SIMPL Tutorial

---------------------------------------------------------------------------------------------------

Overview\*\*\*

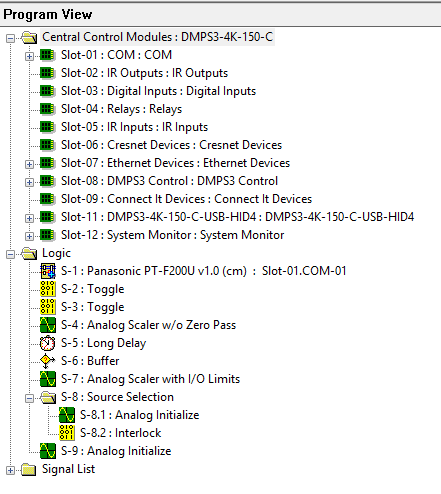
SIMPL is Crestron's main app for programming its systems. It provides the “backend” functionality for setups, and is usually worked in conjunction with VTPro-e, which provides the GUI for the different devices connected to a Crestron setup. While going through this tutorial, if you need to dive deeper into any given subject, click on what you need to know about and push F1. This will pull up Crestron’s manual on the selected subject, and their manuals tend to be very comprehensive (albeit a bit dry and without application, although sometimes they have examples). The purpose of this tutorial is to get you familiar with how Crestron systems are set up, and how the Crestron suite of software works all together. It also will provide several use cases to help you get familiar with how to do things.

# New project steps:

1. Create File -> New SIMPL Program
2. You should see a panel called Device Library on the left of your screen. If this is not the case, you may need to switch to the system view. This is done by clicking on in the top toolbar. System view is not used very often, mostly when you add or remove devices from your program. In order to start programming we need to describe to SIMPL what our setup is. We start by picking our control system. In our case it is a DMPS but it could be any Crestron device with a processor in it (i.e. MPB-10). In the device library pane, go to Control Systems and select DMPS3-4k-350-C. Drag this over to the right pane that is asking for a control system.
3. A gray bar will appear at the top of the screen, and an expandable DMPS label will appear in the pane where you dropped the DMPS. If you expand this, you will see the various parts of a Crestron control system.
4. COM port represents the communication port on the back of the DMPS. This is also the case for IR Outputs, Digital Inputs, Relays, Ethernet Devices, and IR Inputs.
5. Ethernet Devices are any Crestron (or other brand) devices connected to the DMPS3 via Ethernet. (i.e. TSW-1060)
6. DMPS3Control – the devices inside this control where audio, video, and communication go.
   1. Device Control – unused so far
   2. AV Control – controls audio and video
7. Unused so far
8. Look at the back of the DMPS unit, and see where the different devices in your system connect to it. Drag and drop these devices into the appropriate slots on the DMPS (for example, the touchpad into Ethernet devices)
9. The project is now setup, and it is now time to switch to the programming view. To switch to the programming view click on the yellow circuit in the top toolbar 

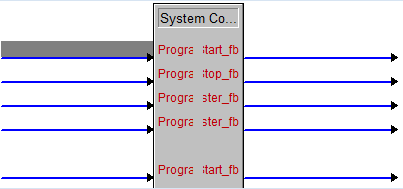
# Program View:

1. When program view is entered you should see three separate windows, **a symbol library**, a **program view**, and a **detail view**. The symbol library contains modules and logic symbols to use in your program. These are essentially digital circuits or very tiny programs that take in digital and analog signals and output the result of their function. The program view represents your system and its components, along with your programs logic symbols and modules. This is where you add logic gates and modules etc. to your system. You add them by dragging them from symbol library, or typing the logic gates name and pressing enter while having the program view selected. You can add things to the logic folder, and they should appear in it once added as shown below.



The signal list folder is SIMPLy a folder that contains all the names of your signals, which are described in the next step.

1. The **detail view** shows the computers representation of logic gates and modules (called SYMBOLS). Inputs are on the left of the gray boxes, and outputs are on the right, as shown below.



If the arrows from the left are filled with white, this means additional signals can be created. You can add them by pressing Alt + ‘+’ (without the ‘’). An example of where this happens is touchpad buttons. Since touchpads can have *n* buttons, you can add buttons. All these lines are called ***SIGNALS****.* There are digital signals and analog signals (and serial signals). The digital signals are represented in blue, and the analog signals are represented in red (serial in black). If you come from programming, digital is like a boolean, analog is like an int, and serial is like a string. All signals have a *DRIVING SOURCE*, and *DESTINATIONS.* Driving source is where a signal is created and given value, destination is where it ends up. To create a signal, click on any of the empty lines connected to a module and start typing. This will name the signal. If you name something to the left of a module, this is a signal destination. Naming something on the right is naming a driving source. Signals can become very messy very quickly, so it’s good to come up with a naming scheme and stick with it. We name signals in the following pattern – *Wheretheycomefrom\_whattheydo*. For example, MPB10\_BTN1\_TVPowerOn. It doesn’t matter what naming scheme you adopt - but be consistent with it or you’ll get confused quickly. Try and avoid naming multiple driving sources the same thing, it becomes hard to predict what the system will do very quickly. It will be very common, however, for signals to have multiple destinations, and this is A-OK.

***TIPS FOR SIGNALS/SYMBOLS***: To comment out a signal, name it //. You can also right click modules and select the “make complete” option. This will comment out unused signals and is useful on occasion. WITH A SIGNAL SELECTED, pushing F2 will show its driving sources and destinations, which can be very clarifying. If you select a signal and push F3, it will highlight the signal red everywhere it is on the screen. This is valuable if you are trying to trace a signal through your program.

# SIMPL+:

SIMPL+ is something meant to complement SIMPL, not replace it. SIMPL+ allows you to program your own logic gates with a C like syntax. You can then use these modules in your program like you could an AND gate. This is useful for things that can be tricky to do with pure digital circuitry. For example, I used a SIMPL+ module to slowly ramp the volume, so that when the program unmutes it fades in, opposed to cutting to full volume immediately. The guide for SIMPL+ is very comprehensive and included in the knowledge DB. It consists of a SIMPL+ Guide, and a language reference guide. When used in conjunction these give you everything you need to know. Most common features will not require SIMPL+ modules, but if you want to add flair to your program they are a good route to go.

# VTPro-e:

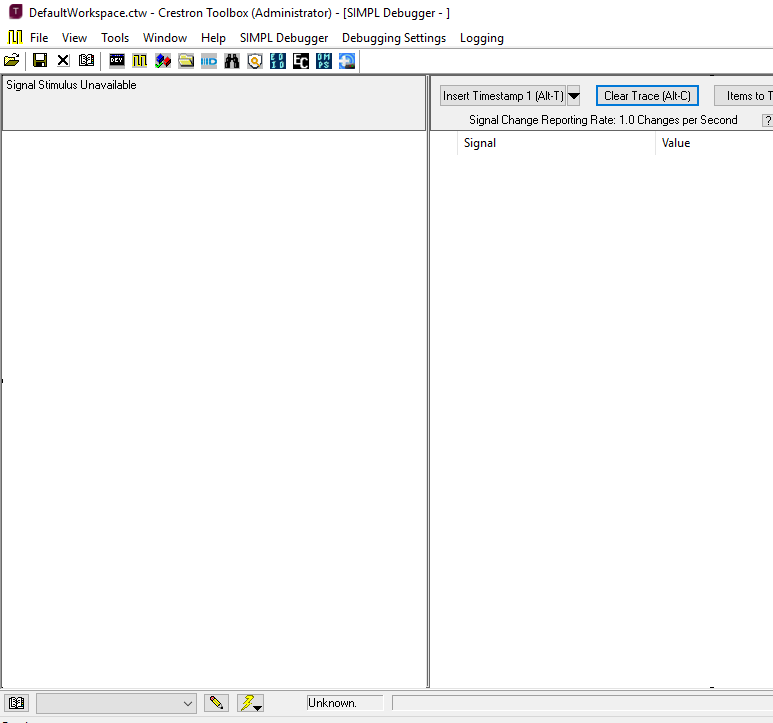
VTPro is for creating the GUI’s for Crestron devices (like touchpads). Like SIMPL, you can upload your VTPro programs directly to Crestron systems. This is done in VTPro and not in SIMPL. **THE BRIDGE BETWEEN VTPro AND SIMPL ARE THE DIGITAL AND SMART JOINS.** We’ve most commonly used digital joins. Essentially what happens is VTPro registers the buttons with the program running on the DMPS unit. If you have a touchpanel, for example, it should have been added into your system in SIMPL. If you double click it in the program view in SIMPL, the symbol that pops up will have button presses. These will correspond to your digital joins from 1,2…. To *n* from top to bottom. **WARNING:** Most touchpads have some ***RESERVED*** joins for the hardbuttons on the touchpad (back button, home button, etc.). Leave these joins alone, as in don’t assign the same digital joins the hardbuttons use to buttons in your program or weird things happen.

# Toolbox:

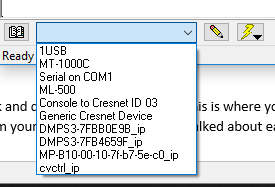
Toolbox is essentially the control hub of your DMPS system. Toolbox is what finds your devices through a network, is what sends your programs, can access command line on the DMPS and devices, change their configurations through the web etc. You will use toolbox a lot, especially because it has a DEBUGGER! A big part of Toolbox is the address book. This is basically a directory to your DMPS system, and stores the IPs of your Crestron devices. You can generate the address book from device discovery. To get to device discovery click on the in the top toolbar, then click discover devices. Ensure the devices in your Crestron system and your PC are all recognized by the firewall. You may have to talk to some IT guy to do this if you don’t have the permissions. To generate an address book, just click address book at the bottom of the list of devices found.

# Debugging:

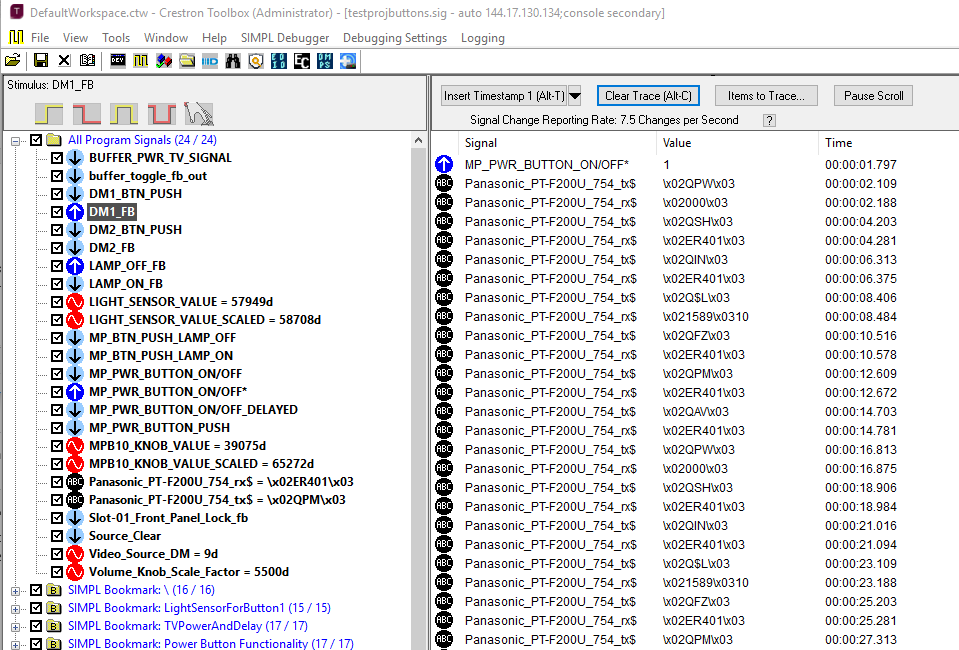
Any programmer (worth their salt) knows that a debugger is necessary for solving advanced problems. Luckily, toolbox has a debugger with it to test your SIMPL programs. To use it, after uploading your SIMPL program, go to toolbox. You should see a symbol in the top that looks like this: When you click on this button you should see the following screen:



In the bottom left you can see the book and drop down bar next to it. This is where you select which device to debug! Devices are taken from your address book, which we talked about earlier in the tutorial.



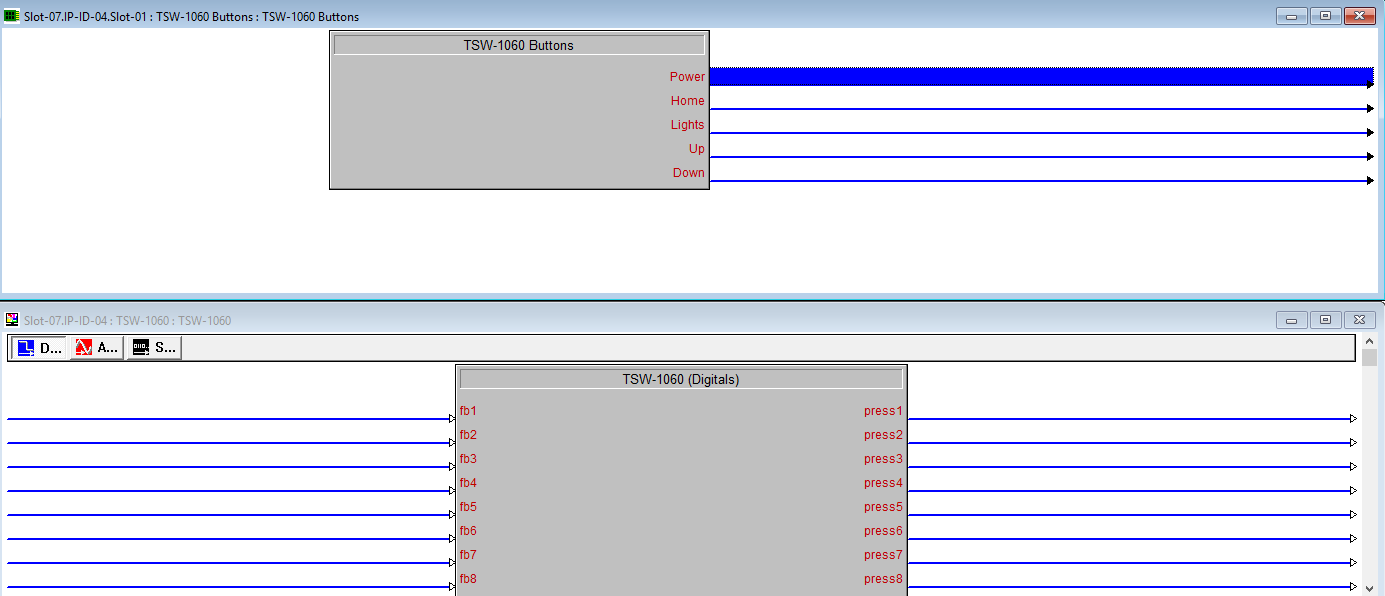
After selecting the device you want to debug, you will be met with a screen that looks like this: (Also, if you see a prompt to synchronize signals, press yes)



The right window changes the state and changing state of signals in your program. You can unselect signals if they are crowding the output window by unchecking the box to the left of the signal name. Many symbols in SIMPL will describe behavior as activating on “the rising edge” of a wave etc. These wave behaviors can be simulated by clicking the buttons above the signal list on the left while having a signal selected. For example DM2\_BTN\_PUSH is a button that sends a digital signal when it gets pushed down. To simulate this, I would click the picture of the hand. You can also use the debugger to send analog and serial values.

# SIMPL EXAMPLES

## Touch-Panel, VTPro, and Interlocks!



Above is an example of some touch-panel symbols in SIMPL, specifically the TSW-1060. It is connected by Ethernet to a DMPS3-4k-350-C. The symbol on top represents the hard buttons on the touch-panel, and the symbol on the bottom represents the digital joins for buttons inside the program that runs ON the touch-panel. In the past, we’ve encountered strange behaviors when we utilize digital joins 1-5 (Power, Home, Lights, Up, Down) so if you use this panel avoid using those symbols. We’re going to hook up some of these **signals** from the **bottom symbol** to our VTPro project, and implement some source selection. Below is shown the VTPro project: