

## Introduction

The new H4TH words to communicate with the WIZ chip are described below. The TCP/IP commands can handle just one socket, which has the advantage that it doesn't have to be specified ;-)

## Accessing WIZ's Memory

---

WZ!	Stores two byte items into the WIZ memory locations addressed by the top stack item.
WZ@	Fetches the value (two bytes) at the WIZ memory locations addressed by the top stack item, and places it on the stack.
CWZ!	Stores one byte item into the WIZ memory location addressed by the top stack item.
CWZ@	Fetches the value (one byte) at the WIZ memory location addressed by the top stack item, and places it on the stack.
WIR@	Fetches the value of the Interrupt Register, and places it on the stack.
WIR!	Resets the bits of the interrupt register as present in the top stack item.
WSR@	Fetches the value of the Status Register, and places it on the stack.

---

Note: The latter three are abbreviations in the sense that the address doesn't have to be specified.

## Initialization

---

WZINIT Resets the WIZ and its settings to defaults. This may be done by the extended ROM automatically. If the reset is unsuccessful, ERROR W is given.

---

## WIZ Configuration

---

HW!	Sets the hardware address to the top six stack items.
IP!	Sets the IP address to the top four stack items.
SM!	Sets the subnet mask to the top four stack items.
GW!	Sets the gateway's address to the top four stack items.
HW?	Shows the configured hardware address.
IP?	Shows the configured IP address.
SM?	Shows the configured subnet mask.
GW?	Shows the configured gateway address.

---

Note: HW?, IP?, SM?, and GW? values are not placed on the stack.

The hardware address is initialized by WZINIT to the string "ZX81 ". This is a locally administered address. The last two bytes may be randomized by using: TIME D@ D WZ!

The subnet masked is initialized by WZINIT to 255.255.255.0.

So the minimal required setting is the IP address. To communicate with the outside world, also the gateway must be set.

## TCP/IP Commands

---

OPEN	Opens a socket.
PORT	Sets the local port to the top stack (two-byte) item (WZINIT initializes it to 8181).
DPORT	Sets the destination port to the top (two-byte) stack items.
DIP	Sets the destination IP address to the top four stack items.
LISTEN	Waits for a connection to the local port. A socket is opened if needed.
CONNECT	Connects to the address specified by DIP, to the port specified by DPORT, using the local port specified by PORT. A socket is opened if needed.
DISCON	Sends a connection termination request.
CLOSE	Closes the socket.
SEND*	Sends an amount of data as specified by the top stack item (done automatically by C>W and ED>W, see below).
RECV*	Process the fact that data has been received (done automatically by W>C and W>ED, see below).

---

Note: The above commands do not only send the command to the WIZ, but also check if it is possible and results in the desired state. (\*) SEND and RECV are currently not in the dictionary as the commands below use these automatically.

## Interfacing with H4TH Data

---

C>WZ	Moves a character string to WIZ's transmit buffer, and sends it.
WZ>C	Waits for data and moves it from WIZ's receive buffer to the character stack.
ED>WZ	Copies the data in the editor to WIZ's transmit buffer, and sends it.
WZ>ED	Waits for data and moves it from WIZ's received buffer to the editor.

---

## Higher Level Commands

---

PING	Sends PING request to the address specified by DIP.
GET	Attempts to get a file from the server specified by DIP. The file name is copied from the editor Pad, placed there by C>B (see below) or Shift/3.

---

## Handy Ones

---

CR>C	Places a carriage control (CR) and line feed (LF) on the character stack (handy for HTML header lines).
C>B	Copies the character string to the editor Pad.

---