

CD600 Communication Driver

Driver for Serial Communication with
CD600 devices

Contents

INTRODUCTION2

GENERAL INFORMATION.....3

 DEVICE CHARACTERISTICS3

 LINK CHARACTERISTICS3

 DRIVER CHARACTERISTICS3

 CONFORMANCE TESTING4

INSTALLING THE DRIVER5

CONFIGURING THE DRIVER6

 SETTING THE COMMUNICATION PARAMETERS6

 CONFIGURING THE DRIVER WORKSHEETS8

EXECUTING THE DRIVER 13

TROUBLESHOOTING 14

SAMPLE APPLICATION 16

REVISION HISTORY..... 17

Introduction

The CD600 driver enables communication between the Studio system and CD600 devices using Smar proprietary protocol , according to the specifications discussed in this document.

This document was designed to help you install, configure, and execute the CD600 driver to enable communication with these devices. The information in this document is organized as follows:

- **Introduction:** Provides an overview of the CD600 driver documentation.
- **General Information:** Provides information needed to identify all the required components (hardware and software) used to implement communication between Studio and the CD600 driver.
- **Installing the Driver:** Explains how to install the CD600 driver.
- **Configuring the Driver:** Explains how to configure the CD600 driver.
- **Executing the Driver:** Explains how to execute the driver to verify that you installed and configured the driver correctly.
- **Troubleshooting:** Lists the most common error codes for this protocol and explains how to fix these errors.
- **Sample Application:** Explains how to use a sample application to test the CD600 driver configuration.
- **Revision History:** Provides a log of all modifications made to the driver and the documentation.



Notes:

- This document assumes that you have read the “Development Environment” chapter in the *Studio Technical Reference Manual*.
- This document also assumes that you are familiar with the Windows NT/2000/XP environment. If you are unfamiliar with Windows NT/2000/XP, we suggest using the **Help** feature (available from the Windows desktop **Start** menu) as you work through this guide.

General Information

This chapter explains how to identify all the hardware and software components used to implement communication between the Studio CD600 driver and the Smar CD600 devices.

The information is organized into the following sections:

- Device Characteristics
- Link Characteristics
- Driver Characteristics

Device Characteristics

To establish communication, you must use devices with the following specifications:

- **Manufacturer:** Smar
- **Compatible Equipment:**
 - CD600
- **Programming Software:** -
- **Device Runtime Software:** None

For a list of the devices used for conformance testing, see “Conformance Testing”.

Link Characteristics

To establish communication, you must use links with the following specifications:

- **Device Communication Port:** Serial Port
- **Physical Protocol:** RS-232
- **Logic Protocol:** Smar Proprietary
- **Specific PC Board:** ICS2.0 (external RS232/RS485 converter)

Driver Characteristics

The CD600 driver is composed of the following files:

- **CD600.INI:** Internal driver file. *You must not modify this file.*
- **CD600.MSG:** Internal driver file containing error messages for each error code. *You must not modify this file.*
- **CD600.PDF:** Document providing detailed information about the CD600 driver.
- **CD600.DLL:** Compiled driver.

Notes:

- All of the preceding files are installed in the /DRV subdirectory of the Studio installation directory.
- You must use Adobe Acrobat® Reader™ (provided on the Studio installation CD-ROM) to view the *CD600.PDF* document.

You can use the CD600 driver on the following operating systems:

- Windows NT/2K/XP

For a list of the operating systems used for conformance testing, see “Conformance Testing” on page 4.

Conformance Testing

The following hardware/software was used for conformance testing:

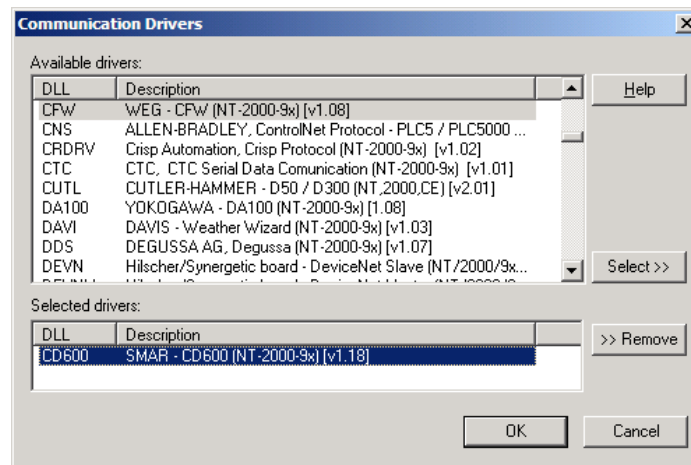
- **Equipment:** CD600
- **Configuration**
 - **COM Port:** COM 1
 - **Baud Rate:** 19200
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **Parity:** Even
- **Cable:** Use specifications described in the “Link Characteristics” section
- **Operating System (development):** Windows XP with Service Pack 1
- **Operating System (target):**
 - Windows 2000
 - Windows XP
- **Studio Version:** 6.0
- **Driver Version:** 1.18

Installing the Driver

When you install Studio version 5.1 or higher, all of the communication drivers are installed automatically. You must select the driver that is appropriate for the application you are using.

Perform the following steps to select the driver from within the application:

1. Open Studio from the **Start** menu.
2. From the Studio main menu bar, select **File** → **Open Project** to open your application.
3. Select **Insert** → **Driver** from the main menu bar to open the *Communication drivers* dialog.
4. Select the CD600 driver from the *Available Drivers* list (as shown in the following figure), and then click the **Select** button.



Communication Drivers Dialog Box

5. When the CD600 driver displays in the **Selected Drivers** list, click the **OK** button to close the dialog.



Note:

It is not necessary to install any other software on your computer to enable communication between the host and the device.



Attention:

For safety reasons, you must use special precautions when installing the physical hardware. Consult the hardware manufacturer's documentation for specific instructions in this area.

Configuring the Driver

After opening Studio and selecting the CD600 driver, you must configure the driver. Configuring the CD600 driver is done in two parts:

- Specifying communication parameters
- Defining tags and controls in the *STANDARD DRIVER SHEETS* (or Communication tables)

Worksheets are divided into two sections, a *Header* and a *Body*. The fields contained in these two sections are standard for all communications drivers — except the **Station**, **Header**, and **Address** fields, which are driver-specific. This document explains how to configure the **Station**, **Header**, and **Address** fields only.

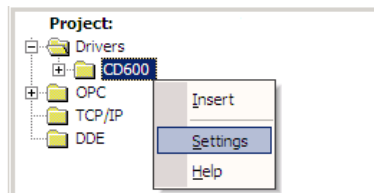
Note:

For a detailed description of the Studio *STANDARD DRIVER SHEETS*, and information about configuring the standard fields, review the product's *Technical Reference Manual*.

Setting the Communication Parameters

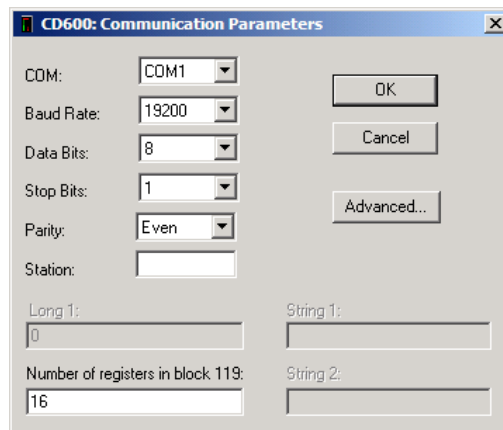
Use the following steps to configure the communication parameters, which are valid for all driver worksheets configured in the system:

1. From the Studio development environment, select the *Comm* tab located below the *Workspace*.
2. Click on the *Drivers* folder in the *Workspace* to expand the folder.
3. Right-click on the CD600 subfolder and when the pop-up menu displays, (as shown in the following figure) select the **Settings** option.



Select Settings from the Pop-Up Menu


The CD600: *Communications Parameters* dialog displays (as follows).




Communication Parameters Dialog

4. Specify the custom parameters as noted in the following table:


Parameters	Default Values	Valid Values	Description
Station	0	0	Not used for this driver
Number of registers in block 119	0	Any number	Number of LI's configured in block 119 of CD600 device. Example: 16

 **Note:**
The device must be configured with *exactly the same* parameters that you configured in the CD600 *Communication Parameters* dialog.

5. Click the **Advanced** button on the *Communication Parameters* dialog to open the *Advanced Settings* dialog and configure the settings that are necessary.

 **Notes:**

- Do not change any of the other *Advanced* parameters at this time. You can consult the Studio *Technical Reference Manual* for information about configuring these parameters for future reference.
- Generally, you must change the *Advanced* parameter settings if you are using a DCE (Data Communication Equipment) converter (232/485 for example), modem, and so forth between the PC, driver, and the host. You must be familiar with the DCE specifications before adjusting these configuration parameters.

 **Attention:**
The Echo option for the serial communication of the CD600 device must be deactivated, otherwise the CD600 driver may not work properly. Contact SMAR for more information about how to deactivate Echo.

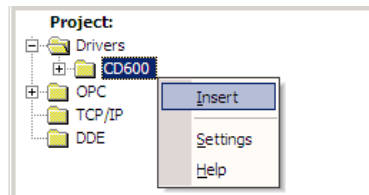
Configuring the Driver Worksheets

This section explains how to configure the *STANDARD DRIVER SHEET*s (or communication tables) to associate application tags with the device addresses. You can configure multiple Driver Worksheets — each of which is divided into a *Header* section and *Body* section.

Configuring the *STANDARD DRIVER SHEET*

Use the following steps to create a new *STANDARD DRIVER SHEET*:

1. From the Studio development environment, select the *Comm* tab, located below the *Workspace* pane.
2. In the *Workspace* pane, expand the *Drivers* folder and right-click the CD600 subfolder.
3. When the pop-up menu displays (as shown in the following figure), select the **Insert** option.



Inserting a New Worksheet

 **Note:**

To optimize communication and ensure better system performance, you must tie the tags in different driver worksheets to the events that trigger communication between each tag group and the period in which each tag group must be read or written. Also, we recommend configuring the communication addresses in sequential blocks to improve performance.

The *STANDARD DRIVER SHEET* displays (similar to the following figure).

The screenshot shows a configuration window for a driver. It has several sections:

- Description:** A text box containing "Driver Test" and a checkbox for "Increase priority".
- Read Trigger:** Fields for "RdTrg" (containing "EnRd"), "EnRd" (containing "EnRd"), "RdCmpl" (containing "RdCmpl"), and "RdSt" (containing "RdSt").
- Write Trigger:** Fields for "WrTrg" (containing "EnWr"), "EnWr" (containing "EnWr"), "WrCmpl" (containing "WrCmpl"), and "WrSt" (containing "WrSt").
- Station:** A field containing "8".
- Header:** A field containing "5:120".
- Min/Max:** Two empty fields with "Min:" and "Max:" labels.
- Table:** A table with 5 columns: Index, Tag Name, Address, Div, and Add. It contains 7 rows of data.

	Tag Name	Address	Div	Add
1	Tag[1]	KP1		
2	Tag[2]	TR1		
3	Tag[3]	TD1		
4	Tag[4]	KP2		
5	Tag[5]	TR2		
6	Tag[6]	TD2		
7	Tag[7]	KP3		

STANDARD DRIVER SHEET

In general, all parameters on the Driver Worksheet (except the **Station**, **Header**, and **Address** fields) are standard for all communication drivers, but they will not be discussed in this document. For detailed information about configuring the standard parameters, consult the *Studio Technical Reference Manual*.

- Use the following information to complete the **Station**, **Header**, and **Address** fields on this worksheet.

Station field: Specify the device using the following syntax:

<PLC ID>

Where:

- PLC ID** is the Station number of CD600 device. IA Parameters in configuration of CD600 device.

Header field: Specify the Header using the following syntax:

<Loop>:<Block> (For example: 5:119)

Where:

- Loop** is the loop number. (From 0 up to 5)
- Block** is the block number. (From 1 up to 121)

When communicating with more than one CD600 device simultaneously, it is possible to configure the number of LIs and Number of Alarms for the block 119 of each CD600 in the **Header** field, according to the following syntax:

<Loop>:119:<Number of Lis>:<Number of Alarms>

Where:

- **Loop** is the loop number. (0 up to 5)
- **Number of LIs** is the number of configured LI's in the block 119
- **Number of Alarms** is the number of Alarms

If the **Number of LIs** is not configured in Header, the driver uses the value configured in **Communication Parameters** and the **Number of Alarms** will be set up to 20.


To write a configuration file to CD600 device, configure the text **CONF** in the **Header** field.

After editing the **Header** field, Studio checks the syntax to determine if it is valid. If the syntax is incorrect, Studio automatically inserts the default value in the **Header** field.

Also, you can type a tag string in brackets {**Tag**} into the **Header** field, but you must be certain that the tag's value is correct and that you are using the correct syntax, or you will get an invalid Header error.

Address field: Use this field to associate each tag to its respective device address.

Type the tag from your application database into the **Tag Name** column. This tag will receive values from or send values to an address on the device. The address must comply with the following syntax:

 **Note:**

- Read Only addresses (do not support writing commands):
 - <Loop>:119 (All addresses)
 - <Loop>:120 (Address = RCP)
- Write Only addresses (do not support reading commands):
 - <Loop>:109 (Address = RCP)
 - <Loop>:110 (Address = RCP)
 - <Loop>:121 (All addresses)

Header	Address	Valid Range	Comments
<Loop>:<Block>	<Register Number>	0 .. <N>	When configuring a Block lower than 119 (from 0 up to 118), the address field must be configured with the number of the register from the CD600 controller that you want access. The maximum register number <N> for each block depends on the configuration of the CD600 controller. Example (first register from the Block 27 of the Loop 5): Header = 5:27 Address = 0
<Loop>:109	RCP	-	When configuring RCP in the Address field of the block 109, the tag configured in the Tag Name column of the driver worksheet (String type) must hold the new name of the recipe stored in the CD600 controller. When the user triggers a writing command, Studio changes the recipe name in the CD600 controller according to this tag value.
<Loop>:110	RCP	-	When configuring RCP in the Address field of the block 110, the tag configured in the Tag Name column of the driver worksheet (String type) must hold the new name of the recipe stored in the CD600 controller. When the user triggers a writing command, Studio changes the recipe name in the CD600 controller according to this tag value.

Header	Address	Valid Range	Comments
<Loop>:119	LI<Number>	LI1 .. LI40	LI1 .. LI32 = Analog value (Word 0 .. 31). LI33 .. LI40 = Digital value (Word 32). Example (LI4 from the Block 119 of the Loop 5): Header = 5:119 Address = LI4
	LR<Number>	LR1 .. LR4	Digital value (Word 32). Example (LR2 from the Block 119 of the Loop 5): Header = 5:119 Address = LR2
	MV<Number>	MV1 .. MV4	Digital value (Word 32). Example (MV1 from the Block 119 of the Loop 5): Header = 5:119 Address = MV1
	AL<Number>	AL1 .. AL20	Analog value (Word 33 .. 52). Example (AL6 from the Block 119 of the Loop 5): Header = 5:119 Address = AL6
	ST<Number>	ST1 .. ST20	Analog value (Word 53 .. 54). Example (ST16 from the Block 119 of the Loop 5): Header = 5:119 Address = ST16
	TOT<Number>	TOT1 .. TOT8	Analog value (Word 55 .. 62). Example (TOT4 from the Block 119 of the Loop 5): Header = 5:119 Address = TOT4
	DIN<Number>	DIN1 .. DIN4	Digital value (Word 63). Example (DIN2 from the Block 119 of the Loop 5): Header = 5:119 Address = DIN2
	DOUT<Number>	DOUT1 .. DOUT8	Digital value (Word 63). Example (DOUT7 from the Block 119 of the Loop 5): Header = 5:119 Address = DOUT7
	RCP	-	When configuring RCP in the Address field of the block 120, the tag configured in the Tag Name column of the driver worksheet (String type) will receive the name of the recipe stored in the CD600 controller.

Header	Address	Valid Range	Comments
<Loop>:120	KP<Number>	KP1 .. KP8	Example (KP2 from the Block 120 of the Loop 5): Header = 5:120 Address = KP2
	TR<Number>	TR1 .. TR8	Example (TR5 from the Block 120 of the Loop 5): Header = 5:120 Address = TR5
	TD<Number>	TD1 .. TD8	Example (TD3 from the Block 120 of the Loop 5): Header = 5:120 Address = TD3
	BIAS<Number>	BIAS1 .. BIAS8	Example (BIAS8 from the Block 120 of the Loop 5): Header = 5:120 Address = BIAS8
	RCP	-	When configuring RCP in the Address field of the block 120, the tag configured in the Tag Name column of the driver worksheet (String type) will receive the name of the recipe stored in the CD600 controller.
<Loop>:121	CR<Number>	CR1 .. CR12	Example (CR4 from the Block 121 of the Loop 5): Header = 5:121 Address = CR4
	CMV<Number>	CMV1 .. CMV4	Example (CMV1 from the Block 121 of the Loop 5): Header = 5:121 Address = CMV1
	CLR<Number>	CLR1 .. CLR4	Example (CLR3 from the Block 121 of the Loop 5): Header = 5:121 Address = CLR3
	AL<Number>	AL1 .. AL20	Example (AL18 from the Block 121 of the Loop 5): Header = 5:121 Address = AL18
	AIN<Number>	AIN1 .. AIN20	Example (AIN9 from the Block 121 of the Loop 5): Header = 5:121 Address = AIN9
	DIN<Number>	DIN1 .. DIN20	Example (DIN12 from the Block 121 of the Loop 5): Header = 5:121 Address = DIN12
	XMV<Number>	XMV1 .. XMV4	Example (XMV4 from the Block 121 of the Loop 5): Header = 5:121 Address = XMV4
CONF	-	-	When configuring CONF in the Header field, the value configured in the Address column (if any) is ignored. The tag configured in the Tag Name column of the driver worksheet (String type) must hold the path and name of the configuration file which will be downloaded to the CD600 controller. The configuration file is generated with the CD600 programming tool.

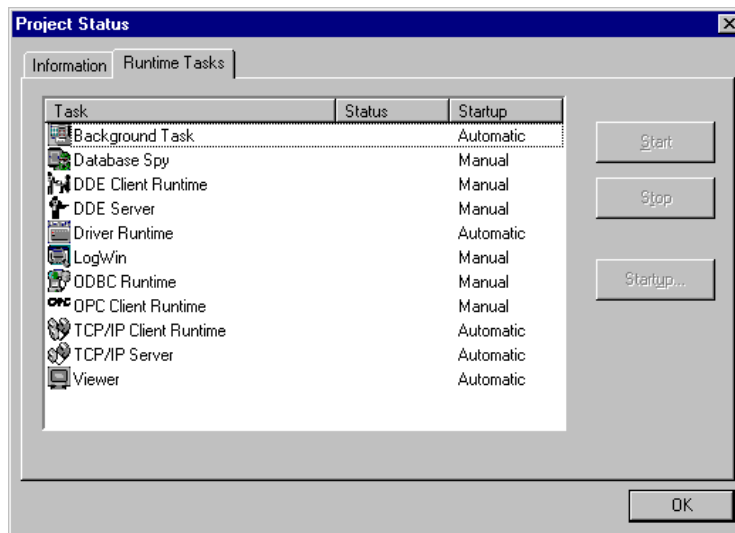
Executing the Driver

After adding the CD600 driver to a project, Studio sets the project to execute the driver automatically when you start the run-time environment.

To verify that the driver run-time task is enabled and will start correctly, perform the following steps:

1. Select **Project** → **Status** from the main menu bar.

The *Project Status* dialog box displays, as follows.



Project Status Dialog Box

2. Verify that the *Driver Runtime* task is set to **Automatic**.
 - If the setting is correct, click **OK** to close the dialog box.
 - If the **Driver Runtime** task is set to **Manual**, select the **Driver Runtime** line. When the **Startup** button becomes active, click the button to toggle the *Startup* mode to **Automatic**.
3. Click **OK** to close the *Project Status* dialog.
4. Start the application to run the driver.

Troubleshooting

If the CD600 driver fails to communicate with the device, the tag you configured for the **Read Status** or **Write Status** fields will receive an error code. Use this error code and the following table to identify what kind of failure occurred.

Error Code	Description	Possible Causes	Procedure to Solve
0	OK	Communication without problems	None required.
10	Invalid Header	Specified an invalid header in the Driver Worksheet.	Type a valid Header.
11	Invalid Address	Specified an invalid address in the Driver Worksheet.	Type a valid Address.
12	Invalid Block 119 Address	Specified an invalid Block 119 address in the Driver Worksheet.	Type a valid Block 119 address.
13	Invalid Block 120 Address	Specified an invalid Block 120 address in the Driver Worksheet.	Type a valid Block 120 address.
14	Invalid Block 121 Address	Specified an invalid Block 121 address in the Driver Worksheet.	Type a valid Block 121 address.
15	No block in Loop	Block not configured in specified Loop	Check the equipment configuration.
17	Try read Block 121	Could not read an "Write only" block	Block 121 is Write Only.
18	Try write Block 119	Could not write an "Read only" block	Block 119 is Read Only.
19	Invalid configuration file path	Invalid configuration file path for Recipe (RCP) commands	Check the path of configuration file.
-15	Timeout Start Message	<ul style="list-style-type: none"> ▪ Disconnected Cables ▪ PLC is turned off, in stop mode, or in error mode ▪ Wrong station number ▪ Wrong RTS/CTS control settings 	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.
-17	Timeout between rx char	<ul style="list-style-type: none"> ▪ PLC in stop mode or in error mode ▪ Wrong station number ▪ Wrong parity ▪ Wrong RTS/CTS configuration settings 	<ul style="list-style-type: none"> ▪ Check cable wiring ▪ Check the PLC state – it must be RUN. ▪ Check the station number ▪ Check the configuration. See <i>Studio Technical Reference Manual</i> for information about valid RTS/CTS configurations.

⇒ **Tip:**

You can verify communication status using the Studio development environment *Output* window (*LogWin* module). To establish an event log for **Field Read Commands**, **Field Write Commands**, and **Serial Communication** right-click in the *Output* window. When the pop-up menu displays, select the option to set the log events. If you are testing a Windows CE target, you can use the Remote LogWin of Studio (**Tools** → **Remote Logwin**) to get the log events from the target unit remotely.

If you are unable to establish communication with the PLC, try to establish communication between the PLC Programming Tool and the PLC. Quite frequently, communication is not possible because you have a hardware

or cable problem, or a PLC configuration error. After successfully establishing communication between the device's Programming Tool and the PLC, you can retest the supervisory driver.

To test communication with Studio, we recommend using the sample application provided rather than your new application.

If you must contact us for technical support, please have the following information available:

- **Operating System** (type and version): To find this information, select **Tools** → **System Information**.
- **Studio version**: To find this information, select **Help** → **About**.
- **Driver Version**: To find this information, read the full description of the driver on the **Communication Drivers** Dialog Box.
- **Communication Log**: Displays in the Studio *Output* window (or *LogWin* window) when the driver is running. Be sure to enable the **Field Read Commands**, **Field Write Commands**, and **Serial Communication** for the LogWin window.
- **Device Model** and **Boards**: Consult the hardware manufacturer's documentation for this information.

Sample Application

You will find a sample application for drivers in the `/COMMUNICATION EXAMPLES/<Driver Name>` directory. We strongly recommend that you check if there is a sample application for this driver and use it to test the driver before configuring your own customized application, for the following reasons:

- To better understand the information provided in the section of this document.
- To verify that your configuration is working satisfactorily.
- To certify that the hardware used in the test (device, adapter, cable, and PC) is working satisfactorily before you start configuring your own, customized applications.



Note:

This application sample is not available for all drivers.

Use the following procedure to perform the test:

1. Configure the device's communication parameters using the manufacturer's documentation.
2. Open and execute the sample application.



Tip:

You can use the sample application screen as the maintenance screen for your custom applications.

Revision History

Doc. Revision	Driver Version	Author	Date	Description of changes
-	1.01	Marcos Taccolini	May/20/1996	Added reset serial always on read
-	1.02	Roberto V. Junior	Jul/10/1997	Utilized UniComm Functions
-	1.03	Marcos Taccolini	Aug/14/1997	Fixed HandInts, TX Message
-	1.04	Roberto V. Junior	Jan/15/1998	Fixed GetAddress to LI33 and include Trace of RX message in Proto function.
-	1.05	Roberto V. Junior	Feb/27/1998	<ul style="list-style-type: none"> ▪ Fixed Command 27 ▪ Fixed Float Value of block 109 ▪ Include Send of Recipe Name
-	1.06	Roberto V. Junior	Mar/2/1998	Include Command to Send Config file to CD600
-	1.07	Roberto V. Junior	Mar/23/1998	Include Configuration of limits of LIs in Header
-	1.08	Roberto V. Junior	Mar/27/1998	Include Configuration of number of Alrs in Header.
-	1.09	Luis F. Rodas	Abr/7/1998	Changed time-slices.
-	1.10	Luis F. Rodas	May/13/1998	Fixed bug with ControlWindows function.
-	1.11	Luis F. Rodas	Nov/16/1998	Fixed bug with block 120(KPs, TRs, TDs) and block 119 (ST1..ST20, DIN, DOUT)
-	1.12	Roberto V. Junior	Nov/24/1998	Include Set of DTR.
-	1.14	Roberto V. Junior	Aug/17/1999	Fixed Bug in CRC value. (CRC == SOH)
-	1.17	Roberto V. Junior	Jun/26/2001	Fixed some bug like: Address of 119 block and performance in the timeout.
A	1.18	Fabio H. Y. Komura	Apr/02/2004	<ul style="list-style-type: none"> ▪ Fixed issue when writing specific float point values (e.g.: Value 2.54). ▪ Fixed conversion issue.
B	2.0	Paulo Balbino	Mar/6/2012	<ul style="list-style-type: none"> ▪ Released