Contents

[Introduction: 2](#_Toc93993361)

[HARDWARE: 2](#_Toc93993362)

[PCB 18.16 2](#_Toc93993363)

[SPI0 3](#_Toc93993364)

[Ethernet 3](#_Toc93993365)

[UART: 3](#_Toc93993366)

[RS-485 unit 3](#_Toc93993367)

[SCHEMATIC 3](#_Toc93993368)

[SOFTWARE 3](#_Toc93993369)

[IDE / Language 3](#_Toc93993370)

[MODBUS 4](#_Toc93993371)

[Modbus RTU 4](#_Toc93993372)

[Modbus RTU Address 4](#_Toc93993373)

[TEST CASE 4](#_Toc93993374)

[Modbus TCP 4](#_Toc93993375)

[Modbus TCP Address 4](#_Toc93993376)

[TEST CASE 4](#_Toc93993377)

[LIBRARIES 5](#_Toc93993378)

[RTOS 5](#_Toc93993379)

# Introduction:

The PCB18.16 is a sensor board with some GPIO’s and a 7-segment display.

The board uses a hybrid RP2040 development board for IO’s, this is the wiznet w5100s-evb-pico, it is identical for all intentional purposes to the RP2040 board with the exception that is has ethernet on SPI1.

Visual studio example:

<https://docs.wiznet.io/Product/iEthernet/W5100S/w5100s-evb-pico>

Arduino example:

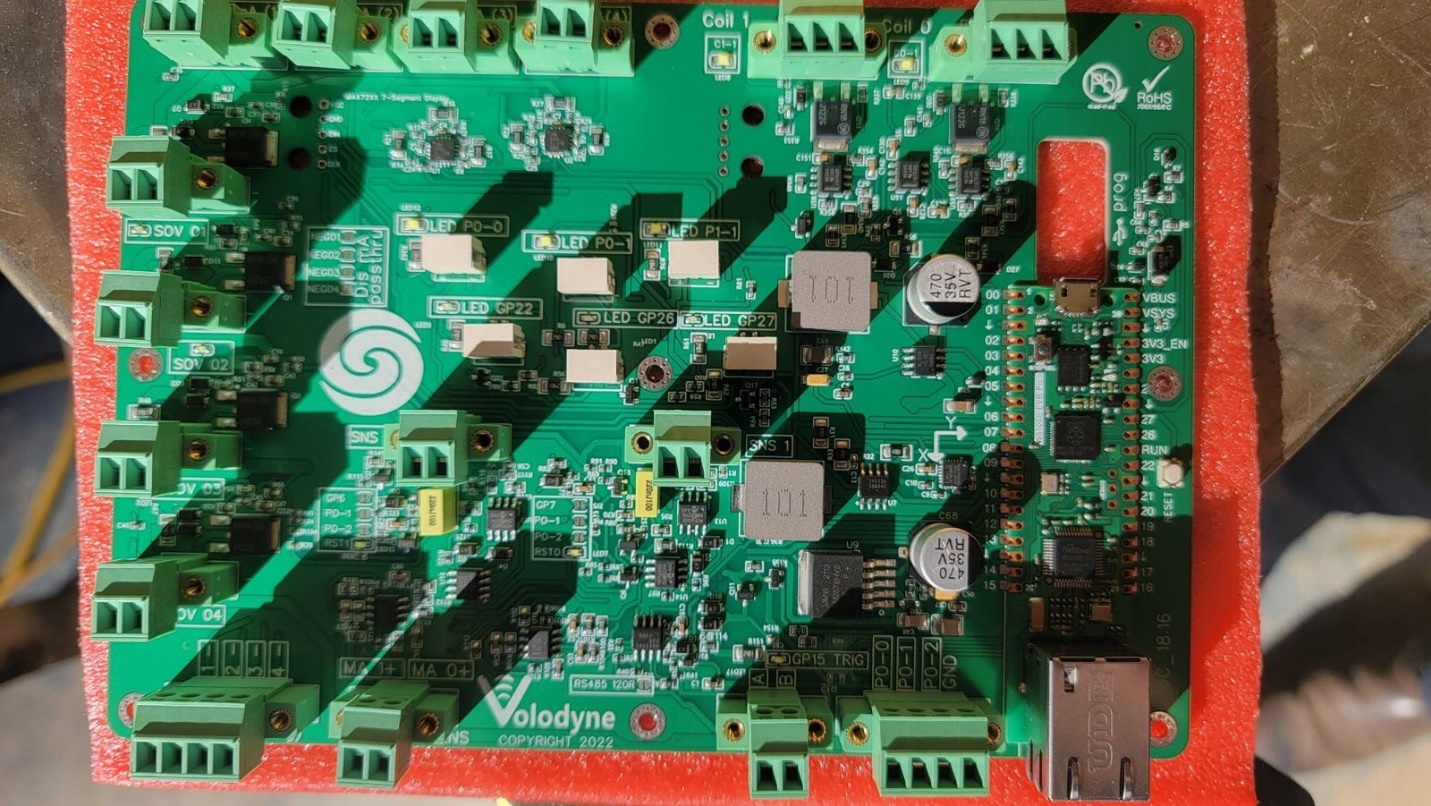
<https://maker.wiznet.io/2021/11/29/w5100s-evb-pico-with-the-arduino-ide/>

Sample code for the ethernet board:

<https://github.com/WIZnet-ArduinoEthernet/Ethernet/archive/master.zip>

# HARDWARE:

# PCB 18.16



A PCB was developed using the Wiznet w5100s-evb-pico board, the PCB 18.16 has the following,

## SPI0

### Ethernet

Model: W5100s

The W5100s ethernet module from wiznet is incorporated into the W5100s-EVB-PICO board and uses SPI0

## UART:

GPIO4 - DI, GPIO5 - RO, GPIO14 - DE

### RS-485 unit

Model:MAX3485ESA

RE/DE: GP4

RO: GP5

DI: GP14

RS-485 is used for Modbus RTU, the Modbus function is described later.

### SCHEMATIC

A schematic is attached Schematic 18.16.

# SOFTWARE

## IDE / Language

The preferred IDE is VS Code as recommended by the PI Organization with C++ code.

Arduino is supported and many examples exist written using the Arduino IDE, if it is not possible to use the IDE then the Arduino IDE can be used. Provided that the code is documented accordingly.

# MODBUS

## Modbus RTU

Using the MAX485 chip and a Modbus library the firmware should report coil status, report registers from sensors and software variables and be able to be writer to coils and registers from a master PLC over RS485.

### Modbus RTU Address

The Modbus RTU address shall be static and set in firmware, the address shall be periodical displayed on the display.

### TEST CASE

For the initial test case global variables will be set on the RP2040 for bits and integers. These values will be displayed via serial from the while loop and be accessible as read coils / registers in Modbus. When the PLC reads the Modbus value and output to the serial shall print the activity.

For write coils and registers the serial shall print Modbus function and the value of the coil register

## Modbus TCP

Using the W5100s ethernet chipset and a Modbus library the firmware should report coil status, report registers from sensors and software variables and be able to be writer to coils and registers from a master PLC over TCP.

### Modbus TCP Address

The Modbus TCP address shall be static in the range of 192.168.1.2 – 192.168.1.100, subnet 255.255.255.0 and set in firmware, the address shall be periodical displayed on the display.

### TEST CASE

For the initial test case global variables will be set on the RP2040 for bits and integers. These values will be displayed via serial from the while loop and be accessible as read coils / registers in Modbus. When the PLC reads the Modbus value and output to the serial shall print the activity.

For write coils and registers the serial shall print Modbus function and the value of the coil register

# LIBRARIES

Libraries should be installed for Modbus TCP and RTU and tested, where possible functions should be written to seamlessly read sensors, read EEPROM, write EPPROM. This can be expanded further.

# RTOS

RTOS or dual core application will be considered once the Bare Metal application has been completed.