MightyWatt Communication Protocol Description

Board revision: 2.5
Minimum firmware version: 2.5.5
Guide revision: A (2016-06-08)

Introduction

The communication protocol was created to have low overhead because the serial communication on Arduino is relatively slow. Periodic data transfers are using unsigned raw data, no negative numbers are transferred. Some one-shot functions transfer data as ASCII text.

Abbreviations

In the following text, there will be some abbreviations and terms:

Abbreviation or term	**Meaning**
MSB	most significant bit/byte
LSB	least significant bit/byte
Firmware	Arduino sketch
Host	Controlling computer
Load	MightyWatt

Watchdog

The load has a watchdog, which will set the load to zero current after a certain period. To reset the watchdog, it is necessary to send commands through the serial line. The purpose is to automatically stop the load when the communication is lost. Watchdog may be disabled in the firmware. Otherwise, approximately one transfer every four seconds is required to keep the watchdog counter from overflowing.

Data transfer description

Each data transfer is initiated by the host, load only responds to requests.

First byte is always a command byte. This byte determines what the load is supposed to do. It is a bit field.

Bit	7	6	5	4	3	2	1	0
Function	ST/SD	DSL MSB	DSL LSB	C4 (MSB)	C3	C2	C1	C0 (LSB)

ST/SD: Set or send bit. 1 = SET command, 0 = SEND command.

DSL: Data stage length in bytes.

C4–C0: Command ID.

Each command byte can be followed by up to 3 data bytes (send everything in one transfer). The number of data bytes is DSL. Data stage is sent MSB first.

SEND command

This command will instruct the load to send data to the host. What data is returned depends on the value of Command ID. Three Command IDs are supported:

31: IDN. This is Identify. It returns “MightyWatt” as ASCII text, including newline. It is suitable for the identification of the connected device.

30: QDC. This is Query Device Capabilities. This returns 9 lines of ASCII text with the following data:

Line	Data
0	Firmware version (as defined in firmware)
1	Board revision (as defined in firmware)
2	Maximum current supported by DAC (calculated from calibration values), mA
3	Maximum current supported by ADC (calculated from calibration values), mA
4	Maximum voltage supported by DAC (calculated from calibration values), mV
5	Maximum voltage supported by ADC (calculated from calibration values), mV
6	Maximum power (as defined in firmware)
7	Input resistance of voltmeter (as defined in firmware)
8	Temperature threshold for overheat (as defined in firmware)

28: SERIES_RESISTANCE_ID. This returns the value of series resistance used to compensate for cables (or any other) series resistance in 4-wire mode for the correct evaluation of dissipated power. Value is returned as ASCII text, the unit is mΩ.

0 (or anything undefined): Measurement report in 7 bytes.

Byte	Data
0	Measured current MSB, mA
1	Measured current LSB, mA
2	Measured voltage MSB, mV
3	Measured voltage LSB, mV
4	Measured temperature, °C
5	Remote status (0 = local, 1 = remote)
6	Load status

Load status is a bit field with flags indicating whether something wrong has happened. If it reads zero, everything is OK. The bits in load status have the following meaning:

Bit	Meaning
0	Current overload
1	Voltage overload
2	Power overload
3	Overheat
4–7	Reserved

SET command

Each SET command will return the measurement report so there is no need to request it after SET command has been sent.

Commands are followed by 1 to 3 byte data stage.

0: Constant current mode. This command is followed by 2 byte data stage:

Byte	Data
0	Set current MSB
1	Set current LSB, mA

1: Constant voltage mode. This command is followed by 2 byte data stage:

Byte	Data
0	Set voltage MSB
1	Set voltage LSB, mV

2: Constant power mode. This command is followed by 3 byte data stage:

Byte	Data
0	Set power MSB
1	Set power
2	Set power LSB, mW

3: Constant resistance mode. This command is followed by 3 byte data stage:

Byte	Data
0	Set resistance MSB
1	Set resistance
2	Set resistance LSB, mΩ

4: Constant voltage mode with inverted phase. This command is followed by 2 byte data stage:

Byte	Data
0	Set voltage MSB
1	Set voltage LSB, mV

5: Maximum power point tracker. This command is followed by 2 byte data stage:

Byte	Data
0	Starting current MSB
1	Starting current LSB, mV

28: Series resistance set. This command is followed by 2 byte data stage:

Byte	Data
0	Set series resistance MSB
1	Set series resistance LSB, mΩ

29: Remote set. This command is followed by 1 byte data stage:

Byte	Data
0	Set remote function: 0 = local (2-wire mode), anything else = remote (4-wire)

Examples

1) **Set constant voltage, 6.5 V:**
   command byte: 0b**11000001** (bit 7: 1 = SET; bit 6 and 5: 0b10 = 2 = number of following data bytes; bit 4–0: 1 = constant voltage mode)
   data byte 0: 0b**00011001** (25 = 25×256 = 6400 mV)
   data byte 1: 0b**01100100** (100 = 100 mV)
   After the completion of this transfer the load will return the measurement report automatically.
2) **Read device capabilities:**
   command byte: 0b**00011110** (bit 7: 0 = SEND, bit 6 and 5: 0 = no data stage; bit 4–0: 30 = QDC)
   After returning the device capabilities, there will be no measurement report transfer. However, the watchdog will be reset.
3) **Read measurement report:**
   command byte: 0b**00000000** (bit 7: 0 = SEND, bit 6 and 5: 0 = no data stage; bit 4–0: 0 = request measurement report)