MightyWatt R3 Communication Protocol Description

Board revision: 3.0
Minimum firmware version: 3.0.0
Arduino IDE tested: 1.8.1 (Arduino.cc)
Guide revision: A (2017-02-10)

Introduction

This document describes how MightyWatt R3 communicates with computer control application. Periodic data transfers use unsigned raw data; no negative numbers are transferred. Functions that should be called once during the initialization return data as ASCII text.

To check the integrity of data, CRC-16 CCITT (polynomial 0x1021) is appended as the last two bytes (LSB first) of each data transfer from computer to MightyWatt and each **Measurement and status** report from MightyWatt to computer. CRC is not used for the initialization string transfers (**IDN**, **QDC** and possible **Error messages** list).

Abbreviations

The following abbreviations are used through the text:

Abbreviation or term	**Meaning**
MSB	Most significant bit/byte
LSB	Least significant bit/byte
Firmware	Arduino sketch
Host	Controlling computer
Load	MightyWatt R3
CRC	Cyclic redundancy check, specifically the CRC-16 CCITT

Watchdog

The load has a watchdog, which will set the load to zero current after a certain period (COMMUNICATION_WATCHDOG_TIMEOUT in CommunicationWatchdog.h). 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 or become 100% corrupted.

Data transfer description

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

First byte is a header which determines what the load is supposed to do. Header is a bit field.

Bit	7	6	5	4	3	2	1	0
Function	R/W	DL MSB	DL LSB	C4 (MSB)	C3	C2	C1	C0 (LSB)

R/W: READ or WRITE bit. 1 = Write to load (set something), 0 = Read from load (get data).

DL: Data stage length. 0 = 0 bytes (no data stage), 1 = 1 byte, 2 = 2 bytes, 3 = 4 bytes.

C4–C0: Command ID.

Each header can be followed by up to 4 data bytes. The number of data bytes is defined by DL. Data stage is sent LSB first. After the optional data bytes, the CRC is appended. Thus, one transfer from computer to the load is composed of header, optional data stage and CRC.

READ command

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

1: Measurement and status report:

Byte	Data
0	Measured current LSB, µA
1	Measured current, µA
2	Measured current, µA
3	Measured current MSB, µA
4	Measured voltage LSB, µV
5	Measured voltage, µV
6	Measured voltage, µV
7	Measured voltage MSB, µV
8	Measured temperature, °C
9	Status flags (see below)
10	Error flags (see below) LSB
11	Error flags
12	Error flags
13	Error flags MSB

Status flags is a bit field:

Bit	Meaning
0	0 = constant current mode, 1 = constant voltage mode
1	0 = high voltage range, 1 = low voltage range
2	0 = high current range, 1 = low current range
3	0 = LED off, 1 = LED on
4	0 = fan off, 1 = fan on
5	0 = 2-wire mode, 1 = 4-wire mode
6–7	reserved

Error flags is a bit field; each set bit means that the corresponding error is active. The error description can be obtained by command "4", *Error messages*.

2: IDN. This is Identify. Load returns "MightyWatt R3" as ASCII text, including CR+LF. It is suitable for the identification of the connected device. No CRC is transmitted in this response.

3: QDC. This is Query Device Capabilities. This returns 10 lines of ASCII text described below. No CRC is transmitted in this response.

Line	Data
0	Calibration date (from Configuration.h)
1	Firmware version (from Configuration.h)
2	Board revision (from Configuration.h)
3	Maximum current supported by DAC (from calibration values in Configuration.h), μA
4	Maximum current supported by ADC (from calibration values in Configuration.h), μA
5	Maximum voltage supported by DAC (from calibration values in Configuration.h), μV
6	Maximum voltage supported by ADC (from calibration values in Configuration.h), μV
7	Maximum power (from Configuration.h), μW
8	Input resistance of voltmeter (from Configuration.h), mΩ
9	Temperature threshold for overheat (from Limiter.h), °C

4: Error messages. This returns text description of possible errors reported by the load. Command returns up to 32 lines. No CRC is transmitted in this response. The actual errors are then reported in a bit field in **Measurement and status** report.

WRITE command

Commands are followed by a 1, 2 or 4-byte data stage and end with a CRC.

1: Constant current mode. This command is followed by a 4-byte data stage:

Byte	Data
0	Set current LSB, µA
1	Set current, µA
2	Set current, µA
3	Set current MSB, µA

2: Constant voltage mode. This command is followed by a 4-byte data stage:

Byte	Data
0	Set voltage LSB, µV
1	Set voltage, µV
2	Set voltage, µV
3	Set voltage MSB, µV

3: Constant power mode. This command is followed by a 4-byte data stage:

Byte	Data
0	Set power LSB, µW
1	Set power, µW
2	Set power, µW
3	Set power MSB, µW

4: Constant resistance mode. This command is followed by a 4-byte data stage:

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

5: Software-controlled constant voltage mode. This command is followed by a 4-byte data stage:

Byte	Data
0	Set voltage LSB, µV
1	Set voltage, µV
2	Set voltage, µV
3	Set voltage MSB, µV

6: Maximum power point tracker. This command is followed by a 4-byte data stage:

Byte	Data
0	Initial voltage LSB, µV
1	Initial voltage, µV
2	Initial voltage, µV
3	Initial voltage MSB, µV

If the initial voltage is zero, the load will attempt to set it automatically – to 90% of the open-circuit voltage.

10: Series resistance. This command is followed by a 4-byte data stage:

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

11: 2-wire or 4-wire voltage sensing. This command is followed by a 1-byte data stage:

Byte	Data
0	0 = set 2-wire mode (default), 1 = set 4-wire mode

12: Measurement speed. This command is followed by a 1-byte data stage:

Byte	Data
0	0 = internal ADC autoranging & filter* disabled (not used by the Windows control application), 1 = internal ADC autoranging enabled, filter* disabled, 2 = internal ADC autoranging & filter* enabled (default)

*) Filter is an internal (in the firmware) 42-period (~300 ms) triangular weighted moving average filter.

13: Fan rules. This command is followed by a 1-byte data stage:

Byte	Data
0	0 = always on (default), 1 = automatic – cool, 2 = automatic – quiet

14: LED rules. This command is followed by a 1-byte data stage:

Byte	Data
0	Flag word, defines when LED is on; rules can be combined

15: LED brightness. This command is followed by a 1-byte data stage:

Byte	Data
0	Sets duty cycle of the PWM that controls LED brightness; 255 = 100%

Examples

1) **Set constant voltage, 6.5 V:**
   command byte: 0b**11100010** (bit 7: 1 = WRITE; bit 6 and 5:
   0b11 (3) = number of following data bytes is 4; bit 4–0: 0b10 (2) = constant voltage mode)
   data byte 0: 0b**10100000** (160 ⇨ 160 µV)
   data byte 1: 0b**00101110** (46 ⇨ 46·256 = 11 776 µV)
   data byte 2: 0b**01100011** (99 ⇨ 99·256^2 = 6 488 064 µV)
   data byte 3: 0b**00000000** (0)
   CRC LSB: 0b**01000111** (71)
   CRC MSB: 0b**01010110** (86)

2) **Read device capabilities:**
   command byte: 0b**00000011** (bit 7: 0 = READ, bit 6 and 5: 0 = no data stage;
   bit 4–0: 0b11 (3) = QDC)
   CRC LSB: 0b**01100011** (99)
   CRC MSB: 0b**00110000** (48)