

The general information of WF1 Modules

- The general information of WF1 Modules
 - Module's Pin Map
 - Notes
 - Update the firmware
 - Update the firmware of the module
 - Other methods to update the firmware
 - I2C for modules
 - Read module information by host
 - Module has new information, host needs to read immediately
 - UART
 - USB Device Stack
 - UART To USB
 - SPI
 - Universal Registers
 - I2C Protocol
 - Name definition
 - Document Convention
 - Registers Table
 - Module No
 - High 8 Bit
 - Low 8 Bit
 - Number of Registers
 - Functions Used
 - Pin Status
 - Pin not used (0b00)
 - Pin was connected, but not used (0b01)
 - Pin is used, and can be modified by the host (0b10)
 - Pin is used, and cannot be modified by the host (0b11)
 - Hardware Version
 - Software Version
 - Checksum
 - Interrupt Triggered
 - Module Name

RIMU's host and modules primarily communicate via the I2C bus, although there are also some special modules. The specific module selection depends on factors such as the required functionality, cost, protection level, and application scenario.

This document focuses on customizable modules that use a custom I2C protocol, such as the DIMMER5 module and the 8-channel IO expansion module.

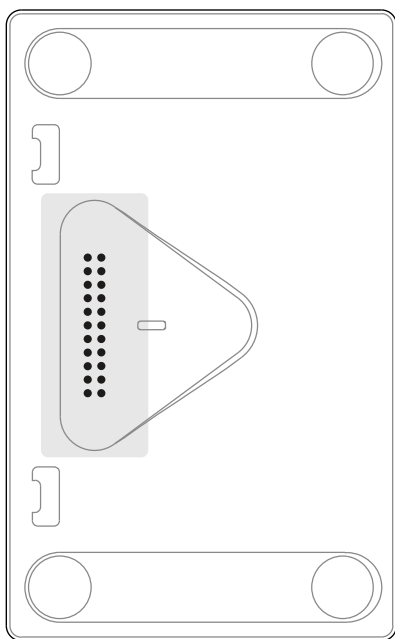
Currently, all modules use the Atmega328p as the main processor. The chip handles as many events as possible to reduce the host's IO usage. Communication between the modules and the host is done using the I2C protocol

for tasks like controlling IO ports and reading sensor data. If the Atmega328p is unable to meet the requirements, the host's IO ports can be used instead. However, the IO ports can be switched, allowing the host to support multiple devices.

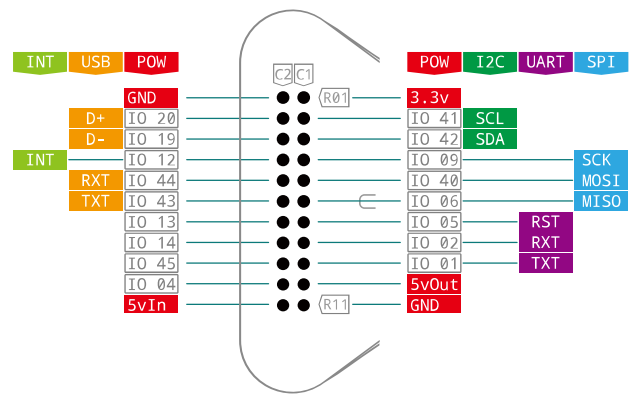
Module's Pin Map

Different modules may use different pins, and we try to use the module's own pins when designing. This reduces the host's IO usage.

	R01	R02	R03	R04	R05	R06	R07	R08	R09	R10	R11
C1	3.3v ¹	IO41	IO42	IO09	IO40	IO06	IO05	IO02	IO01	5V ²	GND
C2	GND	IO20	IO19	IO12 ⁴	IO44	IO43	IO13	IO14	IO45	IO04	5V ³



RIMU Modules' Pinout

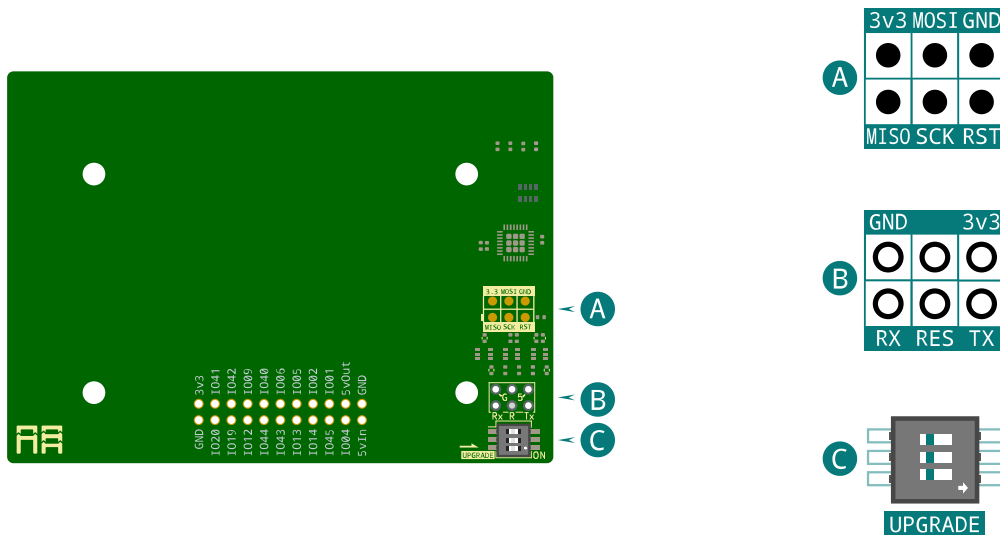


1. 3.3v(C1/R01), 5V voltage from the host, current: 2A(max).
2. 5vOut(C1/R10), The module's power charges the host's battery.
3. 5vIn(C2/R11), 5V voltage from the host, current: 200mA(max).

Notes

1. Maximum output current: 2A with host output.
2. The module's power is supplied by the battery of the host.
3. Maximum output current: 200mA with host output.
4. IO12 is used as an interrupt pin for the host, and the peripheral devices on the host are controlled by the atmega328p processor. When the processor has completed the task, it will send a high signal to notify the host. If the module has processed the signal, it will also send a high signal to notify the host. The host can then read the corresponding register to obtain this information.

Update the firmware



Update the firmware of the module

1. Use a screwdriver to open the module, refer to the diagram C-"UPGRADE" position, and set all three switches to ON position.
2. Put the module's cover on (after the shell is installed, the module can communicate with the host).
3. Click the "update" button on the screen.
4. After the update is successful, remove the module's cover and set the switches back to OFF position.
5. Put the module's cover back on, and the module can communicate with the host again.

Other methods to update the firmware

1. Use a USB-to-UART module to update the firmware. Since the program already has the Arduino firmware, it can be used to update the module's firmware. Refer to the B diagram.
2. Use a USB-ISP module to update the firmware. The USB-ISP module can be used to update the module's firmware. Refer to the A diagram.

I2C for modules

A host can connect one or more modules, and communicate between the host and the modules using the I2C bus. There are two communication methods, user operation through the screen, or the host sending commands to the module and the module reading information.

Read module information by host

The host periodically reads the module information, and if the amount of data to be read is greater than 1, the host needs to shorten the reading interval. Conversely, if the host reads multiple times without any change, the host needs to increase the reading interval to save resources.

I2C is used for communication between the host and one or more modules, with a speed of 400kHz and a voltage of 3.3v. Modules that are designed to work with I2C should use I2C as the protocol. This allows the host

to maximize its IO ports, and utilize its full potential. If no modules are attached, these two IOs can be used for other purposes.

Module has new information, host needs to read immediately

The module pulls up the interrupt pin, notifying the host to read. The interrupt pin is shared by multiple modules. This saves IO ports.

1. The host polls the interrupt bits of each module quickly to read the values.
 1. If non-zero, it means that this module has new values, and the host reads the module's values and clears the interrupt bit.
 2. If zero, skip this module.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
C1	3.3v	IO41	IO42	IO09	IO40	IO06	IO05	IO02	IO01	5V	GND
C2	GND	IO20	IO19	GND	IO44	IO43	IO13	IO14	IO45	IO04	5V
I2C	-	SCL	SDA	-	-	-	-	-	-	-	-
I2C	-	-	-	-	-	-	-	-	-	-	-

UART

Some modules require serial communication, and if RS485, CAN, MAX512, etc. protocols are used, there may also be UART interfaces and RESET pins used to update the module's firmware. Since updating the module's firmware is not commonly used, the TX and RX pins are combined to form a single UART interface, and the Reset (RST) pin is only used when updating the module's firmware.

1. Used for module updates, the switches on the module must be set to ON, and the host can then update the module. After the update is complete, the switches must be set to OFF.
2. Use serial communication with the module, such as RS485, CAN, MAX512, etc. protocols. Serial communication.
3. The Atmega328p can be used to control whether the module is communicating with the host via serial. This allows multiple modules to be used at the same time.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
C1	3.3v	IO41	IO42	IO09	IO40	IO06	IO05	IO02	IO01	5V	GND
C2	GND	IO20	IO19	GND	IO44	IO43	IO13	IO14	IO45	IO04	5V
UART	-	-	-	-	-	-	RST	RX	TX	-	-
UART	-	-	-	-	-	-	-	-	-	-	-

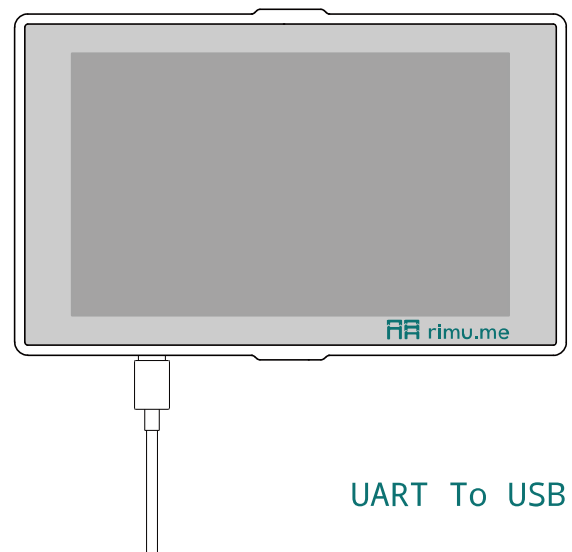
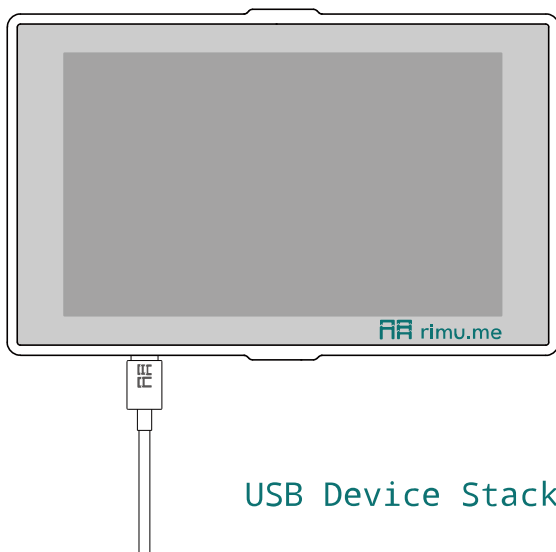
USB Device Stack

The host can be used as a USB device, such as a custom keyboard, mouse, game controller, or USB drive. The host and module communicate using the USB protocol, and use the IO20 (D+) and IO19 (D-) pins. If these types of applications are not needed, they can be used as IO ports.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
C1	3.3V	IO41	IO42	IO09	IO40	IO06	IO05	IO02	IO01	5V	GND
C2	GND	IO20	IO19	GND	IO44	IO43	IO13	IO14	IO45	IO04	5V
USB	-	-	-	-	-	-	-	-	-	-	-
USB	-	D+	D-	-	RX	TX	-	-	-	-	-

UART To USB

The host uses the CH9102X to connect to the ESP32S3's serial (IO44/43), and can be used to update the host's firmware and debug.



1. If the logo and screen are on the same side of the host, it can be used as a keyboard.
2. Otherwise, the USB cable can be used as a serial port, and can be used to update the host's firmware.

SPI

Some modules require faster data transfer, and use the SPI protocol. The SPI pins are defined according to the [Modules Pin Map](#) table, and the chip select pin is controlled by the Atmega328p. This allows a single SPI bus to connect multiple modules.

Universal Registers

I2C Protocol

1. The I2C host address range is 0x60 to 0x6F, corresponding to the 0 to F positions of the switch.
2. The I2C communication rate is 400KHz.
3. The common information register addresses are read-only.

Name definition

1. **Module number:** Module number, different for each module, range: 0x0001-0xFFFF
2. **Communication protocol:** Communication protocol this module has.
3. **Number of registers on the I2C bus:** The number of registers on the I2C bus of this module.
4. **Hardware version:** Module hardware version, module board version.
5. **Software version:** Module software version, firmware version.
6. **Checksum:** Checksum to verify if the module is correct.
7. **Module Name:** Maximum 32 characters, start with WF1-.

Document Convention

0x00 represents the register address, 0x00 represents the register value.

Registers Table

Address	Function	R/W	Comments
0x00/02	Module No	R	
0x01/03	Module No	R	
0x04/05	Number of Registers	R	
0x06	Protocol	R	
0x07/08/09/0A/0B	Pin Status	R	
0x0C	Hardware version	R	
0x0D	Software Version	R	
0x0E	Checksum	R	
0x0F	Interrupt Triggered	R/W	
0x10->0x2F	Module Name	R	

Module No

The module number is unique, ranging from 0x0001 to 0xFFFF. If the same type of module is used, the original number is used. If there is a subsequent version, a new number needs to be used. The assigned number is consistent with the host. This allows the host to jump to the appropriate interface based on the module number, making it easier to operate and use.

To prevent some I2C chips from having the same value in the same register address, WF1 sets duplicate values for the module number. When the host reads, it needs to make the judgment.

1. 0x00 stores the high-order value, and 0x01 stores the low-order value.

2. `0x02` and `0x03` are the second time.

For example, if the module number is `0x1234`, the values of `0x00` and `0x02`, should be `0x12`, and the values of `0x01` and `0x03` should be `0x34`.

High 8 Bit

<code>0x00/0x02</code>	B7	B6	B5	B4	B3	B2	B1	B0
High 8 Bits	0	0	0	0	0	0	0	0

Low 8 Bit

<code>0x01/0x03</code>	B7	B6	B5	B4	B3	B2	B1	B0
Low 8 Bits	0	0	0	0	0	0	0	0

Number of Registers

How many registers are on the I2C bus of this module, stored in `0x04` and `0x05`.

Functions Used

Host and module exchange information, I2C is preferred for communication, but some modules may also use other communication methods, such as `SPI`, `UART`, etc.

The address is `0x06`, each bit represents the communication protocol used by this module, `0` means not used, `1` means used.

<code>0x06</code>	B7	B6	B5	B4	B3	B2	B1	B0
Protocol	UPGRADE	INT	TBD	TBD	TBD	UART	SPI	I2C

1. I2C: This type of module supports I2C protocol.
2. UPGRADE: This type of module can be upgraded by the host.
3. INT: This type of module has an interrupt trigger function, which means that the host can receive a notification when the module's status changes.
4. UART: This type of module supports UART protocol.
5. SPI: Some modules require faster data transfer, which uses SPI protocol.
6. TBD: Reserved bit, not used.

Pin Status

All pins are represented by 2 bits, and the register address is `0x07`, `0x08`, `0x09`, `0x0A`. The definition is as follows:

Pin not used (0b00)

This pin is not connected to any line of this module, and can be used by other modules.

Pin was connected, but not used (0b01)

1. The three pins used for upgrading, IO01 TXT, IO02 RXT, IO05 RST.
2. USB D+ and D- pins, which are directly connected to the USB, cannot be used when using HID function. The host need check the HID function status before using these pins.
3. USB UART Pins, IO43 TXT and IO44 RXT was connected to the USB, but also can be used for modules. The host need check the UART function status before using these pins.
4. The Interrupt pin, IO12, is used for interrupt, and the host's interrupt use the same pin.
5. The pin is connected to the input pin of the IO switch chip, but is not used, such as the number pin of the 74HC4051 chip, which is not selected.

Pin is used, and can be modified by the host (0b10)

1. The pin is connected to the input pin of the IO switch chip, and has been selected, and can be modified by the host.

Pin is used, and cannot be modified by the host (0b11)

- If used as I2C communication, then IO41 and IO42 cannot be modified.
- If used as SPI communication, then IO09, IO40, IO06 cannot be modified.
- If used as UART communication, then IO01 TXT and IO02 RXT cannot be modified.

0x07	B7	B6	B5	B4	B3	B2	B1	B0
Val	IO	05	IO	04	IO	02	IO	01
0x08	B7	B6	B5	B4	B3	B2	B1	B0
Val	IO	13	IO	12	IO	09	IO	06
0x09	B7	B6	B5	B4	B3	B2	B1	B0
Val	IO	40	IO	20	IO	19	IO	14
0x0A	B7	B6	B5	B4	B3	B2	B1	B0
Val	IO	44	IO	43	IO	42	IO	41
0x0B	B7	B6	B5	B4	B3	B2	B1	B0
Val							IO	45

Hardware Version

The address is 0x0C, the first release is 0x00, and the subsequent version number is incremented.

0x0C	B7	B6	B5	B4	B3	B2	B1	B0
Val	0	0	0	0	0	0	0	0

Software Version

The address is `0x0D`, the first release is `0x00`, and the subsequent version number is incremented.

0x0D	B7	B6	B5	B4	B3	B2	B1	B0
Val	0	0	0	0	0	0	0	0

Checksum

The address is `0x0E`, the value is the low 8 bits of the checksum from `0x00` to `0x0D`. The checksum is calculated by adding all the values in the module's registers, and then taking the low 8 bits of the result.

0x0E	B7	B6	B5	B4	B3	B2	B1	B0
Val	0	0	0	0	0	0	0	1

Interrupt Triggered

If the module has information that needs to be processed immediately by the host, by pulling the interrupt pin high, the host is notified. However, since multiple devices are using the interrupt pin, the host needs to check the values of different module registers to quickly determine which module needs to be processed. After the host has processed the interrupt, the host sets this bit to 0, and the module immediately pulls the interrupt pin low. The register address is `0x0F`.

Module Name

The module name is stored in registers `0x10` to `0x2F`, up to 32 bytes, and the name starts with WF1-. This makes it easy to identify different modules when identifying modules.

All private information registers for all modules start from `0x30`.