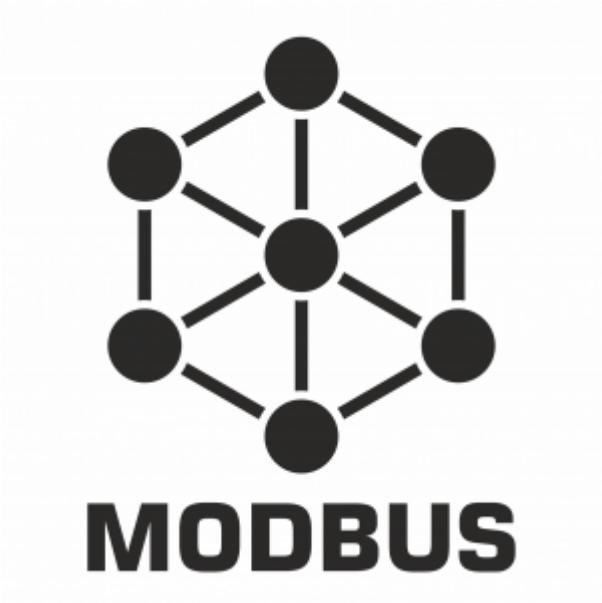
Sommario

RosettaCNC plugin Modbus



Information

Document:	**MDUROSETTACNCMODBUS**
Description:	RosettaCNC Modbus inteface
Link:	http://wiki.rosettacnc.com/en/software/rosettacncmodbus

Release documento	Descrizione	Note	Data
01	First release	/	22/10/2020
02	Align with 1.9.2 release	/	28/04/2021
03	Fix IR_GCODE_CMD address and add IR_EXECUTION_INFO info	/	05/10/2021

Introduction

The RosettaCNC board has a MODBUS® SLAVE server inside to exchange information with a PLC. This function is interesting for implementing machine logic inside the PLC.
The implemented protocol is MODBUS® SLAVE TCP. Port, slave id and refresh time can be set in Board settings panel.

The server share the same IP address used by the card for connection to the software on the PC.

The interface contract between the MODBUS® SLAVE in the CNC board and the product that will implement the MODBUS® MASTER connection is bound to the parameter IR_MODBUS_INTERFACE_LEVEL. The first operation to do as soon as communication with the SLAVE is accessed is to retrieve the contents of this register and adapt the subsequent requests to the relative data table model.

The MODBUS_INTERFACE_LEVEL version 1 is currently available.

In the installation package there is the “extra \ modbusview” folder with a .NET example compiled with Visual Studio 2017 Community version with which it is possible to see how easy it is to access the data of the RosettaCNC card by creating a MODBUS® Master.

What you can do with modbus interface ?

Monitor CNC state

Reading INPUT registers you can monitor the CNC state: positions, state, alarms, gcode line during execution, etc

Activate Alarms/Warnings

Writing on HR_CUSTOM_ALARM_xx or HR_CUSTOM_WARNING_xx you can active an alarm/warning. There are max 8 alarms and 8 warnings. Alarm text can be set in CNC control software and is multi language.

Exchange data with Gcode

Bi directinal shared memory is provided. Gcode can write values that can be so visible to the modbus (see G100 P1000 and IR_SHARED_MEMORY registers). Modbus can write values that can be visible to the Gcode (see M167 P1000 and HR_SHARED_MEMORY registers).

Take care that shared memory is retentive so PLC have to context values with CNC states and other informations.

I/O expansion

Use virtual inputs / outputs concept to expando CNC I/O. In this case PLC have only to copi modbus data to real electrical resource.

Function control

Use vitual inputs outputs concept to operate with normal I/O feature. For example if you want PLC to start a Gcode execution, map the Start input function on first virtual input. When PLC active the relative virtual bit CNC starts execution.

If you want to inform PLC about AUX2 output state, map AUX2 to a virtual outputs.

Registers Table

```
; modbus input registers table
IR_MODBUS_INTERFACE_LEVEL       30001   ; input register [W]: modbus interface level

IR_FIRMWARE_VERSION             30002   ; input register [W]: firmware version:
                                        ;   b15..b08 = version
                                        ;   b07..b00 = release

IR_SERIAL_NUMBER                30003   ; input register [L]: serial number

IR_MACHINE_POSITION_X           30005   ; input register [L]: machine position X [mm/1000]
IR_MACHINE_POSITION_Y           30007   ; input register [L]: machine position Y [mm/1000]
IR_MACHINE_POSITION_Z           30009   ; input register [L]: machine position Z [mm/1000]

IR_TARGET_POSITION_X            30011   ; input register [L]: target position X [mm/1000]
IR_TARGET_POSITION_Y            30013   ; input register [L]: target position Y [mm/1000]
IR_TARGET_POSITION_Z            30015   ; input register [L]: target position Z [mm/1000]
IR_TARGET_POSITION_A            30017   ; input register [L]: target position A [°/1000]
IR_TARGET_POSITION_B            30019   ; input register [L]: target position B [°/1000]
IR_TARGET_POSITION_C            30021   ; input register [L]: target position C [°/1000]

IR_PROGRAM_POSITION_X           30023   ; input register [L]: program position X [mm/1000]
IR_PROGRAM_POSITION_Y           30025   ; input register [L]: program position Y [mm/1000]
IR_PROGRAM_POSITION_Z           30027   ; input register [L]: program position Z [mm/1000]
IR_PROGRAM_POSITION_A           30029   ; input register [L]: program position A [°/1000]
IR_PROGRAM_POSITION_B           30031   ; input register [L]: program position B [°/1000]
IR_PROGRAM_POSITION_C           30033   ; input register [L]: program position C [°/1000]

IR_WORKING_OFFSET_X             30047   ; input register [L]: working offset X [mm/1000]
IR_WORKING_OFFSET_Y             30049   ; input register [L]: working offset Y [mm/1000]
IR_WORKING_OFFSET_Z             30051   ; input register [L]: working offset Z [mm/1000]
IR_WORKING_OFFSET_A             30053   ; input register [L]: working offset A [°/1000]
IR_WORKING_OFFSET_B             30055   ; input register [L]: working offset B [°/1000]
IR_WORKING_OFFSET_C             30057   ; input register [L]: working offset C [°/1000]

IR_ACT_VELOCITY_X               30059   ; input register [L]: actual velocity X [mm/min]
IR_ACT_VELOCITY_Y               30061   ; input register [L]: actual velocity Y [mm/min]
IR_ACT_VELOCITY_Z               30063   ; input register [L]: actual velocity Z [mm/min]
IR_ACT_VELOCITY_A               30065   ; input register [L]: actual velocity A [°/min]
IR_ACT_VELOCITY_B               30067   ; input register [L]: actual velocity B [°/min]
IR_ACT_VELOCITY_C               30069   ; input register [L]: actual velocity C [°/min]

IR_MAIN_CNC_STATE               30071   ; input register [W]: main cnc state
                                        ;    0: Init
                                        ;    1: Init Fieldbus
                                        ;    2: Alarm
                                        ;    3: Idle
                                        ;    4: Homing
                                        ;    5: Jog
                                        ;    6: Run
                                        ;    7: Pause
                                        ;    8: Limit
                                        ;    9: Measuring tool offset
                                        ;    10: Scan 3D
                                        ;    11: Safety Idle
                                        ;    12: Change Tool
                                        ;    13: Safety
                                        ;    14: Waiting Main Power
                                        ;    15: Retract tool

IR_CNC_STATES                   30072   ; input register [W]: cnc states
                                        ;    b00: AllAxesHomed
                                        ;    b01: ProgramExecutionEndedSuccesfully
IR_ALARM_CODE                   30073   ; input register [W]: alarm code
IR_ALARM_INFO_FIELD_A           30074   ; input register [W]: alarm info field A
IR_ALARM_INFO_FIELD_B           30075   ; input register [W]: alarm info field B

IR_GCODE_LINE                   30076   ; actual gcode line [L]

IR_EXECUTION_INFO               30078   ; b00-b07: buffer level [%], b08-b15: program transferred [%]
```

```
IR_VIRTUAL_OUTPUTS              30079   ; virtual outputs 1 to 16 mask
                                        ;   b0 = virtual outputs 1
                                        ;   b1 = virtual outputs 2
                                        ;   ...
                                        ;   b15 = virtual outputs 16

IR_SHARED_MEMORY_01             30080   ; shared memory 32 bit G100 P1000 A parameter
IR_SHARED_MEMORY_02             30082   ; shared memory 32 bit G100 P1000 B parameter
IR_SHARED_MEMORY_03             30084   ; shared memory 32 bit G100 P1000 C parameter
IR_SHARED_MEMORY_04             30086   ; shared memory 32 bit G100 P1000 D parameter
IR_SHARED_MEMORY_05             30088   ; shared memory 32 bit G100 P1000 E parameter
IR_SHARED_MEMORY_06             30090   ; shared memory 32 bit G100 P1000 F parameter
IR_SHARED_MEMORY_07             30092   ; shared memory 32 bit G100 P1000 H parameter
IR_SHARED_MEMORY_08             30094   ; shared memory 32 bit G100 P1000 I parameter

IR_GCODE_CMD                    30096   ; actual gcode command [L]

; modbus holding registers table
HR_CUSTOM_ALARM_01              40001   ; set to non zero to sign custom alarm 01
HR_CUSTOM_ALARM_02              40002   ; set to non zero to sign custom alarm 02
HR_CUSTOM_ALARM_03              40003   ; set to non zero to sign custom alarm 03
HR_CUSTOM_ALARM_04              40004   ; set to non zero to sign custom alarm 04
HR_CUSTOM_ALARM_05              40005   ; set to non zero to sign custom alarm 05
HR_CUSTOM_ALARM_06              40006   ; set to non zero to sign custom alarm 06
HR_CUSTOM_ALARM_07              40007   ; set to non zero to sign custom alarm 07
HR_CUSTOM_ALARM_08              40008   ; set to non zero to sign custom alarm 08

HR_CUSTOM_WARNING_01            40009   ; set to non zero to sign custom warning 01
HR_CUSTOM_WARNING_02            40010   ; set to non zero to sign custom warning 02
HR_CUSTOM_WARNING_03            40011   ; set to non zero to sign custom warning 03
HR_CUSTOM_WARNING_04            40012   ; set to non zero to sign custom warning 04
HR_CUSTOM_WARNING_05            40013   ; set to non zero to sign custom warning 05
HR_CUSTOM_WARNING_06            40014   ; set to non zero to sign custom warning 06
HR_CUSTOM_WARNING_07            40015   ; set to non zero to sign custom warning 07
HR_CUSTOM_WARNING_08            40016   ; set to non zero to sign custom warning 08

HR_SHARED_MEMORY_01             40018   ; shared memory 32 bit M167 P1000 (#5740)
HR_SHARED_MEMORY_02             40020   ; shared memory 32 bit M167 P1000 (#5741)
HR_SHARED_MEMORY_03             40022   ; shared memory 32 bit M167 P1000 (#5742)
HR_SHARED_MEMORY_04             40024   ; shared memory 32 bit M167 P1000 (#5743)
HR_SHARED_MEMORY_05             40026   ; shared memory 32 bit M167 P1000 (#5744)
HR_SHARED_MEMORY_06             40028   ; shared memory 32 bit M167 P1000 (#5745)
HR_SHARED_MEMORY_07             40030   ; shared memory 32 bit M167 P1000 (#5746)
HR_SHARED_MEMORY_08             40032   ; shared memory 32 bit M167 P1000 (#5747)
HR_VIRTUL_INPUTS                40034   ; virtual inputs 1 to 16 (mask)
                                        ;   b0 = virtual inputs 1
                                        ;   b1 = virtual inputs 2
                                        ;   ...
                                        ;   b15 = virtual inputs 16
```

Notes

1. To guarantee the atomicity of the LONG data (32bit), the CNC set busy status during the internal updating of the registers. If a MODBUS® MASTER tries to access the information as long as the SLAVE is in the busy state, it will obtain the BUSY state and must remain in polling so that the SLAVE is not unlocked.
2. If you want to use holding registers data you have to set “Enable write operations” in Board settings panel.
3. Take care that shared memory is retentive so PLC have to context values with CNC states and other informations.

Limitation

To optimize the internal data retrieval operations, the SLAVE supports the multiple reading of INPUT register and HOLDING register for a maximum of 32 registers at a time.