Host Modbus API

Modbus Master API is implemented in myCNC software.

Host computer with myCNC able to read **Input registers** and write data **Holding registers** from/to modbus Slave device.

User API is available through the Software PLC.

Register Name	Address	Description
GVAR_XMODBUS_SERVER_ADDR	5750	Writing to this register sets **Modbus Server Address**
GVAR_XMODBUS_VALUE	5751	Writing to this register sets a **Value** that will be sent while **write to Modbus Server**
GVAR_XMODBUS_RW_READY	5752	
GVAR_XMODBUS_READY_READ	5753	Ready Read flag. The value is automatically cleared ion start of Modbus read operation. If Read is finished successfully, the value is set to “1”. If Read operation is failed, the value is set to “-1”.
GVAR_XMODBUS_WRITE	5755	Writing to this register initiates Modbus Write operation. Written value is Modbus Server Register address (for Write). A value defined in register 5751 will be sent
GVAR_XMODBUS_READ	5756	Writing to this register initiates Modbus Read operation. Written value is a Modbus Server Register address (for Read). Writing to this register will clear READY_READ register (5753). The READ_READY register value will be set to “1” when Read completed successfully. The READ_READY register will be set to “-1” if Read Error happens
GVAR_XMODBUS_ADDR2READ	5761	Writing to this Register sets **Address of Global register** to put a result of **Modbus Read** operation
GVAR_XMODBUS_READ_INPUT_BITS	5763	Read the input bits. This will read the digital inputs/outputs in the
GVAR_XMODBUS_READ	5764	Read Modbus register (specify the particular register to read such as 5764,0)

Example:

- Need to write a value of “999” to Modbus Server with Address #7 to register #25,
- then read register #24 and save it in global variable register #500

```
//Write
  gvarset(5750,7);  //Set Modbus Server Address
  gvarset(5751,99); //Set a value to write
  gvarset(5755,25); //Write to Modbus regsiter #25

//Read
  gvarset(5752,500); //Set Register #500 to receive Modbus Read result
  gvarset(5756,24);  //Read Modbus register #24

  do
  {
    //wait ready_ready
    timer++;
  }while(gvarget(5753)==0);

  if (gvarget(5753)<0)
  {
    exit(99); //Modbus Ready Error
  };

  a=gvarget(500); //Read modbus result can be used by reading redister #500
```

Another read/write example for WP9038ADAM with digital inputs/outputs and analog inputs

The analog inputs in this example are the holding registers, while the bits are the digital inputs and outputs which are dealt with through read/write commands. The code is using a 3 ms delay in order to loop the timer and to allow the Modbus device to receive/send all the necessary commands.

```
main()
{


//Write
  gvarset(5750,34);  //Set Modbus Server Address (device ID is set to be 34
in this case)
  gvarset(5752,1);  //Indicator whether the operation is complete (0 for
complete, 1 for not complete)

  timer=0;

  address=500; //the variable address is specified to the 500
  gvarset(5761,address);  //Address to store read value

count=0;
  do
  {
  gvarset(5761,address);  //Address to store read value

  gvarset(5751,count); //Set a value to write (selects which value will be
written)
  gvarset(5756,0); //Set a value to write (command to write the value that
was set)

  timer=3;do{timer--; }while(timer>0); //0.3 ms timer to loop

// Read Bits
```

```
  gvarset(5763,0);  //Read Modbus register #0
  timer=3;do{timer--; }while(timer>0);

// Read Holding Registers

  gvarset(5761,address+1);  //Address to store read value will be variable
501
  gvarset(5764,0);  //Read Modbus register #0
  timer=3;do{timer--; }while(timer>0);

// Read Holding Registers

  gvarset(5761,address+2);  //Address to store read value will be variable
502
  gvarset(5764,1);  //Read Modbus register #1
  timer=3;do{timer--; }while(timer>0);

// Read Holding Registers

  gvarset(5761,address+3);  //Address to store read value will be variable
503
  gvarset(5764,2);  //Read Modbus register #2
  timer=3;do{timer--; }while(timer>0);

// Read Holding Registers

  gvarset(5761,address+4);  //Address to store read value will be variable
504
  gvarset(5764,3);  //Read Modbus register #3
  timer=3;do{timer--; }while(timer>0);

count++;
gvarset(507,count); //specifies the value for the 507 address

   }while(1);

 exit(99);
};
```

The specifications for the Modbus devices that are currently sold through the myCNC shop can be found here: Modbus Devices available

From:
http://cnc42.com/ - **myCNC Online Documentation**

Permanent link:
http://cnc42.com/mycnc/host_modbus_api

Last update: **2019/08/01 15:26**