PiRail

A raspberry pi controlled model railway

Contents

[General Overview 2](#_Toc505889974)

[Websocket Protocol 2](#_Toc505889975)

[Admin Controls (0x80 Flag) 2](#_Toc505889976)

[Train (0x40 Flag) 3](#_Toc505889977)

[Track (0x20 Flag) 3](#_Toc505889978)

[General (0x10 Flag) 4](#_Toc505889979)

[Circuit of RNet 5](#_Toc505889980)

[Protocol of RNet 6](#_Toc505889981)

[General 6](#_Toc505889982)

[Output 6](#_Toc505889983)

[Input 7](#_Toc505889984)

[Set Device Parameters 8](#_Toc505889985)

[EEPROM on Arduino 9](#_Toc505889986)

# General Overview



# Websocket Protocol

Each Websocket message is send as a binary packet.

You can register to certain topics as a Websocket client using the protocol properties. 255 registers you to all the topics.

The messages are divided into 4 different categories. The system checks for the first set bit. The packet belong to a category if the first bit equals the flag.

## Admin Controls (0x80 Flag)

### Clear track

Delete all the track from current layout.

0x80

### Reload track

Rescan the track for modules. They still need to be ‘joined’ together.

0x81

### Reload Previous track

Reload a layout from a previous setup. They don’t have to be ‘joined’ together

0x82

### Reset Switches

Reset all switches to default

0x83

### Toggle (Light) Output

Toggle output of modules on or off.

0x84

### All train back to depot

Send all trains back to depot, only if depot space is available.

0x88

### Force Switch

Force a switch to a state independent of reserved state.

0xA0

### Emergency Stop, Admin authority

Set emergency stop. Can only be release with admin authority (the send code).

0x90 Admin Code 2 bytes (0-65534)

### Emergency Release, Admin authority

Release Emergency stop.

0x91 Admin Code 2 bytes (0-65534)

### Enable Admin authority for this connection

0xFF Password hash length Password hash

## Train (0x40 Flag)

### Set new train

0x40

### Speed control

0x41

### Function control

0x42

### Operation change

0x43

### Add Route

0x44

### Delete Route

0x45

## Track (0x20 Flag)

### Toggle Switch *Mostly from client to server* Toggle a switch from diverging to straight or vice versa

|  |  |  |
| --- | --- | --- |
| 0x20 | AAAA AAAA | xBBB BBBB |

AAAA AAAA = Module number of switch  
 xBBB BBBB = Switch ID

### MSSwitch toggle up *Mostly from client to server* Increase the state of a MSSwitch

|  |  |  |
| --- | --- | --- |
| 0x21 | AAAA AAAA | xBBB BBBB |

AAAA AAAA = Module number of switch  
 xBBB BBBB = MSSwitch ID

### MSSwitch toggle down *Mostly from client to server* Decrease the state of a MSSwitch

|  |  |  |
| --- | --- | --- |
| 0x22 | AAAA AAAA | xBBB BBBB |

AAAA AAAA = Module number of switch  
 xBBB BBBB = MSSwitch ID

### Set switch *Mostly from client to server* Set a (MS)Switch to a specific state. 0 = Straight, 1 = Diverging.

|  |  |  |  |
| --- | --- | --- | --- |
| 0x23 | AAAA AAAA | BCCC CCCC | DDDD DDDD |

AAAA AAAA = Module number of switch  
 Bxxx xxxx = Set if MSSwitch  
 xCCC CCCC = (MS)Switch ID  
 DDDD DDDD = New state

### Set switch reserved

0x24

### Set switches for route *Mostly from client to server* Server Calculates a route from point A to point B and set the switches accordingly and reserves them.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x25 | AAAA AAAA | BBBB BBBB | CCCC CCCC | DDDD DDDD |

AAAA AAAA = Module number of point A  
 BBBB BBBB = Block of point A  
 CCCC CCCC = Module number of point B  
 DDDD DDDD = Block of point B

**Broadcast track occupation***Only from server to client*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0x26 | AAAA AAAA | BBBB BBBB | CxxD EEEE | FFFF FFFF | AAAA AAAA… |

AAAA AAAA = Module number of block  
 BBBB BBBB = Block ID  
 Cxxx xxxx = Direction of block, set when block is reversed  
 xxxD xxxx = Set when block is occupied  
 xxxx EEEE = State of block: 0000 free, 0001 amber, 0010 red, 0011 unknown, 0100 ghost, 0101 blue  
 FFFF FFFF = ID of train, 0 if not occupied  
A – F is repeated for all blocks

**Broadcast states of switches***Only from server to client*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0x27 | AAAA AAAA | BCCC CCCC | DDDD DDDD | EEEE EEEE | AAAA AAAA… |

AAAA AAAA = Module number of switch  
 Bxxx xxxx = Set if it is a Multi-state switch  
 xCCC CCCC = (MS)Switch ID  
 DDDD DDDD = New State  
 EEEE EEEE = Number of states, only for a MSSwitch, skip for a normal switch  
A – E is repeated for all (ms)switches

## General (0x10 Flag)

### Emergency stop

Set emergency stop, can be released by everyone

0x10

### Emergency stop

Release emergency stop. Can release emergency stop commenced by admin

0x11

### New Message

0x12

### Message Clear

0x13

# Circuit of RNet

There are 4 RNet devices in the network: COM interface, RNet controller, RNet output module, RNet input module.

# Protocol of RNet

## General

### Report ID

|  |  |  |
| --- | --- | --- |
| 0x00 | DevID | Checksum |

All devices sends this packet at startup, so that the controller knows which modules are present.

### Set Emergency STOP

|  |  |
| --- | --- |
| 0x01 | Checksum |

### Release Emergency STOP

|  |  |
| --- | --- |
| 0x02 | Checksum |

### Set Power ON

|  |  |
| --- | --- |
| 0x03 | Checksum |

### Set Power OFF

|  |  |
| --- | --- |
| 0x04 | Checksum |

### Set Acknowledge

|  |  |  |
| --- | --- | --- |
| 0x7F | DevID | Checksum |

## Output

### Toggle Single Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x10 | DevID | Output Low | Output High | Checksum |

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

Toggles the output of the specified ID.

### Pulse Single Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x11 | DevID | Output Low | Output High | Checksum |

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

Send a pulse on the output of the specified ID.

### Toggle Blink Single Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x12 | DevID | Output Low | Output High | Checksum |

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

Toggles the output of the specified ID.

### Toggle Multiple Addresses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x13 | Length | DevID | Output1 Low | Output1 High | … | Checksum |

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

Toggles multiple addresses

### Set All output Addresses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x14 | Length | DevID | Output1 | Output2 | … | Checksum |

Output1: Output & 0x7F  
Output2: (Output >> 7) & 0x7F  
Output3: (Output >> 14) & 0x7F

Set all output to the specified bytes. Output 1 contains the ID of 0-6, output 2 contains 7-13, and so on. NOT IMPLEMENTED

### Set Blink mask of all addresses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x15 | Length | DevID | Output1 | Output2 | … | Checksum |

Output1: Output & 0x7F  
Output2: (Output >> 7) & 0x7F  
Output3: (Output >> 14) & 0x7F

Set the blink mask to the specified bytes. Output 1 contains the ID of 0-6, output 2 contains 7-13, and so on. NOT IMPLEMENTED

### Post all Output

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x16 | Length | DevID | Output1 | Output2 | … | Checksum |

Output1: Output & 0x7F  
Output2: (Output >> 7) & 0x7F  
Output3: (Output >> 14) & 0x7F

### Request Read all Output

|  |  |  |
| --- | --- | --- |
| 0x47 | DevID | Checksum |

***Response:*** Post all Output (0x07)

## Input

### Post Single Input Address

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x05 | DevID | Output Low | Output High | Checksum |

DevID: ID & 0x7F

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

### Post Multiple Addresses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x06 | Length | DevID | Output1 Low | Output1 High | … | Checksum |

DevID: ID & 0x7F

Output Low: ID & 0x7F  
Output High: (ID >> 7) & 0x7F

### Post All input Addresses

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x07 | Length | DevID | Output1 | Output2 | … | Checksum |

DevID: ID & 0x7F

Output1: Output & 0x7F  
Output2: (Output >> 7) & 0x7F  
Output3: (Output >> 14) & 0x7F

### Post all Input

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x08 | Length | DevID | Output1 | Output2 | … | Checksum |

DevID: ID & 0x7F

Output1: Output & 0x7F  
Output2: (Output >> 7) & 0x7F  
Output3: (Output >> 14) & 0x7F

### Request Read all Inputs

|  |  |  |
| --- | --- | --- |
| 0x4C | DevID | Checksum |

DevID: ID & 0x7F

***Response:*** Post all Input (0x07)

## Set Device Parameters

***The device acknowledges all the changes to its parameters with the acknowledge packet***

### Change Device ID

|  |  |  |  |
| --- | --- | --- | --- |
| 0x50 | Old DevID | New DevID | Checksum |

### Set number of input modules and output modules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0x51 | DevID | Input Modules | Output Modules | Checksum |

### Set blink Interval

|  |  |  |  |
| --- | --- | --- | --- |
| 0x52 | DevID | Interval scalar | Checksum |

Blink interval (ms) =

### Set Pulse length

|  |  |  |  |
| --- | --- | --- | --- |
| 0x53 | DevID | Pulse scalar | Checksum |

Pulse duration (ms) =

### Set Check input Interval

|  |  |  |  |
| --- | --- | --- | --- |
| 0x54 | DevID | Interval scalar | Checksum |

Check interval (ms) =

### Post All EEPROM variables

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0x55 | Length | DevID | Byte 0 | Byte 1 | … | Checksum |

### Request All EEPROM Values

|  |  |  |
| --- | --- | --- |
| 0x59 | DevID | Checksum |

## EEPROM on Arduino

|  |  |  |
| --- | --- | --- |
|  | 0x00 | 0x10 |
| 0x0 |  |  |
| 0x1 | Device ID |  |
| 0x2 | Input Devices |  |
| 0x3 | Output Devices |  |
| 0x4 | Blink scalar |  |
| 0x5 | Pulse scalar |  |
| 0x6 | Check scalar |  |
| 0x7 |  |  |
| 0x8 |  |  |
| 0x9 |  |  |
| 0xA |  |  |
| 0xB |  |  |
| 0xC |  |  |
| 0xD |  |  |
| 0xE |  |  |
| 0xF |  |  |