

User and Installation Guide: Arduino Modbus Data Logger for PM8000

An electronic version of this manual can be found at: <https://git.warky.dev/vivarox>

1. Introduction

This guide will walk you through the setup and usage of the Arduino-based Modbus data logger for the Schneider PowerLogic PM8000. This project allows you to read data from the PM8000 using Modbus protocol and log it to an SD card with timestamps.

2. Hardware Requirements

- Arduino Board (MEGA 2560 recommended, UNO for simpler projects)
- RS485 to TTL Module
- SD Card Module
- SD Card (formatted as FAT32)
- RTC (Real Time Clock) Module (DS3231)
- Power Supply (9V AC Adapter with barrel jack)
- Two LEDs for status indication
- Jumper wires
- Breadboard (optional, for prototyping)

3. Hardware Setup

3.1 Wiring Instructions

1. RS485 Module to Arduino:

- RO (Receiver Output) → Arduino RX (pin 8)
- DI (Driver Input) → Arduino TX (pin 7)
- DE (Driver Enable) & RE (Receiver Enable) → Arduino digital pin 4
- VCC → 5V on Arduino
- GND → GND on Arduino
- A & B (RS485 differential pair) → Modbus device (PM8000)

2. SD Card Module to Arduino:

- VCC → 5V on Arduino
- GND → GND on Arduino
- MOSI → MOSI (pin 51 on MEGA, pin 11 on UNO)
- MISO → MISO (pin 50 on MEGA, pin 12 on UNO)
- SCK → SCK (pin 52 on MEGA, pin 13 on UNO)
- CS (Chip Select) → digital pin 10

3. RTC Module to Arduino:

- VCC → 5V on Arduino
- GND → GND on Arduino
- SDA → SDA (pin 20 on MEGA, pin A4 on UNO)
- SCL → SCL (pin 21 on MEGA, pin A5 on UNO)

4. LED Indicators:

- LED A → digital pin 3
- LED B → digital pin 5

3.2 RS485 Connection to Modbus Device

1. Locate the RS485 connection points on your Modbus device (PM8000):

- There should be terminals labeled '+' (positive) and '-' (negative).
- These correspond to the 'A' and 'B' lines of the RS485 communication.

2. Connect the RS485 module to the Modbus device:

- Connect the ‘A’ wire from the RS485 module to the ‘+’ terminal on the Modbus device.
- Connect the ‘B’ wire from the RS485 module to the ‘-’ terminal on the Modbus device.

3. Optional ground connection:

- If you experience communication issues, you may need to connect a ground wire.
- Connect the GND from the RS485 module to the GND or shield terminal on the Modbus device.

4. Troubleshooting RS485 connection:

- If you’re experiencing communication issues, try reversing the ‘A’ and ‘B’ wires.
- Some devices may label their terminals differently, so reversing the connections might be necessary.

Note: Always refer to the specific documentation of your Modbus device (PM8000) for the correct terminal labels and any device-specific connection requirements.

3.3 Power Connection

Connect the 9V AC Adapter to the Arduino’s power jack.

3.4 SD Card Preparation

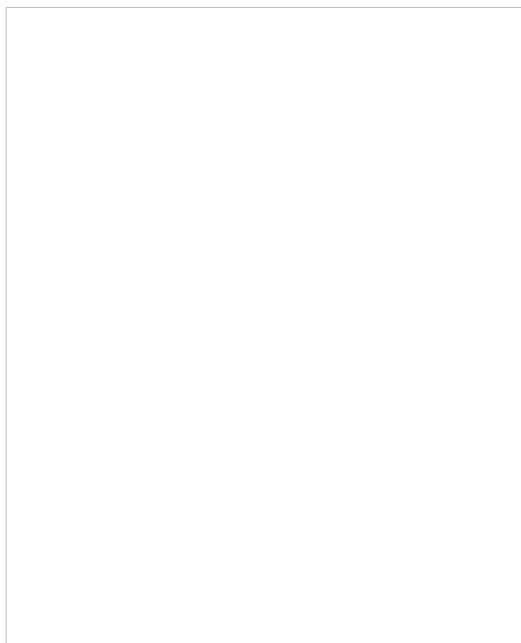
1. Insert the SD card into your computer.
2. Format the SD card as FAT32:
 - On Windows: Right-click the SD card in File Explorer, select “Format”, and choose “FAT32” as the file system.
 - On Mac: Use Disk Utility, select the SD card, click “Erase”, and choose “MS-DOS (FAT)” as the format.
 - On Linux: Use the `mkfs.vfat` command in terminal.
3. After formatting, safely eject the SD card from your computer.
4. Insert the formatted SD card into the SD Card Module connected to your Arduino.

3.5 Assembled Device Images

Below are images of the fully assembled Arduino Modbus Data Logger, with annotations highlighting key connections:

1. Overall Assembly

- Full assembly of the Arduino Modbus Data Logger
- SD Card Module wired to Arduino
- RS485 module connected to Arduino and Modbus device
- RTC module wired to Arduino



2. Front View



3. SideView



Note: These images serve as a visual guide for the assembly process. Always double-check your connections against the wiring instructions provided earlier in this guide.

To see videos of the device working, goto: <https://git.warky.dev/vivarox/arduino/src/branch/main/docs/video>

4. Software Setup

4.1 Required Libraries

Install the following libraries using the Arduino IDE Library Manager:

- ModbusMaster
- SdFat
- RTCLib
- NeoSWSerial

4.2 Arduino IDE Setup

1. Download and install the Arduino IDE from [arduino.cc](https://www.arduino.cc)
2. Open the Arduino IDE
3. Go to Tools → Board and select your Arduino board (MEGA 2560 or UNO)

4. Go to Tools → Port and select the COM port for your Arduino

4.3 Uploading the Code

1. Download the project files (main sketch and `register_map_pm8000.h`)
2. Open the main sketch in the Arduino IDE
3. Verify that the `register_map_pm8000.h` file is in the same directory as the main sketch
4. Click the “Upload” button to compile and upload the code to your Arduino

5. Usage Instructions

1. Ensure all hardware connections are secure
2. Insert the FAT32-formatted SD card into the SD card module
3. Power on the Arduino using the AC adapter
4. The system will automatically begin logging data:
 - LED A will blink to indicate successful data writing and transmission
 - LED B will light up if there are any errors (SD card issues, RTC problems, Modbus communication errors)
5. Data will be logged to a file named `pm8k_YYYYMMDD.csv` on the SD card
6. To retrieve data, power off the Arduino and remove the SD card
7. Insert the SD card into a computer to view and analyze the logged data

6. Troubleshooting

- If LED B is constantly lit:
 - Check SD card insertion and formatting (ensure it's FAT32)
 - Verify RTC module connections
 - Ensure proper Modbus wiring to the PM8000
- If no data is being logged:
 - Check the Modbus device ID and baud rate in the code
 - Verify the register map in `register_map_pm8000.h`
 - Ensure the SD card is properly formatted as FAT32
- If you're experiencing Modbus communication issues:
 - Verify the RS485 connections are correct (A to +, B to -)
 - Try reversing the A and B connections
 - Add a ground wire between the RS485 module and the Modbus device if not already present
 - Check the baud rate settings in the code match the Modbus device settings
 - Ensure the correct Modbus device ID is set in the code
- For detailed error messages, connect the Arduino to a computer and open the Serial Monitor in the Arduino IDE (set to 9600 baud)

7. Customization

To modify the registers being read:

1. Open `register_map_pm8000.h` in a text editor
2. Add, remove, or modify register entries in the `registers` array
3. Reupload the code to the Arduino

Note: Be mindful of memory limitations, especially when using an Arduino UNO.

8. Maintenance

- Regularly check the SD card for available space
- Replace the SD card or transfer data to a computer when it's nearly full
- Keep the hardware clean and dry to ensure longevity

Further help

For technical support or further customization, please contact Vivarox EMS or Warky Devs.
Via support@warky.dev

Designed by www.warky.dev

