Controls Fundamentals Part 1: Control Points

…As Explained Like an Online Recipe (but you can't "Jump to the Recipe")

By Danny Richardson, PE

Have you ever wondered how a building controls system works? Or perhaps a better question is, have you ever walked into a building and wondered what the heck the BAS is doing? I've been there and I've been designing controls systems for over a decade. So, what's an engineer, architect, building manager, and/or concerned individual to do?

Join me on this journey to understand the vast world of HVAC controls. This series of articles will explain the fundamentals of controls systems, their purposes, and how to be successful with your project goals.

As a warning, this article is going to focus primarily on control points. This may seem like a boring topic, but it's critically important in understanding building controls systems. The important piece is learning **how to classify control points**. You can even practice classifying control points by looking at HVAC equipment, or building systems, or even devices in your own home! What kind of control point is a light switch? What about a thermostat?[1]

The Basics

For the uninitiated, the BAS is the Building Automation System, also known as the controls system. There are many other names and acronyms for this: **EMS** (energy management system), **FCS** (facilities control system), **BMS** (building management system), the list goes on. As Shakespeare once said, a controls system by any other name would operate the same.[2] Whichever name the system uses, the operating principles are the same.

Let's start with the **purpose** of a controls system. Why do they even exist? The core principle is to *reduce* energy usage in the building. Stories within our industry say that controls systems started gaining popularity in the 1970's, after the 1973 Oil Crisis, and the original purpose was just to turn stuff off at night to save energy. Very basic, but it worked! Later, more complex controls strategies were developed, such as turning stuff off on the weekends.[3]

Modern control systems primarily use electronic controls referred to as **DDC**, or direct digital controls. DDC controls can be programmed to do what we want. The previous generation of electronic controllers typically didn't allow full programming; they would only allow specific parameters to be changed, which would limit the controls systems. The programmability of DDC gives designers much more flexibility in achieving proper control, including efficiency and keeping people comfortable. It also allows operators to view real-time information, make changes on the fly, and receive alarms when stuff isn't quite working properly. So, how does all of this work?

---

[1] There are unconfirmed reports that the author is really fun at parties.

[2] "Like many things, the trouble with quotes on the internet is that you can never be assured of their authenticity." – Abraham Lincoln, 1863.

[3] The author is being facetious here, but the advent of the time-clock and scheduling was a massive advancement for controls that we rely on today!

Everything in a controls system comes down to its points (we finally got here!). **A point is any piece of data, command, or setpoint that the controls system interacts with[4].** For example, a temperature reading is a point, and so is the command to run a fan VFD at 50% speed. The term "point" is very broad and all-encompassing, which can be confusing because everything will appear to be a control point.[5]

Inputs and Outputs

Points can be divided into different categories based on what they do, and how they do it. What a point does can be classified as an **input** or an **output** – does the controls system read the data point, or is it telling a device to do something? Inputs are typically sensors and switches, such as a temperature sensor or a high limit pressure switch. Outputs are commands, so the controls system is telling a device to do something – like open a valve, turn on a fan, or slow down a pump.

When thinking about inputs and outputs, it is always based on the perspective of the controller. Is the controller reading something (such as a temperature measurement), or is it commanding something (such as turning on a fan)?

An example of an output command is to turn on a fan. A corresponding input could be an airflow proving switch (or maybe just feeling the wind in your hair).

Analog and Binary Points

How a point works can be classified as either binary or analog. **Binary points** are exactly as the name implies: it's either a 1 (work) or a 0 (don't work), just like a binary bit. These points can be commands (outputs) such as on/off or open/close, or they can be data points (inputs) such as a switch. Examples of binary inputs include airflow proving switches (they only send a signal when there is airflow) and high limit pressure switches (they only send a signal when the duct pressure is too high). Examples of binary outputs include commanding an actuator to open or close, or commanding a piece of equipment to turn on or off (such as a fan or pump).

Analog points are a bit more exciting – they can operate across a range from 0% to 100%. This allows the controls system to have more detailed inputs, and to have finer output control over the devices that need it. Analog points can also be inputs or outputs. Some common analog inputs are temperature sensors, pressure sensors, and airflow measuring stations. Common analog outputs are modulating actuator commands (such as economizer dampers or control valves) and VFD speed commands.

---

[4] The poet John Dryden would hate this author's writing because prepositions can go *wherever I want them to*.

[5] Spoiler alert: everything *is* a control point.

Now that we know what points are, let's ask the really important question: what do we do with all of those points? If you have read the author's previous article[6], you'll know that the answer to everything is "it depends." The answer here depends on what we want to do with our building, how we want the HVAC system(s) to operate, and what we want from the controls system.

Let's look at an example of a packaged DX rooftop unit (RTU) that is constant volume, no economizer, no exhaust, one compressor, and a gas furnace. Units like this are generally understood to be *working* or *not working*.[7] However, there is a lot more going on under the hood! The diagram on the following page provides a visual aid for the various points discussed below.

We start under the outside air (OA) intake hood, where the damper actuator for the OA damper lives. That actuator requires a **binary output** command to open/close the OA damper. If this unit had an economizer instead, the actuator would require an **analog output** since we would need to modulate the OA damper: closed, minimum OA for code requirements, and fully open for economizer. Note that if the unit has a separate return air (RA) damper actuator, it will also require a point to match the OA damper actuator.

Moving along to the next component in the system, we have the filters. Typically, packaged RTUs won't have a dirty filter switch. But, for the sake of learning, let's assume this unit does have a dirty filter switch. That would be a **binary input**, since it sends a signal only when the static pressure differential across the filter gets too high.[8] The alternative would be a differential pressure sensor, so that the BAS could read the actual pressure differential across the filter in real-time. In that case, it would be an **analog input** since the reading would have a range (not just dirty/not dirty).

The next thing in our packaged RTU is the DX cooling coil. The coil itself doesn't have anything that we can control, but it's connected to something that we can: the compressor. In our hypothetical unit, let's assume there is a single compressor. We would need to tell that compressor to be on or off, which is a **binary output**. If desired, we could have the controls contractor add a current transducer to the compressor power wiring so we can see if the compressor is running or not. This would be a **binary input** point to the BAS, and it would be helpful for making sure the unit is working. For example, if we're telling the compressor to run but it doesn't pull any power, then we know something is wrong.

Next up in the unit is the gas furnace. This is an easy one from a controls perspective. Most gas furnaces will have two stages of heating. The controls system would be commanding each stage of heating to turn on, which makes both stages **binary outputs.**

---

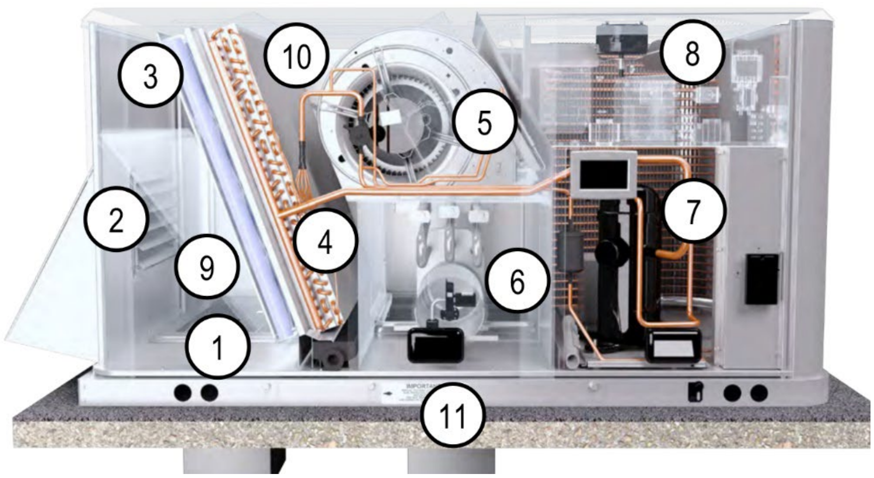
[6] Battle Royale: Air-Cooled Vs. Water-Cooled Chillers, https://eeace.com/battle-royale-air-cooled-vs-water-cooled-chillers/

[7] Both of these states are generally assumed to be the fault of the mechanical engineer.

[8] A dirty filter switch works by measuring the differential pressure across the filter, which increases as the filter gets dirty. This means that it becomes harder to pull air across the filter as it holds more dust particles. The switch is set to a pre-determined value so that when the filter is at a specific pressure drop (or "dirtiness"), the switch will close and send a signal to change the filter.

In larger and/or more complex units, there is the possibility of having a modulating gas burner, which would require a single **analog output** to command.

The last main component in the packaged RTU is the supply fan. I mentioned that our example unit will be constant volume, meaning the fan will have only one speed: on. If the fan needs to be on, the BAS will send it a signal. If the fan needs to be off, the BAS just removes that signal. Therefore, the point would be a **binary output** to command the fan on/off. Just like with the compressor, we could put a current transducer on the power wiring to the fan motor to monitor its status. And just like with the compressor, that status monitor would be a **binary input**. As an exercise, what points would be required if this fan instead had a VFD controlling its speed? Well, we would need to give the VFD a speed command which can range from 0% speed (0 Hz) to 100% speed (usually 60 Hz)[9], so that would be an **analog output.**



No.	Description	Point Type	Why?
1	Return Air Damper Actuator	Analog Output	Command the actuator for damper position.
2	Outside Air Damper Actuator	Analog Output	Command the actuator for damper position.
3	Filter Switch	Binary Input	Dirty filter switch alarms when the filters are dirty.
4	DX Cooling Coil	--	See compressors (7) below.
5	Supply Fan	Binary Output	Command the fan on or off.
6	Gas Furnace	Binary Output	Command the furnace on or off (one output per stage).
7	DX Compressor	Binary Output	Command the compressor on or off. Serves DX coil.
8	Condenser Fan	Binary Output	Typically, we don't directly control this. Runs with the compressor.
9	Mixed Air Temperature	Analog Input	Temperature sensor reading.
10	Coil Leaving Air Temperature	Analog Input	Temperature sensor reading.
11	Unit Discharge Air Temperature	Analog Input	Temperature sensor reading.

Image used is from The Trane Company: System Catalog – Zoned Rooftop Systems, July 2022. https://elibrary.tranetechnologies.com/public/commercial-hvac/Literature/Product%20Catalog/APP-PRC005J-EN_07152022.pdf . **Annotations are by EEA**

---

[9] The speed range for VFDs is somewhat complicated. A VFD can go down to 0%, but in reality, 30% (around 18 Hz) is much more common to protect the motors. On the other end, 100% speed at the VFD is noted here as "usually 60 Hz" because it is becoming increasingly common to have fan motors designed for speeds above 60 Hz. This is more common in fan arrays.

At this point, you might be wondering: what about all the temperature sensors? I've saved those for the end since we can round them up quickly. Each temperature sensor would be giving a temperature reading to the BAS, so each one would be an **analog input**. We could have a mixed air temperature sensor, a cooling coil leaving air temperature sensor, and a unit discharge air temperature sensor. How do we know which ones we need? That is a much larger "it depends" answer, but in general, we want to prioritize our points based on:

1. What points are needed to properly control the unit
2. What points would be nice to have
3. What points are unnecessary (and therefore don't need to be included)

Since this is a constant volume RTU, technically it is not necessary to have any internal air temperature sensors to control the unit (the unit will only control to room temperature). However, it would be quite nice to have all of those temperature sensors included, as that would greatly help with monitoring and diagnostics if something goes wrong. You could also say they are unnecessary and therefore they can be ignored for this unit.

To summarize everything up to this point, there are a total of 4-point types: binary inputs, binary outputs, analog inputs, and analog outputs. Determining which point types are needed is based on how the equipment needs to operate, and which devices will be used. A table of common point examples is below:

Binary Inputs	Analog Inputs	Binary Outputs	Analog Outputs
Alarms	Temperature Sensor	Start/Stop Command	Actuator Position Command
High/Low Limit Switches	Airflow Measurement Station	Open/Close Command	VFD Speed Command
Dirty Filter Switch	Water Flow Meter		
Freezestat (Switch)	Humidity Sensor		
Current Transducer (Status)	Pressure Sensor		

This concludes our first article on **controls fundamentals**. Understanding controls points is the foundation of understanding how controls systems work, and how we can make them work for us. Did you make it this far? If you did, join us next time in **Part 2** to learn about controllers, why they're important, and how they communicate with each other to make the system work!