2x16 IO Eth V4.1Programming Manual

Features:

- Communication over Ethernet
- TCP/UDP server
- Configurable IP address
- Human readable commands
 and messages
- Extension board support
- Grouped output control
- I2C, RS485 protocol translator

Preparing the software tools:



The Ethernet connection is implemented with a WIZnet W7500P-S2E IC. The related overview and datasheet are available on the following links:

W7500P overview

W7500P-S2E Datasheet [En]

Updating/flashing the application firmware on the W7500P can be done either through the ISP header, either with the help of a preloaded bootloader (if present in the chip's memory).

 The ISP update is done through a WIZnet UART header on the 2x16IO Card PCB (highlighted with blue in the next picture) and the W7500 ISP Tool for flashing the W7500P-S2E. The binary file, the software tool and documentations can be downloaded from the following links:

Firmware Binary (WIZISP – boot and application firmware) V 1.3.3				
W7500 ISP Tool				
App. hints - How to use ISP Tool				





2x16 IO Eth V4.1Programming Manual

Elle ISP Help	Ic		
Step 1 - Serial Option		Step 2 - Erase	Step 3 - Code Read Lock or Data R/W Lock
Serial Port Baud Rate	COM4 V Refresh	 Erase All Data/Code Memory Erase All Code Memory 	□ All Code Read Lock/Data Read Write Lock □ All Code Read Unlock/Data Read Write Unlock
Open	Close	Erase Data0(0x0003FE00) Erase Data1(0x0003FF00)	Code Read Lock All Code Write Lock Data1 Read Lock Data0 Read Lock
		Start Step2	Data1 Write Lock Data0 Write Lock
ļ			Start Step3
		Step 5 - Select the bir	Browse
		19 18 17 16 Verify after program	imming 🔽 Write MainFlash 🗌 Write DataFlash 🕓 Reset
GN	ISP BOC		Start Step5
			Start All Steps

2. A regular firmware update (when the bootloader is present in the chip's memory) is done by a special software tool provided by the WIZnet manufacturer. The binary file, the software tool and documentations can be downloaded from the following links:

Firmware Binary (application firmware) V 1.3.3
WIZnet-S2E-Tool-GUI V 1.5.4
App. hints - How to use WIZnet S2E Tool
Commands for configuring the WIZnet circuit [En]

Configuring the 2x16IO Card:

The configuration tool software is used to configure the 2x16IO Card's TCPIP/UDP settings in four mandatory steps, followed by pressing the [Apply Settings] button on the GUI:

 Once powered and plugged in to an Ethernet network a search is initiated for the 2x16IO Card on the network. This can be performed with the [Device Search] button on the config tool's GUI. Once the search is done the 2x16IO



2x16 IO Eth V4.1Programming Manual

Card's MAC address should appear in the [Search results] list of the tool.

e Option Help							
Device Search Apply Settings	Firmware Upload	Reset Device	Factory Reset	Save Config	Load Config	\bigcirc	Exi
92.168.1.125:Realtek PCIe G ~	Basic settings Op	otions User I/O					
arched results: 1	Device informatio	in	Network	configuration			
Mac address Name	Туре	WIZ750SR	0	Static 🔿 🛛	HCP		
00:08:DC:74:9D:27 WIZ750SR	Version	1.3.3	Local IP 192.168.3		3.1.52		
			Subpo	1 mark 255 255	5 255 0		
	Search identificat	tion code	Subile	233.235	.233.0		
		Gateway 192.168.1.1			8.1.1		
	(Max. 8 bytes)		DNS se	erver 8.8.8	.8		
	Channel #0						
	Status:	OPEN	Serial options		Timer interval		
	Serial interface:	RS-232/TTL	Baud rate	115200 ~	Inactivity timer	0	se
	Operation mode		Data bit	8 ~	Reconnection interval	3000	m
	C)						
	TCP client	O SSL TCP clier	Parity	NONE ~	TCD Keen alive in	the second	
	 TCP client TCP server TCP mixed 	MQTT client	Parity Stop bit	NONE ~	TCP Keep-alive in	nterval	
arch ID code	 TCP client TCP server TCP mixed UDP 	 SSL TCP clier MQTT client MQTTS client 	Parity Stop bit Flow control	NONE ~ 1 ~	TCP Keep-alive in	terval	
arch ID code	 TCP client TCP server TCP mixed UDP 	 MQTT client MQTTS client 	Parity Stop bit Flow control	NONE ~ 1 ~ NONE ~	TCP Keep-alive in	7000	m
earch ID code D cod D cod D cod	TCP client TCP server TCP mixed UDP Local port 5000	SSL TCP clier MQTT client MQTTS client Fix port Fix port	Parity Stop bit Flow control Serial data par	NONE V I NONE V cking condition	 TCP Keep-alive in Enable Initial Interval Retry interval 	7000 5000	m
earch ID code D cod Show che earch method	TCP client TCP server TCP mixed UDP Local port 5000 Remote host / Po	SSL I CP clier MQTT client MQTTS client Fix port	Parity Stop bit Flow control Serial data par Character	NONE I NONE King condition	TCP Keep-alive in C Enable Initial Interval Retry interval	7000 5000	m
earch ID code D cod Show cha earch method O UDP broadcast	TCP client TCP server TCP mixed UDP Local port S000 Remote host / Po Remote host	SSL ICP clier MQTT client MQTTS client MQTTS client Fix port 192.168.11.3	Parity Stop bit Flow control Serial data par Character Size	NONE ~ 1 ~ NONE ~ cking condition 0D 0	TCP Keep-alive in Fnable Initial Interval Retry interval	7000 5000	m

Note: for computers connected to more networks than one, this search might not work on all networks but rather the primary one only. In that case recommendation is to configure the unit on the primary network and then switch it to the network where it will be used.

2. TheWIZ7500P communication distinguishes two different TCP/IP port referenced to the same IP address (for example 192.168.1.52 marked with red in the following picture). The 50001 port is for configuration over TCP unicast frames (marked with blue in left lower corner of the following picture), while the 5000 port is for data communication (marked with green in the following picture).



2x16 IO Eth V4.1Programming Manual

Option Help						
Device Search Apply Setting	s Firmware Upload	Reset Device	Factory Reset	Save Config	Load Config	Exif
2.168.1.125:Realtek PCIe G	 Basic settings Op 	tions User I/O				
rched results: 1	Device informatio	n	Network	configuration		
Mac address Name 00:08:DC:74:9D:27 WIZ750SR	Туре	WIZ750SR	•	Static 🔘 🕻	DHCP	
	Version	1.3.3	Local 1	P 192.16	8.1.52	
	- Search identificat	ion code	.255.0			
		Show c	Gatew Gatew	8.1.1		
	(Max. 8 bytes)	DNS server 8.8.8				
	Channel #0					
	Charmer #0					
	Status:	OPEN	Serial options		Timer interval	
	Status: Serial interface:	OPEN RS-232/TTL	Serial options Baud rate	115200 ~	Timer interval	sec
	Status: Serial interface: Operation mode	OPEN RS-232/TTL	Serial options Baud rate Data bit	115200 ~ 8 ~	Timer interval Inactivity timer 0 s Reconnection 3000 r	seo
	Status: Serial interface: Operation mode	OPEN RS-232/TTL	Serial options Baud rate Data bit Parity	115200 ~ 8 ~ NONE ~	Timer interval Inactivity timer 0 s Reconnection 3000 r	sec
	Status: Serial interface: Operation mode () TCP client O TCP server	OPEN RS-232/TTL SSL TCP clier MQTT client MOTTC client	Serial options Baud rate Data bit Parity Stop bit	115200 ~ 8 ~ NONE ~	Timer interval Inactivity timer 0 s Reconnection 3000 r TCP Keep-alive interval	sec
web TD code	Status: Serial interface: Operation mode TCP client TCP server TCP mixed	OPEN RS-232/TTL SSL TCP clier MQTT client MQTTS client	Serial options Baud rate Data bit Parity Stop bit	115200 × 8 × NONE × 1 ×	Timer interval Inactivity timer 0 s Reconnection 3000 r TCP Keep-alive interval	sec
wich ID code	Status: Serial interface: Operation mode TCP client TCP server TCP server UDP	OPEN RS-232/TTL SSL TCP clier MQTT client MQTTS client	Serial options Baud rate Data bit Parity Stop bit Flow control	115200 ~ 8 ~ NONE ~ 1 ~ NONE ~	Timer interval Inactivity timer 0 s Reconnection 3000 r Interval 3000 r TCP Keep-allve interval CP Keep-allve interval CP Keep-allve interval CP Keep-allve interval CP Keep-allve interval	sec ms
rch ID code	Status: Serial interface: Operation mode TCP client TCP client TCP mixed UDP Local port 5000	OPEN RS-232/TTL SSL TCP clier MQTT client MQTTS client Fitx port	Serial options Baud rate Data bit Parity Stop bit Flow control	115200 > 8 > NONE > 1 > NONE >	Timer interval Inactivity timer 0 s Reconnection 3000 r TCP Keep-alive interval Enable Initial Interval 7000 r Retry interval 5000 r	sec ms ms
rch ID code	Status: Serial interface: Operation mode TCP client TCP client TCP mixed UDP Local port 5000 Remote host / Po	OPEN RS-232/TTL SSL TCP clier MQTT client MQTTS client Ftx port	Serial options Baud rate Data bit Parity Stop bit Flow control Serial data pa Character	115200 8 NONE 1 NONE cking condition	Timer interval Inactivity timer 0 s Reconnection 3000 r TCP Keep-alive interval Enable Initial Interval 7000 r Retry interval 5000 r	sec ms ms
arch ID code	Status: Serial interface: Operation mode TCP client TCP client TCP mixed UDP Local port 5000 Remote host / Po	OPEN RS-232/TTL SSL TCP clier MQTT client MQTTS client Fix port rt 192 168 11 2	Serial options Baud rate Data bit Parity Stop bit Flow control Serial data pa Character	115200 ~ 8 ~ NONE ~ 1 ~ NONE ~ cking condition	Timer interval Inactivity timer 0 s Reconnection 3000 r TCP Keep-alive interval Enable Initial Interval 7000 r Retry interval 5000 r	sec ms ms

3. For the WIZ7500P and MCU communication the serial port parameters should be set to 115200baud with 8 data bits, without parity, using 1 STOP bit and without Flow Control.

Serial options			Timer inter
Baud rate	115200	~	Inactivity t
Data bit	8	\sim	Reconnect interval
Parity	NONE	~	TCP Keep-a
Stop bit	1	~	Enable
Flow control	NONE	~	Initial Inte
-Serial data pack		Retry inte	
Character	0D		

4. The last mandatory setting is to configure the [Expansion GPIO] ports of the WIZ7500P to ensure that the PIC MCU is released from reset. For this [GPIO A] should be selected as [Digital Output] and set to [Low] value. As a new feature



2x16 IO Eth V4.1Programming Manual

the system MCU can be remotely reset through the Ethernet interface either using direct WIZnet commands sent to the configuration port (default at 50001) or by using the configuration tool GUI (WIZnet S2E Configuration Tool V 1.5.4). Resetting the MCU is done by setting GPIO A first to High than to Low level.

Firmwar Upload	re Reset Device	Į.	Factor Reset	y 🖵	Ū	Save Config
Basic settings	Options User I/0	2				
Refresh time	() 1 sec () 5 s	ec C	10 sec) 30 se	c	
Expansion GPI	0					
GPIO A	Digital Output 🛛 🗸	Low	\sim	0		
GPIO B	Digital Input \sim	Low	\sim	1		
GPIO C	Digital Input \sim	Low	\sim	1		Digital input: Read only
GPIO D	Digital Input \checkmark	Low	\sim	1		Analog input: R/W

Note: As the last final step don't forget to press the [Apply Settings] button on the GUI.



Once the connection and hardware setup is done for the 2x16IO Card, the device will be available to respond over Ethernet network with using any TCP/IP (or UDP) tool able to transmit and receive data packages in the required format. For example, the Hercules software tool or Putty console are both handy tools for this purpose:





2x16 IO Eth V4.1Programming Manual

IO Card Protocol

The protocol is human readable, uses ASCII coded characters and each message ends with a Carriage Return (CR) character.

Every valid response starts with ">" character. If an error occurs or invalid command was entered the protocol returns a "!" character.

NOTE: some commands are firmware version dependent (for this you will see something like v5.0+ meaning that the command works from firmware 5.0 and up.

For connection, the previously configured IP address and Port number should be used. After connecting to the IO Card over TCP/IP and sending the basic **VER** command followed by a Carriage Return character (VER<CR>) a valid response should be got from the IO Card containing the firmware version.

😵 Hercules SETUP utility by HW-group.com – 🗆 🗙						
UDP Setup Serial TCP Client TCP Server UDP Test Mode About						
Received/Sent data		TCP				
Connecting to 192.168.1.52		Module IP	Po	rt		
VER>VER:5.00		192.168.1.52	50	00		
		Ping	🗶 Dis	connect		
		TEA authorizati	on			
		TEA key				
		1: 01020304	4 3: 090A0	DBOC		
		2: 05060708	3 4: OD 0E	0F10		
		Authorization c	ode			
				a		
	Г	PortStore test				
		🖂 NVT disabl	е			
		Recei	ved <u>t</u> est data			
<u> </u>	Г	Redirect to L	IDP			
Send						
VER <cr></cr>		Send	1Wgr	oup		
RS485WRHEX 09 67 65 74 66 77 76 65 72 0A <cr></cr>	HEX	Send	www.HW-gro	up.com		
RS485GETBAUD <cr></cr>	HEX	Send	ercules SETU Version	IP utility 3.2.8		



Command		Description		
VER <cr></cr>	Get firn	nware version		
Evample:	Send	VER <cr></cr>		
example.	Receive	>VER:5.00		
INO <cr></cr>	Get Input states Main Board			
Evampla:	Send	IN0 <cr></cr>		
example.	Receive	>IN0:00000000001000		
IN1 <cr></cr>	Get Input states Extension Board 1			
Evample:	Send	IN1 <cr></cr>		
example.	Receive	>IN1:000000000001000		
IN2 <cr></cr>	Get Inp	put states Extension Board 2		
Evample:	Send	IN2 <cr></cr>		
example.	Receive	>IN2:00000000001000		
OUTnn s <cr></cr>				
nn - channel number(116) s - state (0,1)	Set ON	/OFF digital output		
Evample	Send	OUT01 1 <cr></cr>		
LXUMPIE.	Receive	>OUT01 1		
PWRn s <cr></cr>				
n - channel number(12) s - state (0,1)	Set ON	/OFF High-Power output		
Evample:	Send	PWR1 1 <cr></cr>		
example.	Receive	>PWR1 1		
INA <cr></cr>	Get an	alog Data		
Example:	Send	INA <cr></cr>		
LXUMPIE.	Receive	>INA:1952 1955 1981 2007		
IND <cr></cr>	Get all digital input values at once from mainboard and from the two extensio boards			
Example:	Send	IND <cr></cr>		
	Receive	>IND:0 32 32 32 0 01		
Note:	Respor (8 bits	each) for the 16+16+16 inputs		



Command	Description				
	(the values should be the same reported by INO, IN1, IN2)				
CLEAR <cr></cr>	Clear all outputs				
Example:	Send CLEAR <cr></cr>				
Liumpie.	Receive > CLEAR				
SETBYMASK 1234 5678 9ABC [FFFF FFFF FFFF] <cr></cr>	Set all outputs at once (v4.2+)				
Example:	Send SETBYMASK 10 10 10 10 10 10 10 10 10 10 10 10 10				
	Receive >SETBYMASK 0010 0010 0010				
Note:	First 3 mandatory parameters: hexadecimal output values for a base card and two extensions (1st bit is OUT1). Second 3 optional parameters: masks considered 0xFFFF by default (permits setting only the selected bits in the masks instead of all bits). The response will contain the three output registers for card and two extensions after the input values were applied. All parameters and the response are in hexadecimal.				
GETOUT <cr></cr>	Get all outputs at once (v4.2+)				
Example:	SendGETOUT <cr>Receive>GETOUT 0010 0010 0010</cr>				
Note:	Response contains all outputs in one response (hexadecimal).				
	The values are the last set ones, there is no HW support to measure back the real digital outputs.				
SETLATCH ddd <cr></cr>	Setting latches				



Command		Description		
ddd - hexadecimal 64 bit number (with swapped LSB order comparing to INO - least significant 1st), therefore 1st byte is IN1- IN8, IN9-IN16, IN17-24, IN25- 32 etc the number is 64byte to support 2nd extension too.				
Example for latching IN1:	Send	SETLATCH 010000000000000CR>		
Note:	This command also resets curr latches (used to reset the latched p at the start of every sequence).			
I2CEXT [Fffff] [Waadddd] [Raann] <cr> ffff – I2C clock frequency in kHz W - write, R - read, F – I2C clock frequency in kHz aa - slave I2C address dddd - data to write nn - number of bytes to read</cr>	Generio	c write and Read I2C Port		
Example for reading 2 bytes from all I2C	Send	I2CEXT W2000 R2002 W2100 R2102 W2200 R2202 W2300 R2302 <cr></cr>		
expanders present on the I2C input extension:	Receive	>I2CEXT W2000 R200000 W2100 R210000 W2200 R220000 W2300 R230000		
Example for writing the 12C expanders present	Send	I2CEXT W20020000 W21020000 W22020000 W23020000 <cr></cr>		
on the 12C output extension:	Receive	>12CEXT W2000 R200000 W2100 R210000 W2200 R220000 W2300		



Command	Description			
		R230000		
Example for changing	Send	I2CEXT F100 W20020000 <cr></cr>		
frequency and write two bytes to register 02:	Receive	>I2CEXT F03E W20020000		
Note:	The F03E response for F100 contains f BRG register value (3E in this case). The value depends on MCU clo frequency (12.8MHz) ex.: I2CEXT F1 W2400 R2402 W25 R2502 W2600 R2602 W2700 R2702 <cr >I2CEXT W20020000 W200600 W21020000 W21060000 W220200 W22060000 W23020000 W23060000</cr 			
PING <cr></cr>	Ping su	pport		
	Send	PING <cr></cr>		
example:	Receive	>PONG		
RS485SETMODE [mode] <cr></cr>	Setting mode	RS485 receiver communication		
Example:	Send	RS485SETMODE 1 <cr></cr>		
	Receive	> RS485SETMODE OK <cr></cr>		
Note:	The RS4	185 receiving communication		
	mode can be :			
	interpre termino Data sh commo	text (mode=0) data received is eted as text, communication is ated by <lf> character ('\n'). hould be read by the RS485TEXT and.</lf>		
	interpre	hex (mode=1) data received is eted as hex, the received buffer		



Command	Description		
	is presented as whatever is received at the moment when it is read by the ethernet interface with the RS485RDHEX command.		
	Default value is text mode.		
RS485GETMODE <cr></cr>	Getting RS485 receiver communication mode		
Example:	Send Receive	RS485GETMODE <cr> > RS485GETMODE 1<cr></cr></cr>	
Note:	Default	value is text (mode=0).	
RS485SETBAUD [baudrate] <cr></cr>	Set the desired RS485 baudrate.		
Example:	Send Receive	RS485SETBAUD 115200 <cr> > RS485SETBAUD 117647<cr></cr></cr>	
Note:	Response contains the actual baudrate set and calculated. Default baud rate is 115200.		
RS485GETBAUD <cr></cr>	Get the	RS485 baudrate.	
Example:	Send Receive	RS485GETBAUD <cr> > RS485GETBAUD 117647<cr></cr></cr>	
Note:	Response contains the actual baudrate set and calculated. Default baud rate is 115200.		
RS485WRTEXT [message] <cr></cr>	Write text message on RS485		
Example:	Send Receive	RS485WRTEXT getresistance <lf><cr> >RS485WRTEXT OK<cr></cr></cr></lf>	
Note:	This example sends a request to the RDECADE24BIT unit connected on		



Command	Description		
	RS485 to the IOCard to query the actual resistance setting. Note that RDECADE24BIT unit uses <lf> ('\n')for message termination.</lf>		
RS485RDTEXT <cr></cr>	Read text message on RS485		
Example:	Send Receive	RS485RDTEXT <cr> >RS485RDTEXT OK getresistance 1666665<lf><cr></cr></lf></cr>	
Note:	This example gets the response sent from the RDECADE24BIT unit connected on RS485 to IOCard. The response is stored in the IOCard buffer. Note that RDECADE24BIT unit uses <lf> ('\n')for message termination.</lf>		
RS485WRHEX			
[length][msg] <cr> Example:</cr>	Write he Send Receive	exadecimal message on RS485 RS485WRHEX 09 67 65 74 66 77 76 65 72 0A <cr> >RS485WRHEX OK<cr></cr></cr>	
Note:	length – hexadecimal value, contains the length of the following message bytes message bytes – the bytes as text (2 character) separated by space character The result of this message on RS485 would be to send 9 bytes "getfwver\n"		
RS485RDHEX <cr></cr>			



Command	Description	
	Write hexadecimal message on RS485	
Example:	Send	RS485RDHEX <cr></cr>
	Receive	>RS485RDHEX 15 4F 4B 20 67 65
		74 66 77 76 65 72 20 31 2E 30 30
		30 30 30 30 0A <cr></cr>
Note:	first byte (hexadecimal representation) in the response is the length of the following message bytes. The result of the received message on RS485 would be that it received 21 bytes "OK getfwver 1.000000\n"	