Tower Serial Communication Protocol V1.3

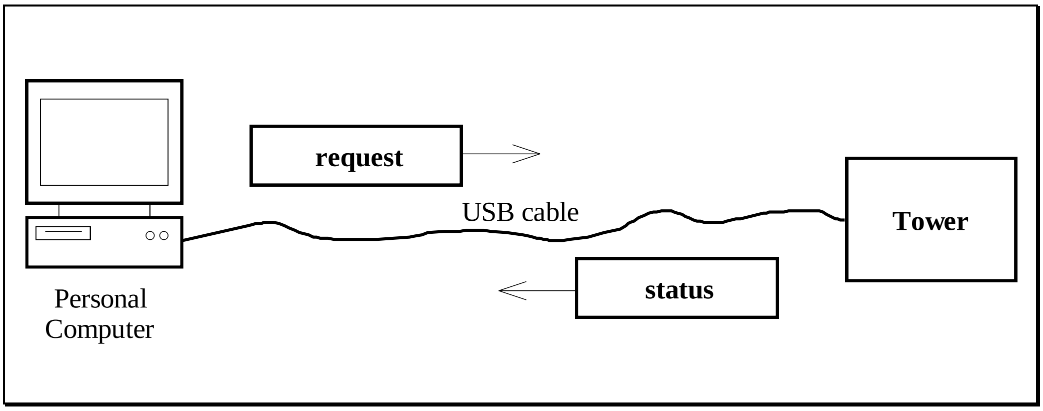
Conceptual view. PC and Tower communication. Packet structure. Packets transmitted from Tower to PC. Packets transmitted from PC to Tower. Packet acknowledgement. PC communication parameters

Introduction

This document outlines the communication protocol between the PC and the Tower microcontroller board.

Conceptual View

The PC or Tower can initiate a transfer of serial information at any time (asynchronously) using the Universal Serial Bus (USB). Information is transferred in packets, with each packet consisting of 5 bytes. A conceptual view of communication between the PC and the Tower is shown below:



Conceptual view of information transfer between PC and Tower

Figure P.1

Sometimes a transmitted packet will request a response from the receiver. In this case, a response to the command is initiated once the command has been carried out (or attempted to be carried out).

P.2

PC and Tower Communication

PC and Tower communicating

A typical picture of the PC and Tower communicating is shown below:

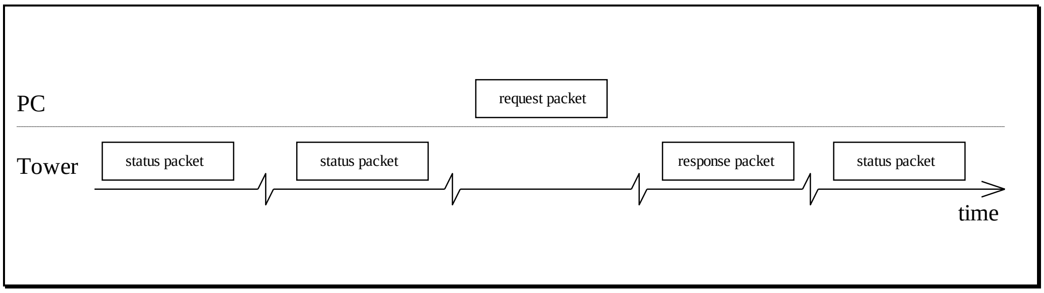


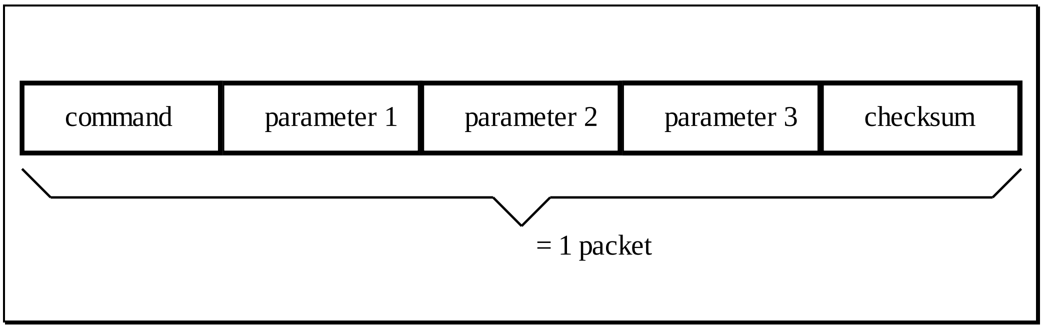
Figure P.2

The Tower is continually updating the PC with information regarding the state of the various inputs and outputs of the module, such as push buttons, LEDs, A/D results and PWM duty cycles. This state information is only sent on start up and when the state of the Tower module changes – this prevents the PC from receiving many packets with the same information, which wastes processing time.

The PC will occasionally send a request packet to the Tower, asking it to carry out some task. It may or may not specify for the Tower to acknowledge that the request has been carried out successfully. If the PC does specify an acknowledgement, then the Tower will respond with a response packet.

Packet Structure

Each packet of information contains 5 bytes as follows:



Packet structure

Figure P.3

The command byte contains a command number, as listed in the tables that follow. The use of the three parameter bytes depends on the particular command being sent. The checksum is the exclusive-or (XOR) of the four preceding bytes and is used as a simple means to detect most transmission errors (corruption of a packet) and for packet synchronization.

Packets Transmitted from Tower to PC

The following table lists the packets that can be sent by the Tower to the PC:

Command	Data Stream
0x04	**Tower Startup** Parameter 1: 0 Parameter 2: 0 Parameter 3: 0 <u>Note:</u> *The Tower will issue this command upon startup to allow the PC to update the interface application and the Tower. Typically, setup data will also be sent from the Towe to the PC.*
0x08	**Flash – Read byte** Parameter 1: address offset (0-7) Parameter 2: 0 Parameter 3: data
0x09	**Special – Master alarm** Parameter 1: 12 Parameter 2: 0 Parameter 3: 2 <u>Note:</u> *If the PC receives this then the Tower is overloaded.*
0x09	**Special – Tower version** Parameter 1: ‘v’ = version Parameter 2: Major Version Number Parameter 3: Minor Version Number (out of 100) <u>Note:</u> *e.g. V1.3 has a major version number of 1 and a minor version number of 30.*
0x0A	**Protocol – Mode** Parameter 1: 1 Parameter 2: 0 = asynchronous 1 = synchronous Parameter 3: 0
0x0B	**Tower Number** Parameter 1: 1 Parameter 2: LSB Parameter 3: MSB <u>Note:</u> *The Tower number is an unsigned 16-bit number.*
0x0C	**Time** Parameter 1: hours (0-23) Parameter 2: minutes (0-59) Parameter 3: seconds (0-59)
0x0D	**Tower Mode** Parameter 1: 1 Parameter 2: LSB Parameter 3: MSB <u>Note:</u> *The Tower mode is just an unsigned 16-bit number – it is usually declared as an enumerated type.*

Example

For example, if the Tower were informing the PC that its version number is 1.3, then the packet would be:

0x09	0x76	0x01	0x1E	0x60
command	parameter 1	parameter 2	parameter 3	checksum

Example packet sent from Tower

Figure P.4

Packets Transmitted from PC to Tower

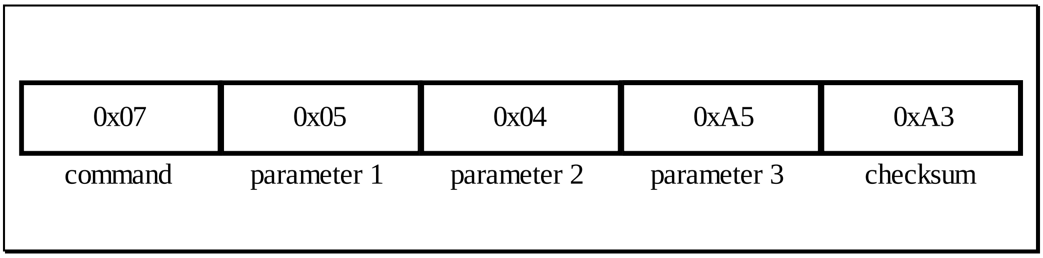
The following table lists the packets that can be sent by the PC:

Command	Data Stream
0x04	**Special - Get startup values** Parameter 1: 0 Parameter 2: 0 Parameter 3: 0 *Note: The PC will issue this command upon startup to retrieve the state of the Tower to update the interface application.*
0x07	**Flash – Program byte** Parameter 1: address offset (0-8) Parameter 2: 0 Parameter 3: data *Note: The address offset has a range of 0x00 to 0x08.* *An address offset of 0x08 will ERASE the* entire *Flash sector.*
0x08	**Flash – Read byte** Parameter 1: address offset (0-7) Parameter 2: 0 Parameter 3: 0 *Note: The address offset has a range of 0x00 to 0x07.*
0x09	**Special – Get values** Parameter 1: ‘g’ Parameter 2: ‘i’ Parameter 3: CR *Note: The Tower will send switch, counter and analog values, which the PC may or may not use.*
0x09	**Special - Start bootloader** Parameter 1: ‘b’ Parameter 2: ‘l’ Parameter 3: CR
0x09	**Special – Toggle debug mode** Parameter 1: ‘d’ Parameter 2: ‘j’ Parameter 3: CR
0x09	**Special – Get version** Parameter 1: ‘v’ Parameter 2: ‘x’ Parameter 3: CR
0x0A	**Protocol – Mode** Parameter 1: 1 = get Protocol mode 2 = set Protocol mode Parameter 2: 0 = asynchronous for a “set”, 0 for a “get” 1 = synchronous for a “set”, 0 for a “get” Parameter 3: 0

0x0B	**Tower Number** Parameter 1: 1 = get Tower number 2 = set Tower number Parameter 2: LSB for a “set”, 0 for a “get” Parameter 3: MSB for a “set”, 0 for a “get” *Note: The Tower number is an unsigned 16-bit number.*
0x0D	**Tower Mode** Parameter 1: 1 = get Tower mode 2 = set Tower mode Parameter 2: LSB for a “set”, 0 for a “get” Parameter 3: MSB for a “set”, 0 for a “get” *Note: The Tower mode is just an unsigned 16-bit number – it is usually declared as an enumerated type.*

Example

For example, if the PC were requesting the Tower to write 0xA5 to EEPROM address 0x405, then the packet would be:



0x07	0x05	0x04	0xA5	0xA3
command	parameter 1	parameter 2	parameter 3	checksum

Example packet sent from PC

Figure P.5

Packet Acknowledgement

The command ID has bit 7 (the most significant bit) reserved for packet acknowledgement purposes. The transmitter sets bit 7 to a 1 to request an acknowledgement, and leaves it cleared to 0 for no required acknowledgement.

The receiver, upon receiving a packet with an acknowledgement request, will firstly attempt to carry out the requested command. It will then either set bit 7 to indicate that the command was carried out successfully (an ACK), or clear it to 0 to indicate that the command could not be carried out (a NAK), and send the packet back to the requester.

For example, if the PC sends the following packet to write 0xA5 to Flash address offset 0x0005:

0x87	0x05	0x00	0xA5	0x27
command	parameter 1	parameter 2	parameter 3	checksum

Figure P.6

then the Tower will respond with the same packet, indicating it was carried out successfully.

However, if the PC asked to write to address 0x1001 (which does not physically exist), then the Tower would respond with:

0x07	0x01	0x10	0xA5	0xB3
command	parameter 1	parameter 2	parameter 3	checksum

Figure P.7

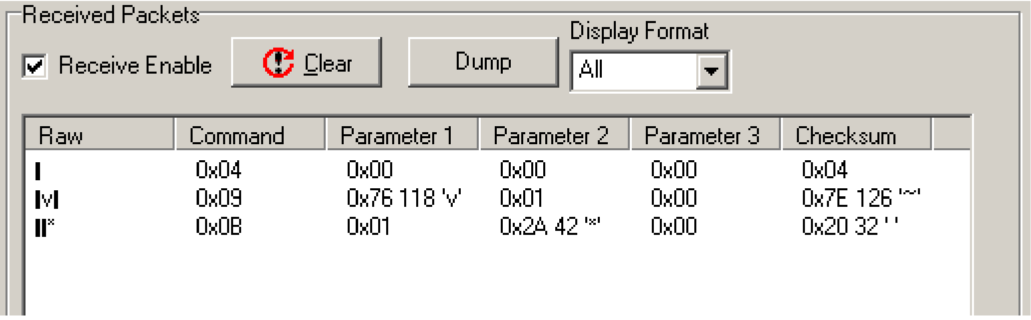
Note that the most significant bit of the command has been cleared, indicating that the requested command could not be carried out.

Example

The Tower PC Interface sends a normal “Startup” packet:

0x04	0x00	0x00	0x00	0x04
command	parameter 1	parameter 2	parameter 3	checksum

The Tower responds with:

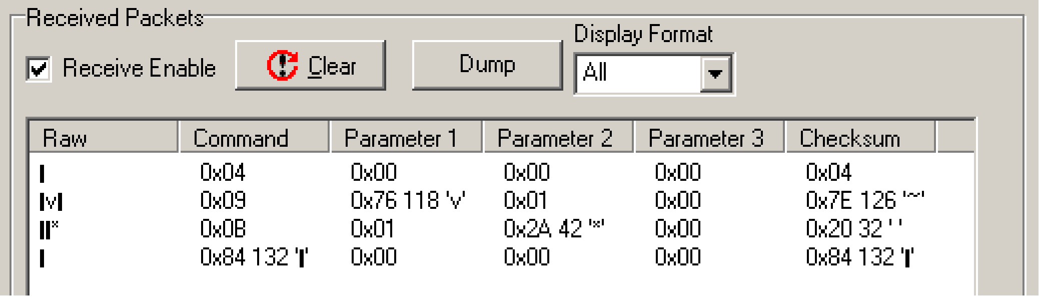


Raw	Command	Parameter 1	Parameter 2	Parameter 3	Checksum
I	0x04	0x00	0x00	0x00	0x04
IvI	0x09	0x76 118 'v'	0x01	0x00	0x7E 126 '~'
II*	0x0B	0x01	0x2A 42 '*'	0x00	0x20 32 ' '

The Tower PC Interface now sends a “Startup” packet with an acknowledgement request (the MSB is set in the command):

0x84	0x00	0x00	0x00	0x84
command	parameter 1	parameter 2	parameter 3	checksum

The Tower responds with:



Raw	Command	Parameter 1	Parameter 2	Parameter 3	Checksum
I	0x04	0x00	0x00	0x00	0x04
IvI	0x09	0x76 118 'v'	0x01	0x00	0x7E 126 '~'
II*	0x0B	0x01	0x2A 42 '*'	0x00	0x20 32 ' '
I	0x84 132 'I'	0x00	0x00	0x00	0x84 132 'I'

Note that the Tower board has responded in the normal way first – it sends the “traditional” 3 packet response – and then it sends an **additional** packet to act as the acknowledgement packet, and in this case with the top bit set to indicate success.

PC Communication Parameters

Baud Rate and Virtual Com Ports

The USB drivers on the PC side implement a virtual COM port.

Baud Rate and Data Format

The baud rate used is selectable and is either 38400 or 115200 baud.

The data format used is 8N1 (8 data bits, no parity, 1 stop bit).