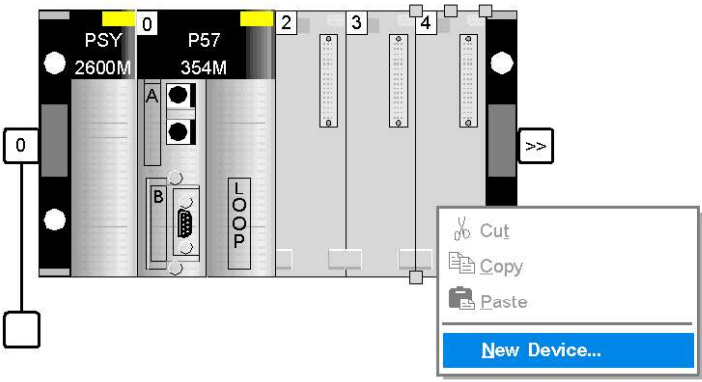
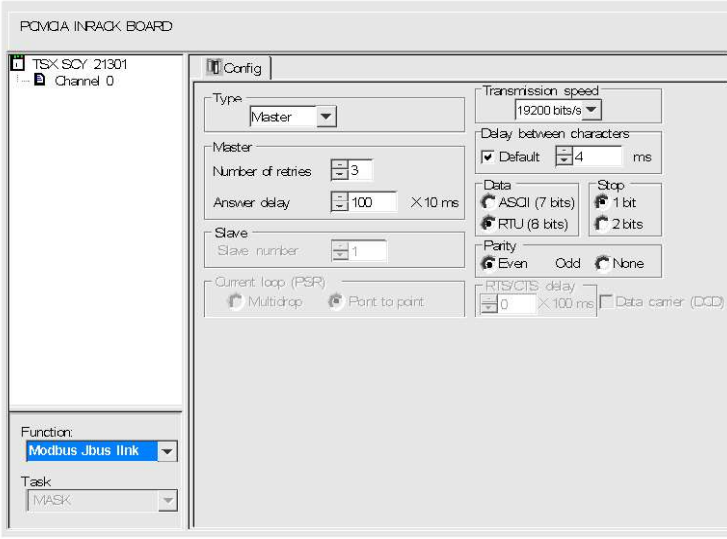
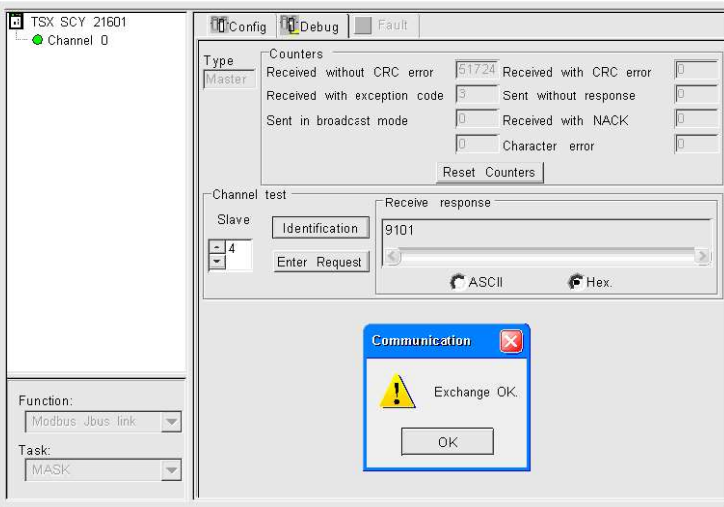


Configure the Communication

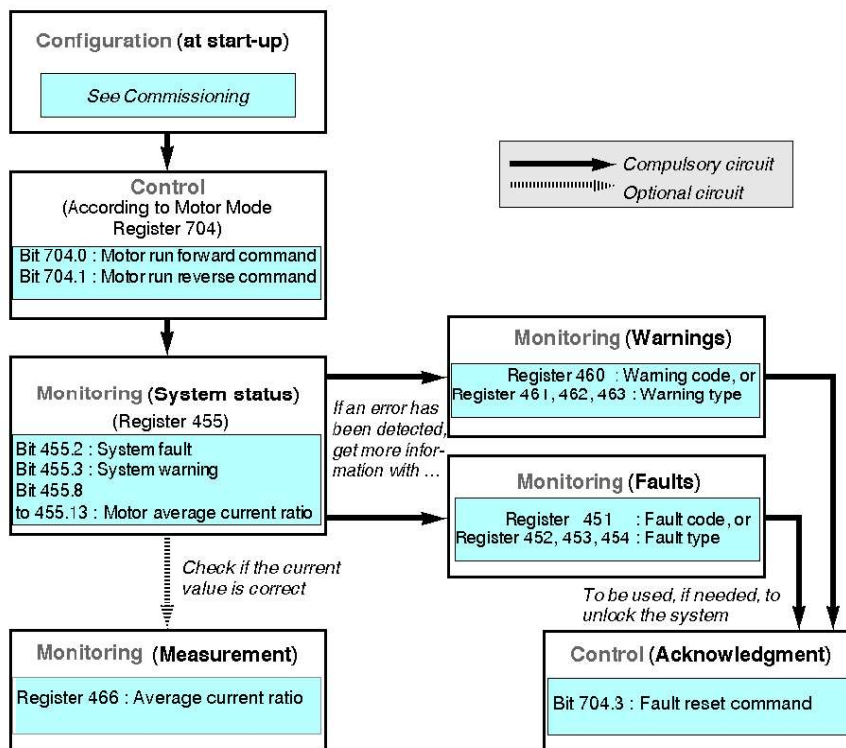
This example describes how to configure communication between a Premium PLC running Unity software and an LTM R controller:

Step	Description
1	Declare the Modbus module in the Unity software: - Right-click on the slot where the module is to be declared and add the module.  - Select TSXSCY 21601 in the Communication family and click OK to confirm.
2	Configure the Modbus module in the Unity software: - On channel 0, select MODBUS/JBUS LINK for the communication protocol. - Select Master mode for the PLC. - Set the transmission speed 19200 and the data format RTU. This is the data format supported by the LTM R controller. - Set the parity to Even. 
3	Save and transfer the configuration to the PLC.

Step	Description
4	Verify the communication operation via the debug screen: • Turn power OFF and ON successively to activate the automatic recognition of transmission speed and data format. • Select the slave address 4 and press the Identification button to activate the identification request transmission. • Wait a few seconds for recognition. Depending on the character format and speed transmission, it may be necessary to press the Identification button twice. 
5	Develop and load the application program, then test it.

Registers for Simplified Operation

Basic setup information using configuration, control and monitoring registers applies to all applications:



Standard Requests on a PLC Platform

1) Example of a Read Operation (Modbus Request Code 3)

The example below describes a READ_VAR request, within a TSX Micro or Premium platform, in order to read the LTM R states at address 4 (slave n° 4) contained in internal word MW0:

Syntax with PL7 software:

```
IF %M0 AND NOT %MW100:X0 THEN READ_VAR(ADR#3.0.4, '%MW', 455, 1, %MW0:1, %MW100:4) :RESET %M0;
EN_IF;
```

- 1 Address of the device with which you wish to communicate: 3 (device address), 0 (channel), 4 (device address on the bus)
- 2 Type of PL7 objects to be read: MW (internal word)
- 3 Address of the first register to be read: 455
- 4 Number of consecutive registers to be read: 1
- 5 Word table containing the value of the objects read: MW0:1
- 6 Read report: MW100:4

Variant with Unity Pro software:

```
IF %M0 AND NOT %MW100.0 THEN
READ_VAR(ADDR( '0.3.0.4' ), '%MW', 455, 1, %MW100:4, %MW0:1); RESET(%M0);
END_IF;
```

2) Example of a Write Operation (Modbus Request Code 16)

The example below describes a WRITE_VAR request, within a TSX Micro or Premium platform, in order to control an LTM R by sending the contents of internal word MW502:

Syntax with PL7 software:

```
IF %M0 AND NOT %MW200:X0 THEN WRITE_VAR(ADR#3.0.4, '%MW', 704, 1, %MW502:1, %MW200:4) :RESET %M10;
EN_IF;
```

- 1 Address of the device with which you wish to communicate: 3 (device address), 0 (channel), 4 (device address on the bus)
- 2 Type of PL7 objects to be written: MW (internal word)
- 3 Address of the first register to be written: 704
- 4 Number of consecutive registers to be written: 1
- 5 Word table containing the value of the objects to be sent: MW502:1
- 6 Write report: MW200:4

Variant with Unity Pro software:

```
IF %M0 AND NOT %MW200.0 THEN
WRITE_VAR(ADDR( '0.3.0.4' ), '%MW', 704, 1, %MW502:1, %MW200:4); RESET(%M0);
END_IF;
```