My First Steps with OpcDbGateway Pt.4

By Jakub Palacka

So far in this series we have made a simple filter, upgraded it into a regulator and then connected that regulator to a database. Now that all this has been accomplished we can move on to the final step, connecting our configuration to an Excel spreadsheet. There is many advantages to being able to write values into a spreadsheet, one of these is the ability to visually represent this incoming data in the form of a graph. So this is what I will show you how to do now. This will be done by creating a new DDE Server and connecting an External DLL, which is included with the program, to the configuration.

1. Open SCADA Configurator and the configuration we made which includes our regulator and it’s database table
2. Right Click **External DLLs** 🡪 **New** 🡪 **External DLL**
   * Name it DDE DLL
   * Under **File Name** open the drop down menu and select the file *RuntimeDde.dll* and press apply
3. Now open an Excel spreadsheet, save it somewhere and keep it open

*Figure 1 Creating a new DDE Server*

1. Go to DDE Servers
   * Right Click **DDE Servers** 🡪 **New** 🡪 **DDE Servers**
   * Name it **Excel** and press **Choose Server**

If there is an Excel spreadsheet open the Service name will have Excel already selected

* Open the drop down menu next to **Topic Name**
* It should show the currently open spreadsheet and all the sheets that exist on it, in this case it’s *[First Steps Spreadsheet.xlsx]Sheet1*
* Choose the sheet you want to use and then click **OK**

1. Next we need to pick which cells on the spreadsheet we want our values to be written into

*Figure 2 Choosing the spreadsheet you want to use*

* + Click **Generate Items**
  + When it asks if you want to generate the required **Memory Operands** automatically then choose **Yes**
  + Now we need to choose the method of how we want to pick the cells we are using, the easiest way is the option Excel Interaction
  + Tick the box next to Excel Interaction

*Figure 3 Choosing the way you want to pick the cells that will be used*

1. Now alt-tab to the spreadsheet which is open and highlight the cells that you want to use. Then return to configurator and press **Next**

*Figure 4 Highlighting cells in Excel*



* + A window will open showing the names of the cells you have selected
  + By default the Data Type of all the cells is **STRING**, change it to **DOUBLE** since this is the type used by all the other **Memory Operands** (MOs) and the database.
  + Since values are going to be written into these cells then set the **Direction** to **Output**
  + Press **Next**, the next window all you need to do is choose the prefix for the **MOs** that are about to be created, I am going to use the prefix *Excel\_*, also change the **Start Index** to a higher number to make sure that it doesn’t use the index of **MOs** that already exist, after you have done this press **Finish**

*Figure 5 Changing the data type and direction of the chosen cells*

1. Now all the **MOs** for the cells in the Excel sheet have been created, if you look under **Sync Controller** 🡪 **Memory Operands** 🡪 **Excel**, you will be able to see them.
2. Now that we have made **MOs** for the cells in the sheet we need to do something similar for the cells in the database. To transfer values between the database and the spreadsheet we need to create a **Database Operand** (DO) for each cell.
3. Go to Sync Controller 🡪 Right Click DB Operands 🡪 New 🡪 DBOperands
   * Name the first one *R1* (Row 1)
   * Choose the **Type** “Cell” as it represents one cell and not an entire column
   * **Table**: ProcessDB.Regulator\_Outputs
   * **Field**: Outpus
   * **Row Select Through**: Row
   * **Row**: 1
4. Repeat this process for Rows 2-10 while changing the **Name** and **Row** number to match

*Figure 6 Making Database Operands*

**Note:** If you want to use the ProcessDB included in this package you will need to change the connection string to match the location of where it is stored

*Figure 7 Database Operands made for each cell used in the database*

1. Now we need to create a new **FB** and commands which will copy the figures from the **DOs** to **MOs** of the spreadsheet.
   * Right Click **Function Blocks** 🡪 **New** 🡪 **Function Block**
   * Create the following commands
     + **Name**: R1
     + **Operation**: SET
     + **Input 1**: Regulator\_Outputs/R1 (**DO**)
     + **Output**: Excel/DDE\_R2C1 (**MO**)
   * Do this for each **DO** 1-10

**Note**: The cells in the Excel sheet start 1 cell lower than those in the database; this is because the heading “Outputs” takes up one cell on the spreadsheet whereas in the database it doesn’t. Because of this *R1* in the database will be mapped to *DDE\_R2C1*, *R2* to *DDE\_R3C1* etc.

1. Finally we need to go to the *Main* **FB** and make a command that **Calls** the function which maps the **DOs** to the **MOs**
   * Create the following command:
     + **Name**: Write\_to\_Excel
     + **Operation**: CALL
     + **Condition**: Utilities/True\_Condition
     + **Function Block**: Write\_to\_Excel

*Figure 8 Calling the function which maps the Database Operands to the Memory Operands of the spreadsheet*

You can also create an initialization **FB** for the **DOs** if you wish, but since the initial state for the regulator is 0 this is eventually transferred over to both the **DOs** of the database and the **MOs** of the Excel spreadsheet so it isn’t necessarily required.

1. Now turn on Monitor View and look at the spreadsheet, you’ll see that the 10 cells under the “Outputs” heading will be filled up with 0
2. If you want to write in a new input go to **Internal OPC Server** 🡪 **Address Space** 🡪 **Regulator Variables**
   * Right Click **Regulator\_Variables.Xn**🡪 **Write Value** 🡪 Type in a value other than 0
3. The new values will start appearing in the spreadsheet, an excel graph can then easily be inserted to represent these values

*Figure 9 New values being written into the spreadsheet and then represented on a graph*

**Note:** You may need to change the decimal operator of excel depending on the language of your operating system