AN3975
Application note

Transparent serial link over ST7590 OFDM PRIME modem

1 Introduction

Nowadays, a lot of power meter manufacturers or smart grid providers are switching from simple networks like RS845 to more sophisticated ones like PRIME. In order to help and inform CTMs in their evaluation phase with transition from a simple protocol to a more sophisticated one, a transparent bridge between RS485 and PRIME would be welcomed. Such a system could easily help to evaluate PRIME protocol in the lab and in the field. STMicroelectronics™, as a chip manufacturer, provides a system-on-chip realizing PRIME protocol ST7590, and the implementation of such a transparent bridge is described in this document. A part of this application note is a zip archive file with a firmware referenced within this document.

Contents

List of tables

List of figures

2 System

2.1 System description

The system being described contains two basic parts:

Power meter part: An interface to a power meter that can communicate with superior system over RS485.

PC part: An interface that can communicate with the PC (using a USB connection) that is a superior system for the power meter.

Both interfaces are interconnected by a power line link using the ST7590 demonstration board. Once there are some data to be sent to the power meter, the PC sends this data to the USB Virtual COM port. In this system, the microcontroller (PC part) takes the data from the USB, encapsulates it into the commands for the first power line modem (base node) and sends it to the power line. Another modem reads the data coming from the power line (service node), and sends them to the microcontroller, power meter part. The microcontroller extracts the data from the power line modem and sends them to UART_A. The RS485 driver is connected to UART_A, so the data sent to the UART_A are converted to RS485 and these data are delivered to the power meter. If the power meter replies, the data comes back through this channel to the PC.

The microcontroller module used in this application is the connectivity gateway (STEVAL-PCC012V1). This module is referred to as CG in the following text.

2.2 System specification

Power meter part (see *Figure 1* and *Figure 7*)

- Power meter with RS485 terminals. Two terminals (+, -) for half duplex RS485
- Line driver for RS485 to UART_A
- Connectivity gateway system with STM32 interfacing RS485 via UART_A
- Connectivity gateway system with STM32 interfacing ST75xx via UART_B
- PLM ST75xx using UART_B and connection to power lines

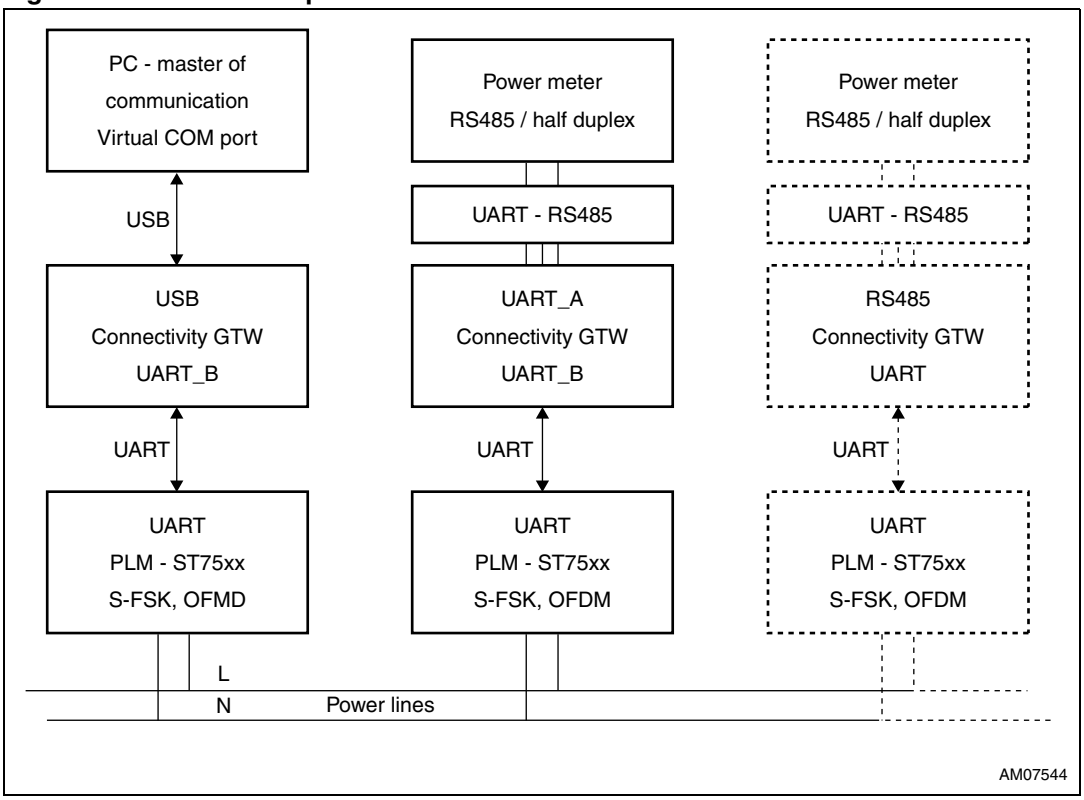
PC part (see *Figure 1*)

- PC with USB
- Connectivity gateway system with STM32 offering USB connection providing Virtual COM port functionality.
- Connectivity gateway system with STM32 interfacing ST75xx via UART_B
- PLM ST75xx using UART_B and connection to power lines

Modulation and modem specifications

- ST7590, OFDM modulation, PRIME protocol, one logical channel opened by service node.

Figure 1. RS485 over power line



3 System implementation - HW

3.1 Power meter (RS485) part

Figure 2 shows the HW implementation of the power meter part (see *Figure 7*). It consists of:

- Connectivity gateway demonstration board (converts command coming from UART (UART_B) to UART (UART_A) of the RS485 module).
- RS485 module (converts UART (UART_A) to RS485 and vice versa).

Figure 2. Connectivity gateway board with RS485 extension

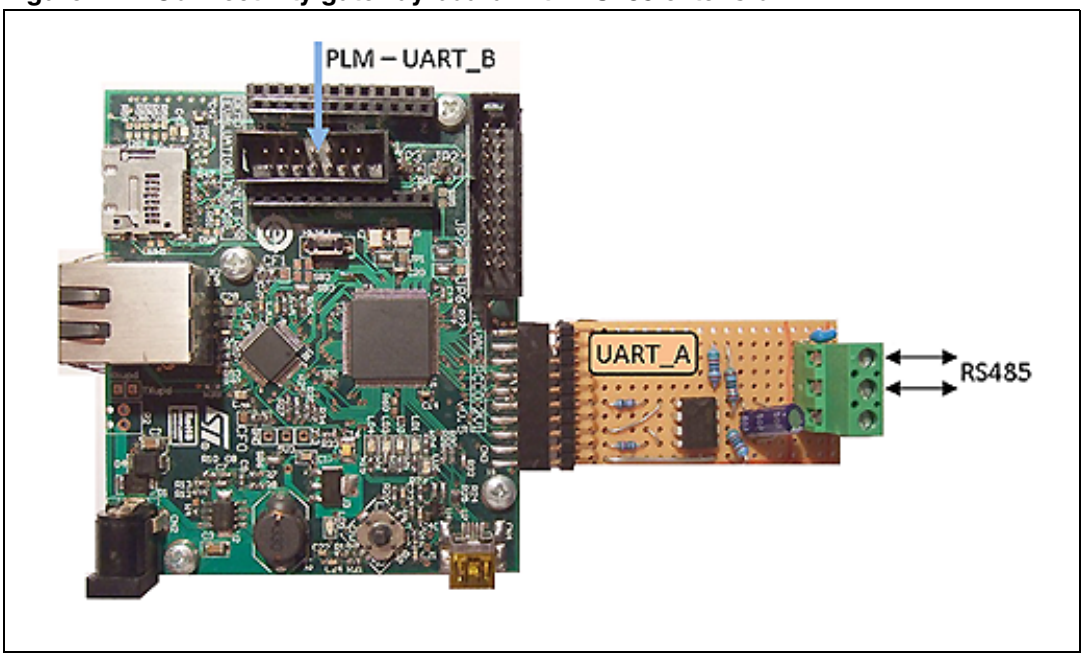


Figure 3 shows the schematic of the RS485 module. The connection of the pins for the RS485 driver is given in *Table 1*. The matching resistor R_b can be simply disconnected by jumper J1 which is not depicted on the schematic.

Figure 3. ST485ABDR in SO8 package - RS485 module schematic

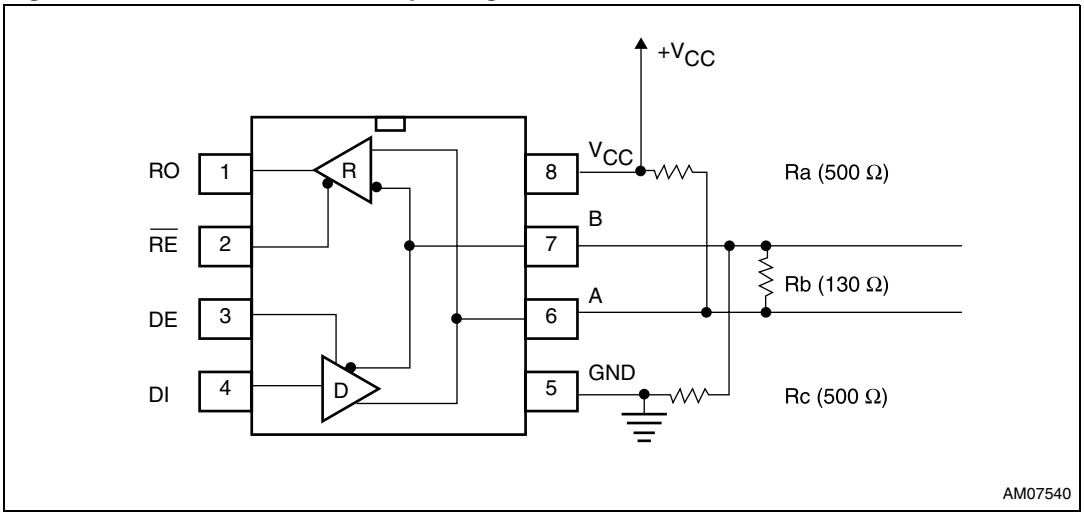


Figure 4 visualizes *Table 1* showing the physical position of the signals located on the extension connector CN3 of the connectivity gateway.

Figure 4. CG (connectivity gateway) CN3 connector

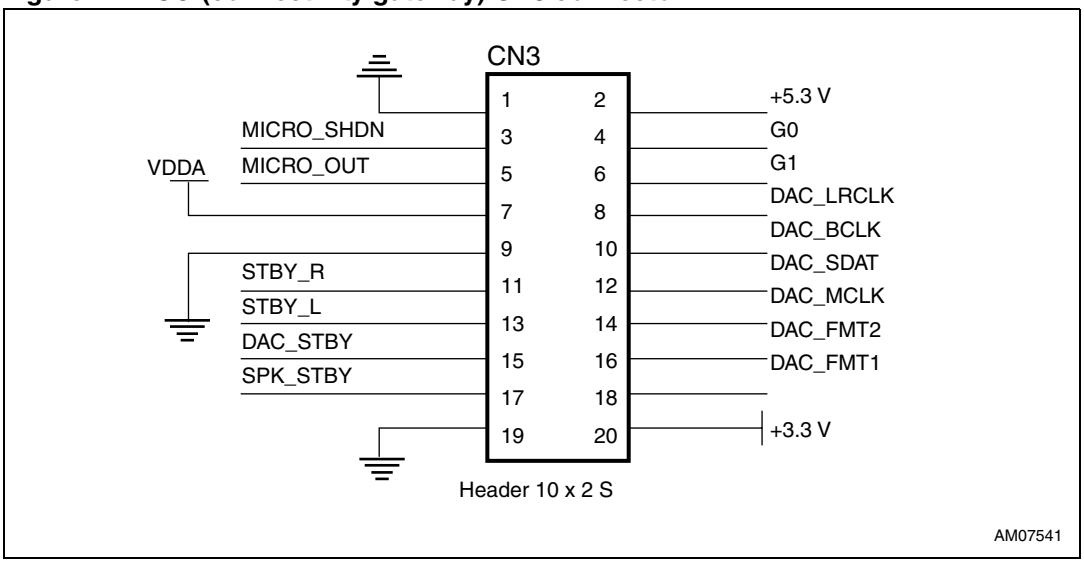
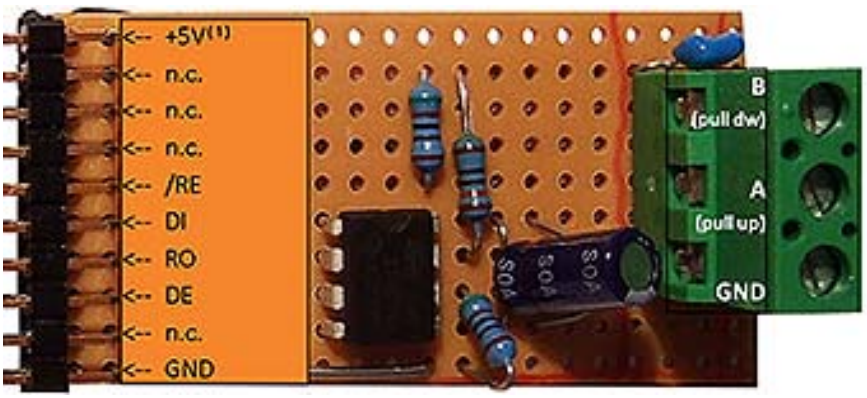


Table 1. Signals of interconnection of STEVAL-PCC012V1 and RS485 module

Signal	CN3 (CG)	RS485
USART_RX	DAC_STBY - PD6 - PIN 15	R0
GPIO	STBY_R - PD4 - PIN11	$\overline{RE}$
GPIO	SPK_STBY - PD7 - PIN17	DE
USART_TX	STBY_L - PD5 - PIN13	DI
+5 V	PIN 2	V_{CC}, +5 V
GND	PIN 19	GND

Figure 5 shows the physical implementation and signal pinout listed in *Table 1* of the RS485 module. Outputs of the RS485 driver, the RS485 bus, are bonded to the connector depicted on the right in *Figure 5*. The RS485 bus has two lines named A and B. The common ground is also bonded out at the same connector. Two coupling capacitors 10 µF and 100 nF are connected between +5 V and GND. We recommend adding 1 kΩ serial resistors to the signals between the microcontroller board CG and RS485 driver (namely, signals RO, RE, DE, DI) during the debugging phase in order to avoid damage of used chips. The resistors are not necessary and can be removed once the application works as expected.

Figure 5. Module signal pinout



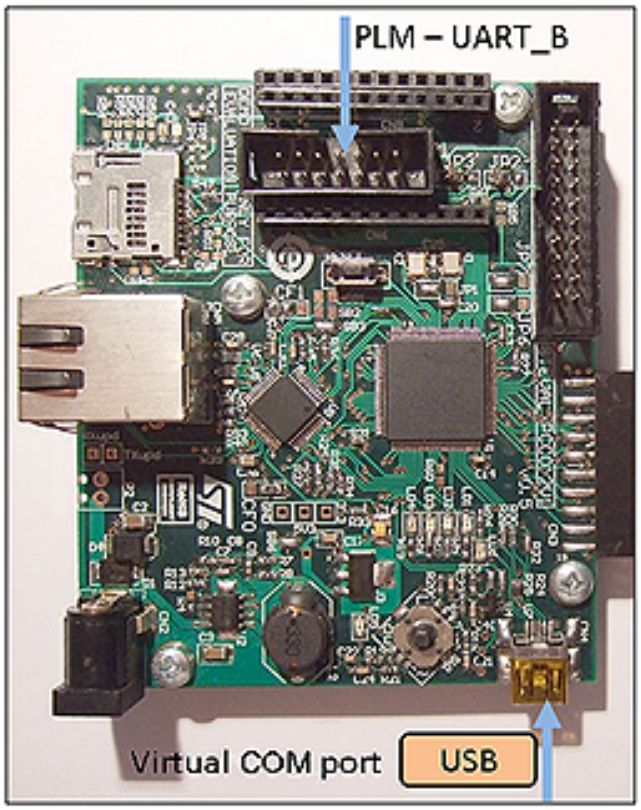
1. Signal +5 V is connected to the bottom pin of the header.

3.2 PC (HyperTerminal) part

Figure 6 shows the HW implementation of the PC part (see *Figure 7*). It consists of:

- Connectivity gateway demonstration board (converts command coming from UART (UART_B) to USB.

Figure 6. Connectivity gateway board with interfacing USB and PLM



4 System implementation - FW

4.1 PC (HyperTerminal) part

The firmware of the transparent link application for the PC part provides conversion of the data coming from the USB to commands sent over UART (UART_B) to the power line mode. The firmware also contains the USB stack for Virtual COM port class - see *Figure 7*.

Firmware project for PC part details

Programming environment

IAR™ project written in embedded workbench® for ARM® IAR 5.50

C Project, location of the project file:

/PC_USB_to_PRIME/Project/Virtual_COM_Port/EWARMv5/VirtualCOMPort.eww

(After decompression of the archive that comes with this application note.)

4.2 Power meter (RS485) part

The firmware of the transparent link application for the power meter part provides conversion of the power line data commands coming from the UART (UART_B) to data sent over UART (UART_A) to RS485 module which may be connected to any device using RS485, e.g. power meter - see *Figure 7*.

Firmware project for power meter part details

Programming environment

IAR project written in embedded workbench for ARM IAR 5.50

C Project, location of the project file:

PMeter_RS485_to_PRIME/Project/Prime_to_RS485/EWARMv5/VirtualCOMPort.eww

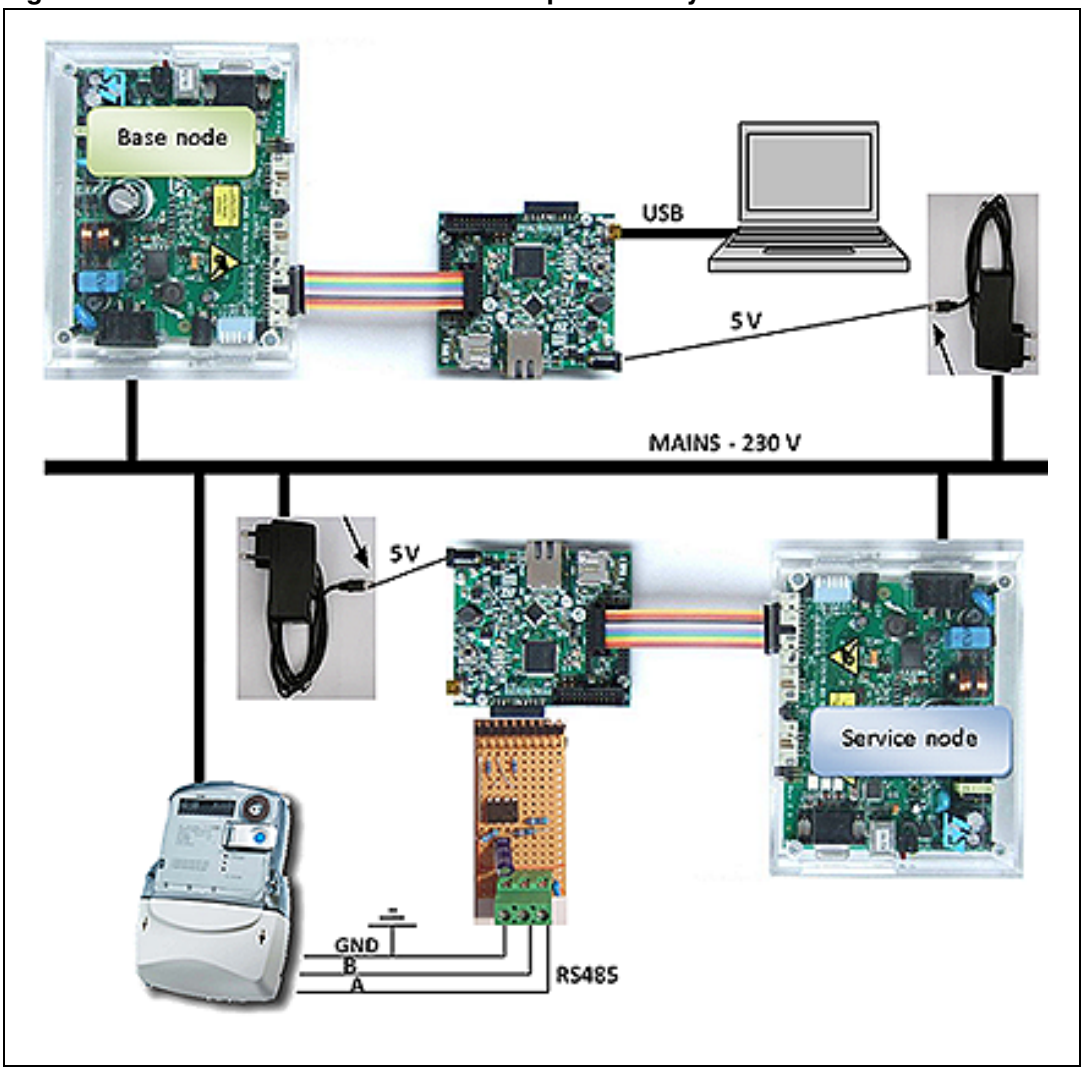
(After decompression of the archive coming with this application note.)

5 System setup

5.1 HW interconnection

Figure 7 shows the complete setup of the application for transparent serial link over power line modem. The top half of the figure shows the so called PC part of the application, the bottom half shows the power meter part. After HW setup of such a system, it is possible to continue with FW and SW installation.

Figure 7. HW interconnection of the complete test system

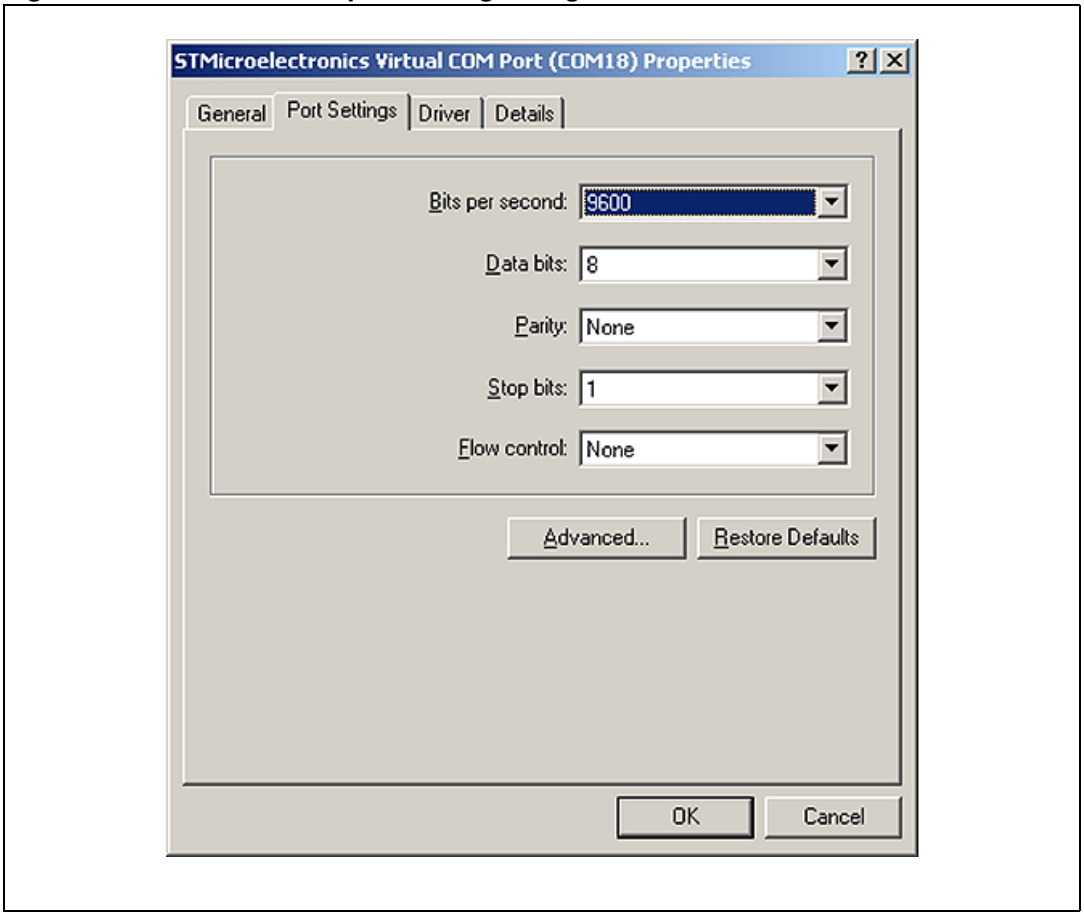


5.2 Communication setup

In order to be able to run the application, the following steps must be completed after correct setup of the HW:

1. Flash the FW: PC_USB_to_PRIME project into the connectivity gateway (STEVAL-PCC012V1) node connected to ST7590 base node.
2. Flash the FW: PMeter_RS485_to_PRIME project into the connectivity gateway (STEVAL-PCC012V1) node connected to ST7590 service node.
3. Power up both the connectivity gateways (STEVAL-PCC012V1) (with 5 V DC power supply).
4. Power up the power meter.
5. If necessary, install the Virtual COM port driver supplied by STMicroelectronics. This driver is supplied within the firmware package.
6. Run Windows® HyperTerminal on the computer connected by a mini-USB cable to the connectivity gateway (STEVAL-PCC012V1) connected to base node.
7. Setup the Virtual COM port setting according to needs (“Speed”, “Data bits”, “Parity”, etc.):

Figure 8. Windows COM port setting dialog



8. Open the Virtual COM port in Windows HyperTerminal or another application used for AMR (automatic meter reading).
9. Power up the power line modems from mains and wait for modem interconnection.

Table 2. LED behavior on CG for PC part of the application

PCC012V1 - base node (PC part)		
LED	**Behavior**	**Meaning**
LED2 (red)	Blinks	ST7590 modem not found
All LEDs	Off	Waiting for logical channel
LED1 (orange)	Shines	Channel established

Table 3. LED behavior of CG for power meter part of the application

PCC012V1 - service node (power meter part)		
LED	**Behavior**	**Meaning**
LED2 (red)	Blinks	ST7590 modem not found
LED3 and 4	Shine	Requesting for logical channel
LED1 (orange)	Shines	Channel established

10. After LED1 (orange) on both connectivity gateways (STEVAL-PCC012V1s) shines, continue to follow this list.
11. Connect Windows HyperTerminal or the application to the Virtual COM port 9.
12. Send or receive data using Windows HyperTerminal or the application to or from the power meter.

6 Tests

6.1 Signal shape and settings

Figure 9 and *Figure 10* show typical waveforms on the RS485 bus when a data is sent to UART_A. Idle state, start bit, data bits, parity and stop bit can be tracked in these figures. The following examples show the correct position and behavior of the parity bit. Once a byte with the value 65 (that is equal to the ASCII code character A) is sent from the hyper terminal, it is possible to see that the parity bit was set to 0. Once a byte with the value 49 (that is equal to the ASCII code character 1) is sent, it is possible to see that the parity bit was set to 1.

UART_A (RS485) setting: 9600 baud, 7 data bits, even parity, 1 stop bit.

Figure 9. Even parity bit P = 0 on the eighth position of the data bit

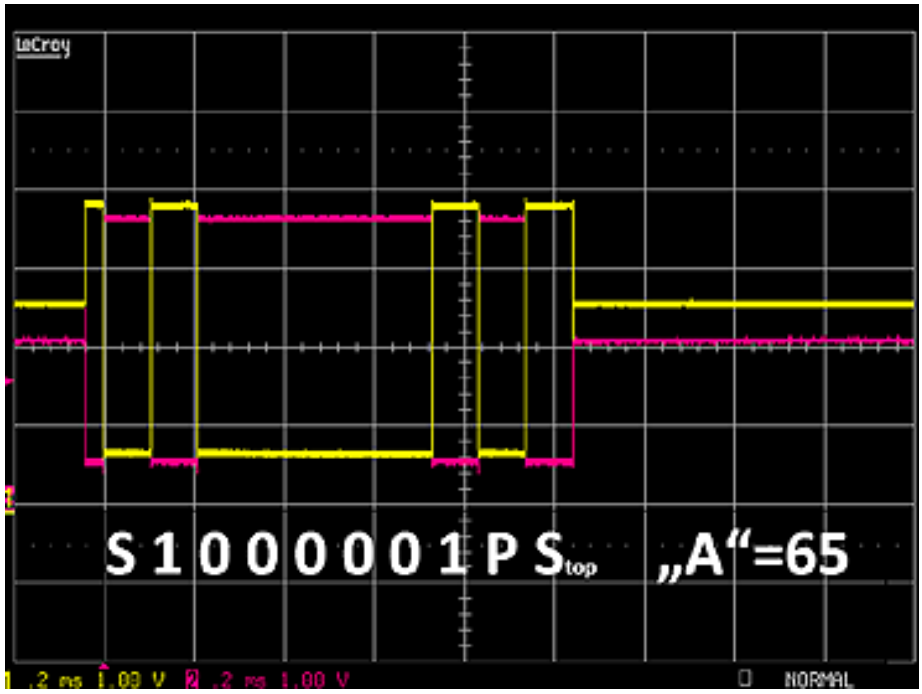
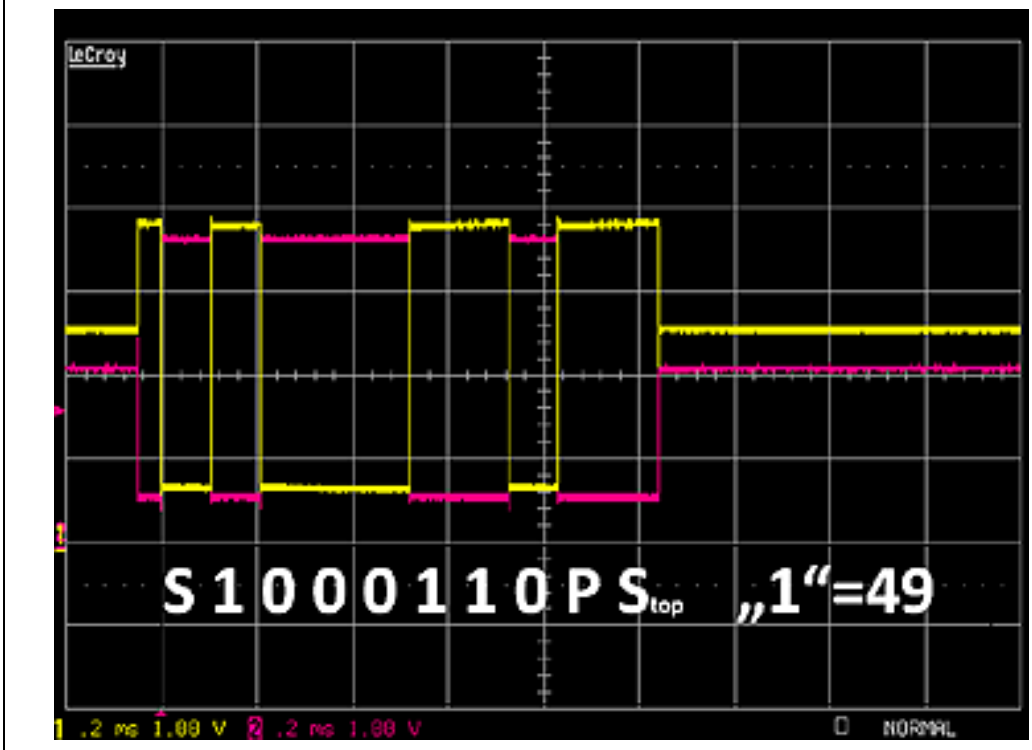


Figure 10. Even parity bit P = 1 on the eighth position of the data bit



6.2 Sending short data from HyperTerminal to RS485 bus using matching resistor

Figure 11 shows typical waveforms measured by the scope on the RS485 bus when one-byte data is sent from PC HyperTerminal to the whole system.

Figure 11. RS485 - one-byte transmission - jumper J1 closed, matching resistor 120 connected

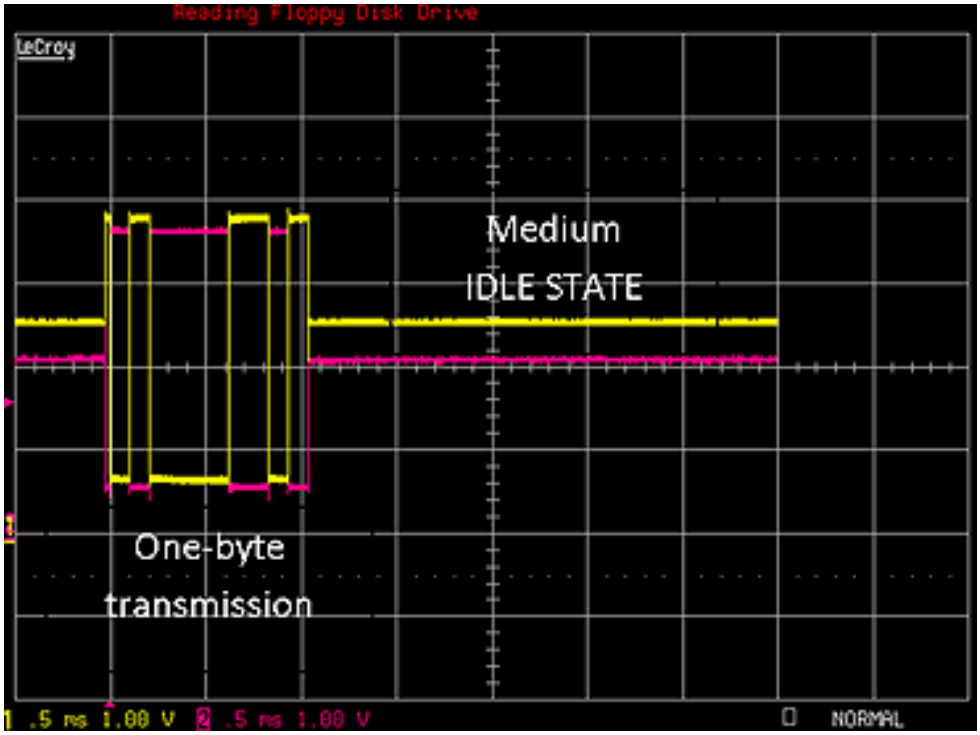
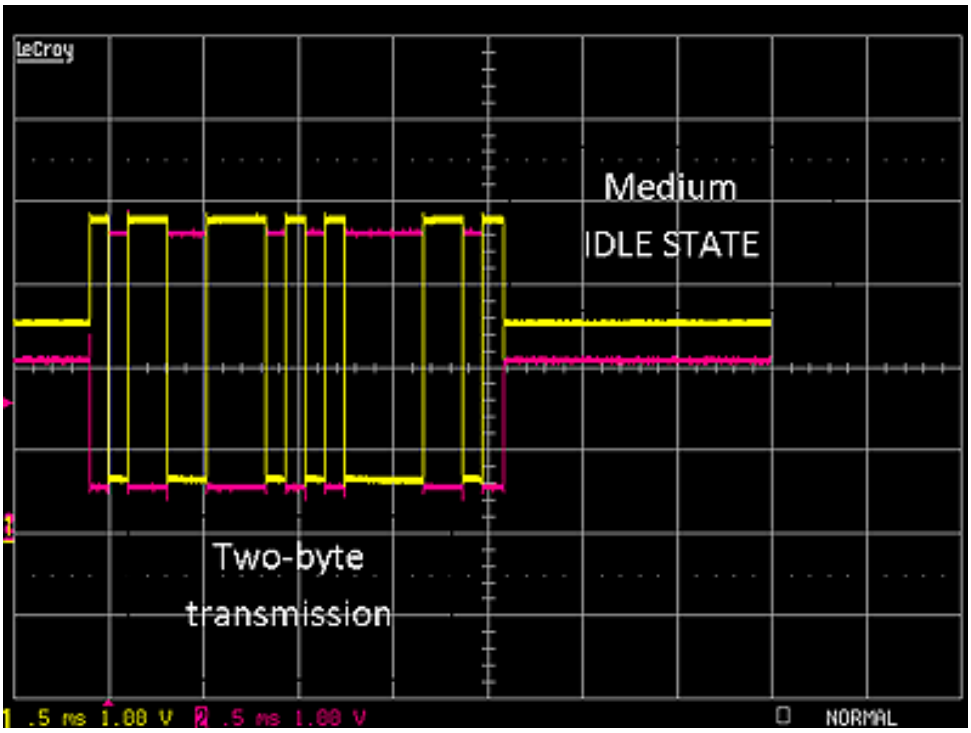


Figure 12 shows typical waveforms measured by the scope on the RS485 bus when two-byte data are sent from PC HyperTerminal to the whole system.

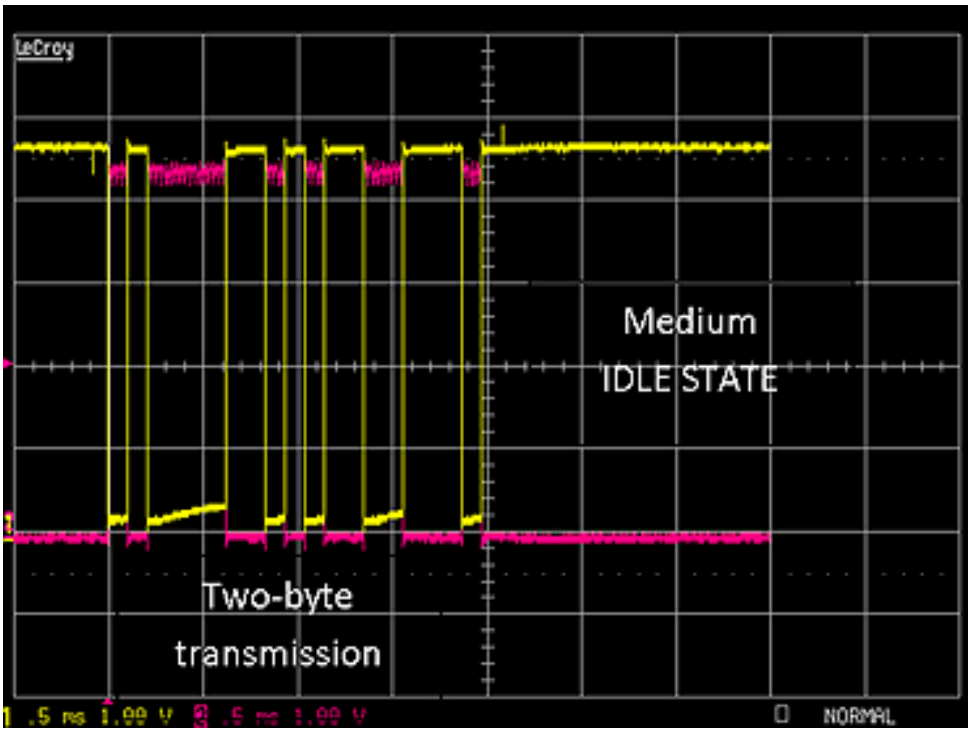
Figure 12. RS485 - two-byte transmission - jumper J1 closed, matching resistor 120 connected



6.3 Sending short data from HyperTerminal to RS485 bus not using matching resistor

Figure 13 shows the influence of disconnection of the matching resistor on RS485 bus lines.

Figure 13. RS485 - two-byte transmission - jumper J1 opened, matching resistor 120 not connected



6.4 Closed loop communication test of 320-byte message

In order to test the whole transparent link before involving the RS485 device, e.g. power meter and automated meter reading SW on the PC side, it is necessary to perform a test that proves the whole chain is working.

Closed loop communication test can be performed by two approaches:

- FW: It can be switched on by changing the FW. See *Appendix A* with guidance on how to update the project file in order to enable closed loop test - see *Figure 14*.
- HW: It can be switched on by changing the HW. The RS485 module should be removed from the CN3 connector of the CG of the power meter part. According to *Table 1*, the USART_TX and USART_RX must be interconnected by a wire - see *Figure 15*.

Figure 14. Closed loop setup for communication test using bigger datafiles, UART loop

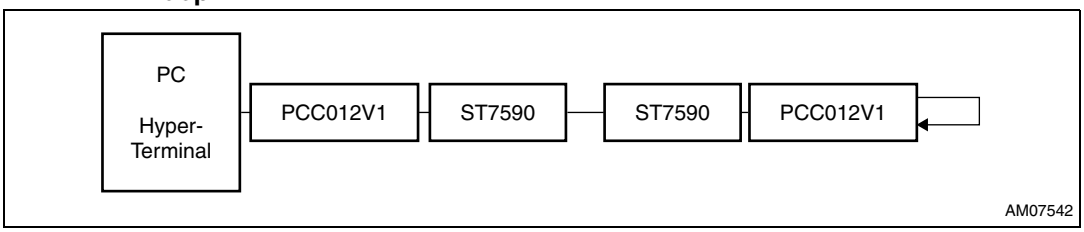
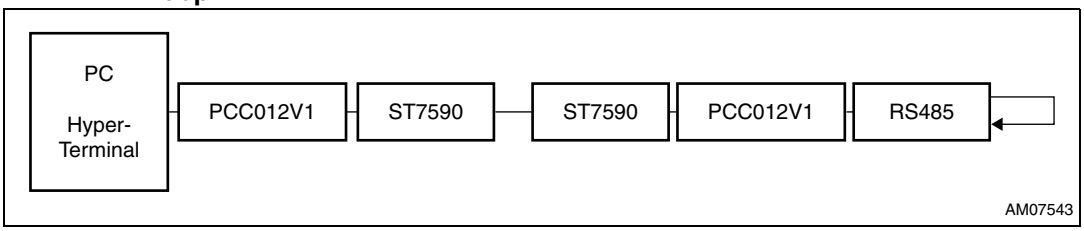
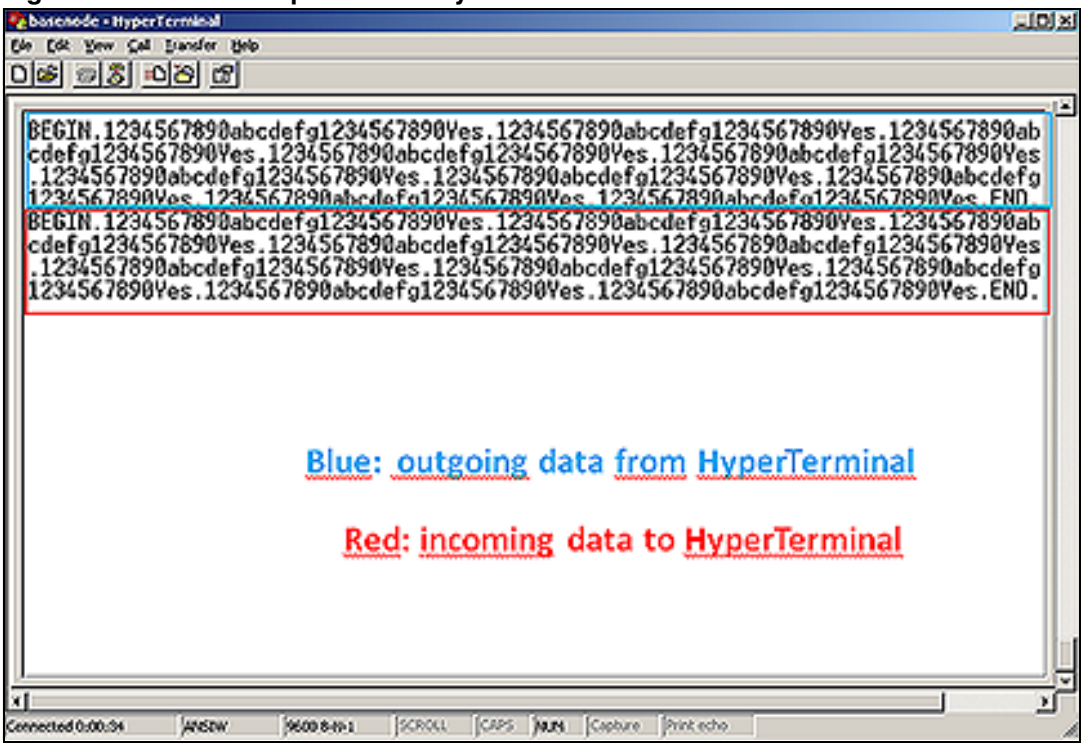


Figure 15. Closed loop setup for communication test using bigger datafiles, RS485 loop



After setting the system to work in closed loop test mode, the test data block can be sent to the system via Windows HyperTerminal, see *Figure 16*. It is recommend to create a file with predefined data to send in order to be able to perform this test quickly and to avoid superfluous typing of the characters with every new test.

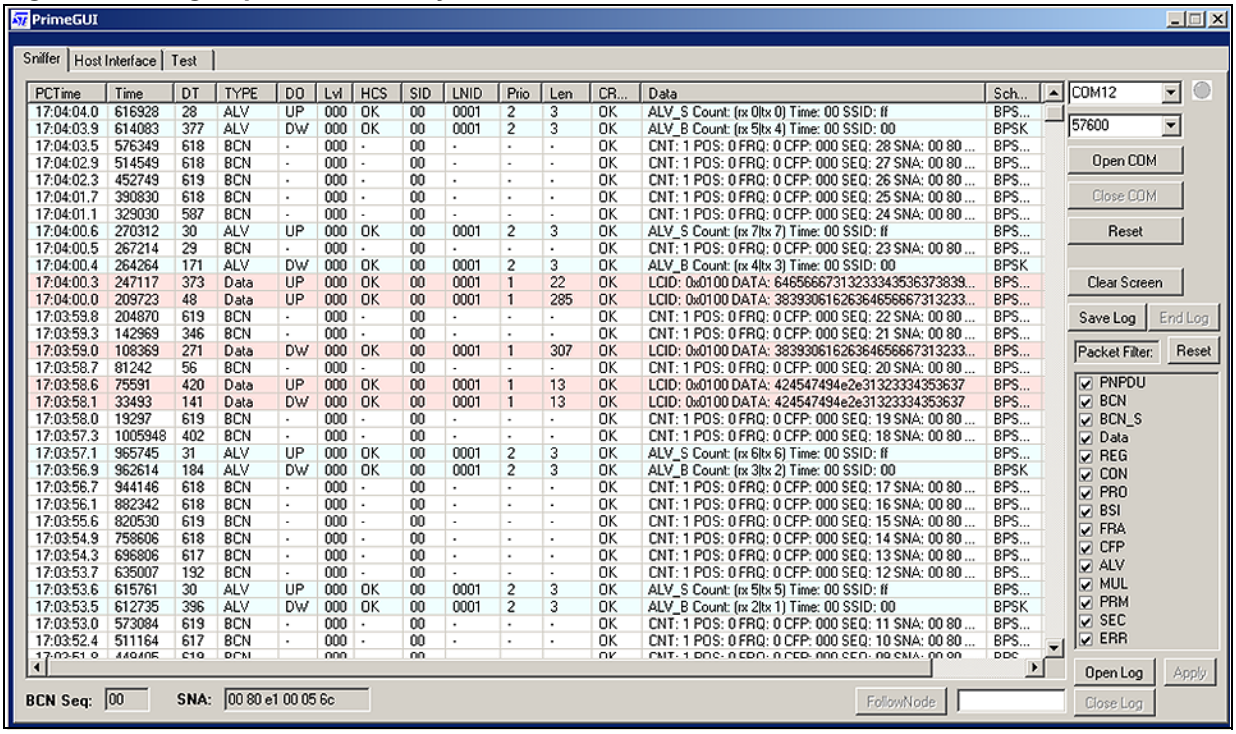
Figure 16. Closed loop test - 320-byte data transfer



In order to have higher control over the system and over the traffic on the power line, one more ST7590 modem configured as a service node should be connected to the power line network. The node should be switched into the sniffer mode by the PC GUI application

supplied with the ST7590 demonstration board. The traffic on the GUI can be observed. Traffic during the test on *Figure 16* is shown in *Figure 17*.

Figure 17. Log of packets - 320-byte data transfer



Packet length in both directions must be 320 in order to pass the test

Column Type: Data means data packets

Column Len: Means length of the data message

Column DO: DW that means data sent from base node to service node (down direction)

For DW packets: Len = 13 + 307 = 320 bytes.

Column DO: DW that means data sent from base node to service node (up direction)

For UP packets: Len = 13 + 285 + 22 = 320 bytes.

6.5 Closed loop communication test of 2565-byte message

Figure 18. Closed loop test - 2,565-Kilobyte data transfer

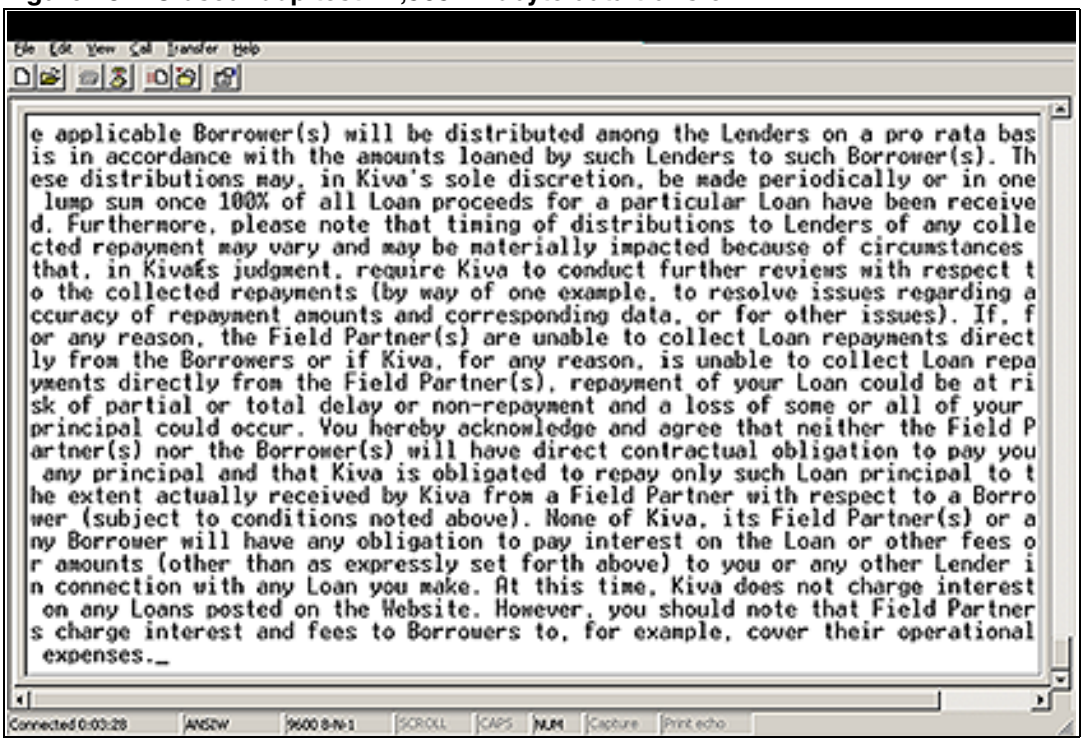
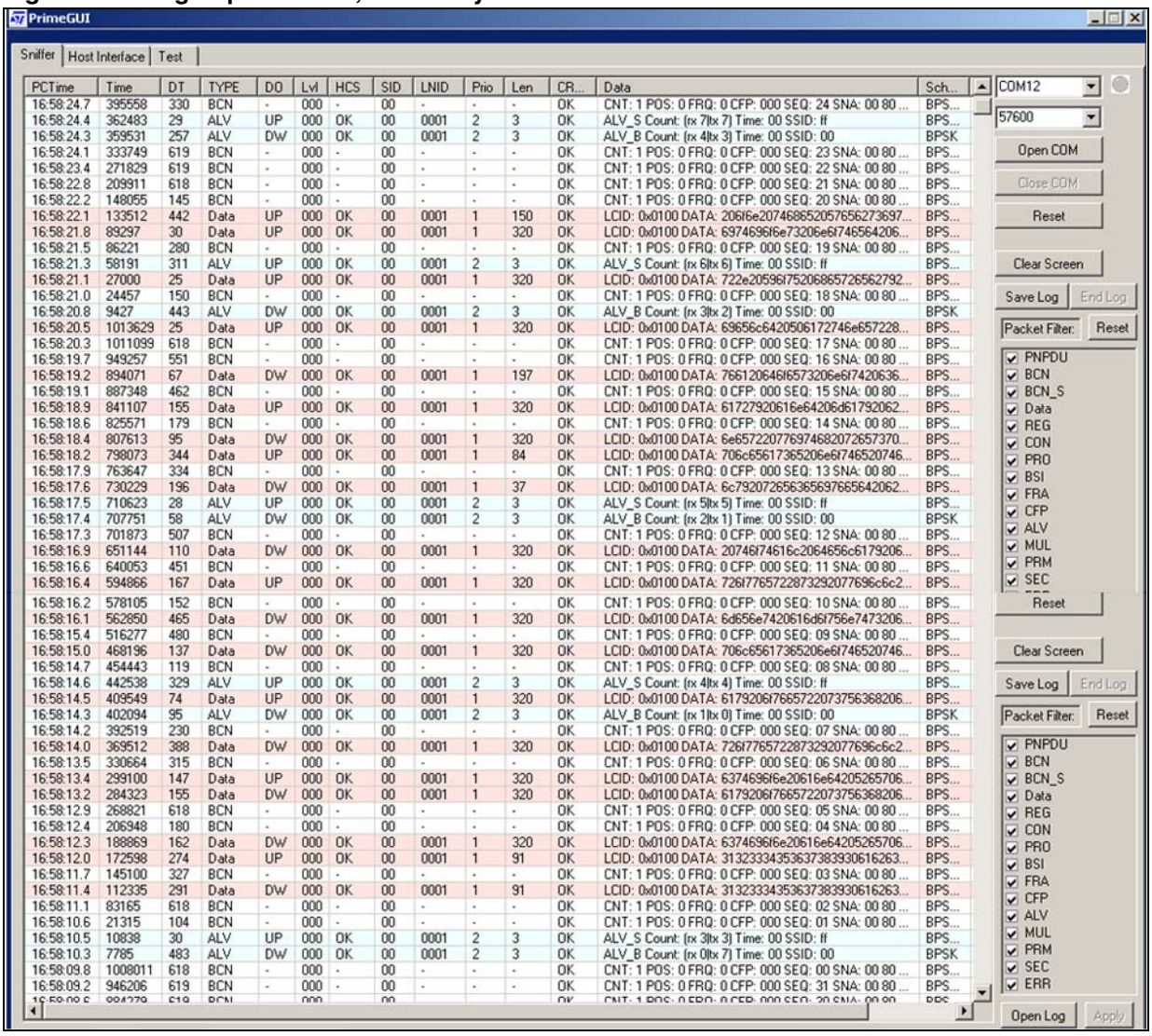


Figure 19. Log of packets - 2,565-Kilobyte data transfer



Packet length in both directions must be 2565 in order to pass the test:

Column Type: Data means data packets

Column Len: Means length of the data message

Column DO: DW that means data sent from base node to service node (down direction)

For DW packets: Len = 91 + 320 + 320 + 320 + 320 + 320 + 320 + 37 + 320 + 197 = 2565 bytes.

Column DO: UP that means data sent from service node to base node (up direction)

For UP packets: Len = 91 + 320 + 320 + 320 + 84 + 320 + 320 + 320 + 320 + 150 = 2565 bytes.

Appendix A

In order to enable or disable close loop test function PLM_To_RS485UART_Send_Data in the function.c file in the PMeter_RS485_to_PRIME project must be updated accordingly.

Function that avoids close loop test

```
Void PLM_To_RS485UART_Send_Data(unsigned char* data_buffer, unsigned short
Nb_bytes)
{
 GPIO_SetBits(nRE_port, nRE_pin);    //Receiver input disable
 GPIO_SetBits(DE_port, DE_pin);      //Driver output enable

 ComWrt_direct (0, data_buffer, Nb_bytes, RS485);
 while(USART_GetFlagStatus(USART2, USART_FLAG_TC) == RESET);

 GPIO_ResetBits(DE_port, DE_pin);    //Driver output disable
 GPIO_ResetBits(nRE_port, nRE_pin);  //Receiver input enable
}
```

Function that enables close loop test

Use only for this test, do not use when communicating with the RS485 device.

```
Void PLM_To_RS485UART_Send_Data(unsigned char* data_buffer, unsigned short
Nb_bytes)
{
 GPIO_ResetBits(nRE_port, nRE_pin);  //Receiver input enable
 GPIO_SetBits(DE_port, DE_pin);      //Driver output enable

 ComWrt_direct (0, data_buffer, Nb_bytes, RS485);
 while(USART_GetFlagStatus(USART2, USART_FLAG_TC) == RESET);
```

Revision history

Table 4. Document revision history

Date	Revision	Changes
16-Nov-2011	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS EXPRESSLY APPROVED IN WRITING BY TWO AUTHORIZED ST REPRESENTATIVES, ST PRODUCTS ARE NOT RECOMMENDED, AUTHORIZED OR WARRANTED FOR USE IN MILITARY, AIR CRAFT, SPACE, LIFE SAVING, OR LIFE SUSTAINING APPLICATIONS, NOR IN PRODUCTS OR SYSTEMS WHERE FAILURE OR MALFUNCTION MAY RESULT IN PERSONAL INJURY, DEATH, OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE. ST PRODUCTS WHICH ARE NOT SPECIFIED AS "AUTOMOTIVE GRADE" MAY ONLY BE USED IN AUTOMOTIVE APPLICATIONS AT USER’S OWN RISK.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.

Information in this document supersedes and replaces all information previously supplied.

The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2011 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com