My First Steps with OpcDbGateway Pt.2

By Jakub Palacka

In the previous article we designed a simple filter; however this filter is quite limited as it can only work when the constant “C” is 0.5. In this article we are going to take a look at how we can expand it’s functionality and turn it into a regulator which makes an output approach an input along a desired curve. By adding a few extra Commands and Memory Operands we can make it work using any value of C less than 1 and therefore regulate the steepness of the curve.

**Memory Operands**

1. We’ll start by starting up the configuration from the previous article and opening the **Sync Controller** field.
2. Go to Memory Operands and create a new folder called *Regulator\_Constants* and *Regulator\_Variables*
   * In *Regulator\_Constants* make a new **MO** called *1-C* with Data Type **DOUBLE**
   * In *Regulator\_Variables* make a new **MO** called *Ynx1-C* with Data Type **DOUBLE**



*Figure 1 Adding regulator variables and constants*

**Constants**

1. Change the **Value** of the constant *C* to something other than 0.5

*Figure 2 Changing value of constant*

**Function Blocks**

1. Next open up **Function Blocks** and create a new one labelled *Initialize\_Regulator*, create the following **Commands**
   * **Name**: Intialize\_1-C
   * **Operation**: SUB
   * **Input1**: 1 (Constant)
   * **Input2**: C (Constant)
   * **Output**: Regulator\_Constants/1-C (**MO**)
   * **Name**: Initialize\_Ynx1-C
   * **Operation**: SET
   * **Input1**: Zero(0) (Constant)
   * **Output**: Regulator\_Variables/Ynx1-C (**MO**)

*Figure 3 Initializing the regulators constant and variable*

1. Now we’ll just go to the *Start* and *Restart* **FBs** and create a command to CALL *Initialize\_Regulator*, do the following;
   * Create a new Command called *Call\_InitializeRegulator*
     + **Operation**: CALL
     + **Condition**: Utilities/True\_Condition (**MO**)
     + **Function Block**: Initialize\_Regulator
   * Don’t forget to copy and paste this command into both the *Start* and *Restart* **FBs**
2. Finally we’ll go to the *Main* **FB** and create a Command which multiplies *Yn* by *1-C*

*Figure 4 Calling the Initialize\_Regulator FB in Start and Restart*

* + **Name**: Mult\_Ynx1-C
  + **Input1**: Regulator\_Variables/Yn (**MO**)
  + **Input2**: Regulator\_Constants/1-C (**MO**)
  + **Output**: Regulator\_Variables/Ynx1-C (**MO**)

*Figure 5 Multiplying Yn by the new constant*

1. Now all we need to do is go to our **MO** *Ynx1-C* and generate an **OPC Item** for it so that we can see the changes that take place when we turn on **Monitor View**
2. Turn on **Monitor View** by clicking the little glasses icon at the top of the window

As you can see I moved the **MO** *Xn* into *Regulator\_Variables* for ease of access. I did this by opening the **MO** Folder *Filter\_Variables* and dragging and dropping the *Xn* **MO** into *Regulator\_Variables*

I also had to go through the **FBs** and commands which use the *Xn* **MO** and change them to ensure that they use the correct **MO**. The Commands that used *Xn* should have **<Not Assigned>** written where previously *Xn* was selected

1. First I went into *Initialize* and changed the **Output** of *Initialize\_Xn* to *Regulator\_Variables/Xn*
2. Second, I went to *Main* and changed the *Input1* of *Add\_Yn-1+Xn* to *Regulator\_Variables/Xn*
3. Finally I went into **Internal OPC Server** 🡪 **Address Space**🡪 **Filter\_Variables**, deleted the old *Xn* **OPC item** and created a new one from the *Regulator\_Variables* Folder where the **MO** is located

You could of course leave it as it is and switch between the *Filter\_Variables* and *Regulator\_Variables* folders in the address space when you want to write in a new value but this way both the input and output are in the same window

* + Open the field **Internal OPC Server**🡪**Address Space** 🡪 **Regulator\_Variables**
  + Right Click on **Regulator\_Variables.Xn** 🡪 **Write Value** 🡪 Type in a value other than 0

Regulator\_Variables.Ynx1-C will now begin approaching the number written into Xn

*Figure 6 Running Regulator*