My First Steps with OpcDbGateway Pt.3

By Jakub Palacka

In Pt.1 of this little series I mentioned that OpcDbGateway can store data from it’s Memory Operandsdatabases, now that we have a working regulator to process data with I will show you how to store this data in a normal MS Access database.

1. Start up SCADA Configurator and open the configuration of the regulator that we made in the previous article

The SCADA Configurator comes with an already set up and ready Process Database for it’s alarms and alarm history. It is of course possible to create an entirely new MS Access database and connect it to the configurator but since we already have one ready and connected we might as well use that.

1. Expand the **Process Databases** field, here you see the database (ProcessDB) that’s installed by default. To the right you can see all the tables that already exist in that database.

**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

1. Right Click **ProcessDB** 🡪 **New** 🡪 **Process Tables**

*Figure 1 Opening Process Databases*

* + Name the new table *Regulator\_Outputs*
  + As you can see we can set the maximum **Table Length**, that is the maximum number of rows or records we want our regulator to be able to fill up
  + Set the **Table Length** to 10 and click **Apply**

**Note**: The when naming a table do not use spaces, if spaces are used the name it will cause an error and the table will not be created.

If you look under the heading **Table Information** the **State** might say that the table does not exist. Don’t be worried by this, at the moment we haven’t set up any of the columns in the table yet, when we have created some columns we will be able to press the button **Create** which will fix this. This might also occur if you have put spaces into the name of the table.

If Connection Test says “Can’t connect to database” then do the following:

*Figure 2 Adding a new table to the database*

* Click **ProcessDB** 🡪 **Edit** 🡪 **Test Connection**
* A message should pop up saying that the connection test was successful
* After you have done this go back to the table you have created, **Connection Test** should now say **Connected**

1. To set up the columns you want to use do the following
   * Right Click **Regulator\_Outputs** 🡪 **New** 🡪 **Columns**
   * The first column should always be an *ID* column, so name this column *ID*
   * The **Data Type** for *ID* can be **INTEGER** since it will only be increasing by 1 for every new record
   * **Ordinal nr.** (Ordinal Number) can be left as 0, the Ordinal nr. is the order number of the column, so in this case ID is column 0, click **Apply** to save the changes
2. Set up a second column with the name **Outputs**
   * The **Data Type** that we need is **DOUBLE** since this is the data type we are using in all our **Memory Operands** (MOs) but this type isn’t available in this particular drop down menu so we will need to go into the database ourselves and change it manually. For now you can select **INTEGER** from the drop down menu
   * For this column set the **Ordinal nr.** to 1 since this is the column where values from our regulator will be written into.



*Figure 3 Inserting columns*

1. Go back to the table Regulator\_Outputs, you’ll see that now the button **Create** is notgrayed out anymore, press **Create**
   * This will create the columns we made earlier in the database
2. Now we might as well go to the database itself and set the correct **Data Type** for the **Outputs** column.
   * The default directory for the ProcessDB database is: C:\Users\.......\Documents\SAEAUT SCADA\Data
   * Here you’ll find the MS Access file, double click on the table we just created and select Datasheet View
   * You can change the Data Type of ID to AutoNumber
   * Leave Outputs as Number, but in **Field Properties**, change **Field Size** to **Double**
   * Save the changes and close the database

Figure Changing field size to Double

1. Now there is only one more thing we need to do, we need to set up a command in the **FB** *Main* which will write the values from the regulator into the database.
   * Open **Sync Controller** 🡪 **Function Blocks** 🡪 **Main**
   * Create a new commence, **Right Click Main** 🡪 **New** 🡪 **Commands**

* **Name**: Write\_to\_Database
* **Operation**: WRITE\_ARRAY\_TO\_ACTUALTREND (Database)
* **Process Table**: ProcessDB/Regulator\_Outputs
* **Input 2**: Regulator\_Variables/Ynx1-C
* **Line no**.: 40

*Figure 5 Writing to database function*

The this particular function works is that it writes new values into the cells in the database until it runs out of cells to write into. When all the available cells have been used up the oldest value is removed and all the other values are moved down by 1 cell to make room for the newest value. We have 10 cells available, when the 10th is used up the 1st will be emptied. The 2nd value will move into the 1st, the 3rd into the 2nd etc.

1. To see the regulator write values into the database turn on **Monitor View** by pressing the little glasses icon at the top of the window.
2. Open **Process Databases** 🡪 **ProcessDB** 🡪 **Regulator\_Outputs**
   * Press the button **Show Table** this will show the currently selected table
   * At the moment it’s only writing in zeroes because this is the initial value we have set and we haven’t written in a new one for Xn

*Figure 6 Values being written into the table*

1. Go to **Internal OPC Server** 🡪 **Address Space** 🡪 **Regulator\_Variables**

*Figure 7 New figures overwriting old ones*

* + Write in a new value for the **Item ID** *Regulator\_Variables.Xn*
  + Right Click *Regulator\_Variables.Xn*
  + Write Value 🡪 Type in a value other than 0
  + The Table will now begin filling up with the new values, as you can see here the zeroes are already being overwritten by new figures.