



Users' Guide: Connecting the Modbus Gateway to the X-Series G4 Totalflow and PCCU

Application Note

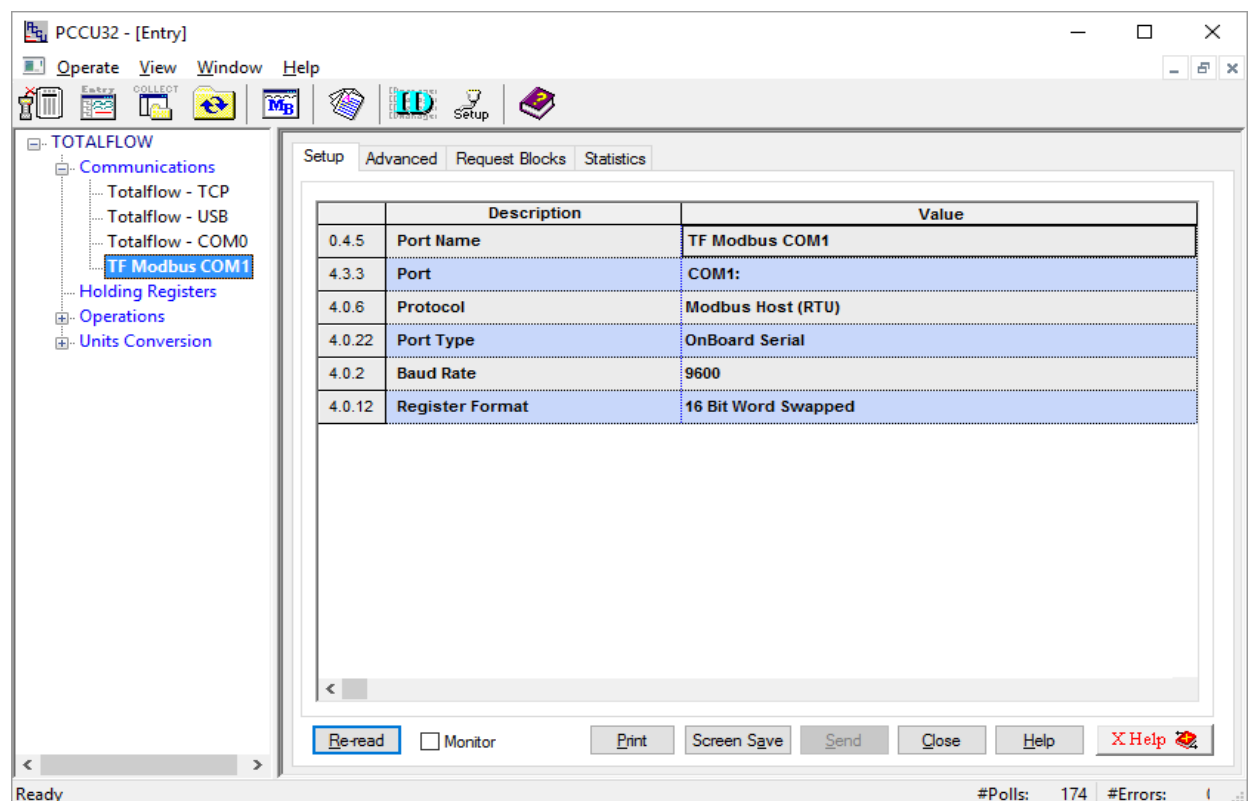
NOTE: In this tutorial, it is assumed that the reader knows how to connect the Totalflow to your PC.

In this tutorial, the Totalflow has an RS485 card in the first communication port, Comm1. Comm1's RS485 terminals are used, connecting GND to ground on the breakout board's ground, BUS+ to A, and BUS- to B.

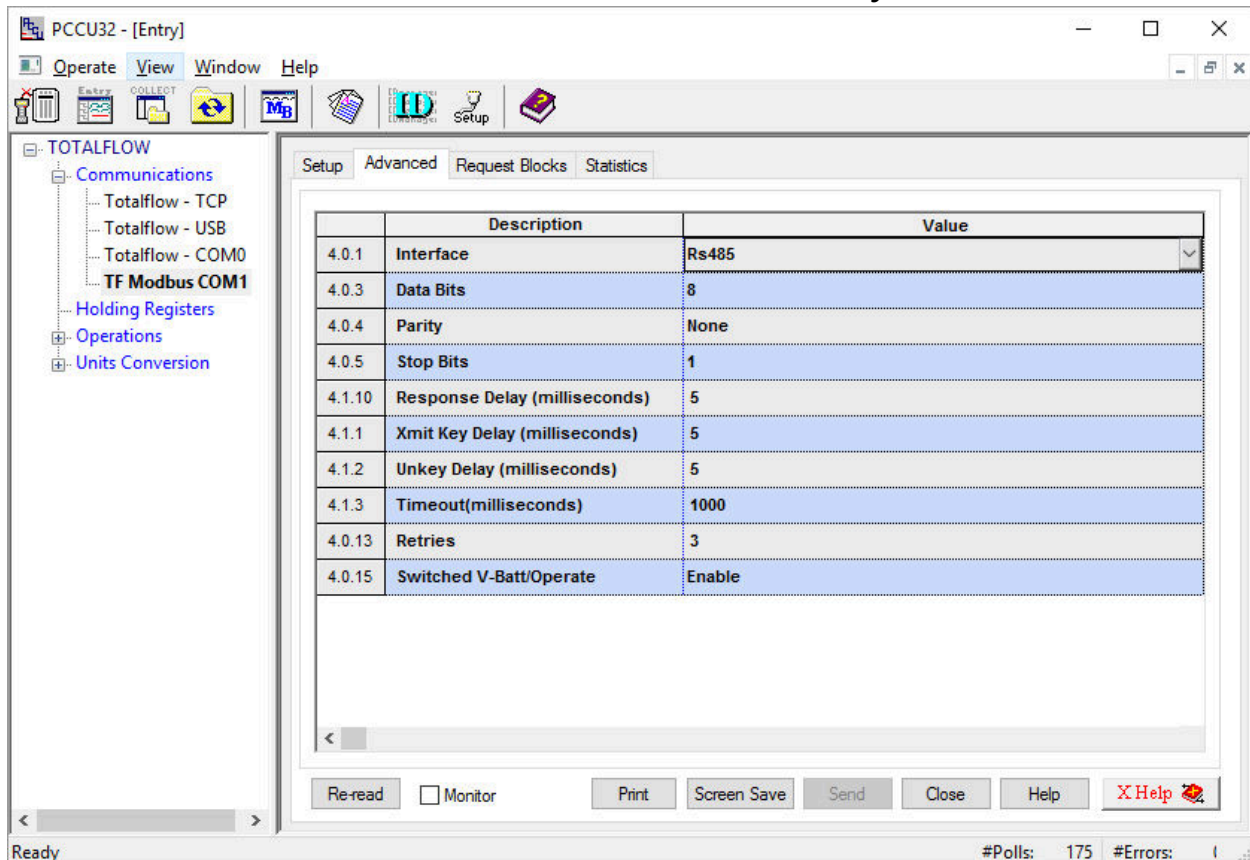
Communication Setup

First, set up communication with the Gateway.

1. Expand **Communications**, then click the entry labeled **COM1** (or something similar). Keep in mind that this has nothing to do with serial port COM1 on your computer! Your Port Name may not match the one shown here.

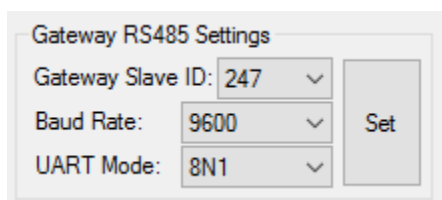


2. Make sure that **Protocol** is set to **Modbus Host (RTU)**, that **Baud Rate** is set to **9600**, and that **Register Format** is set to **16 Bit Word Swapped**.
3. Click the tab labeled **Advanced**. Set the values here **exactly as shown**. **Otherwise, the Totalflow will not be able to read data from the Gateway!**



Now we will set up two register reads. In the first, we will read the testing registers of the Gateway: register 2100 contains the value "2100", 2101 the value "2101", and 2102 the value "2102". (Check the Gateway manual for details.) In the second, we will read the HART variables from a Sentinel HART and verify that they match the values shown in the SignalFire Toolkit.

Part One: Reading Gateway Registers



The Gateway's default RS485 settings are shown here: it uses Slave ID 247, runs at 9600 baud, and uses 8N1 UART. The last two match the settings we set in Configuration; the Slave ID is what we'll use to connect to it.

1. Click on **Holding Registers**.

2. Set **Number of Arrays** to 2 (because we're doing two reads; you can set it to however many you need).
3. Click **Send**. Then, set the arrays' description and name to whatever you like.
4. The Gateway read will be three registers, so set the first array's **Capacity** to 3.
5. The HART read will be four, so set the second array's **Capacity** to 4.
6. Set the first array's **Type** to Uint16 (as in the Gateway manual), and the second array's type to Float, which are the values the HART sensor reports.
7. Click **Send** again.

The screenshot shows a configuration window with three tabs: 'Uint16', 'Float', and 'Capacity'. The 'Capacity' tab is selected. It contains a table with the following data:

	Description	Capacity	Type	Persistence	Name
9.255.255	Number of Arrays	2			
9.255.0	Array 1	3	Uint16	Non-Persistent	Uint16
9.255.1	Array 2	4	Float	Non-Persistent	Float

Below the table is a large empty text area. At the bottom of the window are several buttons: 'Re-read', 'Monitor' (with an unchecked checkbox), 'Print', 'Screen Save', 'Send', 'Close', 'Help', and 'X Help' (with a close icon).

8. Now go to the tab whose name matches the first array's **Name** (which is Uint16 here). Set the Description to whatever you like but take notice of the numbers to the left, here "9.0.something". We will set up the register read so that the three values go into 9.0.0, 9.0.1, and 9.0.2 respectively.

	Description	Value
9.0.0	Uint16 0	0
9.0.1	Uint16 1	0
9.0.2	Uint16 2	0

9. Now go back to the COM1 entry. Click on the tab labeled **Request Blocks**. Click **Add**, enter a file name, then click **OK**. Here, it is called "GWread".

10. To read registers from SignalFire devices, select function **3 – Read Registers** from the dropdown menu.

11. Set **Slave Address**

to the slave ID of the device to read registers from (here, the Gateway is 247).

12. Set **Starting Register** to the first register you want to read **plus one**. Here we want to read registers 2100 to 2102, so I will enter 2101.

13. Set **# Registers** to the number of registers to read, which is three.

14. Set **Register Type** to the value shown in the Gateway Register View or the manual (which is Int16).

15. You can set **Trigger Type** to "Interval", which will show a box below where you can set the sampling interval of the Totalflow. Leave Response Status unchanged. Click **Save**, and put it anywhere you'd like.

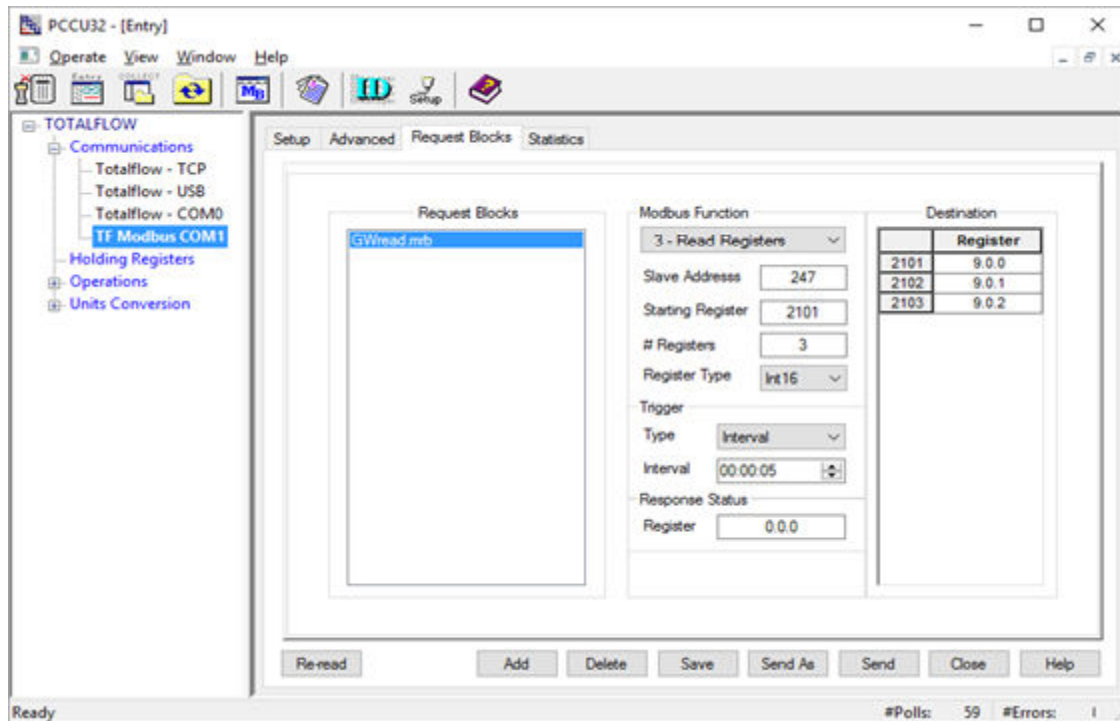
16. Now right click on **Register** under **Destination** and select Auto Fill.

17. Set **Current Cell** to "9.0.0", the first entry from above.

18. Set **Additional Registers to Fill** to 3 (since we have three registers).

19. Click OK. **You may encounter a bug with PCCU where an extra row is added with no number to the left of it. Just right click on the row and delete it.**

20. Click **Save** again, and replace the old file. Then click **Send**.



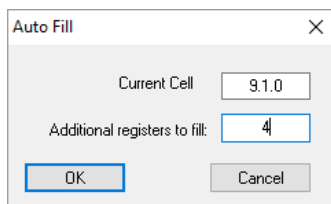
The completed Block Request window.

21. Now go back to **Holding Registers**, select the tab corresponding to the first array (which says 9.0.0), and click **Re-read**. The cells now update with the correct values.

Part Two: Reading HART Registers

Unit16	Float	Capacity
9.1.0	Float 0	0
9.1.1	Float 1	0
9.1.2	Float 2	0
9.1.3	Float 3	0

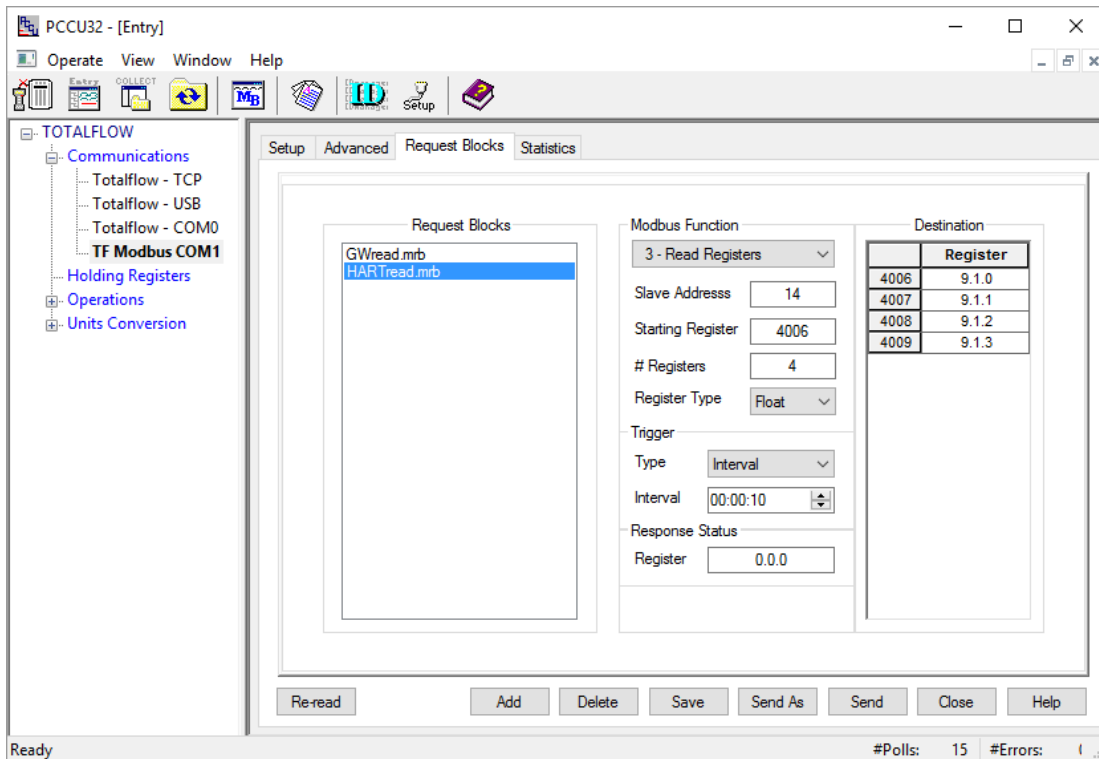
We go through a very similar process for the HART registers. We already set up an array of four Floats above (called "Float"). That tab shows registers from 9.1.0 to 9.1.3, as shown to the left. That means we need to set the register values that way in the **Request Blocks** window.



22. Back in the **Request Blocks** window, click Add and create a file "HARTread".

23. Then set function 3, the slave ID of the device (here 14), starting register **4006** (4005 + 1), and four registers of type **Float**. The interval is set to ten seconds here, but it can be any value.

24. "Auto Fill" is also set to reflect that the register table starts with 9.1.0 and has four entries.



Another completed Block Request window.

25. Click **Save**, and replace any file if it already exists. Then click **Send**.
26. Once that finishes, return to **Holding Registers**, and go to the tab corresponding to the second array ("Float" in this example). Then click **Re-read** and the cells will now update with the correct values.

