Simple Serial Control Protocol

# Introduction

The Simple Serial Control Protocol (SSCP) is a protocol used by a microcontroller attached to a Parallax Wi-Fi Module to configure the wi-fi module and to use it as a bridge for communicating over a wi-fi connection to a network. The Parallax Wi-Fi Module can be used either in SoftAP mode where the module acts as an access point allowing clients to connect to it or in station mode where the module is a client connected to another wi-fi access point.

# Command Syntax

An SSCP command always begins with a prefix character. The default prefix character is 0xFE but that can be changed to ‘$’ to allow interactive typing of commands from a terminal. After the prefix character, there is an alphanumeric command name. If the command takes parameters, they are preceded by a colon and separated by commas. The command is terminated by a \r (0x0D) character.

Here is an example of a command with no parameters:

0xFE POLL\r

Here is an example of a command with parameters:

0xFE ARG:0,gto\r

Note that there should be no space after the 0xFE character and no space before the 0x0D character, nor should there be spaces before or after the colon or commas.

As shorthand, special tokens can be used in place of command names.

Here is how the POLL command above could be expressed using a command name token:

0xFE 0xE7 0,gto\r

Notice that the colon is not necessary or allowed to separate the command token from the arguments.

The full list of tokens and their numeric character codes are given in a later section.

# SSCP Commands

Note: In the command descriptions below, the prefix character (0xFE or ‘$’) and the terminating character \r (0x0D) are left out. You should assume that each command begins with the prefix character and ends with the terminating character.

## General

(empty)

A request that consists of the prefix character (0xFE or ‘$’) followed immediately by \r (0x0D) will do nothing and respond with “S,0”.

## HTTP

### LISTEN:slot,path

slot is a slot number between 0--1

path is the path part of the URL which can end in a ‘\*’ to indicate a wildcard

returns the S,value on success or E,code on errors

### PATH:conn

conn is a connection number between 0--­3

returns the S,path on success or E,code on errors

### ARG:conn,name

conn is a connection number between 0--­3

name is the name of a query argument

returns the S,value on success or E,code on errors

### POSTARG:conn,name

conn is a connection number between 0-­‐3

name is the name of A

returns the S,value on success or E,code on errors

### BODY:conn

conn is a connection number between 0-­‐3

Returns S,count followed by count bytes of payload or E,code on errors

### REPLY:conn,code,count

(followed by count bytes of payload)

conn is a connection number between 0--3

code is the HTTP status code for the reply body is the body of the reply

returns S,0 on success and E,code on errors

## WebSockets

WSLISTEN:slot,path

slot is a slot number  between 0--1

path is the path part of the URL returns OK on success and ERROR on Errors

  TCP

TCPCONNECT:addr,port

addr is an IP address or a domain name

port is the port number

returns S,conn on success and E,code on errors

TCPDISCONNECT:,conn,port

conn a connection number between 0--3

returns S,0 on success and E,code on errors

## Connections

POLL

returns:

G:conn,count\r

For GET requests

P:conn,count\r

For POST requests

W:conn,0\r

For a WebSocket connections

D:conn,count

For data received from on a WebSocket or TCP connection

N:0,

If nothing is available

E:code,

For errors

RECV:conn

conn is a connection number between 0-­‐3

Returns S,count followed by count bytes of payload or E,code on errors

SEND:conn,count

(followed by count bytes of payload)

conn is a connection number between 0-­‐3

payload is the data to send over the connection

returns S,0 on success and E,code on errors

## Miscellaneous

JOIN:ssid,passwd

Join a wi-fi network

All of the parameters listed above in the HTTP Settings section can be used here as well. For example:

GET:ip-address

Gets the module’s station IP address

GET:pause-time

SET:pause-time,ms

Get/set the number of milliseconds to pause after sending a comma or colon in a serial protocol response.

GET:sscp-enable

SET:sscp-enable,enable

Enable/disable the serial protocol. “0” is disable, “1” is enable.

GET:baud-rate

SET:baud-rate,rate

Get/set the baud rate used to communicate with the MCU.

# Tokens

   SSCP\_TKN\_JOIN               = 0xEF,

   SSCP\_TKN\_GET                = 0xEE,

   SSCP\_TKN\_SET                = 0xED,

   SSCP\_TKN\_POLL               = 0xEC,

   SSCP\_TKN\_PATH               = 0xEB,

   SSCP\_TKN\_SEND               = 0xEA,

   SSCP\_TKN\_RECV               = 0xE9,

   SSCP\_TKN\_LISTEN             = 0xE8,

   SSCP\_TKN\_ARG                = 0xE7,

   SSCP\_TKN\_POSTARG            = 0xE6,

   SSCP\_TKN\_BODY               = 0xE5,

   SSCP\_TKN\_REPLY              = 0xE4,

   SSCP\_TKN\_WSLISTEN           = 0xE3,

   SSCP\_TKN\_TCPCONNECT         = 0xE2,

   SSCP\_TKN\_TCPDISCONNECT      = 0xE1,

# Error Codes

   SSCP\_ERROR\_INVALID\_REQUEST      = -1,

   SSCP\_ERROR\_INVALID\_ARGUMENT     = -2,

   SSCP\_ERROR\_WRONG\_ARGUMENT\_COUNT = -3,

   SSCP\_ERROR\_NO\_FREE\_LISTENER     = -4,

   SSCP\_ERROR\_NO\_FREE\_CONNECTION   = -5,

   SSCP\_ERROR\_LOOKUP\_FAILED        = -6,

   SSCP\_ERROR\_CONNECT\_FAILED       = -7,

   SSCP\_ERROR\_SEND\_FAILED          = -8,

   SSCP\_ERROR\_INVALID\_STATE        = -9,

   SSCP\_ERROR\_INVALID\_SIZE         = -10,

   SSCP\_ERROR\_DISCONNECTED         = -11,

   SSCP\_ERROR\_UNIMPLEMENTED        = -12