Remote Control Case Guide (Binary Output Control, Modbus Function Code 05)

This section introduces various remote control scenarios. Each case's project files can be found in the provided "Example Project Files." Choose the closest match and modify as needed.

Case 1: Control one device with 4 IO output channels

(1) Modify `main.lua`

Configure `YK_List`

```lua
-- IEC61850 remote control data point definitions
YK_List =
{
  -- Modbus IO output control unit #1 (4 outputs)
  {"RTU/GGIO1.SPCSO1", ".stVal"}, -- binary IO output control, 0 or 1
  {"RTU/GGIO1.SPCSO2", ".stVal"}, -- binary IO output control, 0 or 1
  {"RTU/GGIO1.SPCSO3", ".stVal"}, -- binary IO output control, 0 or 1
  {"RTU/GGIO1.SPCSO4", ".stVal"}  -- binary IO output control, 0 or 1
}
```

Configure `MB_List`

```lua
-- Modbus remote control data point definitions
MB_List =
{
  -- Modbus IO output control unit #1 (4 outputs)
  {
    -- 9600 bps, no parity, 1 stop bit, function code "05", modbus address 0x08, max
response 100 ms, inter-packet 1000 ms
    com = {"BAUDRATE_9600","NoneParity","StopBit_1","05",0x08,100,1000},
    data =
    {
      {"RTU/GGIO1.SPCSO1",0x0000}, -- write coil, 0 or 1
      {"RTU/GGIO1.SPCSO2",0x0001}, -- write coil, 0 or 1
      {"RTU/GGIO1.SPCSO3",0x0002}, -- write coil, 0 or 1
      {"RTU/GGIO1.SPCSO4",0x0003}  -- write coil, 0 or 1
    }
  }
}
```

Note: Lua syntax — no trailing comma after the last `}`.

(2) Modify `rtu.cid`

Define logical node (IO control unit, 4 channels)

```
<LNodeType desc="IO control unit (4 channels)" id="GGIO_TYPE_4Out" lnClass="GGIO">
  <DO desc="Mode" name="Mod" type="ENC_Mod" />
  <DO desc="Behavior" name="Beh" type="ENS_Beh" />
  <DO desc="Health" name="Health" type="ENS_Health" />
  <DO desc="Nameplate" name="NamPlt" type="LPL_2_NamPlt" />
  <DO desc="Remote control coil 1" name="SPCSO1" type="SPC_Control" />
  <DO desc="Remote control coil 2" name="SPCSO2" type="SPC_Control" />
  <DO desc="Remote control coil 3" name="SPCSO3" type="SPC_Control" />
  <DO desc="Remote control coil 4" name="SPCSO4" type="SPC_Control" />
</LNodeType>
```

Add logical node instance and set `ctlModel`

```
<LN desc="IO control unit #1 (4 ch)" lnClass="GGIO" lnType="GGIO_TYPE_4Out" inst="1" prefix="">
  <DOI name="Mod">
    <DAI name="stVal">
      <Val>on</Val>
    </DAI>
    <DAI name="ctlModel">
      <Val>status-only</Val>
    </DAI>
  </DOI>
  <DOI name="Beh">
    <DAI name="stVal">
      <Val>on</Val>
    </DAI>
  </DOI>
  <DOI name="Health">
    <DAI name="stVal">
      <Val>ok</Val>
    </DAI>
  </DOI>
  <DOI name="SPCSO1">
    <DAI name="ctlModel">
      <Val>direct-with-normal-security</Val>
    </DAI>
  </DOI>
  <DOI name="SPCSO2">
    <DAI name="ctlModel">
      <Val>direct-with-normal-security</Val>
    </DAI>
  </DOI>
  <DOI name="SPCSO3">
    <DAI name="ctlModel">
      <Val>direct-with-normal-security</Val>
    </DAI>
  </DOI>
  <DOI name="SPCSO4">
    <DAI name="ctlModel">
      <Val>direct-with-normal-security</Val>
    </DAI>
  </DOI>
</LN>
```

Configure DataSet

```
<DataSet name="YK_RC" desc="YK_RC">
  <FCDA ldInst="RTU" lnClass="GGIO" fc="ST" lnInst="1" doName="SPCSO1" daName="stVal" />
  <FCDA ldInst="RTU" lnClass="GGIO" fc="ST" lnInst="1" doName="SPCSO2" daName="stVal" />
  <FCDA ldInst="RTU" lnClass="GGIO" fc="ST" lnInst="1" doName="SPCSO3" daName="stVal" />
  <FCDA ldInst="RTU" lnClass="GGIO" fc="ST" lnInst="1" doName="SPCSO4" daName="stVal" />
</DataSet>
```

DataSet is for the Report service and is optional.

Case 2: Control multiple devices, each with 4 IO outputs

Repeat the configuration for devices #1, #2, and #3 in both `YK_List` and `MB_List`, add LN instances for each, and include them in the `YK_RC` DataSet as shown.