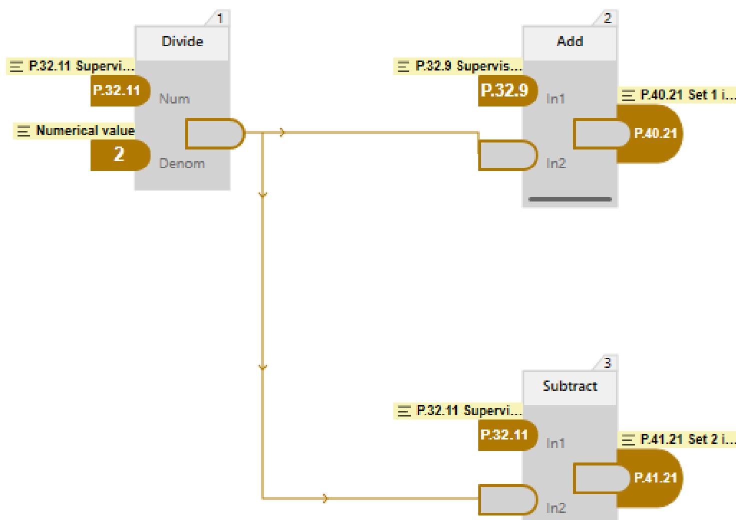

Figure 8: Use two BACnet analog values and write to four drive parameters

Digital input toggle counter value to a BACnet AV

Counters can be useful to better understand the operation of the RTU or troubleshooting issue with inputs, outputs or programming. The drive has toggle counters for the three relay outputs that can provide insight into how often the drive may be turning another device on, off or going into a fault. In this example, digital input DI4 could be connected to a freezestat used to monitor the air temperature entering a cooling coil. When the freezestat contact opens, this will increment the counter and the value will be saved in Data storage 1 (47.01). There is now a record of how often the freezestat switch changes and could be used to address an issue with low air temperature or nuisance shutdowns.

Select the left edge of a binary or Boolean input to reverse its logical input. A dash is normal and a circle is reversed, see Figure 9. In this case, safety interlocks are default normally closed for the drive at DI4, so this input will be true when DI4 is open. The Trigger up block is used and its output is true only once when this DI4 input is true. The output will then be false on the next program cycle so the counter only increments once for each toggle of DI4. The Switch value block output changes between the In1 input when Sel1 is true and the Default input when the Sel1 input is not true. The Switch value block sends a one to the Add block and increments the Data storage 1 value or a zero to stay the same. The data storage parameter can then be assigned to a Data I/O parameter in group 58 and the counter value read from a BACnet AV.

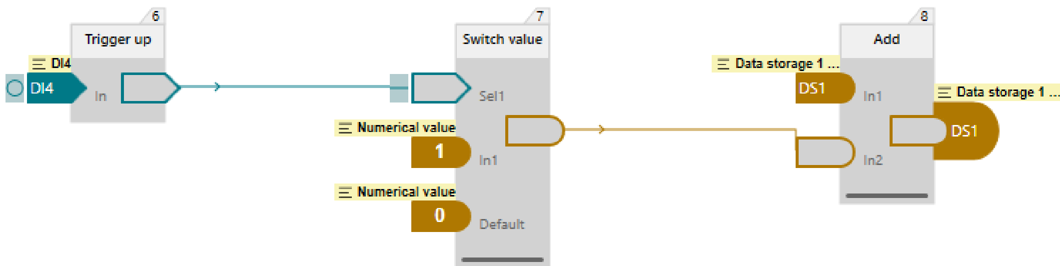

Figure 9: Counter program to monitor for digital input DI4

Control a drive analog output from different drive sources

This example uses one of the drive's External PIDs to control an analog output to modulate the RTU outdoor air damper position. The drive will vary the damper position in response to External PID 3 used for fresh air economizing or free cooling. When economizing is disabled, the analog output needs to be set to the output minimum of External PID 3 which will be used as the outdoor air damper minimum position for ventilation requirements. When the RTU is set to unoccupied, the damper output will then be zero or closed.

In Figure 10, this example uses two AND blocks and one Switch value block. The first AND block requires four inputs to be true then it will enable External PID3 (73.07). The first input is when economizing is enabled from the BAS, communicated to the drive through BACnet object BV34. In the drive, this is User0-command and to read the status Adaptive Programming is looking at the drive status word, bit 11 (6.11.11). For detailed steps to assign these BACnet objects, see ABB Technical Note 130 User-configurable BACnet Objects.

Next, Supervision 1 status (32.1.0) is used to control heat/cool mode following the space temp setpoint examples above. When the supervision status is not true, indicated by the circle left of the input, the RTU is in cooling mode which is required for economizing. The third input is drive running status from Drive status word 1, bit 14 (6.16.14) and this is used to only enable economizing when the supply fan is running. The last input is the status of Ext2 active, which can be commanded in the drive at BACnet BV13 (19.11 set to EFB MCW bit 11). Ext2 active will correspond to when the RTU is occupied while Ext1 will be when unoccupied. When these four inputs are true, the external PID is enabled.

The Switch value block is then used to pass one of the inputs to the output depending on which switch input is true or false. In our example, when the first AND block is true the External PID3 output (73.01) will be written to analog output AO3 (15.82) of a CAIO-01 extension module. If this input selection is not true then the Switch value block checks to see if the second AND block is true. This is when only Ext2 is active and the drive running status is true rather than the conditions from the first AND block; RTU occupied and the fan is running but either heating or economizing disabled. Then the output minimum of External PID3 (73.36) from BACnet AV124 will be written to AO3. When neither AND block is true, the default output of the Switch value block will be a constant value of 0.

An important note about Adaptive Programming, the order of operation is indicated at the top of each function block. The number increases as you add blocks to the right and down. So, for the example in Figure 10, the block number 3 operation will happen first, then block 4, then block 5. This can be important when the output of one block needs to take place before the action or output of another block.

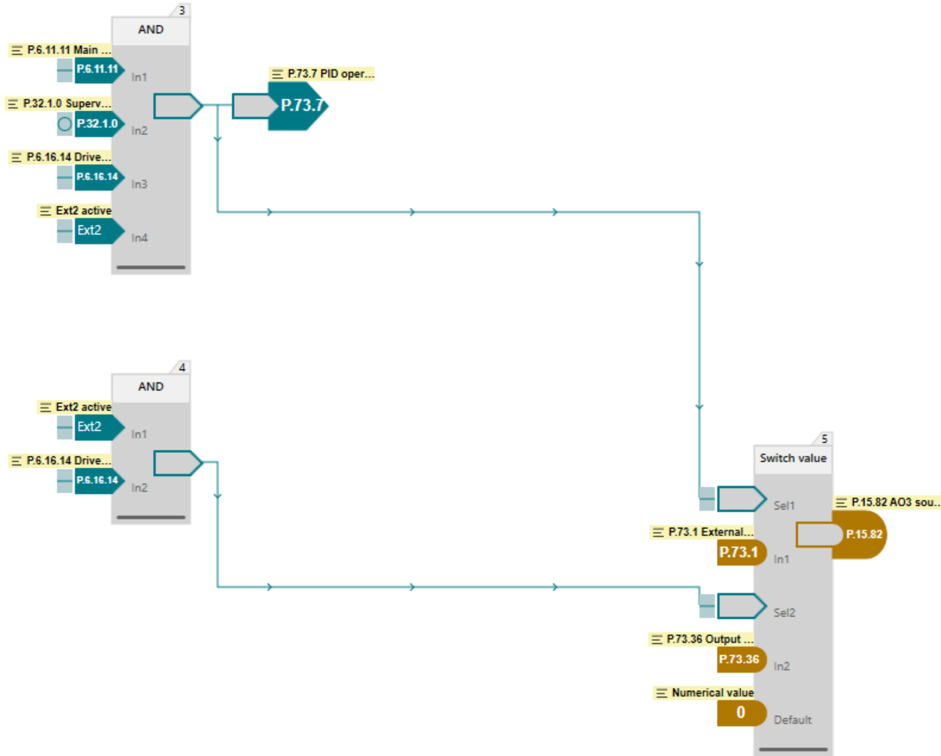

Figure 10: Analog output value changes depending on switch input selection

Sequence program, control parameter changes on triggers

One example for using sequence programming could be a value sent to the drive is changing frequently, such as an analog input. In this example, the analog value being used for space temperature setpoint is now coming into the drive through analog input AI1 rather than a BACnet AV. This could be a digital space temperature sensor with local adjustment for the occupants. In this case, a voltage or milliamp signal may vary slightly causing the parameters in the drive to be changed frequently.

To limit the number of times a new value is being written to a parameter, a trigger can be used to only write the analog input value when the change is greater than a certain step size. In Figure 11, the scaled value for AI1 (12.12) is set to equal space temperature setpoint in degrees Fahrenheit. The setpoint then has the value saved in Supervision Low 1 subtracted to output the difference between the two values. The Abs block is for converting the absolute value from the Subtract block, if the new setpoint is less than the previous and the output is negative. When the difference between space temp setpoint and a new value at AI1 is greater than 0.5 degrees, this will transition the sequence program into state S2.

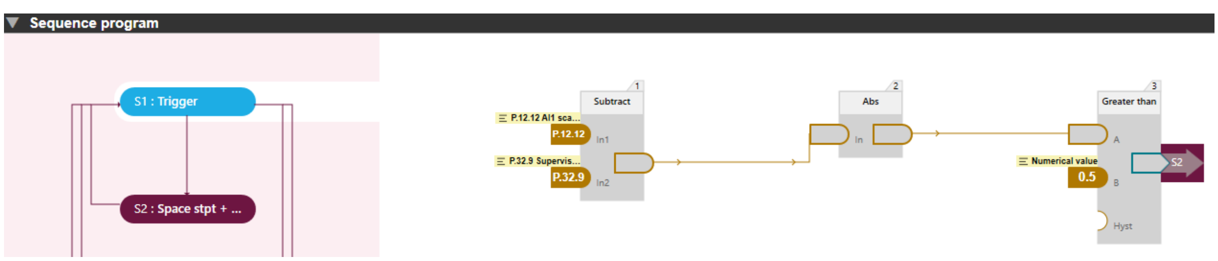

Figure 11: State 1 program to trigger other programs

When the program has moved to state S2, see Figure 12, the input values for the setpoints and offset are written to the drive parameters. After that, the sequence moves back to state S1 and waits again for the input values to change beyond the trigger.

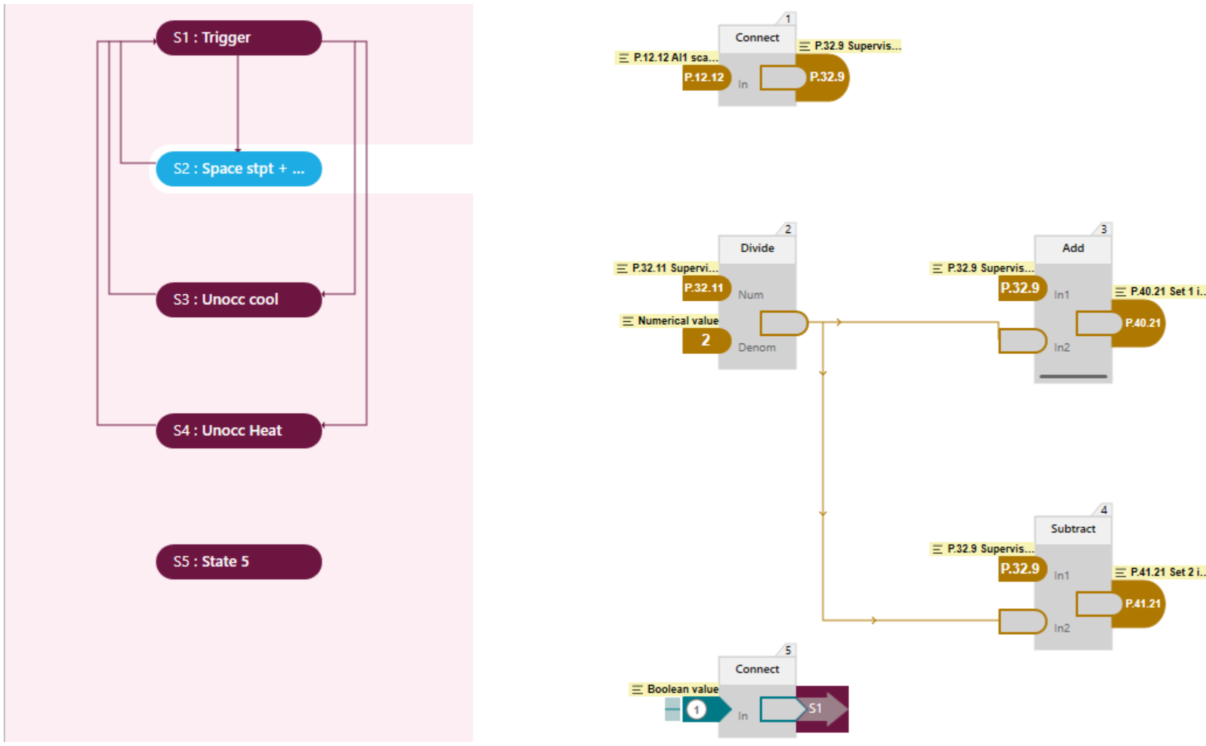

Figure 12: State 2 program to update parameter values and return to state 1

Summary

The ACH580 and ACH180 series of drives contain input, outputs and programmable functions that can be used to operate HVACR systems similar to that of a building automation system controller. Beyond just sending the drive a start, stop and speed command, features and capabilities of the drives can be leveraged to expanding the BAS into the drives as well. Through both physical inputs, output and communication points the drives can become an extension of the BAS or a means to simplify the system by reducing components. Adaptive Programming allows for advanced functions to be programmed within the drive and with proper inputs or outputs act as a standalone controller for the HVACR applications.