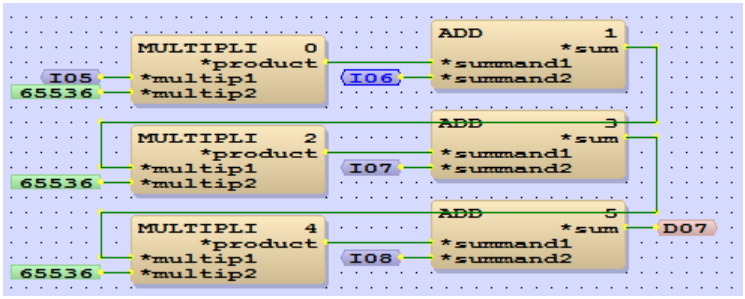
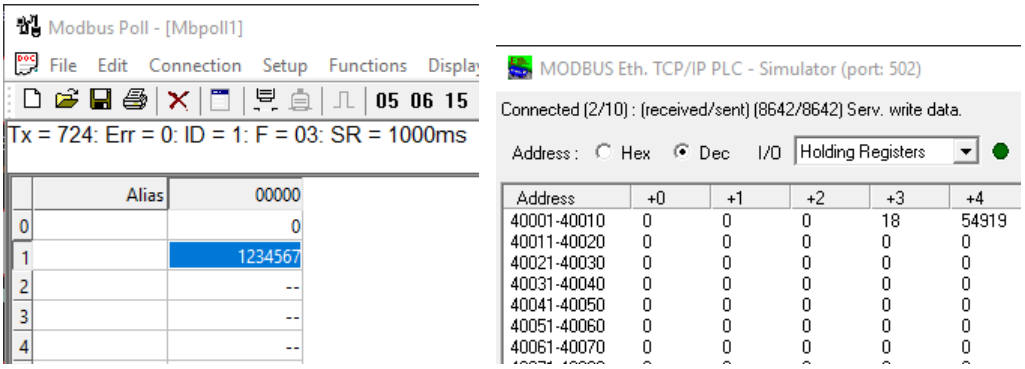
TT190902 – FUP - Modbus 64 Bit Register

1. Sometimes we need to read 64-bit register from a Modbus device, most likely it is the kWh reading from a power meter. But currently in FUP we can only read up to 32-bit Modbus register using type ULI and FL
2. What we can do is to read the 64-bit register as four 16-bit registers using UI format, and then calculate the correct reading in FUP using the calculation below for 64-bit unsigned integer

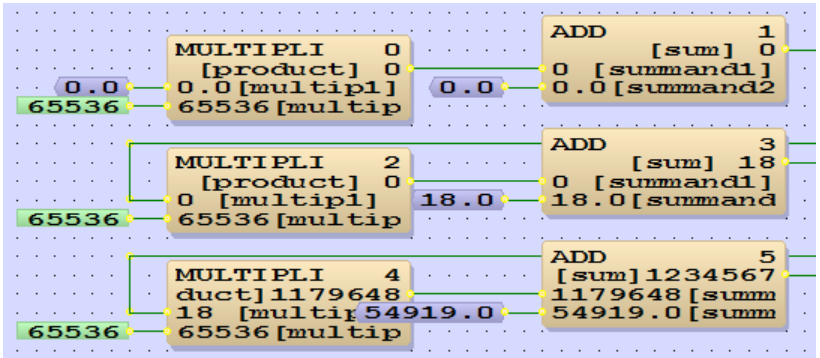


Iden	Desc	Label	consistency	M_SLAVE	M_memory_type	M_VAR_ADR	M_VARTYP	M_FACTOR
A		M64.F:I05	not verified	1	HoldingRegister	1	UI	1
B		M64.F:I06	not verified	1	HoldingRegister	2	UI	1
C		M64.F:I07	not verified	1	HoldingRegister	3	UI	1
D		M64.F:I08	not verified	1	HoldingRegister	4	UI	1

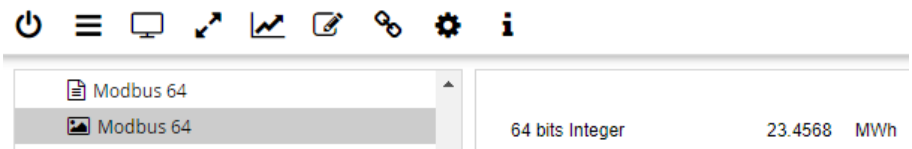
3. To test it, we can use the Modbus testing software we use in TT190801 and TT190803. Set the format in Modbus Poll as “64 Bit Unsigned, Big-endian”



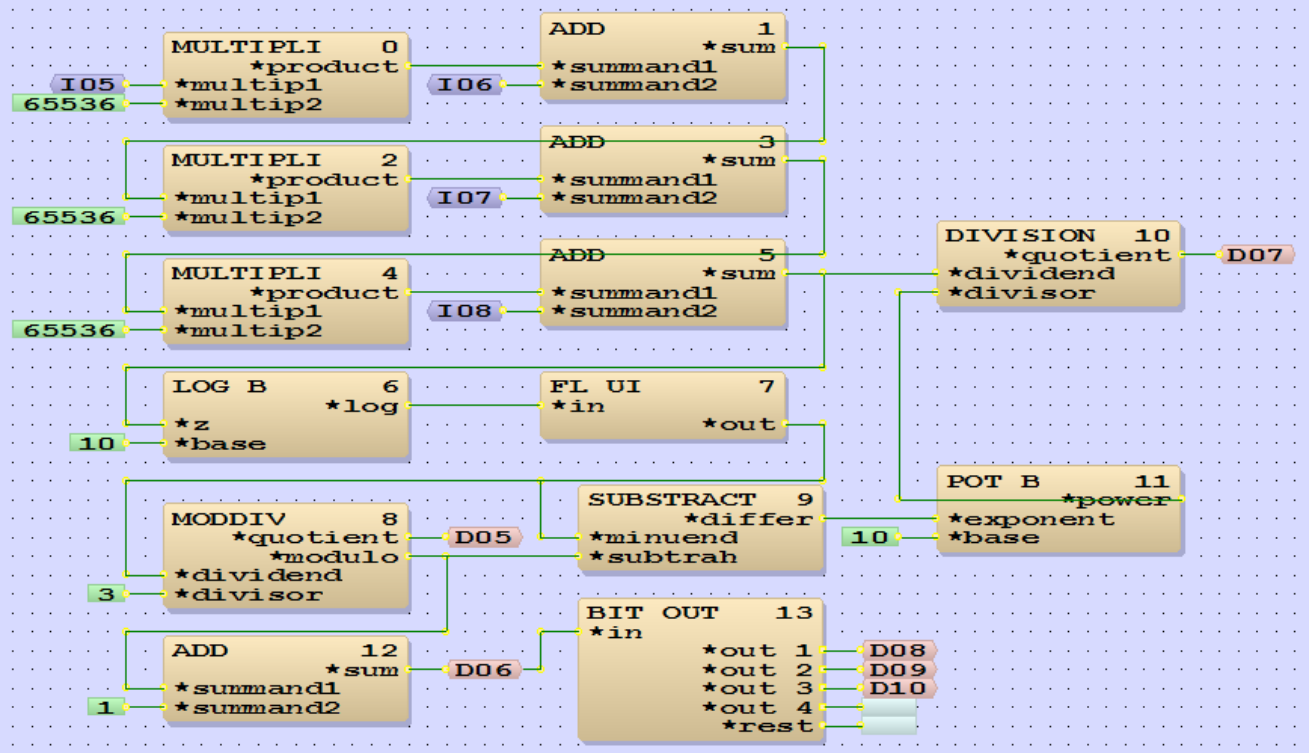
4. For small value, we can show exactly the same number in FUP (like below). But for bigger number, the value will be a bit different, because we use 32-bit floating point format in FUP



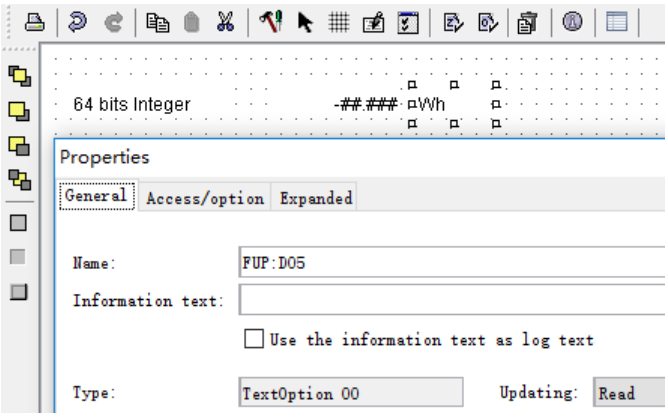
5. To avoid this problem, we have to limit the number of digits (and decimal places) to show in the graphic, like this

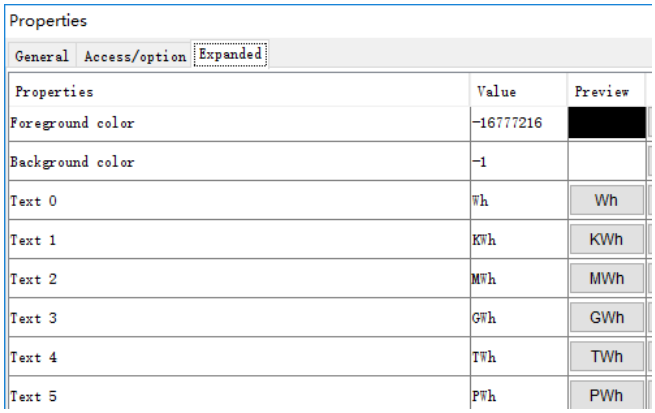


6. We convert the integer number to a floating point number, by using the “log based 10 (LOG_B)” module and divide it by 3, to get the correct unit (e.g. kWh, MWh, etc.). We then restrict the number of digits to 6 by the “BIT_OUT” module together with “Display Element” option in graphic

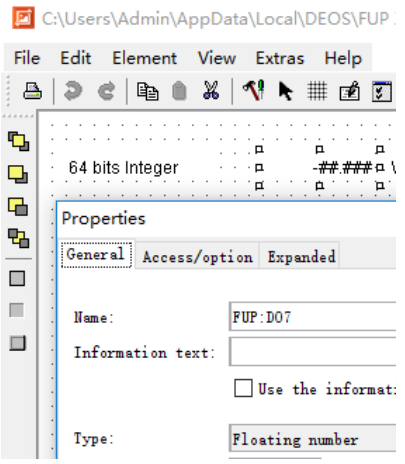


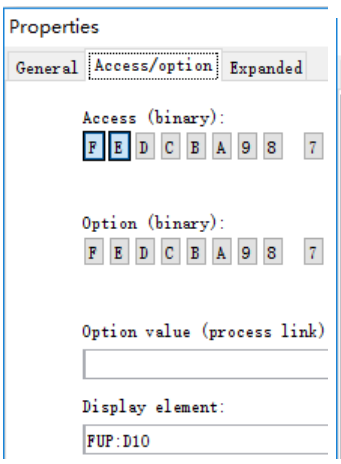
7. In the graphic, we use the “TextOption00” element, link it to “D05”, and assign the corresponding unit to the value in the “Expanded” tab

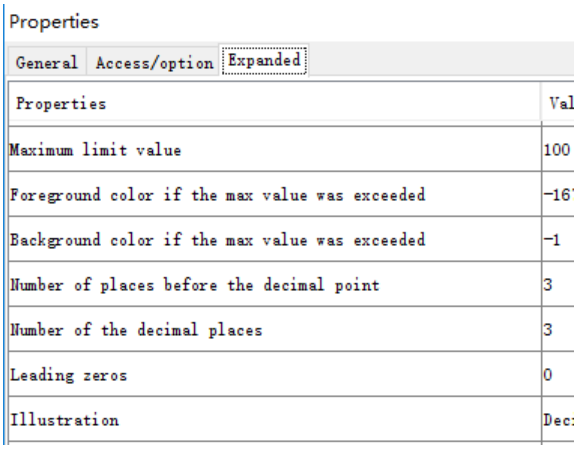




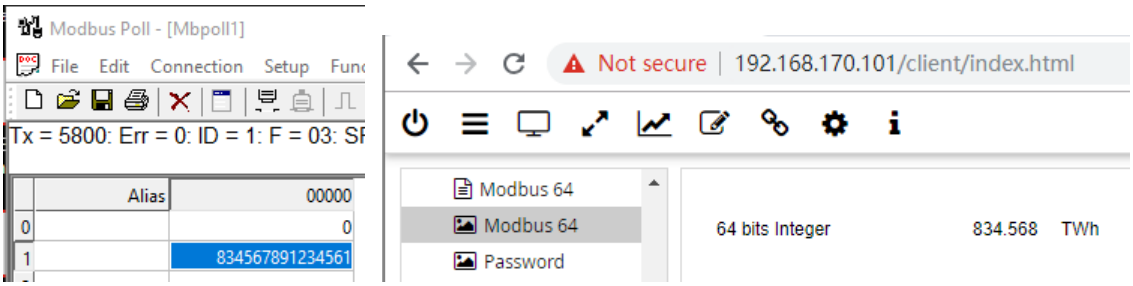
8. For the value, we use multiple elements (link to D07), each showing different number of decimal places, and use D8-D10 to display the correct element using “Display element”



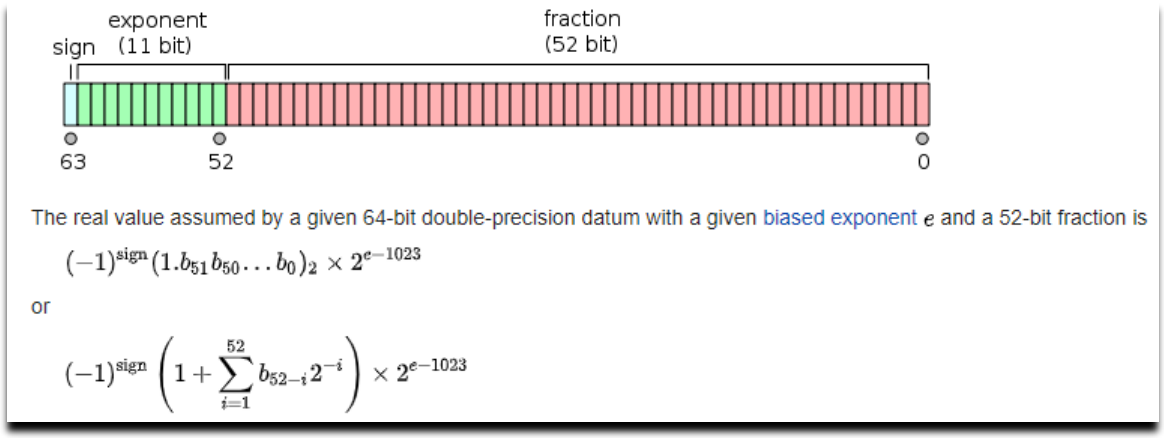




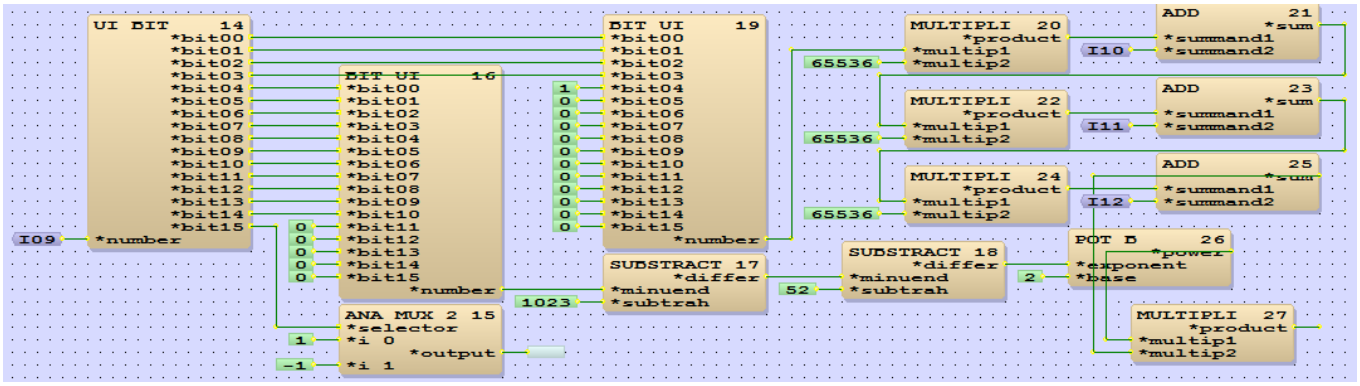
9. Finally, this is the value we show in our graphic with such a big number



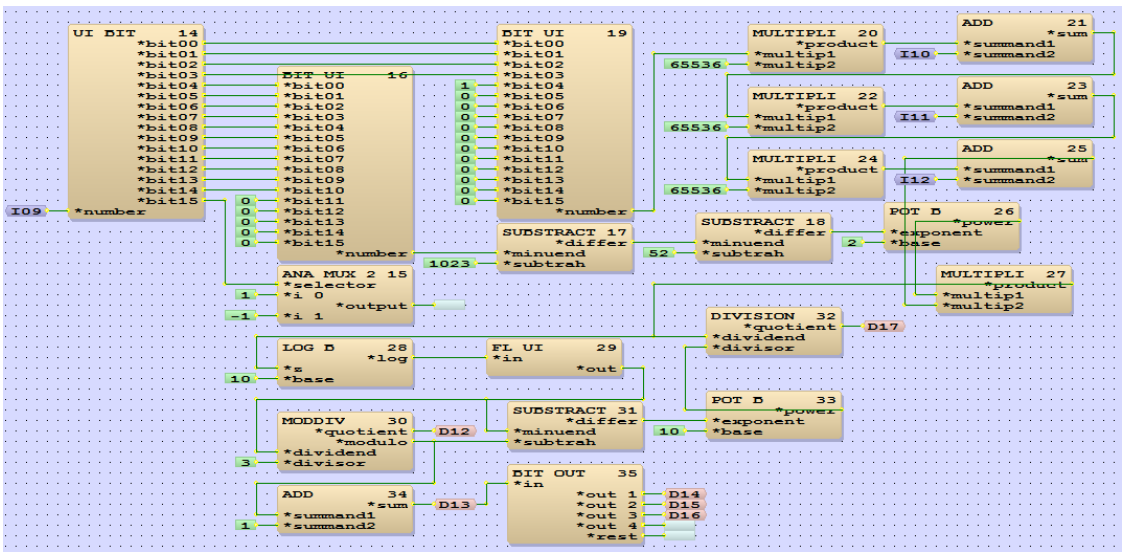
10. For 64-bit floating point number, the calculation is a bit complicate. Here is the formula from Wiki for your reference



11. In FUP, we have to extract the bits from the UI and then recalculate it based on the above formula. Assuming the kWh value are all positive, so just ignore the sign here to make it simple



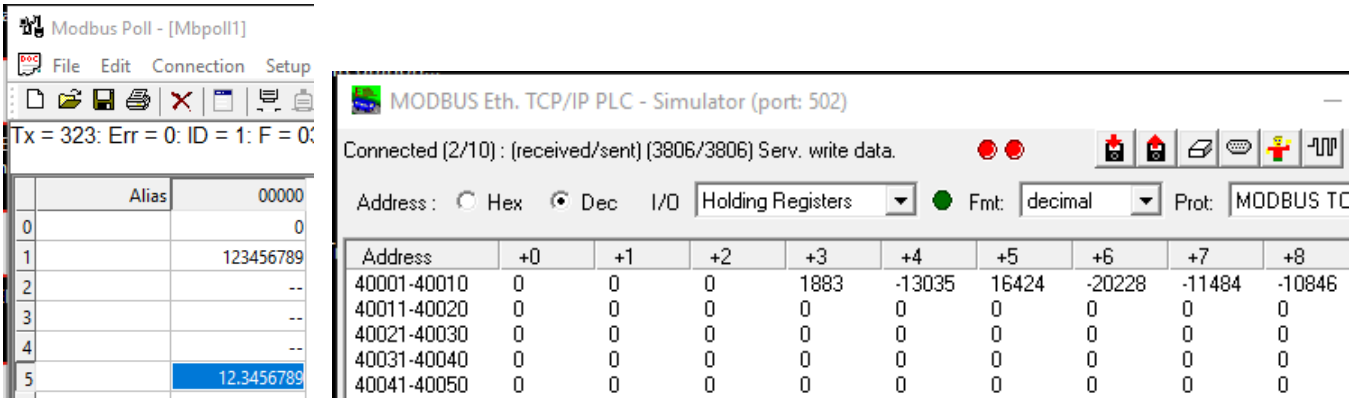
12. Here is the final program



13. Here is the Modbus Integration table

Iden	Desc	Label	consistency	M_SLAVE	M_memory_type	M_VAR_ADR	M_VARTYP	M_FACTOR
A		M64.F:I09	not verified	1	HoldingRegister	5	UI	1
B		M64.F:I10	not verified	1	HoldingRegister	6	UI	1
C		M64.F:I11	not verified	1	HoldingRegister	7	UI	1
D		M64.F:I12	not verified	1	HoldingRegister	8	UI	1
A		M64.F:I05	not verified	1	HoldingRegister	1	UI	1
B		M64.F:I06	not verified	1	HoldingRegister	2	UI	1
C		M64.F:I07	not verified	1	HoldingRegister	3	UI	1
D		M64.F:I08	not verified	1	HoldingRegister	4	UI	1

14. Compile and upload it to the controller. Now test it with the simulation software. Set address 5 to format “64 Bit Double, Big-endian”



15. In the graphic, the values are displayed like this