

IS4350-M1

I2C Modbus TCP/IP Server Module with RJ45 connector

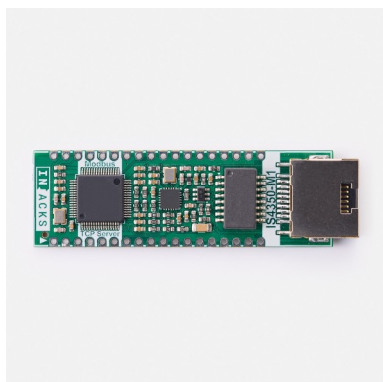
Presentation

The IS4350-M1 is a ready-to-use Ethernet module that turns any microcontroller with an I2C port into a Modbus TCP/IP server. It integrates the IS4350-L64 I2C Modbus TCP/IP Server IC, the LAN8742A 10/100 Ethernet PHY, the Ethernet magnetics, ESD protection and an RJ45 connector with status LEDs on a single, pre-designed and tested board.

Designing an Ethernet interface from scratch requires a PHY, its reference clock, strap configuration, impedance-controlled RMII and MDI routing, magnetics, line termination and a shielded RJ45 connector. The IS4350-M1 contains all of this. Your design only has to provide 3.3 V, two I2C lines and a few configuration pins; the module takes care of the Ethernet link, the TCP/IP stack and the Modbus protocol.

The module has 36 castellated pads on a 2.54 mm pitch, so it can be soldered directly onto a carrier PCB as a surface-mount component, or fitted with pin headers for prototyping and evaluation.

IS4350-M1 Characteristics	
Implemented Function Codes	03 – Read Holding Registers 06 – Write Single Register 16 – Write Multiple Registers
Available Holding Registers	65,500
Modbus Server IP	DHCP / Static (default: DHCP)
Modbus Server Port	502
Host interface	I2C, 100 kHz, 400 kHz, 1 MHz
I2C addresses	24, 25, 26 (selected with the ADR pin)
Supply voltage	3.3 V
Dimensions	20 × 62 mm, 36 flat castellated pads



1 Ordering Information

Standard configuration

The standard configuration refers to the device variant that is mass-produced and offered at the most cost-efficient price.

Orderable Device	Status	Package		Pins	Temperature Range
IS4350-L64 www.inacks.com/is4350	ACTIVE		LQFP64 (10×10 mm)	64	I (Industrial) -40 °C to 85 °C
IS4350-M1C www.inacks.com/is4350-m1	ACTIVE		Module (20×62 mm)	36 + Ethernet Connector	C (Consumer) 0 °C to 70 °C
IS4350-M1I www.inacks.com/is4350-m1	ACTIVE				I (Industrial) -40 °C to 85 °C

Status Description:

ACTIVE: Product recommended for new designs.

SPECIAL: Available only for high-volume pre-orders. Contact sales.

NRND: Device is in production to support existing customers, but it is not recommended for use in new designs.

OBSOLETE: Production of the device has been discontinued.

2 Revision History

This manual and the IS4350-L64 datasheet may be revised without notice. Users are responsible for obtaining and reviewing the latest version of this document and the datasheet before purchasing the product or designing it into any application. The most recent datasheet revision available on the company's website (www.inacks.com) shall be considered the only valid and authoritative source of product information.

Document Revision

Date	Revision Code	Description
October 2026	ISDOC155B	- Initial Release
June 2026	ISDOC155A	- Preview

Module Revision

Date	Revision Code	Description
June 2026	ISB3117C	- Initial Release

Related Documents

Document	Content
IS4350 Datasheet-I2C Modbus TCP Server (ISDOC139) https://inacks.com/is4350	Complete IS4350 description: memory map, register details, I2C protocol, I2C-CRC, watchdog, electrical specifications.
LAN8742A Datasheet (Microchip)	Ethernet PHY used on the module.

3 Specifications

The IS4350-M1 electrical limits are those of the IS4350-L64 (ISDOC139) and of the LAN8742A PHY, both supplied from the module 3V3 pins. The values below apply at the module pins.

Absolute Maximum Ratings

Parameter			Min	Max	Unit
Input Voltage	3V3, SPD, ADR, nRST pins		-0.3	3.6	V
	SCL, SDA, CRC, DFT, INT, RX, TX pins		-0.3	6	
Current Sourced/Sunk by any I/O and Control Pin			—	±20	mA
Total output current Sourced/Sunk by sum of all I/Os and control pins			—	±140	
Temperature	Operating Temperature	IS4350-L64-M1C (Consumer)	0	+70	°C
		IS4350-L64-M1I (Industrial)	-40	+85	
	Storage Temperature		-55	+150	

Exceeding the specifications outlined in the Absolute Maximum Ratings could potentially lead to irreversible harm to the device. It's important to note that these ratings solely indicate stress limits and don't guarantee the device's functionality under such conditions, or any others not specified in the Recommended Operating Conditions. Prolonged exposure to conditions at or beyond the absolute maximum ratings might compromise the reliability of the device.

Recommended Operating Conditions

Parameter	Pins	Symbol	Min	Nom	Max	Unit
Supply Voltage	3V3	V_{DD}	2.7	3.3	3.6	V
I/O Input Voltage	SCL, SDA, CRC, DFT, INT, RX, TX	V_{IO-IN}	-0.3	3.3	5.5	
	SPD, ADR		-0.3	1.65, 3.3	3.6	
	nRST		-0.3	3.3	3.6	
Source/Sink Current	Any I/O Pins	I_{IO-SS}	—	—	±8	mA

Electrical Characteristics

All I/Os are CMOS and TTL compliant.

Parameter		Symbol	Min	Nom	Max	Unit
Current Consumption	$T_J = 25\text{ °C}$	I_{DD}	—	192	385	mA
Input Voltage	Logical High-Level	V_{IH}	$0.7 \times V_{DD}$	—	—	V
	Logical Low-Level	V_{IL}	—	—	$0.3 \times V_{DD}$	

Electrical Specifications Revision A

Modbus Characteristics

Parameter	Characteristics
Implemented Function Codes	- FC03 – Read Holding Registers - FC06 – Write Single Register - FC16 – Write Multiple Registers
Available Holding Registers	65,500
Modbus Unit Identifier	0xFF (Responds to any, as defined in the standard)
Simultaneous Client Connections	1
Typical request rate	≈ 216 to 270 requests/s depending on function code and register count (see ISDOC139, Modbus Performance)

Ethernet Characteristics

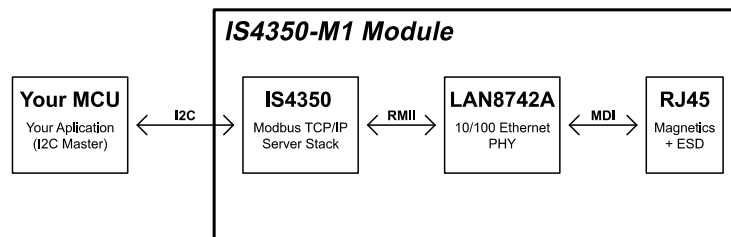
Parameter		Characteristics
Ethernet PHY		Onboard LAN8742A
Modbus Server IP		DHCP / Static (default: DHCP)
Modbus Server Port		502
Ethernet Speed		10/100 Mbps
Status LED		Yellow: LED1 of the LAN8742A (link / activity) Green: LED2 (link speed, on at 100 Mbps)
Ethernet MAC address configuration	Locally Administered	- Manual (manually configured) - Fixed (generated from the IS4350-L64's serial number) - Random (randomly generated)

4 Description

The IS4350-M1 implements a Modbus TCP/IP server. It exposes 65,500 Holding Registers that can be read and written both by your microcontroller, over I2C, and by any Modbus client on the network (a PLC, SCADA system, HMI or PC). The Holding Registers work as a shared memory: your firmware writes measurements for the client to read, and reads the set-points and commands that the client writes.

From your microcontroller's point of view, the IS4350-M1 behaves like an I2C memory with 16-bit addresses and 16-bit registers. No Ethernet driver, TCP/IP stack or Modbus library is needed.

The module contains two interfaces: I2C-serial interface, where your microcontroller is the I2C master and the IS4350-M1 is the slave; and the Ethernet interface (LAN8742A PHY + RJ45).



4.1 Firmware Quick Start

Your firmware only needs two routines: an I2C register write and an I2C register read. Register addresses and register values are 16 bits, sent most significant byte first. A register write therefore sends four bytes after the device address: address high, address low, data high, data low.

1. Power up the module and wait until the IS4350-L64 is ready (about 1 s is enough).
2. Optionally, read `CHIP_ID`. It always returns 150 and confirms that I2C communication works.
3. Write the initial values of your Holding Registers (`HOLD_0 ... HOLD_65499`), so that clients never read uninitialized data.
4. Start the Modbus server: write 1 to `GO_ONLINE`.
5. Poll `STATUS` until the low byte (`STATUS_BITS`) equals 5: the server is ready and listening on port 502.
6. Read the `IP` registers to find the address assigned by DHCP.
7. In the main loop, update the Holding Registers with your data and read the values written by the client.

Fail-safe power-up: `GO_ONLINE` is 0 after every power-up or reset. The Modbus server never starts until your firmware explicitly writes `GO_ONLINE`, so it cannot come online with stale data if your microcontroller has failed.

4.2 Firmware Best Practices

- Leave a few milliseconds between consecutive I2C operations.
- Update the Holding Registers before bringing the server online.
- Stop the server (`GO_ONLINE = 0`) before changing `DHCP`, `IP`, `MASK` or `GATEWAY`.
- Check the `STATUS` register instead of assuming that writing `GO_ONLINE` succeeded.

4.3 Ethernet and Network Configuration

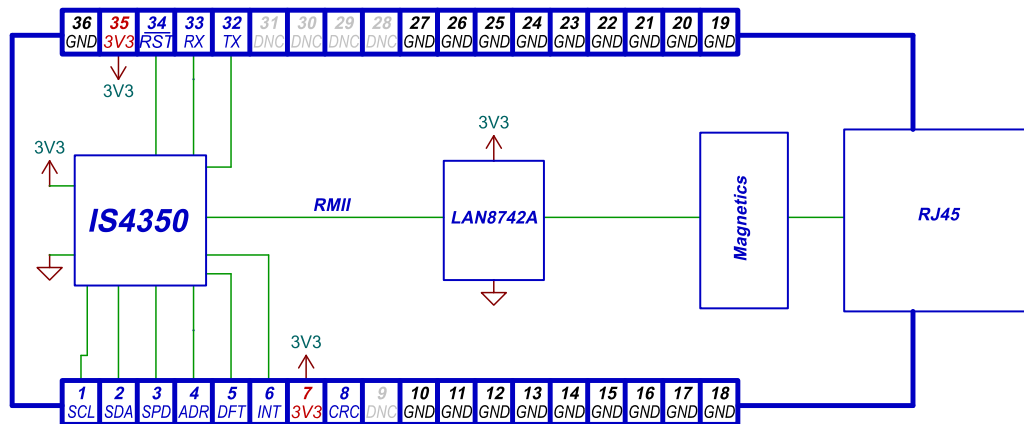
Connect the RJ45 to a switch, router or directly to the Modbus client with a standard Cat5e (or better) cable. The link speed and duplex mode are auto-negotiated (10 or 100 Mbps).

By default, DHCP is enabled and the module obtains its IP address, subnet mask and gateway from the network's DHCP server after it is set online (`GO_ONLINE = 1`). Once online, the assigned values can be read in the `IP`, `MASK` and `GATEWAY` registers.

The MAC address is by default locally administered and generated from the IS4350-L64 serial number.

5 Pinout

The IS4350-M1 has 36 castellated pads on a 2.54 mm pitch. All signal pins are 3.3 V logic unless otherwise noted.



(Image shown for illustrative purposes only) SCH0099B

Pin	Name	Type	Description
1	SCL	Open Drain	I2C serial interface. These pins connect to your microcontroller. Two external pull-up resistors to 3.3 V or 5 V are required.
2	SDA		
3	SPD	Analog Input	I2C serial interface speed selection pin. - Connect to GND for 100 kHz. - Connect to 1.65 V for 400 kHz. - Connect to 3V3 for 1 MHz.
4	ADR	Analog Input	I2C serial interface address selection pin. - Connect to GND for address 24. - Connect to 1.65 V for address 25. - Connect to 3V3 for address 26.
5	DFT	Digital Input	Restore default network configuration. Hold high for 3 s. Optional; internal weak pull-down, leave floating if unused.
6	INT	Digital Output Push-Pull	Goes high for about 100 ms each time a Modbus write (FC06 or FC16) is received. Optional; leave floating if unused.
7, 35	3V3	Supply	3.3V power supply input. Connect both pins.
10–27, 36	GND	Reference	Voltage reference pin (0 V). Connect all pins to the ground plane of your PCB.
8	CRC	Digital Input	I2C CRC protection enable. High: enabled. Low or floating: disabled (internal weak pull-down). Read at start-up only.
32	TX	Digital Output Push-Pull	Provisioning UART port. Used for firmware updates.
33	RX	Digital Input	
34	nRST	Digital Input	Active-low reset of the IS4350-L64 and the LAN8742A. Pull-up on module. Leave floating if unused.
9, 28–31	DNC	Do Not Connect	Leave these pins floating.

For first bring-up, the minimum set of connections is 3V3, GND, SCL, SDA, SPD and ADR. All other pins may be left floating.

5.1 SCL & SDA Pins

I2C-serial interface between your microcontroller (master) and the IS4350-L64 (slave). Both lines are open drain and require external pull-up resistors, to 3.3 V or to 5 V. The module has no I2C pull-ups on board, so the pull-ups must be placed on your PCB. Typical values are 4.7 k Ω for 100 kHz and 2 k Ω for 400 kHz and 1 MHz; adjust them to the bus capacitance.

Your microcontroller must support I2C clock stretching: the IS4350-L64 may hold SCL low while it processes a transaction.

5.2 SPD Pin

Selects the I2C speed and configures the IS4350-L64 input filters accordingly. It is read at power-up only. Do not leave it floating.

I2C speed	SPD connection
100 kHz (Standard Mode)	GND
400 kHz (Fast Mode)	1.65 V: two 10 k Ω resistors, one to 3V3 and one to GND
1 MHz (Fast Mode Plus)	3V3

5.3 ADR Pin

Selects the I2C device address. It is read at power-up only. Do not leave it floating. With three addresses available, up to three IS4350-M1 modules can share the same I2C-serial interface.

I2C address	ADR connection
24	GND
25	1.65 V: two 10 k Ω resistors, one to 3V3 and one to GND
26	3V3

5.4 CRC Pin

Connect to 3V3 to protect every I2C read and write with a CRC-16/Modbus checksum. Leave floating or connect to GND to disable it. The pin is read at start-up only and must not change during operation.

During development, keep CRC disabled until basic communication works, then enable it if your application needs it. The CRC procedure is described in ISDOC139, section I2C-CRC Protection.

5.5 DFT Pin

Restores the factory network configuration and restarts the IS4350-L64 when held high for 3 seconds. Use it to recover a device that has become unreachable because of a wrong IP or MAC configuration. After a restore:

- GO_ONLINE = 0 (server stopped)
- All Holding Registers = 0xFFFF
- DHCP enabled (DHCP = 1)
- MAC address generated from the IS4350-L64 serial number (MAC_CFG = 1)

A recessed push button to 3V3 is recommended in the final product, so that it cannot be pressed by accident. The pin has an internal weak pull-down (≈ 40 k Ω); leave it floating if unused.

5.6 INT Pin

Push-pull output that goes high for about 100 ms whenever a Modbus client sends a write request (FC06 or FC16), even if the written value does not change. Connect it to an interrupt-capable input of your microcontroller to avoid polling the registers. The same information is latched in the INT_BIT (bit 9) of the STATUS register. Leave floating if unused.

5.7 nRST Pin

Active-low reset. On the module, nRST is connected to both the IS4350-L64 and the LAN8742A so that both ICs always restart together. Asserting it restarts the IS4350-L64 firmware; non-volatile settings (IP configuration, MAC configuration) are kept.

It can be left floating (pull-up on module), driven by a GPIO of your microcontroller, or connected to your microcontroller reset line if that line is active-low and 3.3 V compatible. Drive it with an open-drain output, or with a push-pull output that never drives it above 3.6 V.

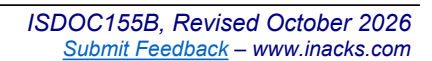
5.8 TX & RX Pins

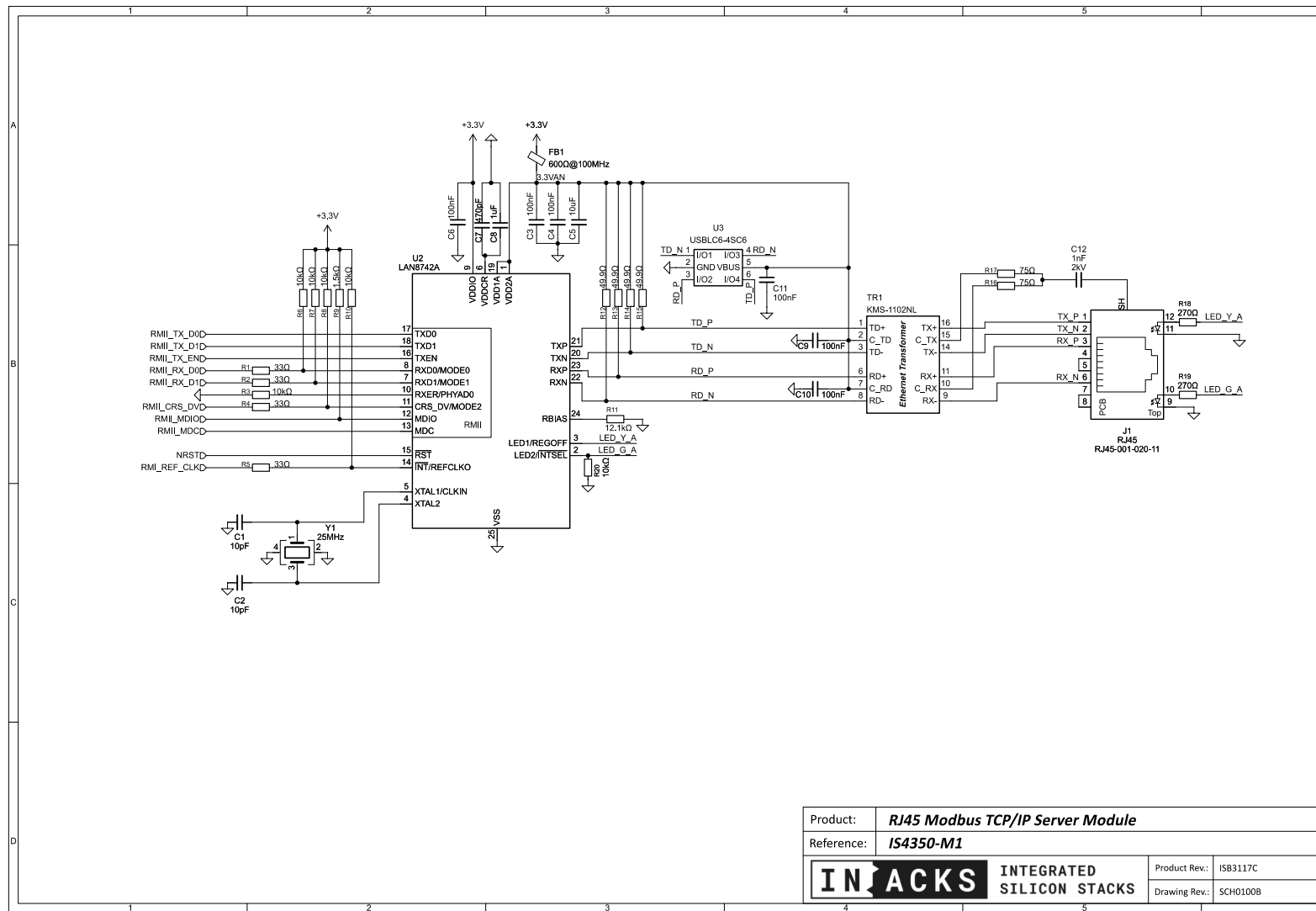
Provisioning UART (TX and RX) used to update the IS4350-L64 firmware with security patches or new releases. Bring these two pins and GND to a 3-pin header on your PCB so that a 3.3 V USB-to-UART adapter can be connected. Connect the adapter RX to the module TX, and the adapter TX to the module RX.

5.9 3V3 & GND Pins

The IS4350-M1 is powered from a single 3.3 V supply applied to the 3V3 pins (7 and 35). The same supply feeds the IS4350-L64 and the LAN8742A. The module generates all other internal voltages itself.

10/20



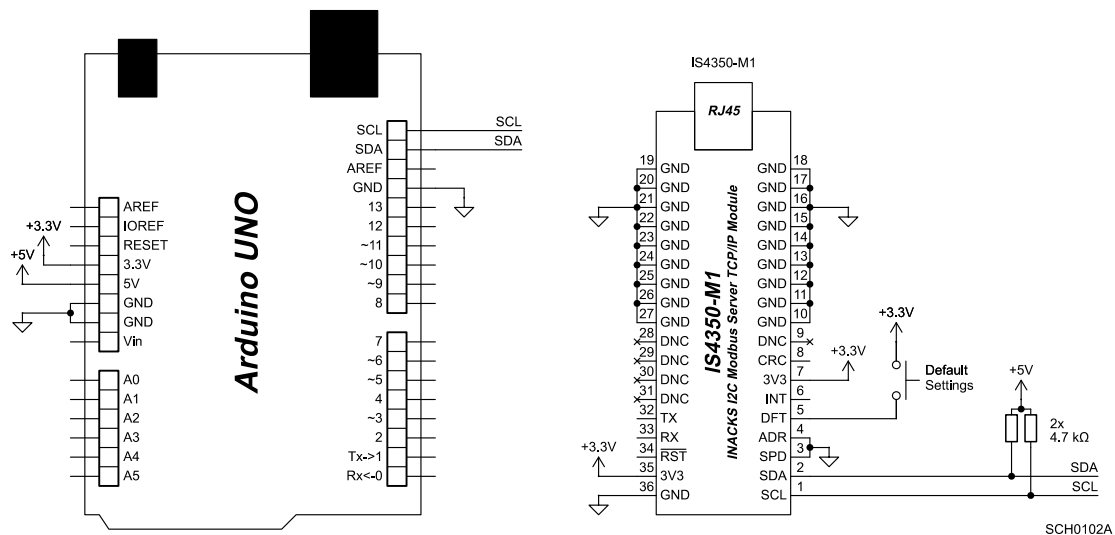


7 Examples

7.1 Arduino Connection Example

The IS4350-M1 connects to any microcontroller or microprocessor with an I2C master that supports clock stretching. No Ethernet peripheral is required on the host.

Arduino Uno	IS4350-M1	Notes
3.3V	3V3 (7, 35)	Some Arduino Uno boards have a 3.3 V output rated for only 50 mA, which is not sufficient to power the IS4350-M1. Check the 3.3 V output current capability of your specific Arduino board before connecting the module.
GND	GND	—
SDA	SDA (2)	4.7 k Ω pull-up to 5 V
SCL	SCL (1)	4.7 k Ω pull-up to 5 V
—	SPD (3) to GND	100 kHz
—	ADR (4) to GND	Address 24



7.2 Arduino Code Example: Temperature Sensor

This example (ISXMPL4350ex3) demonstrates how to use the IS4350-M1 module with an Arduino.

Find the complete example and buy the IS4350-M1 at: www.inacks.com/is4350-m1

```

/*
 * IS4350-M1 - Arduino Uno Temperature Sensor Example
 * -----
 * A Modbus TCP/IP temperature sensor in a few lines:
 * - Reads a TMP36 analog temperature sensor once per second
 * - Writes the temperature to Holding Register 0 (HOLD_0)
 * - Any Modbus TCP/IP client (PLC, SCADA, PC) reads it over Ethernet
 *
 * Wiring:
 *   TMP36 +Vs          -> Arduino 5V
 *   TMP36 Vout         -> Arduino A0
 *   TMP36 GND          -> Arduino GND
 *   (TMP36 flat side facing you, legs down: +Vs, Vout, GND from left to right)
 *
 *   IS4350-M1 3V3 (7, 35) -> external 3.3 V supply (verify current source capability)
 *   IS4350-M1 GND         -> Arduino GND and 3.3 V supply GND
 *   IS4350-M1 SDA (2)     -> Arduino A4, 4.7 kOhm pull-up to 5 V
 *   IS4350-M1 SCL (1)     -> Arduino A5, 4.7 kOhm pull-up to 5 V
 *   IS4350-M1 SPD (3)     -> GND (I2C at 100 kHz)
 *   IS4350-M1 ADR (4)     -> GND (I2C address 24)
 *   IS4350-M1 RJ45        -> network with a DHCP server
 *   All other module pins -> leave floating
 *
 * Reading the temperature from a Modbus client:
 *   IP address : printed on the Serial Monitor (9600 baud)
 *   Port       : 502, Unit ID 255
 *   Function    : 03 - Read Holding Registers, address 0 (HOLD_0,
 *                  shown as 40001 in tools that use 1-based numbering)
 *   Format      : signed 16-bit integer, in tenths of a degree
 *                  (235 = 23.5 °C, -52 = -5.2 °C)
 */

#include <Wire.h>

const uint8_t  IS4350_ADDR   = 24;    // ADR pin tied to GND
const uint16_t REG_HOLD_0    = 0;    // temperature, in 0.1 °C
const uint16_t REG_IP        = 65501; // 65501..65504
const uint16_t REG_GO_ONLINE = 65513;
const uint16_t REG_STATUS    = 65514;
const uint16_t REG_CHIP_ID   = 65522; // always 150

const uint8_t  SENSOR_PIN    = A0;
const long     ADC_REF_MV     = 5000; // 5000 on 5 V boards (Uno), 3300 on 3.3 V boards

// Write one 16-bit register. Address and data are sent MSB first.
void is4350Write(uint16_t reg, uint16_t value) {
    Wire.beginTransmission(IS4350_ADDR);
    Wire.write((uint8_t)(reg >> 8));
    Wire.write((uint8_t)(reg & 0xFF));
    Wire.write((uint8_t)(value >> 8));
    Wire.write((uint8_t)(value & 0xFF));
    Wire.endTransmission();
    delay(5); // a few ms between I2C operations
}

// Read one 16-bit register. Address and data are sent MSB first.
uint16_t is4350Read(uint16_t reg) {
    Wire.beginTransmission(IS4350_ADDR);
    Wire.write((uint8_t)(reg >> 8));
    Wire.write((uint8_t)(reg & 0xFF));
    Wire.endTransmission(false); // repeated START
    Wire.requestFrom(IS4350_ADDR, (uint8_t)2);
    uint16_t value = 0;
    if (Wire.available() >= 2) {

```

```
    value = (uint16_t)Wire.read() << 8; // MSB
    value |= Wire.read();               // LSB
}
delay(5);
return value;
}

// TMP36: 500 mV at 0 °C and 10 mV/°C, so (mV - 500) is the temperature in 0.1 °C.
int16_t readTemperature() {
    long mV = analogRead(SENSOR_PIN) * ADC_REF_MV / 1024;
    return (int16_t)(mV - 500);
}

void setup() {
    Serial.begin(9600);
    Wire.begin(); // 100 kHz, matches SPD tied to GND
    delay(1000);  // let the IS4350 start

    if (is4350Read(REG_CHIP_ID) != 150) {
        Serial.println("IS4350-M1 not detected. Check wiring, pull-ups, SPD and ADR.");
        while (1);
    }

    is4350Write(REG_HOLD_0, (uint16_t)readTemperature()); // valid data before going online
    is4350Write(REG_GO_ONLINE, 1);                        // start the Modbus server

    uint8_t status;
    while ((status = is4350Read(REG_STATUS) & 0xFF) != 5) { // 5 = ready and listening
        Serial.print("Status: ");
        Serial.println(status);
        delay(500);
    }

    Serial.print("Sensor online. IP: ");
    for (uint8_t i = 0; i < 4; i++) {
        Serial.print(is4350Read(REG_IP + i));
        if (i < 3) Serial.print(".");
    }
    Serial.println(", port 502");
}

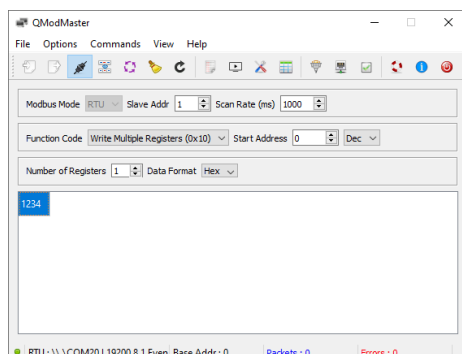
void loop() {
    int16_t temperature = readTemperature(); // in 0.1 °C
    is4350Write(REG_HOLD_0, (uint16_t)temperature); // negative values as two's complement

    Serial.print("Temperature: ");
    Serial.print(temperature / 10.0, 1);
    Serial.println(" °C");
    delay(1000);
}
```

8 Modbus Software

The following third-party software options are provided for reference only. These applications are not developed, maintained, or endorsed by INACKS. We do not guarantee their functionality, compatibility, or compliance with the Modbus standard. Users should evaluate and choose software based on their specific needs.

Modbus Client Software



Title: QModMaster

Description:

QModMaster is a free Qt-based implementation of a Modbus master application. A graphical user interface allows easy communication with Modbus RTU and TCP slaves. QModMaster also includes a bus monitor for examining all traffic on the bus.

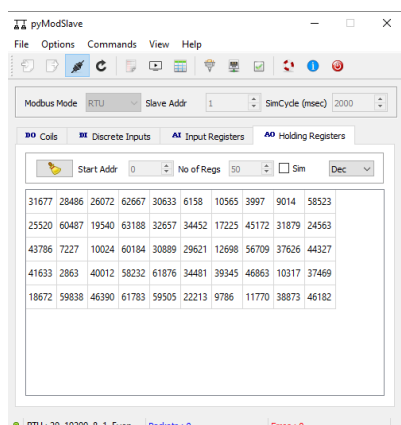
Web:

<https://sourceforge.net/projects/qmodmaster/>

Important:

Go to menu *Options* → *Settings*, and set the parameter *Base Addr* to 0 to avoid confusions with the Modbus registers addresses.

Modbus Server Software



Title: pyModSlave

Description:

pyModSlave is a free python-based implementation of a Modbus slave application for simulation purposes. You can install the python module or use the precompiled (for Windows only) stand alone GUI (Qt based) utility (unzip and run). pyModSlave also includes a bus monitor for examining all traffic on the bus.

Web:

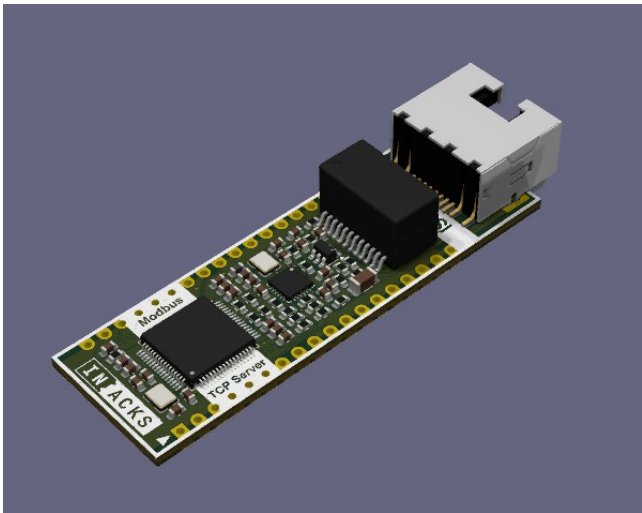
<https://sourceforge.net/projects/pymodslave/>

9 Mechanical Information

9.1 STEP 3D model

A STEP 3D model of the module is available to download at the product page: www.inacks.com/is4350-m1

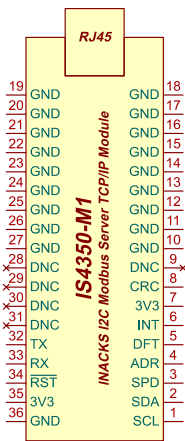
The 3D model is provided for reference only. Some component dimensions, particularly those of the RJ45 connector and magnetics, as well as component appearances, may differ from the actual product and should not be relied upon for critical mechanical verification.



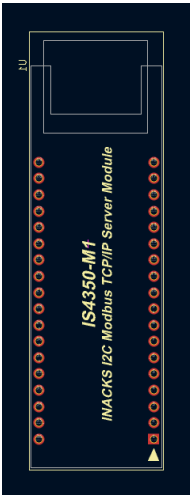
9.2 KiCad Symbol and Footprint

Both the symbol and footprint are available to download from the product page in KiCad format.

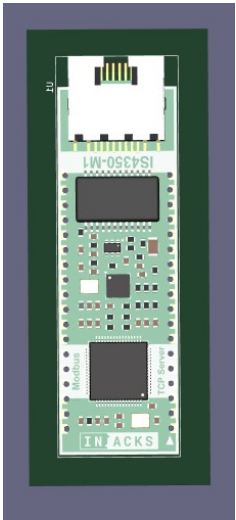
Download IS4350-M1 Symbol and Footprint: www.inacks.com/is4350-m1



IS4350-M1 KiCad Symbol



IS4350-M1 KiCad Footprint



IS4350-M1 KiCad Model

SCH0103A

10 Troubleshooting

Symptom	Possible cause and solution
The module does not acknowledge its I2C address	Check the 3.3 V supply, the pull-up resistors on SCL and SDA, and the ADR setting (24, 25 or 26). SPD and ADR must not be floating; they are read only at power-up, so power-cycle after changing them.
CHIP_ID does not read 150	The I2C read routine must send the 16-bit register address (MSB first) and use a repeated START. Check that the microcontroller supports clock stretching and that SPD matches the bus speed.
Writes seem to be ignored	If CRC is high, every write must end with a valid CRC. Tie CRC low while developing.
STATUS stays at 1	GO_ONLINE has not been written, or the watchdog (GO_ONLINE = 2) expired. Write GO_ONLINE again.
STATUS stays at 3	No DHCP server answered. Connect to a network with DHCP or configure a static IP.
STATUS = 103	Cable disconnected or no link. Check the cable and that the yellow LED is on.
STATUS = 101 or 102	Wrong MAC configuration. Set MAC_CFG = 1 and reset, or use the DFT pin.
The client cannot connect although STATUS = 5	Check that the client uses the IP from the IP registers and port 502, and that both are in the same subnet. Only one client can be connected at a time.
Device unreachable after a network configuration change	Hold DFT high for 3 s to restore DHCP and the default MAC configuration.

Content

Presentation.....	1	5.7 nRST Pin.....	8
1 Ordering Information.....	2	5.8 TX & RX Pins.....	8
2 Revision History.....	3	5.9 3V3 & GND Pins.....	9
3 Specifications.....	4	6 Schematic.....	10
Absolute Maximum Ratings.....	4	7 Examples.....	12
Recommended Operating Conditions.....	4	7.1 Arduino Connection Example.....	12
Electrical Characteristics.....	4	7.2 Arduino Code Example: Temperature Sensor	13
Modbus Characteristics.....	5	8 Modbus Software.....	15
Ethernet Characteristics.....	5	9 Mechanical Information.....	16
4 Description.....	6	9.1 STEP 3D model.....	16
4.1 Firmware Quick Start.....	6	9.2 KiCad Symbol and Footprints.....	16
4.2 Firmware Best Practices.....	6	10 Troubleshooting.....	17
4.3 Ethernet and Network Configuration.....	6	Content.....	18
5 Pinout.....	7	Appendix.....	19
5.1 SCL & SDA Pins.....	8	Documentation Feedback.....	19
5.2 SPD Pin.....	8	Sales Contact.....	19
5.3 ADR Pin.....	8	Customization.....	19
5.4 CRC Pin.....	8	Independence and Trademarks Notice.....	19
5.5 DFT Pin.....	8	Disclaimer.....	20
5.6 INT Pin.....	8		

Appendix

Documentation Feedback

Feedback and error reporting on this document are very much appreciated.

feedback@inacks.com

Sales Contact

For special order requirements, large volume orders, or scheduled orders, please contact our sales department at:

sales@inacks.com

Customization

INACKS can develop new products or customize existing ones to meet specific client needs. Please contact our engineering department at:

engineering@inacks.com

Independence and Trademarks Notice

This company and the products provided herein are developed independently and are not affiliated with, endorsed by, or associated with any official protocol or standardization entity.

All trademarks, names, and references to specific protocols remain the property of their respective owners.

Disclaimer

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, INACKS does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. INACKS takes no responsibility for the content in this document if provided by an information source outside of INACKS.

In no event shall INACKS be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, INACKS's aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of INACKS.

Right to make changes — INACKS reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — INACKS products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an INACKS product can reasonably be expected to result in personal injury, death or severe property or environmental damage. INACKS and its suppliers accept no liability for inclusion and/or use of INACKS products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Quick reference data — The Quick reference data is an extract of the product data given in the Limiting values and Characteristics sections of this document, and as such is not complete, exhaustive or legally binding.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. INACKS makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using INACKS products, and INACKS accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the INACKS product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

INACKS does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using INACKS products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). INACKS does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — INACKS products are sold subject to the general terms and conditions of commercial sale, as published at <http://www.inacks.com/comercialsaleterms>, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. INACKS hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of INACKS products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — This INACKS product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. INACKS accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

Protocol Guidance Disclaimer: The information provided herein regarding the protocol is intended for guidance purposes only. While INACKS strive to provide accurate and up-to-date information, this content should not be considered a substitute for official protocol documentation. It is the responsibility of the client to consult and adhere to the official protocol documentation when designing or implementing systems based on this protocol.

INACKS make no representations or warranties, either expressed or implied, as to the accuracy, completeness, or reliability of the information contained in this document. INACKS shall not be held liable for any errors, omissions, or inaccuracies in the information or for any user's reliance on the information.

The client is solely responsible for verifying the suitability and compliance of the provided information with the official protocol standards and for ensuring that their implementation or usage of the protocol meets all required specifications and regulations. Any reliance on the information provided is strictly at the user's own risk.

Certification and Compliance Disclaimer: Please be advised that the product described herein has not been certified by any competent authority or organization responsible for protocol standards. INACKS do not guarantee that the chip meets any specific protocol compliance or certification standards.

It is the responsibility of the client to ensure that the final product incorporating this product is tested and certified according to the relevant protocol standards before use or commercialization. The certification process may result in the product passing or failing to meet these standards, and the outcome of such certification tests is beyond our control.

INACKS disclaim any liability for non-compliance with protocol standards and certification failures. The client acknowledges and agrees that they bear sole responsibility for any legal, compliance, or technical issues that arise due to the use of this product in their products, including but not limited to the acquisition of necessary protocol certifications.