# IEC 61850 and Modbus Data Type Mapping Rules

For example, in the IEC 61850 telemetering data point list `YC_List` below, we can see two data types: `"FLOAT32"` and `"INT32"`. In the Modbus telemetering list `MB_List`, we can see: `"U_AB", 2` (meaning a 16-bit unsigned integer that should be converted with two decimal places, i.e., multiplied by 0.01) → corresponds to `"FLOAT32"`; `"U_AB", 0` (a pure 16-bit unsigned integer) → corresponds to `"INT32"`.

```
-- IEC61850 telemetering data point definitions
YC_List =
{
  -- Transformer #1
  {"RTU/GGIO1.AnIn1", ".mag.f", "FLOAT32"}, -- floating point, phase A temperature
  {"RTU/GGIO1.AnIn2", ".mag.f", "FLOAT32"}, -- floating point, phase B temperature
  {"RTU/GGIO1.AnIn3", ".mag.f", "FLOAT32"},  -- floating point, phase C temperature
  {"RTU/GGIO1.Inc1", ".stVal", "INT32"}, -- integer, phase A sensor signal strength
  {"RTU/GGIO1.Inc2", ".stVal", "INT32"}, -- integer, phase B sensor signal strength
  {"RTU/GGIO1.Inc3", ".stVal", "INT32"}  -- integer, phase C sensor signal strength

}

-- Modbus telemetering data point definitions
MB_POLL_TIME_MS = 5000 -- minimum polling period for all RS-485 devices below: 5000 ms (5 s); may be longer if a device communicates poorly
MB_INTERVAL_MS = 100 -- minimum interval between RS-485 transactions
MB_List =
{
  -- Transformer #1
  {
    -- 115200 bps, no parity, 1 stop bit, function code "03", Modbus address 0x02, max response wait 100 ms, inter-packet delay 1000 ms
    com = {"BAUDRATE_115200","NoneParity","StopBit_1","03",0x02,100,1000},
    data =
    {
      {"RTU/GGIO1.AnIn1",0x0030,"U_AB",2}, -- phase A temperature, 2 decimals (actually U_AB integer × 0.01)
      {"RTU/GGIO1.AnIn2",0x0031,"U_AB",2}, -- phase B temperature, 2 decimals (actually U_AB integer × 0.01)
      {"RTU/GGIO1.AnIn3",0x0032,"U_AB",2}, -- phase C temperature, 2 decimals (actually U_AB integer × 0.01)
      {"RTU/GGIO1.Inc1",0x0060,"U_AB",0}, -- phase A signal strength, integer, 0 decimals
      {"RTU/GGIO1.Inc2",0x0061,"U_AB",0}, -- phase B signal strength, integer, 0 decimals
      {"RTU/GGIO1.Inc3",0x0062,"U_AB",0}  -- phase C signal strength, integer, 0 decimals
    }
  }
}
```

Below we describe this mapping in detail. In scenarios that involve numeric values such as `telemetering` and `setpoint (control)`, IEC 61850 currently supports two data types: `FLOAT32` and `INT32`, which already cover most use cases. Although Modbus defines more data formats, after normalization they can be easily mapped one-to-one to IEC 61850 `FLOAT32` and `INT32`. Please keep this in mind when defining your data.

| Modbus type | Description | Corresponding IEC 61850 type |
|---|---|---|
| "S_AB" | 16-bit signed integer, AB byte order | `INT32` when decimals = 0 |
| "S_BA" | 16-bit signed integer, BA byte order | `INT32` when decimals = 0 |
| "U_AB" | 16-bit unsigned integer, AB byte order | `INT32` when decimals = 0 |
| "U_BA" | 16-bit unsigned integer, BA byte order | `INT32` when decimals = 0 |
| "UL_ABCD" | 32-bit unsigned integer, ABCD byte order | `INT32` when decimals = 0 |
| "UL_CDAB" | 32-bit unsigned integer, CDAB byte order | `INT32` when decimals = 0 |
| "UL_BADC" | 32-bit unsigned integer, BADC byte order | `INT32` when decimals = 0 |
| "UL_DCBA" | 32-bit unsigned integer, DCBA byte order | `INT32` when decimals = 0 |
| "L_ABCD" | 32-bit signed integer, ABCD byte order | `INT32` when decimals = 0 |
| "L_CDAB" | 32-bit signed integer, CDAB byte order | `INT32` when decimals = 0 |
| "L_BADC" | 32-bit signed integer, BADC byte order | `INT32` when decimals = 0 |
| "L_DCBA" | 32-bit signed integer, DCBA byte order | `INT32` when decimals = 0 |
| "F_ABCD" | 32-bit floating-point, ABCD byte order | `FLOAT32` |
| "F_CDAB" | 32-bit floating-point, CDAB byte order | `FLOAT32` |
| "F_BADC" | 32-bit floating-point, BADC byte order | `FLOAT32` |
| "F_DCBA" | 32-bit floating-point, DCBA byte order | `FLOAT32` |
| "D_ABCDEFGH" | 64-bit double, ABCDEFGH byte order | `FLOAT32` |
| "D_GHEFCDAB" | 64-bit double, GHEFCDAB byte order | `FLOAT32` |
| "D_BADCFEHG" | 64-bit double, BADCFEHG byte order | `FLOAT32` |
| "D_HGFEDCBA" | 64-bit double, HGFEDCBA byte order | `FLOAT32` |
| "BIT" | Must be used for reading coils/discrete inputs | For telesignalling/telecontrol only |