

# **PLCmodule-C14**

## **Systems Manual**

**Edition Juli 2013**

In this manual are descriptions for copyrighted products which are not explicitly indicated as such. The absence of the trademark (©) symbol does not infer that a product is not protected. Additionally, registered patents and trademarks are similarly not expressly indicated in this manual

The information in this document has been carefully checked and is believed to be entirely reliable. However, SYS TEC electronic GmbH assumes no responsibility for any inaccuracies. SYS TEC electronic GmbH neither gives any guarantee nor accepts any liability whatsoever for consequential damages resulting from the use of this manual or its associated product. SYS TEC electronic GmbH reserves the right to alter the information contained herein without prior notification and accepts no responsibility for any damages which might result.

Additionally, SYS TEC electronic GmbH offers no guarantee nor accepts any liability for damages arising from the improper usage or improper installation of the hardware or software. SYS TEC electronic GmbH further reserves the right to alter the layout and/or design of the hardware without prior notification and accepts no liability for doing so.

© Copyright 2013 SYS TEC electronic GmbH. rights – including those of translation, reprint, broadcast, photomechanical or similar reproduction and storage or processing in computer systems, in whole or in part – are reserved. No reproduction may occur without the express written consent from SYS TEC electronic GmbH.

Inform yourselves:

Contact	Direct	Your local distributor
Address:	SYS TEC electronic GmbH Am Windrad 2 D-08468 Heinsdorfergrund GERMANY	Please find a list of our distributors under:  <a href="http://www.systemec-electronic.com/distributors">http://www.systemec-electronic.com/distributors</a>
Ordering Information:	+49 (0) 37 65 / 38 600-0 <a href="mailto:info@systemec-electronic.com">info@systemec-electronic.com</a>	
Technical Support:	+49 (0) 37 65 / 38 600-0 <a href="mailto:support@systemec-electronic.com">support@systemec-electronic.com</a>	
Fax:	+49 (0) 37 65 / 38 600 4100	
Web Site:	<a href="http://www.systemec-electronic.com">http://www.systemec-electronic.com</a>	

7<sup>th</sup> Edition Juli 2013

---

---

<b>1</b>	<b>Preface.....</b>	<b>1</b>
<b>2</b>	<b>Overview of the PLCmodule-C14.....</b>	<b>3</b>
2.1	Block Diagram .....	4
2.2	View of the PLCmodule-C14.....	4
<b>3</b>	<b>System Description.....</b>	<b>5</b>
3.1	Connectors on the PLCmodule-C14.....	5
3.1.1	Power Supply VCPU, VIO.....	5
3.1.2	Digital Inputs and Outputs .....	5
3.1.2.1	24V Digital Inputs DI0..23 .....	6
3.1.2.2	24VDC Counter Inputs C0..3.....	7
3.1.2.3	Relay Outputs REL0..3 .....	8
3.1.2.4	24VDC Digital Transistor Outputs DO0..15 ...	8
3.1.2.5	PWM Outputs P0..1 .....	9
3.1.3	Analog Inputs and Outputs.....	10
3.1.3.1	Analog Inputs AI0..3.....	10
3.1.3.2	Analog Outputs AO0 .. 1 .....	10
3.1.4	Serial Interfaces.....	11
3.1.4.1	RS-232 Interfaces ASC0 .. 3.....	11
3.1.4.2	CAN Interface CAN0 .. 1 .....	12
3.1.4.3	Ethernet Interface.....	12
3.2	Programming the PLCmodule-C14.....	13
3.3	The PLCmodule-C14 working as CANopen Master .....	13
3.4	Start-up Characteristics and Restart .....	14
3.5	Switching from RS-232 to CANopen and/or Ethernet-Version. ....	15
3.6	Firmware Expansions .....	15
<b>4</b>	<b>Addressing PLC Inputs and Outputs.....</b>	<b>17</b>
4.1	Fundamentals.....	17
4.2	Addressing Compact Control Units .....	17
4.3	Representation of Variables of the CAN1 interface in Marker space of Process Image.....	21
<b>5</b>	<b>Control Units on the PLCmodule-C14.....</b>	<b>34</b>
5.1	Run/Stop/MRES Switch.....	34
5.2	Run LED (green).....	35
5.3	Error LED (red).....	36
5.4	HEX-Encoding Switch and DIP Switch.....	38
5.4.1	HEX-Encoding Switches.....	38
5.4.2	DIP Switch .....	38
5.4.3	Communication Parameter Configuration.....	39
5.4.3.1	Parameters for RS-232 Communication .....	39
5.4.3.2	CANopen communication via CAN0 .....	40
5.4.3.3	CANopen communication via CAN1 .....	41

---

---

5.4.3.4	Ethernet communication .....	42
<b>6</b>	<b>Operating the PLCmodule-C14.....</b>	<b>43</b>
6.1	Connecting the CAN Interfaces .....	43
6.2	Initial Startup of the PLC .....	43
6.3	Resetting the Control Unit .....	46
<b>7</b>	<b>Using the PLC Simulator .....</b>	<b>50</b>
7.1	Starting the Simulator .....	50
7.2	Using the Simulator.....	51
7.3	Simulating Compact Control Units .....	53
<b>8</b>	<b>Initial Configuration using the Configuration Command Shell... 55</b>	
8.1	Command description .....	56
8.1.1	GET_DEV_INFO .....	56
8.1.2	GET_DEV_CONFIG.....	56
8.1.3	SET_IP_CONFIG .....	57
8.1.4	SET_CAN_CONFIG .....	58
8.1.5	DEL_CAN_CONFIG.....	58
8.1.6	GET_RTC .....	59
8.1.7	SET_RTC.....	59
8.1.8	DEL_PLC_PROG.....	60
8.1.9	DEL_NVDATA .....	60
8.1.10	GET_LAST_ERROR.....	60
8.1.11	SET_PWF_MODE.....	61
8.1.12	SET_WDG_MODE .....	62
8.1.13	EXIT.....	62
8.1.14	HELP.....	62
<b>9</b>	<b>Device pinout.....</b>	<b>63</b>
<b>10</b>	<b>Technical Specifications .....</b>	<b>69</b>
<b>Index</b>	<b>.....</b>	<b>73</b>

---

Figure 1: Block Diagram of the PLCmodule-C14.....	4
Figure 2: Top-view of the PLCmodule-C14 (top side).....	4
Figure 3: Schematic of 24VDC digital inputs DI0..23 .....	6
Figure 4: Schematic of 24V counter inputs C0..3.....	7
Figure 5: Schematic of relay outputs REL0..3.....	8
Figure 6: Schematic of 24VDC digital outputs DO0..15.....	9
Figure 7: Schematic of 24VDC PWM output P0..1.....	10
Figure 8: Schematic of analog input 0..10V AI0..3 .....	10
Figure 9: Schematic of analog output 0..10V AO0..1 .....	11
Figure 10: Start-up Characteristics of the PLCmodule-C14.....	14
Figure 11: HEX-Encoding Switch and DIP Switch.....	38
Figure 12: PLC Version info dialog window.....	46
Figure 13: PLC Simulator .....	50
Figure 14: Built-in Configuration Command Shell .....	55
Figure 15: Device pinout overview.....	63

Table 1: Alternative functions defined for inputs DI21..23 .....	7
Table 2: IO Address table.....	18
Table 3: Data format of IOs in the Process Image .....	20
Table 4: Overlapping of the CANopen Address space (Subindex) in Marker space for (Input- data) .....	21
Table 5: Overlapping of the CANopen Address space (Subindex) in Marker space for (Output - data) .....	22
Table 6: Assignment of CANopen Objekt Dictionary to input data in Marker space forCANopen RPDO .....	25
Table 7: Assignemt of CANopen Objekt Dictionary to Output data in Marker space, for CANopen TPDO.....	27
Table 8: Default communication and mapping parameter, CAN1, RPDO29	
Table 9: Default communication and mapping parameter, CAN1, TPDO31	
Table 10: Functions and Positions of the RUN/STOP Switch.....	35
Table 11: Run LED Modes .....	36
Table 12: Error LED Modes.....	37
Table 13: Configuring the RS-232 Baudrate.....	40
Table 14: Configuring the CANopen Bitrate .....	41
Table 15: Simulator Addresses for PLC modules <sup>(1)</sup> see description) .....	53
Table 16: Environmental Parameters .....	69
Table 17: Communication Interfaces .....	69
Table 18: IO configuration.....	71
Table 19: PLC specific parameters .....	71

## 1 Preface

This manual describes only the functions of the PLCmodule-C14. The microcontroller is not described herein. Additional controller- and board-level information and technical descriptions can be found in appropriate microcontroller Data Sheets/User's Manuals. Refer to the "*COMBI*module-C14" Hardware Manual for more information on the board's functions. Additional descriptions about the *IEC61131-3 Programming System* for PLCs can be found in the "*OpenPCS*" Manuals as well as the "*CANopen Extension for IEC61131*" Software Manual.

In this hardware manual and in the attached schematics, low active signals are denoted by a "/" in front of the signal name (i.e.: /RD). A "0" indicates a logic-zero or low-level signal, while a "1" represents a logic-one or a high-level signal.

### Declaration of the Electro Magnetic Conformity for the PLCmodule-C14



The PLCmodule-C14 (henceforth product) is designed for installation in electrical appliances or as dedicated Evaluation Boards (i.e.: for use as a test and prototype platform for hardware/software development) in laboratory environments.

**Note:**

It is necessary that only appropriately trained personnel (such as electricians, technicians and engineers) handle and/or operate these products.

SYS TEC products fulfill the norms of the European Union's Directive for Electro Magnetic Conformity only in accordance to the descriptions and rules of usage indicated in this hardware manual (particularly in respect to the connectors, power connector and serial interface).

Implementation of SYS TEC products into target applications, as well as user modifications and extensions of SYS TEC products, is subject to renewed establishment of conformity to, and certification of, Electro Magnetic Directives. Users should ensure conformance following any modifications to the products as well as implementation of the products into target systems.

The PLCmodule-C14 is one of a series of SYS TEC automation control products that can be fitted with different controllers and, hence, offers various functions and configurations. The available programming system according to the IEC61131-3 standard and use of the CAN bus – a standard network interface for various automation control solutions – together with the CANopen network protocol significantly reduce the development time while using off-the-shelf hardware solutions. The implemented on-board PLC functions with optional CANopen network layer makes complicated microcontroller firmware design unnecessary.



## **2 Overview of the PLCmodule-C14**

The PLCmodule-C14 is a compact PLC for universal processing purposes of standard industrial signals. The module is designed for a great variety of uses, such as main control unit in measurement, control and data processing applications. The PLCmodule-C14 is also an ideal solution for distributed field bus systems in conjunction with other components and systems of the SYS TEC automation line.

The PLCmodule-C14 design is primarily based on the proven SYS TEC microcontroller core boards and makes use of the Infineon XC161 microcontroller resources. Some of the microcontroller's I/O ports are interrupt capable, thus enabling very short response times.

Peripheral sensors, actuators and control devices can be easily connected to the board using lug connector strips. Use of stable, removable screw thimbles enables easy exchange of the connected sensors and actuators. The board is installed in an industry proven casing and can be mounted on a DIN/EN chassis bar.

## 2.1 Block Diagram

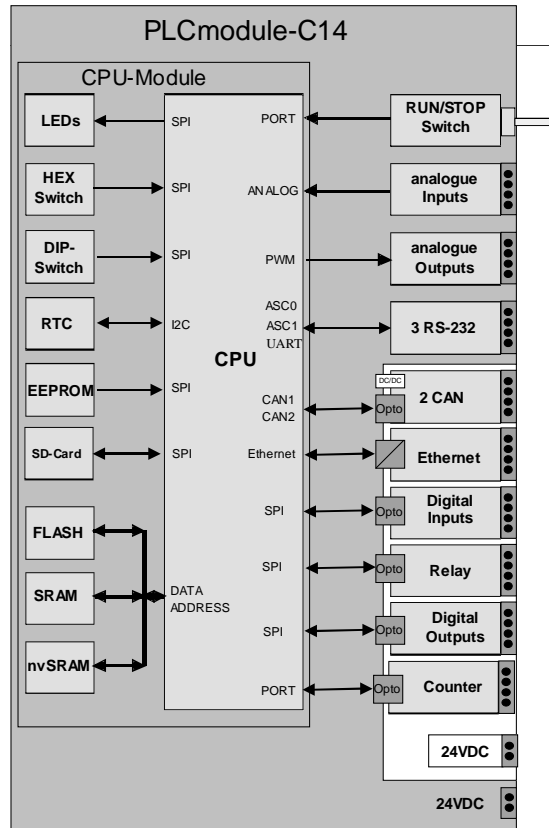


Figure 1: Block Diagram of the PLCmodule-C14

## 2.2 View of the PLCmodule-C14



Figure 2: Top-view of the PLCmodule-C14 (top side)

## 3 System Description

### 3.1 Connectors on the PLCmodule-C14

The connectors available on the PLCmodule-C14 are described in this section.

#### 3.1.1 Power Supply VCPU, VIO

The power supply input **VCPU** supplies the microcontroller, the core circuitry, the analog in/outputs and the relay on the PLCmodul-C14. A DC voltage power source in the range of 24V  $\pm$ 20% can be connected to this input. The typical current draw is  $I_{VCPU}$ .

The power supply input **VIO** supplies the transistor outputs (**DO0..15**) and the PWM outputs (**P0..1**). A DC voltage power source in the range of 24V  $\pm$ 20%, can be connected to this input. The current draw depends on the power consumption of the devices connected to the transistor outputs.

Power supply inputs are protected against polarity reversal in each case by a diode in the positive voltage input. Do not connect power supply connections to analog inputs or outputs as this could destroy the analog circuitry.

#### 3.1.2 Digital Inputs and Outputs

The following sections describe the hardware interface to access the peripheral IOs. The structure and the usage of the corresponding process image are explained in more detail in *section 4*.

### 3.1.2.1 24V Digital Inputs DI0..23

The PLCmodule-C14 provides a maximum of 24 digital inputs. The inputs are spitted in groups of four inputs each (Group 1: DI0...DI3, Group 2: DI4...DI7, ect.).

The groups are separated from one another and from the microcontroller core via an opto-coupler device. All inputs within a group use the same ground signal.

Signal voltage definition:

- Input voltage > 13 VDC: represents '1' in the input process image
- Input voltage < 5 VDC: represents '0' in the input process image

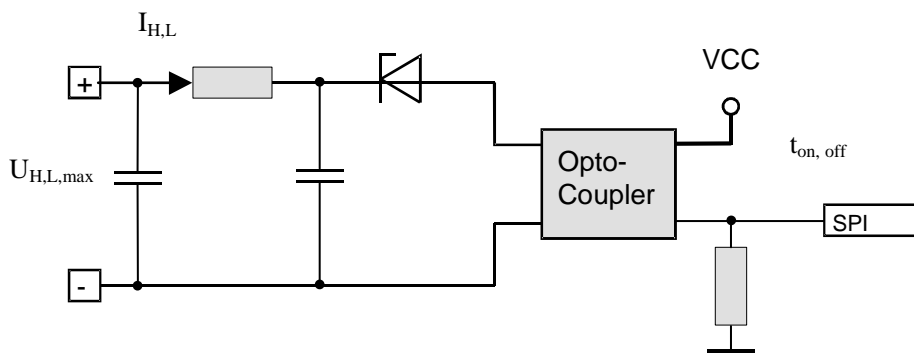


Figure 3: Schematic of 24VDC digital inputs DI0..23

Alternative Functions:

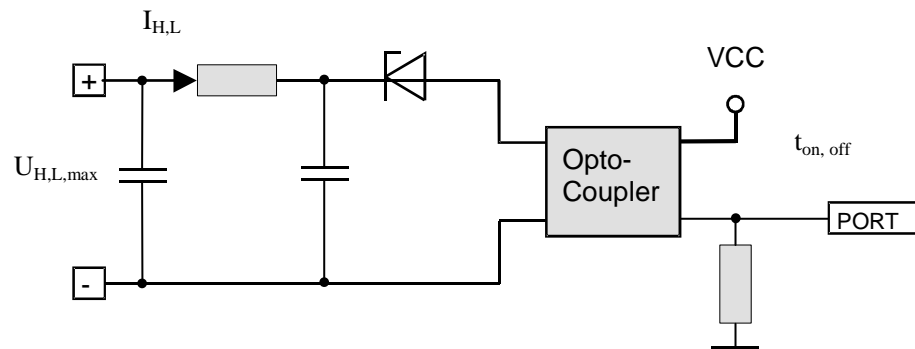
There are alternative functions available for digital inputs DI21 till DI23. These functions are selectable via software.

Input	Can also be used as:
DI21	- Counter input for counter C3, - Interrupt input
DI22	- Up/Down-control input for counter C2 or - Incremental encoder
DI23	- Up/Down control input for counter C0 - Incremental encoder

Table 1: Alternative functions defined for inputs DI21..23

### 3.1.2.2 24VDC Counter Inputs C0..3

The PLCmodule-C14 comes with 3 fast counter inputs. These are directly connected to the microprocessor portpins. The internal structure is shown in *Figure 4*.



*Figure 4: Schematic of 24V counter inputs C0..3*

The counter inputs are electrical isolated from another and the microcontroller core via an opto-coupler device. All inputs are high-active.

Trigger threshold:

- Input voltage > 13 VDC: '1'
- Input voltage < 5 VDC: '0'

Voltage range, input current and delay time can be found in *section 10*.

Also the counter inputs C0 and C2 have UP/DOWN control inputs to select the count direction. (*see Table 1*)

Alternative Functions:

The counter inputs can alternatively be used as digital or interrupt input. (*see above*).

### 3.1.2.3 Relay Outputs REL0..3

The PLCmodule-C14 provides 4 relay outputs. The maximum current on the relay outputs is 3A at 250VAC.

Contact arrangement:

Relay 0 .. 2: 1 closing contact  
 Relay 3: 1 closing / 1 opening contact

- Value '1' in the output process image represents:  
 Relay contact REL<sub>x</sub>\_3 closed  
 Relay contact REL3\_1 opened
- Value '0' in the output process image represents  
 Relay contact REL<sub>x</sub>\_3 opened  
 Relay contact REL3\_1 closed

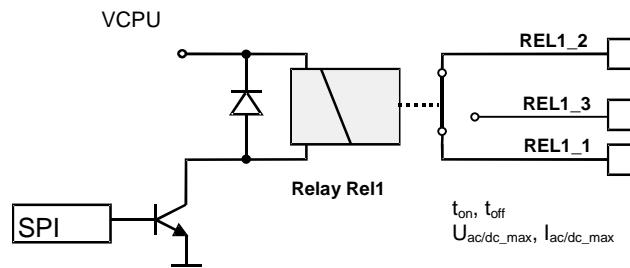


Figure 5: Schematic of relay outputs REL0..3

Mechanical, electrical and timing characteristics can be found in section 10.

### 3.1.2.4 24VDC Digital Transistor Outputs DO0..15

The PLCmodule-C14 provides 16 outputs for 24VDC digital signals. These outputs are active high and protected against short circuit. The maximum current on the 24VDC outputs is 0.5A at ohmic, inductive or capacitive load.

All outputs are galvanic isolated from the microcontroller core. Electrical power is provided through connector VIO.

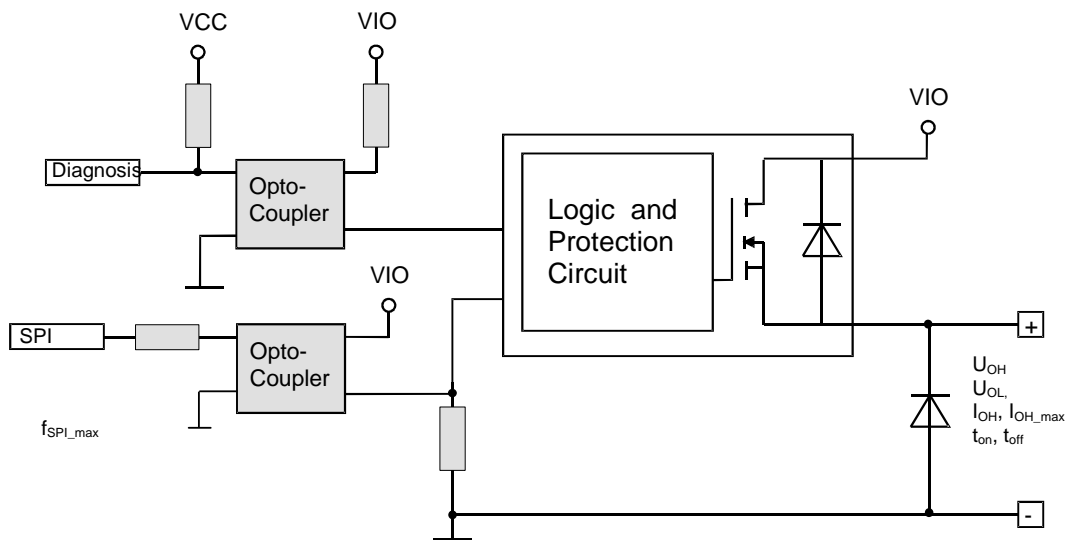


Figure 6: Schematic of 24VDC digital outputs DO0..15

The line drivers are protected against overvoltage, inverse-polarity, and temperature excursion.

Any overload or failure is reported to the CPU by a special Diagnosis Signal (see Figure 6).

### 3.1.2.5 PWM Outputs P0..1

The PLC PLCmodule-C14 provides two low active PWM outputs. These outputs switch the Ground potential when activated, therefore the load must be connected to the 24VDC supply. The maximum current load on the PWM outputs is 0.5 A for each channel.

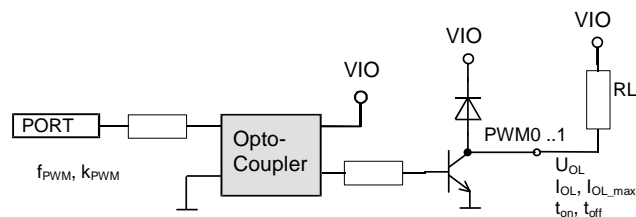


Figure 7: Schematic of 24VDC PWM output P0..1

All outputs are galvanic isolated from the microcontroller core protected against polarity reversal. The PWM outputs are connected to the ground signal of VIO.

### 3.1.3 Analog Inputs and Outputs

The PLCmodule-C14 provides 4 analog inputs and 2 analog outputs.

#### 3.1.3.1 Analog Inputs AI0..3

The standard configuration of the PLCmodule-C14 provides 4 analog inputs offering an input voltage range of 0..10V and a resolution of 10-bit.

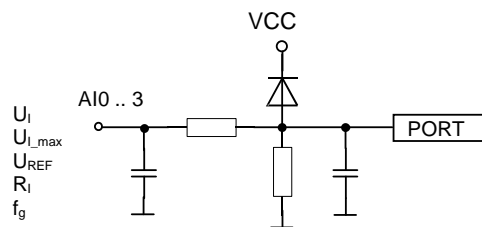


Figure 8: Schematic of analog input 0..10V AI0..3

#### 3.1.3.2 Analog Outputs AO0 .. 1

Two analog output signals can be generated on the PLCmodule-C14. The microcontroller provides a Compare/Capture unit that allows generation of pulse width modulated (PWM) signals. The on-board circuitry, with an active low-pass and operational amplifier, supports analog output signals in the range of 0..10V with a resolution of 10-bit and an accuracy of  $\pm 1\%$ .



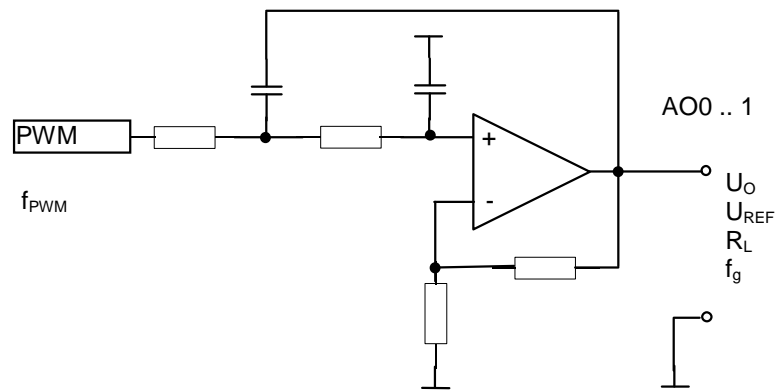


Figure 9: Schematic of analog output 0..10V AO0..1

### 3.1.4 Serial Interfaces

#### 3.1.4.1 RS-232 Interfaces ASC0 .. 3

The PLCmodule-C14 comes with 3 serial RS-232 interfaces (ASC0 .. 2).

**ASC0:** This interface is used for communication between PC and PLC (Special PLC firmware version for RS-232 access is required), firmware updates and PLC program download and debugging. The RS-232 signals RxD, TxD and GND are available at the RJ11 connector. You need to use the adapter cable supplied with the PLCmodule for connecting the device to the host-PC. A shielded serial extension cable (1:1) can be used to extend the cable.

**ASC1:** This interface is used for communication between PLC and another users device. The RS-232 level signals RxD, TxD and GND are available at the RJ11 connector. A shielded serial cable should be used for connecting the board to the device. The shield needs to be connected to PE on one side of the cable in order to ensure proper functioning.

**ASC2:** This full modem capable RS-232 interface is used for communication between PLC and an users device. The RS-232 level signals RxD, TxD, DTR, DCD, RTS, CTS, DSR and GND are available at the RJ45 connector. A shielded serial cable should be

used for connecting the board to a host-PC. The shield needs to be connected to PE on one side of the cable in order to ensure proper functioning.

#### **3.1.4.2 CAN Interface CAN0 .. 1**

The microcontroller is equipped with two integrated CAN interfaces called **CAN0** and **CAN1**. The signals CAN0\_HIGH, CAN0\_LOW, CAN1\_HIGH, CAN1\_LOW and CAN\_GND are galvanic decoupled and accessible on a removable connector.

##### **CAN Bus Cable**

It is recommended to use a twisted pair CAN bus cable, terminated with a resistor of 120 Ohm between CAN\_H and CAN\_L at both ends. According to CiA recommendation DRP 303-1 CAN ground should be included and connected. Please refer to the corresponding CiA standards for further information.

#### **3.1.4.3 Ethernet Interface**

The PLCmodule-C14 provides a 10Base-T Ethernet interface. The interface is accessible on a RJ-45 connector. Any appropriate standard CAT-5 Ethernet cable can be used.

### 3.2 Programming the PLCmodule-C14

The PLCmodule-C14 is programmed with the integrated IEC61131-3 programming system *OpenPCS*. For instructions on how to use *OpenPCS*, consult the manuals included with the software package. Depending on the installed module firmware, communication between the programming PC and the PLC device is established over RS-232, CANopen or an Ethernet connection.

The firmware integrated of the PLCmodule-C14 is based on the standard firmware for SYS TEC PLC systems, and therefore behaves identically to other control systems from SYS TEC with regard to the appearance of the process image (*see section 4.2*) as well as the function of the user interface.

Multiple function blocks for accessing the CANopen network are available to the PLC programmer when using the PLCmodule-C14 with the CANopen version of the firmware. These CANopen extensions are described in the manual "*CANopen Extensions for IEC61131-3*" (L-1008).

### 3.3 The PLCmodule-C14 working as CANopen Master

On PLCmodule-C14 supporting CANopen communication, the CANopen Master functionality for **CAN0** of the PLC unit can be enabled by setting DIP-Switch 4. For **CAN1** the master functionality can be set via Configuration Command Shell (*see section 0*).

**Note:**

Detailed information about the PLC's CANopen functions can be found in the manual "*CANopen Extension for IEC61131-3*" (L-1008).

### 3.4 Start-up Characteristics and Restart

On power-on and reset the PLC firmware first tests if a valid control program is located in the non-volatile memory upon connection of the operating voltage, or after a reset. If an executable program was found, the firmware starts executing the program. During this procedure, all variables are reset to their initial values. This is the same value as these variables have immediately after program download.

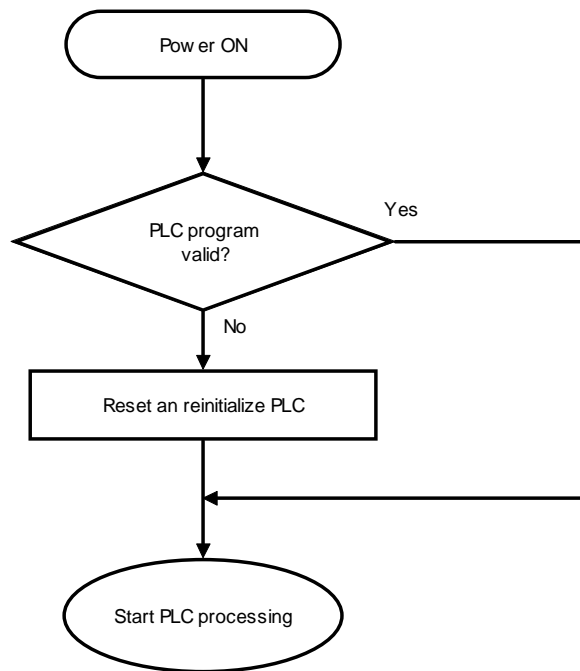


Figure 10: Start-up Characteristics of the PLCmodule-C14

If the firmware has found a valid control program in the non-volatile memory, this program is restarted and processed as long as the Run/Stop switch is in the "Run" position. Otherwise the reactivated program is ready for execution and the PLCmodule-C14 is in Stop mode.

### **3.5 Switching from RS-232 to CANopen and/or Ethernet-Version**

The PLCmodule-C14 is available in three different firmware variants supporting communication via either RS-232, Ethernet or CANopen. Exchanging the communication protocol requires a firmware update with the applicable PLC firmware as well as the right hardware configuration (i.e. Ethernet and CAN controllers) on the module.

### **3.6 Firmware Expansions**

As an additional service, SYS TEC offers expansion or custom-specific adaptation of the PLC firmware. As an alternative, the user can also perform additional firmware extensions and adaptations. The latter option requires purchase of the firmware in library format. The interfaces for the process image, as well as to the firmware's function blocks, are provided in source. The complete firmware package is also available in source code.

The firmware, located in the Flash memory, can be exchanged over the RS-232 interface only.



## 4 Addressing PLC Inputs and Outputs

### 4.1 Fundamentals

Addressing the PLC inputs and outputs with a PLC program occurs by defining the following information:

- **Data Direction:**
  - %I = Input
  - %Q = Output
  - %M = Memory
- **Access Width:**
  - X = Bit
  - B = Byte (8 bit)
  - W = Word (16 bit)
  - D = Dword (32 bit)
- **Address:** "cascaded" address specification, for example: "*Module.Channel Number*"

It is in the user`s responsibility to define the number of positions used for addressing. The IEC61131-3 standard recommends a two step addressing scheme in various examples.

### 4.2 Addressing Compact Control Units

The manufacturer of compact PLC units (for example the PLCmodule-C14) defines the addresses that are used to access the inputs and outputs on the device in accordance with the IEC61131-3 standard for PLC programming. *Table 2* shows the addressing scheme for all IO of the PLCmodule-C14.

To guarantee compatibility, all compact control units from SYS TEC provide a unique process image structure.

<b>Input</b>	<b>Address</b>	<b>Output</b>	<b>Address</b>
DI0	%IX0.0	DO0	%QX0.0
DI1	%IX0.1	DO1	%QX0.1
DI2	%IX0.2	DO2	%QX0.2
DI3	%IX0.3	DO3	%QX0.3
DI4	%IX0.4	DO4	%QX0.4
DI5	%IX0.5	DO5	%QX0.5
DI6	%IX0.6	DO6	%QX0.6
DI7	%IX0.7	DO7	%QX0.7
DI8	%IX1.0	DO8	%QX1.0
DI9	%IX1.1	DO9	%QX1.1
DI10	%IX1.2	DO10	%QX1.2
DI11	%IX1.3	DO11	%QX1.3
DI12	%IX1.4	DO12	%QX1.4
DI13	%IX1.5	DO13	%QX1.5
DI14	%IX1.6	DO14	%QX1.6
DI15	%IX1.7	DO15	%QX1.7
DI16	%IX2.0		
DI17	%IX2.1	REL0	%QX2.0
DI18	%IX2.2	REL1	%QX2.1
DI19	%IX2.3	REL2	%QX2.2
DI20	%IX2.4	REL3	%QX2.3
DI21	%IX2.5		
DI22	%IX2.6	AO0	%QW8.0
DI23	%IX2.7	AO1	%QW10.0
DI24	%IX3.0 (C0)		
DI25	%IX3.1 (C1)		
DI26	%IX3.2 (C2)		
AI0	%IW8.0		
AI1	%IW10.0		
AI2	%IW12.0		
AI3	%IW14.0		
C0	%ID40.0 (DI24)		
C1	%ID44.0 (DI25)		
C2	%ID48.0 (DI26)		
C3	%ID52.0 (DI21)		

Table 2: IO Address table



**Note:**

The I/Os specified in the addressing table (Table 2) describe the real inputs and outputs available on the PLCmodule-C14. PLC programs that only make use of the available peripherals hence use the standard process image, can be used on any SYS TEC PLC hardware providing the corresponding I/O functions. Additional compilation of the program is thereby not necessary.

The user should note that not all PLC-modules provide the same number of inputs and outputs even if the process image is identical. The value zero will be returned when reading an input value for an unavailable input channel. Writing to an unavailable output does not result in any reaction.

The process image addresses for accessing inputs and outputs on various PLC-modules are listed in Table 2. The first address entry always specifies a byte address; the second entry references the applicable bit. For example, the address %QX1.3 entry specifies access to the fourth bit in the second byte (counting starts with "0" for the first byte) of the output address space, or in other words the output bit number 12 of the compact control unit.

The following addressing variants are possible when accessing **digital inputs and outputs** (bit addressable inputs and outputs):

**Bit access:** Each digital input and output can be accessed individually by bit access. The addresses shown in *Table 2* must be used for bit access.

**Example:**

Input DI0 = %IX0.0      Output DO0 = %QX0.0  
Input DI13 = %IX1.5      Output DO14 = %QX1.6

**Byte/Word access:** A group of digital inputs and outputs can be accessed together by byte or word access. The digital inputs and outputs are located at the beginning of the inputs and outputs section in the process image. Bit positions 0..7 are located in Byte 0, Byte 1 contains Bits 8..15 etc. The bit address for byte and word access is always zero.

**Example:**

Inputs DI0...7           = %IB0.0  
 Output DO8...15       = %QB1.0  
 Input DI0...23         = %IW0.0

The input addresses %ID20.0, %ID24.0, %ID28.0, %ID32.0 (for C0-3) must be used regardless of which actual input matches the counter channel.

**Data Format**

IOs	Access Width	Example <sup>1</sup>
Digital input	1 bit	2#0, 2#1
Digital output	1 bit	
Analog input	15 bit + 1bit sign	16#7FFF → +32767 LSB 16#0000 → 0
Analog output	15 bit + 1bit sign	16#FFFF → -1 LSB 16#8000 → -32768 LSB <sup>2</sup>
Counter input	32 bit	16#00000000 – 16#FFFFFFFF
PWM output	16 bit	16#0000 – 16#FFFF

*Table 3: Data format of IOs in the Process Image*

---

<sup>1</sup> Definition of base (16# = Hexadecimal, 2# = Binary)

<sup>2</sup> The value of the LSB can be found in the Technical Specifications.

### 4.3 Representation of Variables of the CAN1 interface in Marker space of Process Image

Via CAN1 interface the PLCmodule-C14 corresponds to an CANopen device with static Object Dictionary. The number of variables is fixed. There are 252 bytes for input and 252 bytes for output available in Marker space of the PLC Process Image. (see Table 6 and Table 7). Keep in mind, that the CANopen objects for variables of type BYTE, WORD and DWORD represent the same elements in Marker space. That means the CANopen objects are overlapping. A WORD overlaps 2 BYTE variables ect. (see Table 4 and Table 5).

Marker %MB	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0	11.0
Index 16#2000 (BYTE) Subindex	1	2	3	4	5	6	7	8	9	10	11	12
Index 16#2001 (SHORT) Subindex	1	2	3	4	5	6	7	8	9	10	11	12
Index 16#2010 (WORD) Subindex	1		2		3		4		5		6	
Index 16#2011 (INT) Subindex	1		2		3		4		5		6	
Index 16#2020 (DWORD) Subindex	1			2			3					
Index 16#2021 (LINT) Subindex	1			2			3					

Table 4: Overlapping of the CANopen Address space (Subindex) in Marker space for (Input- data)

Marker %MB	256.0	257.0	258.0	259.0	260.0	261.0	262.0	263.0	264.0	265.0	266.0	267.0
Index 16#2030 (BYTE) Subindex	1	2	3	4	5	6	7	8	9	10	11	12

Index 16#2031 (SHORT) Subindex	1	2	3	4	5	6	7	8	9	10	11	12
Index 16#2040 (WORD) Subindex	1		2		3		4		5		6	
Index 16#2041 (INT) Subindex	1		2		3		4		5		6	
Index 16#2050 (DWORD) Subindex	1				2				3			
Index 16#2051 (LINT) Subindex	1				2				3			

Table 5: Overlapping of the CANopen Address space (Subindex) in Marker space for (Output - data)

Communication with other CANopen devices is possible via SDO<sup>1</sup> and PDO<sup>2</sup>. Therefore 16 Receive-PDO (RPDO) and 16- Transmit-PDO (TPDO) are available. The PDO mapping can be done via Function Blocks from within the PLC program. The first set of 4 RPDO and 4 TPDO are preconfigured according to the CANopen „Predefined Connection Set“ as shown in *Table 8* and *Table 9*.

Thus, without doing any configuration 4 RPDO and 4 TPDO are represented in the Marker space of the PLC process image.

Not all variables in Marker space can be mapped to PDO because of limited PDO storage capacity (16 PDO \* 8 Bytes = 128Bytes). However, it is possible to access all variables in OD via SDO.

---

<sup>1</sup> SDO service data object

<sup>2</sup> PDO process data object

<b>Index</b>	<b>Sub-index</b>	<b>Marker-Address</b>	<b>Description in EDS</b>	<b>CANopen Data type</b>
16#2000				
	1	%MB0.0	CAN1InByte0	USIGNED8
	2	%MB1.0	CAN1InByte1	USIGNED8
	3	%MB2.0	CAN1InByte2	USIGNED8
	4	%MB3.0	CAN1InByte3	USIGNED8
	5	%MB4.0	CAN1InByte4	USIGNED8
	6	%MB5.0	CAN1InByte5	USIGNED8
	...	...		
	240	%MB239.0	CAN1InByte239	USIGNED8
16#2001				
	1	%MB0.0	CAN1InSInt0	INTEGER8
	2	%MB1.0	CAN1InSInt1	INTEGER8
	3	%MB2.0	CAN1InSInt2	INTEGER8
	4	%MB3.0	CAN1InSInt3	INTEGER8
	5	%MB4.0	CAN1InSInt4	INTEGER8
	6	%MB5.0	CAN1InSInt5	INTEGER8
	...	...		
	240	%MB239.0	CAN1InSInt239	INTEGER8
16#2010				
	1	%MW0.0	CAN1InWord0	USIGNED16
	2	%MW2.0	CAN1InWord1	USIGNED16
	3	%MW4.0	CAN1InWord2	USIGNED16
	4	%MW6.0	CAN1InWord3	USIGNED16
	5	%MW8.0	CAN1InWord4	USIGNED16
	6	%MW10.0	CAN1InWord5	USIGNED16
	...	...		
	120	%MW238.0	CAN1InWord120	USIGNED16
16#2011				
	1	%MW0.0	CAN1InInt0	INTEGER16
	2	%MW2.0	CAN1InInt1	INTEGER16
	3	%MW4.0	CAN1InInt2	INTEGER16
	4	%MW6.0	CAN1InInt3	INTEGER16
	5	%MW8.0	CAN1InInt4	INTEGER16
	6	%MW10.0	CAN1InInt5	INTEGER16
	...	...		
	120	%MW238.0	CAN1InInt120	INTEGER16
16#2020				
	1	%MD0.0	CAN1InDword0	USIGNED32
	2	%MD4.0	CAN1InDword1	USIGNED32
	3	%MD8.0	CAN1InDword2	USIGNED32

	4	%MD12.0	CAN1InDword3	UNSIGNED32
	5	%MD16.0	CAN1InDword4	UNSIGNED32
	6	%MD20.0	CAN1InDword5	UNSIGNED32
	...	...		
	60	%MD236.0	CAN1InDword60	UNSIGNED32
16#2021				
	1	%MD0.0	CAN1InDInt0	INTEGER32
	2	%MD4.0	CAN1InDInt1	INTEGER32
	3	%MD8.0	CAN1InDInt2	INTEGER32
	4	%MD12.0	CAN1InDInt3	INTEGER32
	5	%MD16.0	CAN1InDInt4	INTEGER32
	6	%MD20.0	CAN1InDInt5	INTEGER32
	...	...		
	60	%MD236.0	CAN1InDInt60	INTEGER32

*Table 6: Assignment of CANopen Objekt Dictionary to input data in Marker space for CANopen RPDO*

<b>Index</b>	<b>Sub-index</b>	<b>Marker-Address</b>	<b>Description in EDS</b>	<b>CANopen Data type</b>
16#2030				
	1	%MB256.0	CAN1OutByte0	USIGNED8
	2	%MB257.0	CAN1OutByte1	USIGNED8
	3	%MB258.0	CAN1OutByte2	USIGNED8
	4	%MB259.0	CAN1OutByte3	USIGNED8
	5	%MB260.0	CAN1OutByte4	USIGNED8
	6	%MB261.0	CAN1OutByte5	USIGNED8
	...	...		
	240	%MB495.0	CAN1OutByte240	USIGNED8
16#2031				
	1	%MB256.0	CAN1OutSInt0	INTEGER8
	2	%MB257.0	CAN1OutSInt1	INTEGER8
	3	%MB258.0	CAN1OutSInt2	INTEGER8
	4	%MB259.0	CAN1OutSInt3	INTEGER8
	5	%MB260.0	CAN1OutSInt4	INTEGER8
	6	%MB261.0	CAN1OutSInt5	INTEGER8
	...	...		
	240	%MB495.0	CAN1OutSInt240	INTEGER8
16#2040				
	1	%MW256.0	CAN1OutWord0	USIGNED16
	2	%MW258.0	CAN1OutWord1	USIGNED16
	3	%MW260.0	CAN1OutWord2	USIGNED16
	4	%MW262.0	CAN1OutWord3	USIGNED16
	5	%MW264.0	CAN1OutWord4	USIGNED16
	6	%MW266.0	CAN1OutWord5	USIGNED16
	...	...		
	120	%MW494.0	CAN1OutWord120	USIGNED16
16#2041				
	1	%MW256.0	CAN1OutInt0	INTEGER16
	2	%MW258.0	CAN1OutInt1	INTEGER16
	3	%MW260.0	CAN1OutInt2	INTEGER16
	4	%MW262.0	CAN1OutInt3	INTEGER16
	5	%MW264.0	CAN1OutInt4	INTEGER16
	6	%MW266.0	CAN1OutInt5	INTEGER16
	...	...		
	120	%MW494.0	CAN1OutInt120	INTEGER16
16#2050				
	1	%MD256.0	CAN1OutDword0	USIGNED32
	2	%MD260.0	CAN1OutDword1	USIGNED32

	3	%MD264.0	CAN1OutDword2	UNSIGNED32
	4	%MD268.0	CAN1OutDword3	UNSIGNED32
	5	%MD272.0	CAN1OutDword4	UNSIGNED32
	6	%MD276.0	CAN1OutDword6	UNSIGNED32
	...	...		
	60	%MD492.0	CAN1OutWord60	UNSIGNED16
16#2051				
	1	%MD256.0	CAN1OutDInt0	INTEGER32
	2	%MD260.0	CAN1OutDInt1	INTEGER32
	3	%MD264.0	CAN1OutDInt2	INTEGER32
	4	%MD268.0	CAN1OutDInt3	INTEGER32
	5	%MD272.0	CAN1OutDInt4	INTEGER32
	6	%MD276.0	CAN1OutDInt6	INTEGER32
	...	...		
	60	%MD492.0	CAN1OutDInt60	INTEGER32

*Table 7: Assignemt of CANopen Objekt Dictionary to Output data in Marker space, for CANopen TPDO*



Index	Sub-index	Value	Description
RPDO1		communication	
16#1400	1	16#200+NodeId	CAN-Identifier
	2	255	transmission type
RPDO2		communication	
16#1401	1	16#300+NodeId	CAN-Identifier
	2	255	transmission type
RPDO3		communication	
16#1402	1	16#400+NodeId	CAN-Identifier
	2	255	transmission type
RPDO4		communication	
16#1403	1	16#500+NodeId	CAN-Identifier
	2	255	transmission type
RPDO1		mapping	
16#1600			
	0	8	number of entries
	1	16#20000108	
	2	16#20000208	
	3	16#20000308	
	4	16#20000408	
	5	16#20000508	
	6	16#20000608	
	7	16#20000708	
	8	16#20000808	
RPDO2		mapping	
16#1601			
	0	8	number of entries
	1	16#20000908	
	2	16#20000A08	
	3	16#20000B08	
	4	16#20000C08	
	5	16#20000D08	
	6	16#20000E08	
	7	16#20000F08	
	8	16#20001008	
RPDO3		mapping	
16#1602			
	0	8	number of entries
	1	16#20001108	
	2	16#20001208	
	3	16#20001308	

	4	16#20001408	
	5	16#20001508	
	6	16#20001608	
	7	16#20001708	
	8	16#20001808	
RPDO4		mapping	
16#1603			
	0	8	number of entries
	1	16#20001908	
	2	16#20001A08	
	3	16#20001B08	
	4	16#20001C08	
	5	16#20001D08	
	6	16#20001E08	
	7	16#20001F08	
	8	16#20002008	

Table 8: Default communication and mapping parameter, CAN1, RPDO

Index	Sub-index	Value	Description
TPDO1		communication	
16#1800	1	16#180+NodeId	CAN-Identifier
	2	255	transmission type
	3	0	inhibit time
	5	0	event timer
TPDO2		communication	
16#1801	1	16#280+NodeId	CAN-Identifier
	2	255	transmission type
	3	0	inhibit time
	5	0	event timer
TPDO3		communication	
16#1802	1	16#380+NodeId	CAN-Identifier
	2	255	transmission type
	3	0	inhibit time
	5	0	event timer
TPDO4		communication	
16#1803	1	16#480+NodeId	CAN-Identifier
	2	255	transmission type
	3	0	inhibit time
	5	0	event timer
TPDO1		mapping	
16#1A00			
	0	8	number of entries
	1	16#20000108	
	2	16#20300208	
	3	16#20300308	
	4	16#20300408	
	5	16#20300508	
	6	16#20300608	
	7	16#20300708	
	8	16#20300808	
TPDO2		mapping	
16#1A01			
	0	8	number of entries
	1	16#20300908	
	2	16#20300A08	
	3	16#20300B08	
	4	16#20300C08	
	5	16#20300D08	
	6	16#20300E08	

	7	16#20300F08	
	8	16#20301008	
TPDO3		mapping	
16#1A02			
	0	8	number of entries
	1	16#20301108	
	2	16#20301208	
	3	16#20301308	
	4	16#20301408	
	5	16#20301508	
	6	16#20301608	
	7	16#20301708	
	8	16#20301808	
TPDO4		mapping	
16#1A03			
	0	8	number of entries
	1	16#20301908	
	2	16#20301A08	
	3	16#20301B08	
	4	16#20301C08	
	5	16#20301D08	
	6	16#20301E08	
	7	16#20301F08	
	8	16#20302008	

Table 9: Default communication and mapping parameter, CAN1, TPDO

Example:

It is assumed CANopen devices with following configuration are connected to CAN1 on the PLCmodule-C14:

CANopen device using „Predefined Connection Set“:

NodeID: 10

1x RPDO: 3 Variables of type UNSIGNED16,  
CAN-Id = 16#20A

1x TPDO: 2 Variables of type UNSIGNED8,  
1 Variable of type UNSIGNED32  
CAN-Id = 16#18A





## 5 Control Units on the PLCmodule-C14

The PLC module provides at least the following control and display units:

- Run/Stop/Module Reset switch
- Run LED (green)
- Error LED (red)
- IO-State LEDs
- Communication state LEDs

Compact PLCs intended for use in network applications, such as CANopen, provide additional configuration components:

- HEX-encoding switch (CAN node number configuration)
- DIP switch (RS-232/CAN baud rate selection, CANopen-Master selection)

The control and display units are described in more detail in the following sections.

### 5.1 Run/Stop/MRES Switch

The Run/Stop switch allows the user to start and suspend the execution of the PLC program on the compact control unit, as well as to reset the board into its factory default state if required (position Mres).

The Run/Stop switch mounted on the local PLC device and the Start and Stop buttons in the "*Test and Commissioning*" (T&C) tool form a logic AND combination. This means, the program execution on the PLC device will start only if both the local Run/Stop switch is in position Run **and** the Start command (cold, warm or hot start) has been executed in the T&C tool (*refer to the OpenPCS User's Manual for details*). The order in which these control elements are set is not of importance. If the local Run/Stop switch is in position Stop, but the

---

T&C tool has released a Run command, then the Run LED (green) will start short blinking. The switch position "Module Reset" (MRES) is provided to reset the PLC unit into its factory default state. The user must adhere to the operating step order as described in *section 6* in order to avoid unintended erasing of the PLC program.

An illustration of the switch positions Run, Stop and Module Reset is provided in *Table 10*.

Switch Function	Switch Position
RUN	right position
STOP	middle position
MRES	left position

*Table 10: Functions and Positions of the RUN/STOP Switch*

## 5.2 Run LED (green)

The Run LED indicates the operational state of the PLC unit with the following possible modes:

LED Mode	Operational State of the PLC
Off	The PLC unit is in "Stop" state: <ul style="list-style-type: none"> <li>no valid program available on the PLC</li> <li>the local Run/Stop switch is in position Run but the T&amp;C tool has released a Stop command</li> <li>an internal error occurred while executing the PLC program, execution stopped</li> </ul>
Short blinking cycles with a pulse ratio of 1:8 (on to off)	The PLC unit is ready but program execution has not been started: <ul style="list-style-type: none"> <li>the T&amp;C tool has released a Run command but the local Run/Stop switch is still in position Stop</li> </ul>
Slow blinking cycles with a pulse ratio of 1:1	The PLC unit is in "Run" state and executes the PLC program.
Fast blinking cycles with a pulse ratio of 1:1	The PLC unit is in "Module Reset" state, <i>see section 6.3</i> .

*Table 11: Run LED Modes*




### 5.3 Error LED (red)

The Error LED indicates the error states of the PLC unit with the following possible modes:

LED Mode	Operational/Error State of the PLC
Off	No error, the PLC unit is in normal "Run" state and executes the PLC program.
Always on	A serious internal error occurred while executing the PLC program. The PLC has been forced to exit Run state (possible reasons: hardware problem, division by zero,....)
Slow blinking cycles with a pulse ratio of 1:1	A network error occurred during communication with the OpenPCS programming system. If the PLC program was already started on the PLC it will continue to execute. The PLC will automatically reset this error state the next time communication with OpenPCS is successful.
Fast blinking cycles with a pulse ratio of 1:1	The PLC unit is in "Module Reset" state, <i>see section 6.3.</i>

Table 12: Error LED Modes

If a serious internal error occurs, such as a hardware problem or division by zero, the PLC will automatically exit Run state and goes into Stop state. This is indicated by an illuminated red Error LED. In this case, the cause of the error is stored in the PLC and can be extracted at a later time. Follow the steps described below to query the error cause:

- Start OpenPCS.
- Select <PLC  Online >.

A dialog box will appear showing error number and a short description of the error cause.

## 5.4 HEX-Encoding Switch and DIP Switch

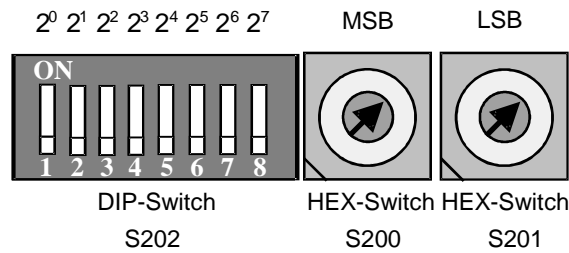


Figure 11: HEX-Encoding Switch and DIP Switch

### 5.4.1 HEX-Encoding Switches

The PLCmodule-C14 is equipped with two HEX-encoding switches marked S200 (MSB) and S201 (LSB). The two HEX-encoding switches are intended for configuration purposes prior to operation rather than implementing the switches as a control unit during runtime.

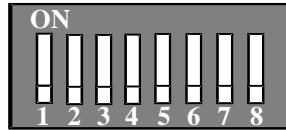
Section 5.4.3 describes the communication parameter configuration.

These switches are used to set up the node address when integrating the PLCmodule-C14 into a CANopen network.

CANopen networks require a unique node number for each individual control unit connected to the CANopen system. Assigning the same node number to two different devices will result in functional problems. Please note that the node numbers 00H and  $\geq 80$ H are reserved and must not be used.

### 5.4.2 DIP Switch

The PLCmodule-C14 is equipped with a 8-position DIP switch at S202. The DIP switch is intended for configuration purposes prior to operation rather than implementing the switches as a control unit during runtime.



DIP	Function	See section:
DIP1	CAN0 or RS232 bitrate	5.4.3.1
DIP2		5.4.3.2
DIP3		
DIP4	CAN0 - CANopen Master (ON/OFF)	5.4.3.2
DIP5	CAN1 (ON/OFF)	5.4.3.3
DIP6	Reserved, OFF	
DIP7	Reserved, OFF	
DIP8	Configuration mode (ON/OFF)	8

Section 5.4.3 describes the communication parameter configuration.

## 5.4.3 Communication Parameter Configuration

### 5.4.3.1 Parameters for RS-232 Communication

#### Configuring the Node Address:

A node address is not required for a peer-to-peer connection, such as an RS-232. Furthermore, the settings on the HEX-encoding switches define the baudrate when using the RS-232 communication.

#### Configuring the Baudrate:

Baudrate configuration for RS-232 communication is done with the DIP switch DIP1 through DIP3, located on the module. The following baudrates are possible:

Baudrate	DIP1	DIP2	DIP3
9600	OFF	OFF	OFF
19200	ON	OFF	OFF
38400	OFF	ON	OFF
57600	ON	ON	OFF
115000	OFF	OFF	ON

Table 13: Configuring the RS-232 Baudrate

Default Settings:

The DIP switches 1-3 are in off position at time of delivery. This configures the PLCmodule-C14 to the default baudrate of 9600 baud.

### 5.4.3.2 CANopen communication via CAN0

Configuring the Node Address:

The CAN node address of the PLC can be configured with the HEX-encoding switches.

**Note:**

Please note that the node numbers 00H and  $\geq 80$ H are reserved and must not be used.

Configuring the Baudrate:

Configuration of the CAN bus baudrate is done with DIP switch DIP1 through DIP3. The following baudrates can be selected:

Bitrate [kBit/s]	DIP1	DIP2	DIP3
10	OFF	OFF	ON
20	ON	OFF	OFF
50	ON	OFF	ON
125	OFF	OFF	OFF
250	OFF	ON	ON
500	OFF	ON	OFF
800	ON	ON	ON
1000	ON	ON	OFF

Table 14: Configuring the CANopen Baudrate

Configuring the CANopen-Master:

Configuration of the CANopen master functionality is done with DIP switch DIP4 for CAN0.

- CAN0: DIP4 = OFF → CANopen Slave  
DIP4 = ON → CANopen Master

Default Settings:

DIP4 is in on position, all other switches are in in off position at time of delivery, while the HEX-encoding switches are set to 20H. This configures the PLC to the following default settings:

- Node Address = 20H
- Bitrate = 125 kBit/s
- CANopen Master @ CAN0 = on
- CAN1 = off

### **5.4.3.3 CANopen communication via CAN1**

On how to configure the communication parameters for the CAN1 interface please refer to *section 0*.

The CAN1 interface is disabled by default. To enable CAN1 the DIP5 has to be set to ON. However, if CAN1 is not needed, DIP5 should be set to OFF for performance reasons.

- CAN1: DIP5 = OFF → CAN1 disabled  
DIP5 = ON → CAN1 enabled

### **5.4.3.4 Ethernet communication**

The Ethernet interface is available in phyPS-412-Z5 version only. It is enabled by default. On how to configure the communication parameters for the Ethernet interface please refer to *section 8.1.3*.

## 6 Operating the PLCmodule-C14

### 6.1 Connecting the CAN Interfaces

The microcontroller is equipped with two integrated FULL-CAN interfaces. The signals CAN0\_HIGH, CAN1\_HIGH, CAN0\_LOW, CAN1\_LOW and CAN\_GND are galvanically decoupled and accessible on the removable connectors according to CiA standard DRP303-1 (open style connector). An external power supply is not required.

The CAN signals can be routed to a DB-9 male connector with signal connection according to the CiA standard DRP303-1:

- CAN<sub>x</sub>\_HIGH at pin 7
- CAN<sub>x</sub>\_LOW at pin 2
- CAN\_GND at pin 6 and/or pin 3

The maximum bitrate supported is 1 MBit/s.

### 6.2 Initial Startup of the PLC

The steps described below are required for the initial startup of the control unit and for communication with the *OpenPCS* programming system.

The initial PLC programming steps are described in more detail in the *OpenPCS Quick Start Instructions* manual.

- **Step 1:**  
The PLCmodule-C14 features two galvanic isolated power supply inputs. Connect the two power supply inputs to a 24VDC source. It is also possible to supply both inputs from one 24VDC source. This, however, would result in shortening the galvanic isolation. We recommend using two independent power supply sources in

order to ensure proper operation of the PLC and galvanic isolation between CPU core circuitry and I/O components.

- **Step 2:**

Connect the PLC device with your host-PC. The connection type, hence the required cable style, depends on the chosen communication protocol. When using the PLC with serial protocol connect the PLC with an RS-232 extension (1:1) cable to your PC. A CAN bus cable is required when using the CANopen version. A Ethernet cable is required when using the Ethernet version.

- **Step 3:**

Configure the desired communication parameter on the PLC in accordance with the chosen network protocol. Refer back to *section 5.4.3* for details on the configuration options.

- **Step 4:**

Select the applicable communication protocol in your *OpenPCS* programming system and configure the matching communication parameters. Follow the steps below:

Start *OpenPCS*. Select <PLC ➤ Connections ...>. A new dialog box "**Connection Setup**" will appear. Click on the "**New**" button. Enter a connection name in the "**Edit Connection**" dialog box. Next click on the "**Select**" button. Select the applicable driver and click on "**OK**". The driver type "**Standard Driver**" must be selected (double-click) when using PLC devices with SYS TEC firmware. Now click on the "**Settings**" button. A new dialog box "**Connection Configuration**" will appear which allows you to specify the desired communication parameters and, if applicable, additional hardware settings. First select one of the available protocols (RS-232, CANopen, UDP/IP, Simulator). Then you can select all other parameters according to your PLC hardware setup.


These "**Global Settings ...**" serve as template for all further connections with the same communication protocol. The parameters, specific for each individual connection, must be set in the "**Hardware Settings ...**" dialog box.

Definition of a node address is required when using a bus system for communication between the PLC and *OpenPCS*. Click on the "**Select Node ...**" button in the "**Connection Configuration**" dialog box and enter the desired node address as hexadecimal value. All extended settings are reserved for customer specific applications and not available in the standard version.

Click on "**OK**" in all the individual dialog boxes in order to close and save the settings.


Highlight the desired resource in the project browser Right-click on this resource and select the context menu item "**Properties**". This will open the "**Edit Resource Specifications**" dialog box. Select the applicable PLC as the hardware module, for example the "*SYSTEC PLCmodule-C14 (phyPS-412-Z4)*".

Choose the previously configured network connection in the corresponding menu. Save and close by clicking on "**OK**".

Now select <PLC  Build All> to start a new compilation of the PLC program code.

- **Step 5:**

Next check the connection between the PLC and the *OpenPCS* programming system. Follow the steps below:


Select <PLC  Online> in the *OpenPCS* programming system. *OpenPCS* will now try to establish communication with the PLC device. If this is successful, the message box "*Connection to the PLC could be established successfully*" will appear indicating that the connection is established.

**Note:**

It may take several seconds until an error message appears if the connection can not be established due to wrong parameter specification or missing power supply on the PLC unit. The system is blocked during this time. The exact delay time depends on the communication protocol used.

Check the PLC hardware properties as described below:



Select <PLC  Hardware Info...> in the *OpenPCS* programming system. The PLC hardware properties will be displayed as shown below if the connection between PLC and *OpenPCS* functions properly.

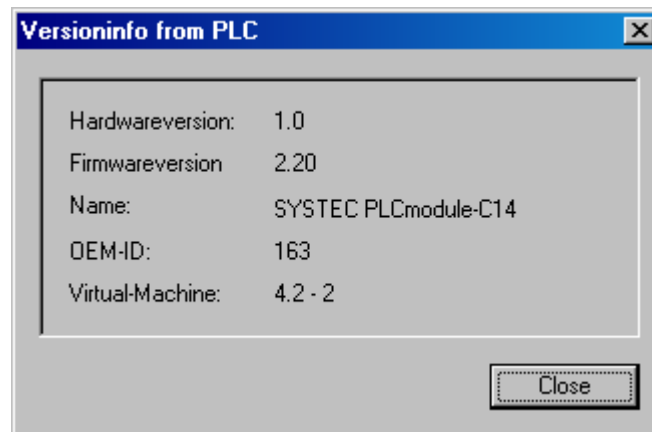


Figure 12: PLC Version info dialog window

### 6.3 Resetting the Control Unit

The PLC device can be reset into its factory default state if required. This results in completely erasing the PLC program stored on the unit and releases the memory. Resetting the PLC device is done locally on the PLC using the Run/Stop switch.

In order to avoid unintended erasure of the control program and, hence, resetting the PLC with the Run/Stop switch, the user must strictly follow the individual steps as described below. This reset procedure can be aborted while executing the individual steps without losing data. The following steps are required for a PLC reset:

- **Step 1:**

Put the Run/Stop switch, located on the PLC, in the "Modul Reset" position (*refer to section 5.1*).

- **Step 2:**

Release a hardware-reset on the PLC device by either pushing the reset button or shortly interrupting the power supply.

⇒The green Run LED will blink with a fast blinking cycle and a pulse ratio of 1:1.

- **Step 3:**

Put the Run/Stop switch, located on the PLC, in the "Run" position.

⇒The red Error LED will blink with a fast blinking cycle and a pulse ratio of 1:1.

- **Step 4:**

Put the Run/Stop **switch back into the "Modul Reset" position within 2 seconds**. The PLC now recognizes the reset request and resets its data memory.

⇒Both the red Error LED and the green Run LED will blink in alternating order.

- **Step 5:**

Now put the Run/Stop switch back into the "Run" or "Stop" position and release a reset in order to render the PLC unit into normal operating state.

The PLC unit will also recognize the reset request in case a reset is released on the control unit (for example caused by a short power interruption) while the Run/Stop switch is in "Modul Reset" position. This will be indicated by a fast blinking green Run LED. If you don't wish to reset the PLC device, just bring the Run/Stop switch back into the "Run" or "Stop" position. The red Error LED will blink for a short time (about 2-3 seconds). The PLC will then automatically abort the reset procedure and start normal execution with unmodified data contents.



## 7 Using the PLC Simulator

The PLC Simulator, integrated in the *OpenPCS* programming system, provides an easy and fast means for testing program algorithms and program debugging for defined input states. Due to the lack of real connection to the control process it is not possible to analyze run-time specific problems.

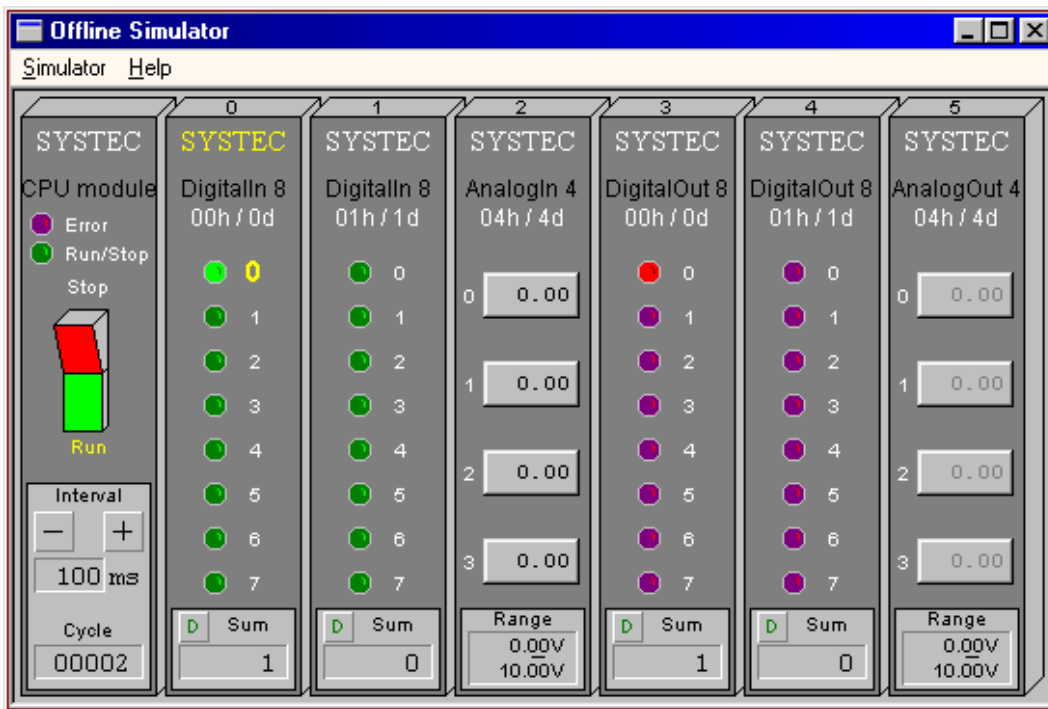


Figure 13: PLC Simulator

### 7.1 Starting the Simulator

Integrating the PLC Simulator in the *OpenPCS* programming system is done with a special network layer that allows data exchange between the "**Test and Commissioning**" (T&C) tool and the Simulator program running on the same PC.

The following steps are required to start the Simulator:

- **Step 1:**  
Start *OpenPCS*.
- **Step 2:**  
Right-click on the desired resource and select the context menu item "**Properties**". This will open the "**Edit Resource Specifications**" dialog box. Select the applicable PLC as the hardware module, for example "*SYSTEC PLCmodule-C14 (phyPS-412-Z3)*". Choose "*Simulation*" for the connection in the corresponding menu. Save and close by clicking on "**OK**".
- **Step 3:**  
Now select <PLC ➔ Build All> to start a new compilation of the PLC program code.
- **Step 4:**  
Select <PLC ➔ Online> in the *OpenPCS* programming system. *OpenPCS* will now try to establish communication with the PLC device. If this is successful, the message box "*Connection to the PLC could be established successfully*" will appear indicating that the connection is established. The Simulator will now start automatically.

## 7.2 Using the Simulator

The Simulator is operated with the help of your mouse and keyboard. The individual control elements can be accessed as listed below:

**Select input:** Selecting the desired input is done with the mouse by clicking on the applicable digital or analog input. Alternatively, the keyboard can be used. Select the desired module with the TAB key and then choose the individual input with the cursor keys "↑" and "↓". The selected input is now clearly marked with a yellow color.

**Digital inputs:** The state of a digital input can be inverted by clicking on the corresponding input symbol or by pressing the space bar on your keyboard. The digital input is in active state with the logic value "1" if the corresponding symbol has a light color. A dark-colored symbol shows an input with the logic value "0".

**Analog inputs:** The desired value can be entered in the selected input field with the keyboard number keys. Valid entries include numbers, comma, decimal point and sign ([0-9] . , + -). Click on any location outside the input field to save and exit the configuration or use the ENTER key or the cursor keys "↑" and "↓". The entered input value is now transferred into the process image.

**Run/Stop Switch:** The Run/Stop switch is used to stop or resume program execution. The desired state can be selected by clicking on the corresponding side of the Run/Stop switch symbol (*see Figure 13*). As an alternative, the "S" (Stop) and "R" (Run) keys can be used.

**Cycle Interval:** Program execution speed can be modified by changing the cycle interval. This allows for easier program analysis. The cycle interval can be altered by clicking on the applicable "+" and "-" buttons in the CPU module section (*see Figure 13*). The "+" (slower) and the "-" keys (faster) can also be used.

**Display Mode:** The digital input value can be displayed in either decimal or hexadecimal notation. The button next to the "Sum" label (*see Figure 13*) shows the current display mode. The letter "D" indicates that decimal notation is used, while "H" shows hexadecimal values. The display mode is altered by clicking this button or by pressing the "X" key.

You also find a complete overview with additional details about using the Simulator in the online help.


### 7.3 Simulating Compact Control Units

Programs written for a PLC module can also be tested with the Simulator prior to downloading to the PLC unit. The modular layout of the Simulator contains various input and output modules that correspond to the inputs and outputs available on the PLC module as shown in *Table 15*. Inputs and outputs displayed in the Simulator but not physically available on the PLC module have no function.

Input/Output on the PLC module	Input/Output Module in the Simulator
DI0 ... DI7	digital input module 0
DI8 ... DI15	digital input module 1
DI16 ... DI23 <sup>(1)</sup>	digital input module 2 <sup>(1)</sup>
AI0 ... AI3	analog input module 3
C0 ... C2 <sup>(1)</sup>	counter channel 6 (counter 1) <sup>(1)</sup>
DO0 ... DO7	digital output module 0
DO8 ... DO15	digital output module 1
P0 ... P1 <sup>(1)</sup>	not supported
AO0 ... AO1	analog output module 3

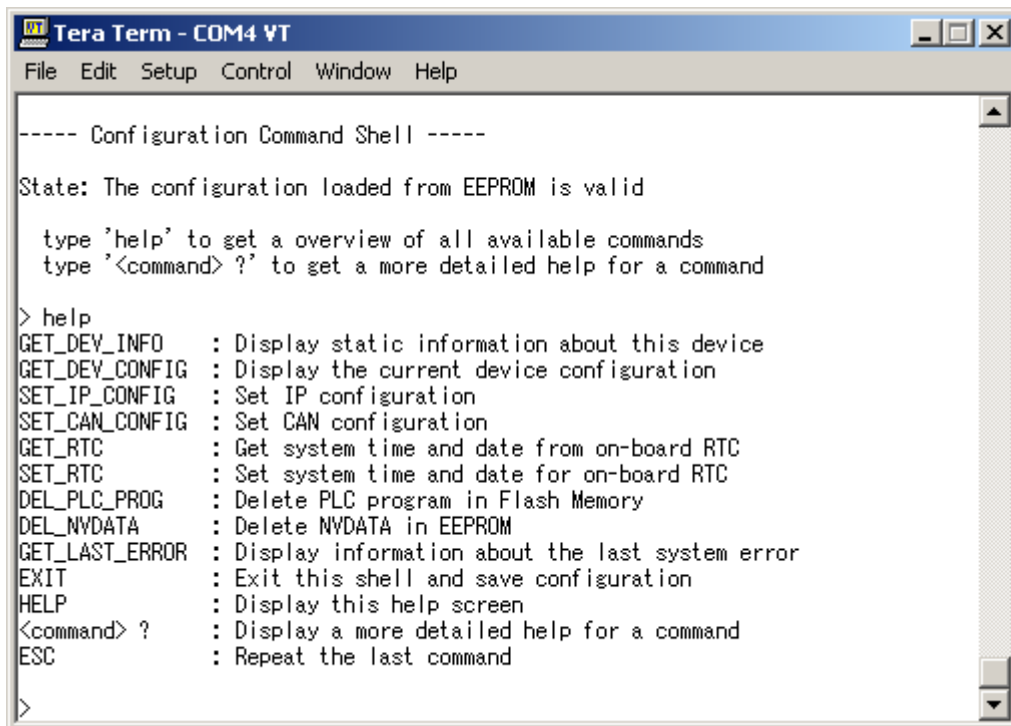
*Table 15: Simulator Addresses for PLC modules <sup>(1)</sup> see description*

In the Simulator default configuration not all available inputs and outputs of the PLC module are displayed. Due to space considerations, only the most commonly used modules are shown. For example, all inputs higher than DI16, as well as the counter channel are not displayed in the Simulator. All hidden modules are marked with in *Table 15*. Alternatively, the Simulator can display all I/O components available on the PLC module. The width of the Simulator window will increase by adding three new modules. In order to activate these additional I/O modules follow the steps below:

Select <Simulator  Show all modules>. The Simulator window will reappear with all available modules displayed in the new window.

## 8 Initial Configuration using the Configuration Command Shell

The Configuration Command Shell (herein after called config shell) of the PLCmodule-C14 provides a set of commands to read and write device configurations such as TCP/IP address settings and CAN interface configurations.



```
---- Configuration Command Shell ----
State: The configuration loaded from EEPROM is valid

type 'help' to get a overview of all available commands
type '<command> ?' to get a more detailed help for a command

> help
GET_DEV_INFO      : Display static information about this device
GET_DEV_CONFIG    : Display the current device configuration
SET_IP_CONFIG     : Set IP configuration
SET_CAN_CONFIG    : Set CAN configuration
GET_RTC           : Get system time and date from on-board RTC
SET_RTC           : Set system time and date for on-board RTC
DEL_PLC_PROG      : Delete PLC program in Flash Memory
DEL_NVDATA        : Delete NVDATA in EEPROM
GET_LAST_ERROR    : Display information about the last system error
EXIT              : Exit this shell and save configuration
HELP              : Display this help screen
<command> ?      : Display a more detailed help for a command
ESC               : Repeat the last command

>
```

Figure 14: Built-in Configuration Command Shell

To access the config shell connect the PLC to the PC via ASC0 using the adapter cable delivered with the module. Open a connection using a terminal program such as Hyperterminal coming with Windows operating systems.

The configuration parameters have to be set to 9600bps, 8N1, no flow control.

Set switch 8 on the DIP-Switch block to ON and reset the device to enter the configuration mode. The config shell status message should come up on your terminal window.



All inputs are case insensitive. The HELP command gives you an overview about all command available. To get help for a specific command just enter HELP followed by the command name, or type the command followed by an “?”. Press Escape to repeat the last command. When you have finished leave the config shell by typing “exit”. The configuration is stored in non-volatile memory and the PLC starts. Do not forget to put switch 8 into its original position (OFF).

## **8.1 Command description**

The following section describes the commands available on config shell for PLCmodule-C14.

### **8.1.1 GET\_DEV\_INFO**

Display static information about this device such as serial number (Device-ID) and firmware version. This command has no parameters.

Example:

```
> GET_DEV_INFO

Device Information:
Type:          SYSTEC PLCmodule-C14 (phyPS-412-Z5)
DeviceID:      1004005
Hardware:      1.00
  CPU Board:   4112.00 (#0BH)
  IO Board:    4103.01 (#F6H)
Firmware:      3.01
Virtual Machine: 4.02 - 2 (#42)
OEM-ID:        163

Serial number: 12345678
ETH0 MacAddr: AA-BB-CC-DD-EE-FF
```

### **8.1.2 GET\_DEV\_CONFIG**

Displays the current device configuration. This command has no parameters. The output also specifies whether the parameter can be configured via config shell (rw – read write) or not (ro – read only).

**Example:**

```
> GET_DEV_CONFIG

Device Configuration:

(ro) ETH0 Enabled:      on
(ro) ETH0 MacAddr:     AA-BB-CC-DD-EE-FF
(rw) ETH0 IpAddr:      192.168.010.130
(rw) ETH0 SubnetMask:  255.255.255.000
(rw) ETH0 DefGateway:  192.168.010.001

(ro) CAN0 Enabled:     on
(ro) CAN0 NodeID:      32 (0x20)
(ro) CAN0 Baudrate:    125
(ro) CAN0 MasterMode:  on

(ro) CAN1 Enabled:     off
(rw) CAN1 NodeID:      48 (0x30)
(rw) CAN1 Baudrate:    125
(rw) CAN1 MasterMode:  on
```

### 8.1.3 SET\_IP\_CONFIG

The SET\_IP\_CONFIG command is used to set the communication parameters for Ethernet interface.

Parameters		Description
IfNum	Mandatory	Interface number (always 0 on PLCmodule-C14)
IpAddr	Mandatory	IP address (i.e. 192.168.10.130)
NetMask	Mandatory	Netmask (i.e. 255.255.255.0)
Gateway	Optional	IP address of default gateway

**Example:**

```
SET_IP_CONFIG 0 192.168.10.130 255.255.255.0 192.168.10.200
```

### 8.1.4 SET\_CAN\_CONFIG

The SET\_CAN\_CONFIG command is used to set the communication parameters of the CAN1-interface.

Parameters		Description
IfNum	Mandatory	Interface number (always 1 on PLCmodule-C14, CAN0 is configured via the DIP-switches and HEX-encoding switches)
NodeID	Mandatory	Node-ID (i.e. 0x21)
Bitrate	Mandatory	CAN-bus bitrate (i.e. 500)
MstMode	Optional	Master mode (on/off)

Example:

```
SET_CAN_CONFIG 1 0x20 500 off
```

### 8.1.5 DEL\_CAN\_CONFIG

The SET\_CAN\_CONFIG command is used to erase the communication parameters of the CAN1-interface.

Parameters		Description
IfNum	Mandatory	Interface number (always 1 on PLCmodule-C14, CAN0 is configured via the DIP-switches and HEX-encoding switches)

Example:

```
DEL_CAN_CONFIG 1
```

### 8.1.6 GET\_RTC

Shows the system time and date from the on-board RTC unit. This command has no parameters.

Example:

```
> GET_RTC  
  
Valid: yes  
Time: 08:54:50  
Date: 2004/08/20
```

### 8.1.7 SET\_RTC

SET\_RTC enables you to set the system time and date for the on-board RTC. The RTC on PLCmodule-C14 is battery backed. The time format is based on 24h. Please refer to the SYS TEC PLC extension Manual (I-1054) for more information on how to accessing the RTC from within the PLC program.

Parameters		Description
hh:mm:ss	Mandatory	hh: hour (0..24) mm: minutes (0..59) ss: seconds (0..59)
yyyy/mm/dd	optional	yyyy: year Mm: month dd: day

Example:

```
SET_RTC 10:15:12 2004/08/20
```

### **8.1.8 DEL\_PLC\_PROG**

The DEL\_PLC\_PROG command enables you to delete the PLC programs stored on the PLC. This gives you the chance to access the PLC, even when your PLC programs have a race condition or are trapped in an endless loop.

This command has no parameters. Before the PLC program archive gets deleted the config shell asks for confirmation.

Example:

```
> DEL_PLC_PROG
Do you really want to delete the PLC program archive (y/n)? y
Ok
```

### **8.1.9 DEL\_NVDATA**

The DEL\_NVDATA command deletes all data(variables) stored in non-volatile memory.

This command has no parameters. Before the PLC program archive gets deleted the config shell asks for confirmation.

Example:

```
> DEL_NVDATA
Do you really want to delete the NVDATA archive (y/n)? y
Ok
```

### **8.1.10 GET\_LAST\_ERROR**

Displays detailed information about the last system error occurred.

Example:

```
> get_last_error

ErrorCode = 0
ErrorText = Controler working normal

Ok
```

### 8.1.11 SET\_PWF\_MODE

By mean of the SET\_PWF\_MODE command the modules behaviour on powerfail events can be defined.

Parameters	Description
ON	Default-setting, powerfail guarding is activated. The device is reset on powerfail event.
MONITOR	Powerfail-events are recognized by the firmware without resetting the device. However, an error message (message box) is sent to the programming system when a powerfail-event has occurred.
OFF	Powerfail-guarding is inactive, the device does not react to any power-fail events.

Example:

```
SET_PWF_MODE MONITOR
```

### 8.1.12 SET\_WDG\_MODE

By mean of the SET\_WDG\_MODE command the modules behaviour on Watchdog-events can be defined.

Parameters		Description
ON		Default-setting, Watchdog guarding is activated. The device is reset on Watchdog-event.
MONITOR		Watchdog-events are recognized by the firmware without resetting the device. However, an error message (message box) is sent to the programming system when a Watchdog-event has occurred.
OFF		Watchdog-guarding is inactive, the device does not react to any Watchdog-events.

Example:

```
SET_WDG_MODE MONITOR
```

### 8.1.13 EXIT

Exit the config shell, save configuration and restart the PLC.

### 8.1.14 HELP

Display command list and short description (see picture above).

## 9 Device pinout

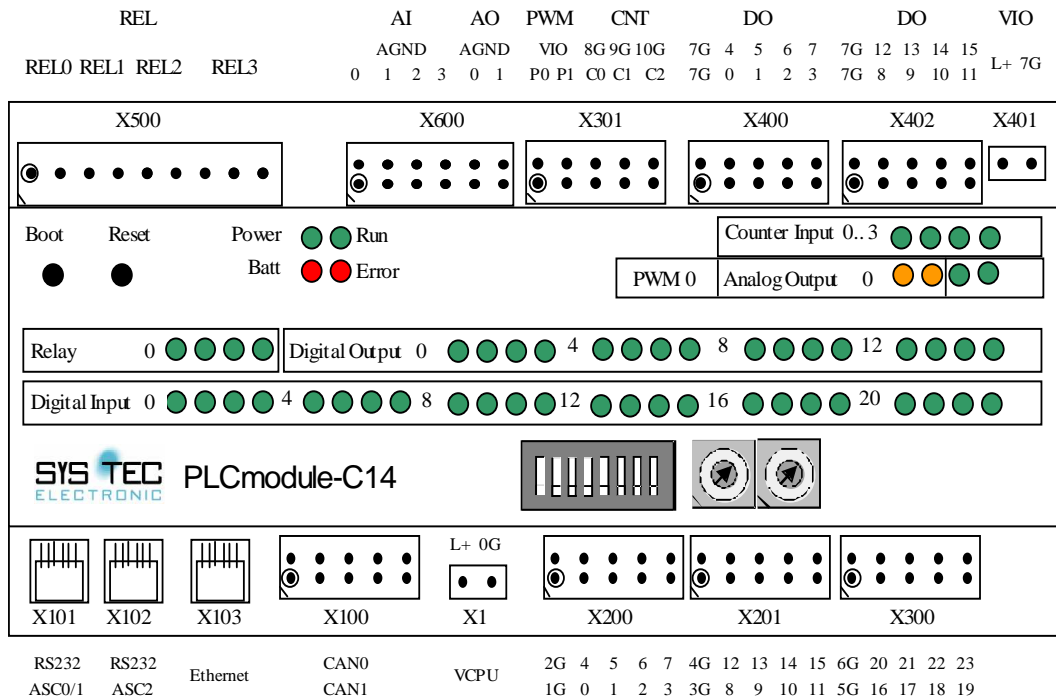


Figure 15: Device pinout overview

Interface	Pin Description	Pin #	Label
power supply CPU	VCPU, +24VDC	X1.1	L+
	Ground GND_VCPU	X1.2	0G
power supply for IOs	VIO, +24VDC	X401.1	L+
	Ground GND_VIO	X401.2	7G
ASC0, RS-232 interface	TxD, RS232 level	X101.2	
	GND	X101.3	
	RxD, RS232 level	X101.4	
ASC1, RS-232 interface	RxD, RS232 level	X101.6	
	TxD, RS232 level	X101.1	
	GND	X101.5	
ASC2, RS-232 interface with modem control lines	DSR, RS232 level	X102.1	
	CTS, RS232 level	X102.2	
	DCD, RS232 level	X102.3	
	RxD, RS232 level	X102.4	
	GND	X102.5	
	TxD, RS232 level	X102.6	



CAN-bus interface CAN0	Reserved	X100.5A	
	CAN_H	X100.4A	
	Reserved	X100.3A	
	CAN_L	X100.2A	
	CAN_GND	X100.1A	
CAN-bus interface CAN1	Reserved	X100.5B	
	CAN_H	X100.4B	
	Reserved	X100.3B	
	CAN_L	X100.2B	
	CAN_GND	X100.1B	
Ethernet Interface	TxD+	X103.1	
	TxD-	X103.2	
	RxD+	X103.3	
	RxD-	X103.6	
Digital Inputs DI0..7	Common ground DI0...DI3	X200.1A	1G
	DI0	X200.2A	0
	DI1	X200.3A	1
	DI2	X200.4A	2
	DI3	X200.5A	3
	Common ground DI4..DI7	X200.1B	2G
	DI4	X200.2B	4
	DI5	X200.3B	5
	DI6	X200.4B	6
	DI7	X200.5B	7
Digital Inputs DI8..15	Common ground DI8..DI11	X201.1A	3G
	DI8	X201.2A	8
	DI9	X201.3A	9
	DI10	X201.4A	10
	DI11	X201.5A	11
	Common ground DI12..DI15	X201.1B	4G
	DI12	X201.2B	12
	DI13	X201.3B	13
	DI14	X201.4B	14
	DI15	X201.5B	15

Digital Inputs DI16..23	Common ground DI16..DI19	X300.1A	5G
	DI16	X300.2A	16
	DI17	X300.3A	17
	DI18	X300.4A	18
	DI19	X300.5A	19
	Common ground DI20..DI23	X300.1B	6G
	DI20	X300.2B	20
	DI21	X300.3B	21
	DI22	X300.4B	22
	DI23	X300.5B	23
Counter Inputs C0..2	C0	X301.3A	24
	Ground C0	X301.3B	8G
	C1	X301.4°	25
	Ground C1	X301.4B	9G
	C2	X301.5°	26
	Ground C2	X301.5B	10G
Digital Outputs DO0..7	Common ground GND_VIO	X400.1A	7G
	DO0	X400.2A	0
	DO1	X400.3A	1
	DO2	X400.4A	2
	DO3	X400.5A	3
	Common ground GND_VIO	X400.1B	7G
	DO4	X400.2B	4
	DO5	X400.3B	5
	DO6	X400.4B	6
	DO7	X400.5B	7
Digital outputs DO8..15	Common ground GND_VIO	X402.1A	7G
	DO8	X402.2A	8
	DO9	X402.3A	9
	DO10	X402.4A	10
	DO11	X402.5A	11
	Common ground GND_VIO	X402.1B	7G
	DO12	X402.2B	12
	DO13	X402.3B	13
	DO14	X402.4B	14
	DO15	X402.5B	15

PWM outputs	PWM output 0	X301.1A	P0
	+24VDC IO	X301.1B	VIO
	PWM output 1	X301.2A	P1
	+24VDC IO	X301.2B	VIO
Analog Inputs AI0..3	AI0	X600.1A	0
	Common ground GND_VCPU	X600.1B	AGND
	AI1	X600.2A	1
	Common ground GND_VCPU	X600.2B	AGND
	AI2	X600.3A	2
	Common ground GND_VCPU	X600.3B	AGND
	AI3	X600.4A	3
	Common ground GND_VCPU	X600.4B	AGND
Analog Outputs AO0..1	AO0	X600.5A	0
	Common ground GND_VCPU	X600.5B	AGND
	AO1	X600.6A	1
	Common ground GND_VCPU	X600.6B	AGND
Dry-Contact Outputs (Relay)	REL0 C	X500.1	
	REL0 NO	X500.2	
	REL1 C	X500.3	
	REL1 NO	X500.4	
	REL2 C	X500.5	
	REL2 NO	X500.6	
	REL3 C	X500.7	
	REL3 NO	X500.8	
	REL3 NC	X500.9	

Tabelle 1: Connector pinout



## 10 Technical Specifications

Environmental Parameters		Typical	Minimum	Maximum
Power Supply	V <sub>CPU</sub>	24VDC	22VDC	30VDC
	V <sub>IO</sub>	24VDC	-20%	+20%
Current Consumption (inactive IOs)	I <sub>CPU</sub>	0.180mA		
	I <sub>IO</sub>	0.180mA		
Temperature Range	Storage temperature	-20°..+70°C		
	Operating temperature	0°..+50°C		
Protection class	Enclosure	IP20		
Weight	without any cable and packing	350g		
Dimensions	Width	160mm		
	Height	75mm		
	Depth	95mm		
Connector type		Spring type connector		

Table 16: Environmental Parameters

Communication Interfaces		Min.	Max.
<b>CAN-Bus</b>			
CAN0/1	Bitrate	10kBit/s	1Mbit/s
	Max. number of nodes		64
	CAN-H, CAN-L short-circuit-proof towards 24V		
<b>Ethernet – 10base T</b>			
	Transmission rate		10Mbit/s
<b>RS-232</b>			
ASC 0/1/2	Baudrate	1200bps	115kbps

Table 17: Communication Interfaces

<b>I/O-configuration</b>			
<b>Digital Outputs DO0 .. 15</b>			
24VDC Output (High Side Switch)	$U_{OH}$ at $I_{OH} = 500 \text{ mA}$	$V_{IO} - 0,16\text{V} < U_{OH} < V_{IO}$	
	$U_{OL}$ at $I_{OL} = 0 \text{ mA}$		0.5V
	Current limitation $I_{OH\_max}$		625mA
	Max. current		8A
	$I_{OL(off)}$		10 $\mu$ A
	$t_{off}$ at $I_{OH} = 500 \text{ mA}$	115 $\mu$ s	190 $\mu$ s
	$t_{on}$ at $I_{OH} = 500 \text{ mA}$	75 $\mu$ s	125 $\mu$ s
<b>Digital Outputs REL0 .. 3</b>			
Relay outputs	Switching Voltage		250AC
	Switching Current		6A
	Durability (mech.) at 5A		1x10 <sup>5</sup>
	$t_{on}$	5ms	
	$t_{off}$	2,5ms	
	Isolation		4kV
<b>Digital Inputs DI0 .. 23</b>			
24VDC-Inputs, pulse switching	$U_{IH}$	15V	30V
	$U_{IL}$	-3V	5V
	$I_{ICH}$	3mA	8,5mA
<b>Counter Inputs C0..3</b>			
24VDC-Inputs, pulse switching	$U_{IH}$	15V	30V
	$U_{IL}$	-3V	5V
	$I_{ICH}$	3mA	8,5mA
Input frequency	f		70kHz
<b>Analog Inputs AI0 .. 3</b>			
0 .. +10V	Measurement range $U_I$	0..+10,18V	
	Destructive voltage $U_{I\_max}$		>30V
	Input resistance $R_I$		49k $\Omega$ $\pm$ 0.1%
	Reference voltage $U_{REF}$	4.572V	$\pm$ 1%
	Physical Resolution		10Bit

<b>PWM outputs 0..1</b>			
24VDC-PWM- Output (Low Side Switch)	$U_{OL}$ at $I_{OL} = -500\text{mA}$		<1V
	$I_{OH(off)}$		20 $\mu\text{A}$
	$I_{OH\_max}$		0.6A
	$t_{on}$ at $I_{OL} = -500\text{mA}$		2.5 $\mu\text{s}$
	$t_{off}$ at $I_{OL} = -500\text{mA}$		3.5 $\mu\text{s}$
	PWM Frequency min.		15Hz
	PWM Frequency max.		15kHz
<b>Analog Outputs AO0 .. 1</b>			
0 .. +10V	Voltage Range $U_O$	0 – +10.18V	
	Output current $I_O$		30mA
	Output capacity		10nF
	Reference voltage $U_{REF}$	4.572V	$\pm 1\%$
	Internal Resolution		10 Bit

Table 18: IO configuration

<b>PLC specific parameters</b>			
memory configuration	PLC program		256kiB
	PLC data		256kiB
	Non-volatile data		24kiB
cycle time for 1000 binary instructions	interpreter code (one CAN instance enabled)		9 ms
	interpreter code (two CAN instances enabled)		15 ms
	native code (one CAN instance enabled)		1-2 ms
	native code (two CAN instances enabled)		3 ms

Table 19: PLC specific parameters

These specifications describe the standard configuration of the PLCmodule-C14 as of the printing of this manual.





---

## Index

- Addressing ..... 17  
Analog Inputs..... 10, 60, 62  
Analog Outputs ..... 10, 60, 63  
ASC..... 61  
Baudrate ..... 35, 51, 61  
bitrate ..... 35, 36, 39, 52  
Block Diagram ..... 4  
CAN Interface..... 12  
CAN012, 13, 35, 36, 39, 51, 52,  
58, 61  
CAN112, 13, 21, 26, 28, 29, 35,  
37, 39, 51, 52, 58  
CAN-Bus..... 61  
CANopen1, 2, 13, 15, 21, 22, 23,  
24, 28, 31, 34, 35, 36, 37, 40  
CANopen Master ..... 13  
Communication Interfaces ..... 61  
Communication Parameter..... 35  
Configuration Command Shell 13,  
49  
configuration mode ..... 49  
Connector pinout..... 57  
Connectors ..... 5  
Control Units..... 17, 31, 48  
Counter Inputs..... 7  
Cycle Interval..... 47  
DEL\_CAN\_CONFIG ..... 52  
DEL\_PLC\_PROG..... 54  
Device pinout ..... 57  
Digital Inputs ..... 5, 6, 58, 59, 62  
Digital Outputs..... 8, 59, 62  
DIP Switch..... 34  
Display Mode..... 47  
EMC ..... 1  
Environmental Parameters ..... 61  
Error LED ..... 31, 33, 43  
Ethernet12, 13, 15, 37, 40, 51, 58,  
61  
EXIT..... 56  
Firmware Expansions..... 15  
Firmware versions..... 15  
Fundamentals ..... 17  
GET\_DEV\_CONFIG..... 50  
GET\_DEV\_INFO ..... 50  
GET\_LAST\_ERROR..... 54  
GET\_RTC ..... 53  
HELP..... 56  
HEX-Encoding Switches ..... 34  
I/O-configuration ..... 62  
Initial Configuration..... 49  
Initial Startup..... 39  
Marker space ..... 21, 22, 23, 24  
MRES Switch..... 31  
Network Protocol ..... 40  
*OpenPCS*1, 13, 31, 33, 39, 40, 41,  
42, 45, 46  
Operating the PLCmodule-C14 39  
Overview ..... 3  
PDO mapping..... 22  
PLC Simulator..... 45  
Power Supply ..... 5, 61  
Preface..... 1  
process image5, 8, 13, 15, 17, 19,  
20, 22, 47  
Programming..... 1, 13  
PWM outputs ..... 5, 9, 10, 60, 63  
PWM Outputs ..... 9  
Relay Outputs..... 8  
Resetting the Control Unit ..... 42  
Resource Specifications ..... 41, 46  
RPDO ..... 22, 23, 26, 28

---

RS-23211, 13, 15, 31, 35, 40, 57, 61	Simulator, Cycle Interval ..... 47
Run LED ..... 31, 32, 43	Simulator, Digital Inputs..... 47
Run/Stop Switch ..... 31, 47	Simulator, Display Mode ..... 47
Selecting the Communication Protocol..... 40	Simulator, Run/Stop Switch ..... 47
Serial Interfaces ..... 11	Start-up..... 14
SET_CAN_CONFIG..... 52	System Description ..... 5
SET_IP_CONFIG..... 51	Technical Specifications ..... 61
SET_RTC ..... 53	Testing the Connection ..... 41
SET_WDG_MODE..... 56	TPDO ..... 22, 24, 28
Simulator..... 40, 45, 46, 47, 48	Using the Simulator ..... 46
Simulator, Analog Inputs..... 47	VCPU..... 5
	VIO ..... 5
	Watchdog ..... 56

---

<b>Document:</b>	PLCmodule-C14
<b>Document number:</b>	L-1044e_7, Edition Juli 2013

---

**How would you improve this manual?**

---

---

---

---

**Did you find any mistakes in this manual?** page

---

---

---

---

**Submitted by:**

Customer number: \_\_\_\_\_

Name: \_\_\_\_\_

Company: \_\_\_\_\_

Address: \_\_\_\_\_

---

**Return to:**           SYS TEC electronic GmbH  
                          August-Bebel-Str. 29  
                          D-07973 Greiz  
                          GERMANY  
                          Fax : +49 (0) 36 61 / 62 79 99

---

Published by

---

© SYS TEC electronic GmbH 2006

**SYS TEC**  
ELECTRONIC  
Ordering No. L-1044e\_7  
Printed in Germany