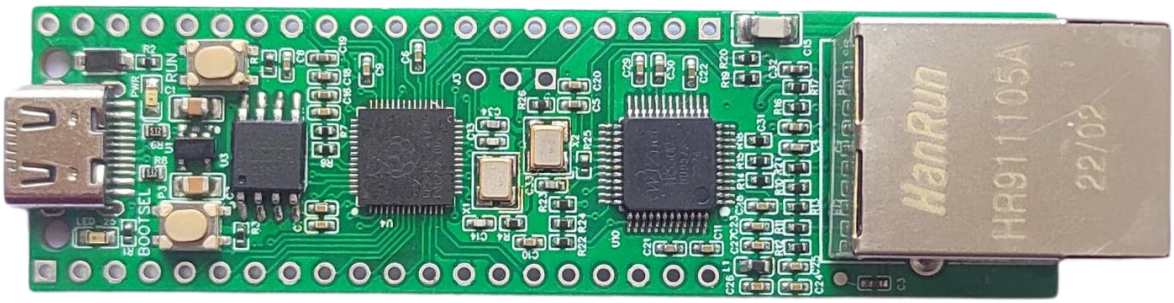
ADIY W5500 EVB PICO RP2040



Description:

ADIY FLY RP2040 Basic Board with Ethernet is a microcontroller evaluation board based on the Raspberry Pi RP2040 and fully hardwired TCP/IP controller W5500 – and basically works the same as Raspberry Pi Pico board but with additional Ethernet via W5500.W5500-EVB-Pico pinout is directly connected to the GPIO of RP2040 .It has the same pinout as the Raspberry Pi Pico board. However, GPIO16, GPIO17, GPIO18, GPIO19, GPIO20, GPIO21 are connected to W5500 inside the board. These pins enable SPI communication with W5500 to use Ethernet function. If you are using the Ethernet function, these pins cannot be used for any other purpose.

The W5500 chip is a Hardwired Internet controller designed as a full hardwired TCP/IP stack with WIZnet technology. W5500 provides Internet connectivity to your embedded systems by using SPI(Serial Peripheral Interface). SPI provides easy connection via external MCU to W5500. The clock speed of W5500 SPI supports up to 80MHz.

Since W5500 integrates the Hardwired TCP/IP stack with 10/100 Ethernet MAC and PHY, it is truly a one-chip solution for the stable internet connectivity. WIZnet's Hardwired TCP/IP stack supports TCP, UDP, IPv4, ICMP, ARP, IGMP, and PPPoE - and it has been proven through various applications over the last decade.

W5500 provides 8 independent SOCKETs to be used simultaneously and 32KB internal memory for data communication. Users can develop an Ethernet application easily by using the simple W5500 SOCKET program instead of handling a complex Ethernet controller. W5500 also provides WOL (Wake on LAN) and a Power Down Mode in order to reduce power consumption.

Features:

1. RP2040 microcontroller with 2MByte Flash
2. Dual-core cortex M0+ at up to 133MHz

3. 264kByte multi-bank high performance SRAM
4. External Quad-SPI Flash with eXecute In Place (XIP)
5. High performance full-crossbar bus fabric
6. 30 multi-function General Purpose IO (4 can be used for ADC)
7. 1.8-3.3V IO Voltage (NOTE. Pico IO voltage is fixed at 3.3V)
8. 12-bit 500 ksps Analogue to Digital Converter (ADC)
9. Various digital peripherals
10. 2 × UART, 2 × I2C, 2 × SPI, 16 × PWM channels
11. 1 × Timer with 4 alarms, 1 × Real Time Counter
12. 2 × Programmable IO (PIO) blocks, 8 state machines total
13. Flexible, user-programmable high-speed IO
14. Can emulate interfaces such as SD Card and VGA
15. Includes W5500
16. Supports Hardwired Internet Protocols: TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
17. Supports 8 Independent Hardware SOCKETs simultaneously
18. Internal 32 Kbytes Memory for TX/ RX Buffers
19. Supports High Speed Serial Peripheral Interface(SPI MODE 0, 3)
20. USB type C port for power and data (and for reprogramming the Flash)
21. 40 pin 21x51 'DIP' style 1mm thick PCB with 0.1" through-hole pins also with edge castellation
22. 3-pin ARM Serial Wire Debug (SWD) port
23. 10 / 100 Ethernet PHY embedded
24. Supports Auto Negotiation
25. Full / Half Duplex
26. 10 / 100 Based
27. Built-in RJ45
28. Built-in LDO

W5500 TCP Function

By setting some register and memory operation, W5500 provides internet connectivity.

Initialization

You must check PHY LINK (0 bit of PHYCFGR) before attempting to make a network connection using sockets.

Basic Setting

For the W5500 operation, select and utilize appropriate registers shown below.

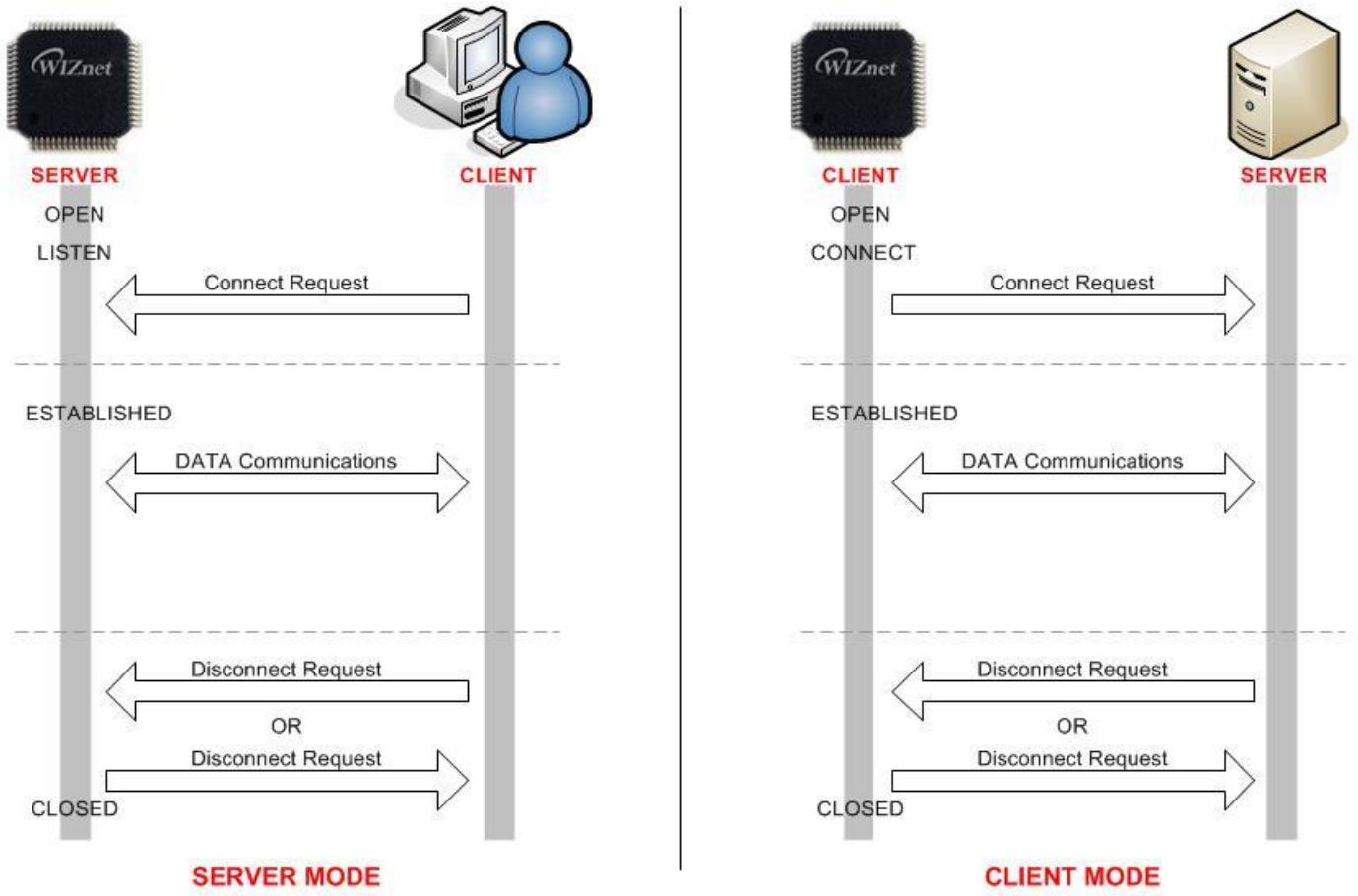
1. Mode Register (MR)
2. Interrupt Mask Register (IMR)
3. Retry Time-value Register (RTR)
4. Retry Count Register (RCR)

Data Communications

After the initialization process, W5500 can transmit and receive the data with others by 'opening' the SOCKET of TCP, UDP, IPRAW, and MACRAW mode. The W5500 supports the independently and simultaneously usable 8 SOCKETS. In this section, the communication method for each mode will be introduced.

TCP

The TCP is a connection-oriented protocol. The TCP makes the connection SOCKET by using its own IP address, port number and destination IP address, port number. Then transmits and receives the data by using this SOCKET. Methods of making the connection to SOCKET are "TCP SERVER" and "TCP CLIENT". It is divided by transmitting the connect-request (SYN packet). The "TCP SERVER" listens to the connect-request from the "TCP CLIENT", and makes connection SOCKET by accepting the transmitted connect-request (Passive-open). The "TCP CLIENT" transmits the connect-request first to "TCP SERVER" to make the connection (Active-open).



Application:

- Industrial ethernet based applications
- To display information via LAN network
- Development board