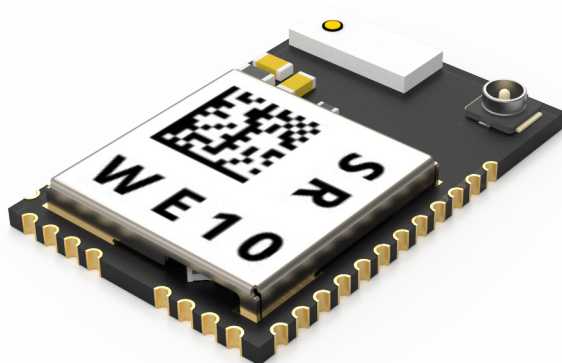


**celium** devices

Short range radio engines



# WE10/20D

Standalone Wi-Fi 802.11 b/g/n and Wi-Fi 802.11 b/g/n+Bluetooth 5.0 Combo Engines

**Command document**

REV 6.1

## Document information

Product family	WE
Product name	WE10/WE20
Document type	Command document
Document number	WE10/WE20 002
Document revision	REV 6.1
Date	24-01-2022

This document applies to the following products

Product Name	Ext. Number	Firmware Version	Status
WE10	Nil	v4.1.0	Initial production
WE20D	Nil	v4.0.0	Initial production

## Document history

Document Name	Revision	Date
WE10/WE20 Command document	REV01	26-12-2020
WE10/WE20 Command document	REV02	26-05-2021
WE10/WE20 Command document	REV03	04-06-2021
WE10/WE20 Command document	REV04	27-07-2021
WE10/WE20 Command document	REV05	21-08-2021
WE10/WE20 Command document	REV 6.1	24-01-2022

# Contents

## 1. Overview

- 1.1 Command format
- 1.2 Response
- 1.3 Events

## 2. Default UART Settings

## 3. Connections

## 4. Commands

### 4.1. System Commands

- 4.1.1 CMD+RESET command to soft reset module
- 4.1.2 CMD+FACRESET Command to reset factory data, uart, ota and fast connect details set to default.
- 4.1.3 CMD+UARTCONF Command to configure UART
- 4.1.4 CMD+GPIO Command to control GPIO

### 4.2. WIFI b/g/n Commands

- 4.2.1 CMD+WIFIMODE Command to set the WiFi mode
- 4.2.2 CMD+CONTOAP Command to connect to an Access Point (AP)
- 4.2.3 CMD+DISCONN Command to disconnect from the AP
- 4.2.4 CMD+SCANAP Command to scan for nearby Access Points (APs)
- 4.2.5 CMD+SETAP Command to configure AP
- 4.2.6 CMD+DHCP Command to set IP Type (DHCP or Static IP)
- 4.2.7 CMD+STAIP Command to set static ip in STA mode
- 4.2.8 CMD+GATEWAYIP Command to set the Gateway IP in AP mode
- 4.2.9 CMD+AUTOCONAP Command to enable auto connection.
- 4.2.10 CMD+MAC Command to set the mac address of the device

### 4.3.TCP/UDP/SSL commands

- 4.3.1 CMD+SETSERVER Command to set TCP/UDP/SSL server
- 4.3.2 CMD+SETCLIENT Command to set TCP/UPD/SSL client
- 4.3.3 CMD+CLOSESKT Command to close TCP/UDP/SSL connection
- 4.3.4 CMD+SKTSENDATA Command to send data via TCP/UDP/SSL connections
- 4.3.5 CMD+SKTRCVDATA Command to get received data from TCP/UDP/SSL connections
- 4.3.6 CMD+SKTAUTORCV Command to enable/disable auto receive data from SKT conn

### 4.4. HTTP Commands

- 4.4.1 CMD+HTTP Command to push data to a web server using HTTP/HTTPs protocol
- 4.4.2 CMD+HTTPDATA Command to make multiple POST and PUT requests in single session
- 4.4.3 CMD+HTTPCLOSE Command to close current HTTP session

#### **4.5 MQTT Commands**

- 4.5.1 CMD+MQTTCONCFG Command to set MQTT connection configuration
- 4.5.2 CMD+MQTTNETCFG command to set MQTT network configuration
- 4.5.3 CMD+MQTTPUB Command to publish data to broker
- 4.5.4 CMD+MQTTSUB Command to subscribe to MQTT topic
- 4.5.5 CMD+MQTTUNSUB Command to unsubscribe to mqtt topic
- 4.5.6 CMD+MQTTSTART Command to start mqtt communication
- 4.5.7 CMD+MQTTSSL Command to set MQTT SSL and CRT config

#### **4.6 BLE Commands (Available Only in WE20D)**

- 4.6.1 CMD+BLEMODE command to select BLE mode
- 4.6.2 CMD+BLESCAN Command to start/ stop BLE scan
- 4.6.3 CMD+BLECON command to connect to BLE device
- 4.6.4 CMD+BLEDISCONN Command to Disconnect from BLE device
- 4.6.5 CMD+BLESERVICE Command to list out supported uuid and characteristic
- 4.6.6 CMD+BLECDATA Command to send data to connected peripheral
- 4.6.7 CMD+BLENTF Command to enable disable notification from central
- 4.6.8 CMD+BLEADV Command to start/stop BLE advertisement
- 4.6.9 CMD+BLEPDATA Command to send data to central device.
- 4.6.10 CMD+BLENAME Command to modify BLE Adv name

#### **4.7 OTA Commands**

- 4.7.1 CMD+OTA Command for over the air update via HTTP protocol
- 4.7.2 CMD+FIRMWARE Command to select firmware image

#### **4.8 Query Commands**

- 4.8.1 CMD?VERSION Command to check Firmware version
- 4.8.2 CMD?UARTCONF Command to check UART configuration
- 4.8.3 CMD?WIFI Command to check WIFI settings and connection
- 4.8.4 CMD?BLENAME command to read BLE ADV name
- 4.8.5 CMD?BLEMODE command to read BLE mode

#### **4.9 Web socket commands**

- 4.9.1 CMD+WSSSTART Command to establish connection with web server
- 4.9.2 CMD+WSSSEND Command to send data to connected web server
- 4.9.3 CMD+WSSCLOSE Command to close web socket connection

#### **4.10 Ping commands**

- 4.10.1 CMD+PING command to request ping response from server

#### **4.11 USER Flash space Config**

- 4.11.1 CMD+USERFLASH command to download data to USER flash using HTTP
- 4.11.2 CMD+USERFLASHREAD command to read data stored in user flash
- 4.11.3 CMD+USERFLASHDEL command to delete data from user flash

## **5. Conditional Commands**

- 5.1 Setting up Static IP.
- 5.2 Setting up Gateway IP
- 5.3 Establishing connection to MQTT broker.
- 5.3 Establishing connection to MQTT broker using CRT.

## **6. Homepage**

- 6.1 Overview
- 6.2 Index page
- 6.3 UPDATE FIRMWARE
- 6.4 CONFIGURE WIFI
- 6.5 USER DATA

## **7. Timing parameters and recommended command flow**

## **8. Response/Status Codes**

## **9. Event Table**

# 1. Overview

WE-XX series of WiFi modules are controlled by a simple, intuitive serial ASCII command interface. There are three different types of data sets that are exchanged between the module and host controller; Commands, Responses and Events.

## 1.1. Commands

These are used to command the WiFi module to perform a particular function. All the commands follow the structure described.

CMD+<command>=<parameter 1>,<parameter 2><CR><LF>

CMD – ‘Op code’ to identify the data stream as a ‘Command’.

<command> - Name of command @refer command list (table x)

<parameters> - Parameters expected by the command @refer command list (table x)

<CR><LF> - Every command ends with “<CR><LF>” characters.

There are 2 types of commands; SET commands and GET commands.

(a) SET commands: These commands are used to set a parameters for a particular function.

eg. “CMD+NAME=test123<CR><LF>”. This command sets the name of the device to ‘test123’.

(b) GET commands: These commands are used to get default/stored parameters from the module.

eg. “CMD?NAME<CR><LF>” This command fetches the name of the module.

WE-XX modules respond to all the commands with a status code.

Total length of any ATCMD should not exceed **1600** bytes for WE10.

Total length of any ATCMD should not exceed **3200** bytes for WE20D.

## 1.2. Response

WE-XX modules send out data packets in response to commands sent by the host controller. All the responses follow the structure described.

RSP=<status>,<parameter 1>,<parameter 2><CR><LF>

RSP – Op code to identify the data stream as a ‘Response’

<status> - Execution status of a particular command that was sent by the host controller. @refer status code list (table x)

<parameters> - Any parameters requested by the host controller @refer command list (table x)

<CR><LF> - Every response ends with “<CR><LF>” characters.

## 1.3. Events

WE-XX module generate events to notify the host controller of special conditions like “peer connection”, “disconnection”, “data reception” etc. All the events follow the structure described

EVT+<event>=<parameter 1>,<parameter 2><CR><LF>

EVT – Op Code to identify the data stream as an ‘Event’

<event> - Name of the event @refer event list (table x)

<parameters> - Any parameters that would be needed by the host controller @refer event list (table x)

<CR><LF> - Every event ends with “<CR><LF>” characters.

## 2. Default UART Settings

Baud rate	38400
HWFC	Disabled
Parity	None
Data bits	8
Stop bits	1

## 3. Connections

### WE10

Pin No.	Pin Name	Description
17	GPIOA_13	RXD
20	GPIOA_14	TXD
09	GPIOA_9	RTS
12	GPIOA_10	CTS
01	CHIP_EN	Chip Enable; Low – OFF, High – ON. Internally Pulled UP

### WE20D

Pin No.	Pin Name	Description
17	PA_13	RXD
20	PA_12	TXD
09	PA_27	RTS
12	PA_28	CTS
01	CHIP_EN	Chip Enable; Low – OFF, High – ON. Internally Pulled UP

## 4. Commands

### 4.1. System command set

#### 4.1.1. CMD+RESET

Command	Description
CMD+RESET<CR><LF>	Command to 'soft reset' the module.

##### USAGE:

CMD+RESET<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

##### Note:

when reset is successful 'EVT+READY' event is generated. @refer Events for details

#### 4.1.2. CMD+FACRESET

Command	Description
CMD+FACRESET<CR><LF>	Command to reset factory data, uart, ota and fast connect details set to default.

##### USAGE:

CMD+FACRESET<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

##### Note:

when reset is successful 'EVT+READY' event is generated. @refer Events for details

#### 4.1.3. CMD+UARTCONF

Command	Description
CMD+UARTCONF=<baudrate>,<databits>,<stop - bits>,<parity>,<flowcontrol>,<configmode><CR><LF>	Command to configure UART



Parameters	
<baudrate>	2400, 4800, 9600, 19200, <b>38400(default)</b> ,57600, 115200, 921600, 1152000
<databits>	5 - 5 bit data 6 - 6 bit data 7 - 7 bit data <b>8 - 8 bit data (default)</b>
<stopbits>	<b>1 - 1 bit stop (default)</b> 2 - 2 bit stop
<parity>	<b>0 - None parity (default)</b> 1 - Odd parity 2 - Even parity
<flowcontrol>	<b>0 – Disable RTS and CTS (default)</b> 1 - Enable RTS and CTS
<configmode>	0 – Set the configuration, do not save in flash.  1 – Save configuration in flash and take effect immediately.  2 – Save configuration in flash and take effect after module resets.

#### USAGE:

CMD+UARTCONF=115200,8,1,0,0,2<CR><LF>

This command sets the baud rate to 115200 and this configuration will take effect only after the module resets.

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### 4.1.4. CMD+GPIO

Command	Description
CMD+GPIO=<mode>,<pin>,<value>,<DIR>,<pull><CR><LF>	Command to control output status of GPIO

Parameters	
<mode>	W - Write R - Read
<pin>	WE10: PA_X should be used PA_X is same as GPIOA_X in datasheet. WE20D: PA_X and PB_X are valid refer WE20 datasheet.  NOTE:pins used for UART,SPI, I2C etc should not be modified using th is command
<value>	1 - High 0 - Low
<DIR>	1 - Pin output
<pull>	0 - Pull None/Pull Default 1 - Pull Up 2 - Pull Down 3 - Open Drain

Response	Description
RSP=00,<value><CR><LF>	<value> - Value set on pin in command

#### USAGE:

CMD+GPIO=W,PA\_4,1,1,0<r><n> - Sets pin PA\_4 to high

CMD+GPIO=W,PA\_4,0,1,0<r><n> - Sets pin\_4 to low

CMD+GPIO=R,PA\_4 - Will give status of PA\_4

## 4.2. WiFi b/g/n command set

### 4.2.1. CMD+WIFIMODE

Command	Description
CMD+WIFIMODE=<mode><CR><LF>	Command to set the WiFi mode
Parameters	
<mode>	<b>1 – Station Mode (default)</b> 2 – Access Point Mode 3 – Concurrent Mode

#### USAGE:

CMD+WIFIMODE=2<CR><LF>

This command puts the module in Access Point mode

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

While using concurrent mode first use CMD+SETAP followed by CMD+CONTOAP.

### 4.2.2. CMD+CONTOAP

Command	Description
CMD+CONTOAP=<ssid>,<password>,<key_id>,<bssid><CR><LF>	Command to connect to an Access Point (AP)
Parameters	
<ssid>	SSID of the AP to which the module needs to be connected.
<password>	Password
	If the AP has an 'Open' connection, No password is required.
<key_id> (optional parameter)	For WEP security, the <key_id> must be 0 – 3. If not set, 0 is used as default.
<bssid> (optional parameter)	Basic Service Set Identifier.
	6 byte Hex number eg: 01FEAB34CCDE

## USAGE:

1. CMD+CONTOAP=CeliumWiFi<cr><lf>

This command connects to 'CeliumWiFi' if the network has no password set.

2. CMD+CONTOAP=CeliumWiFi, 123456789<cr><lf>

This command connects the module to 'CeliumWiFi' if the password of the AP is 123456789.

Response	Description
RSP=<status><CR><LF>	@refer status codes

## Note:

When module connects to the AP, 'EVT+CONTOAP' event is generated. @refer Events for details

when module is not able to connect to the AP within the set time, 'EVT+TIMEOUT' event is generated.  
@refer Events for details

## 4.2.3 CMD+DISCONN

Command	Description
CMD+DISCONN<CR><LF>	Command to disconnect from the Access Point (AP)

## USAGE

CMD+DISCONN<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

## Note:

The module needs to be in 'Station' Mode and needs to be connected to an AP for this command to have effect.

Upon successful disconnection, EVT+DISCONN event is generated. @refer Events for details

## 4.2.4 CMD+SCANAP

Command	Description
CMD+SCANAP<CR><LF>	Command to scan for nearby Access Points (APs)

## USAGE

CMD+SCANAP<CR><LF>

Scans for nearby APs

Response	Description
RSP=<status><CR><LF>	@refer status codes

**Note:**

When an AP is detected 'EVT+SCANAP' is generated. @refer Events for details

When module completes scanning nearby AP EVT+SCANAPDONE is triggered, if APs are detected EVT+SCANAP is generated before EVT+SCANAPDONE. @refer Events for details

#### 4.2.5 CMD+SETAP

Command	Description
CMD+SETAP=<ssid>,<password>,<channel>,<hidden>,<max_connections><CR><LF>	Command to configure AP
Parameters	
<ssid>	Must add prefix '' for special characters, if they are used in APs ssid.
<password> (optional parameter)	If password is set, the security is automatically set to WPA/WPA2. If not the network security is 'open'
<channel> (optional parameter)	Channel Selection (1 – 11)
<hidden> (optional parameter)	<b>0 – Visible (default)</b> 1 – Hidden SSID
<max_connections> (optional parameter)	Maximum number of STAs. 1 – 3 ( <b>default – 3</b> )

**USAGE**

CMD+SETAP=CeliumWiFi,cd123,8,0,3<CR><LF>

This command configures AP to use 'CeliumWiFi' SSID with cd123 as password.

Response	Description
RSP=<status><CR><LF>	@refer status codes

**Note:**

The module should be in AP mode before sending this command.

#### 4.2.6 CMD+DHCP

Command	Description
CMD+DHCP=<mode>,<IP_Type><CR><LF>	Command to set IP Type (DHCP or Static IP)
Parameters	
<mode>	1 – Access Point 2 – Station
<IP_type>	<b>1 – DHCP (Default)</b> 2 – Static

**USAGE**

CMD+DHCP=1,1<CR><LF>

This command sets Dynamic IP to module in AP mode.

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### 4.2.7 CMD+STAIP

Command	Description
CMD+STAIP=<ip>,<gateway_ip>,<subnet_mask_ip><CR><LF>	Command to set static ip in STA mode

Parameters	
<ip>	Sets Static Station IP eg: 192.168.1.2
<gateway_ip> (optional parameter)	Sets Gateway IP
<subnet_mask_ip> (optional parameter)	Sets subnet mask IP

#### USAGE

1. CMD+STAIP=192.168.1.150<CR><LF>

This command sets a static IP for station

2. CMD+DHCP=2,2<CR><LF>

This command makes the static IP effective

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### 4.2.8 CMD+GATEWAYIP

Command	Description
CMD+GATEWAYIP=<start_ip>,<end_ip>,<gateway><CR><LF>	Command to set the Gateway IP in AP mode

Parameters	
<start_ip> (optional parameter)	Sets start IP for the client
<end_ip> (optional parameter)	Sets end IP for the client
<gateway_ip> (optional parameter)	Sets gatewayIP

#### USAGE

CMD+GATEWAYIP= 192.168.99.100,192.168.99.102,192.168.99.1<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### 4.2.9 CMD+AUTOCONAP

Command	Description
CMD+AUTOCONAP=<enable><CR><LF>	Command to enable auto connection.

##### Parameters

<enable>	0 – Disable auto connection. 1 – Enable auto connection.
----------	---

##### USAGE

CMD+AUTOCONAP=1<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

##### Note:

The module should be connected to an AP for this command to be work. @refer CMD+CONTOAP command.

#### 4.2.10 CMD+MAC

Command	Description
CMD+MAC=<mac_address><CR><LF>	Command to set the mac address of the device

##### Parameters

<mac_address>	6 byte mac adress eg: 03ef3b1200df
---------------	---------------------------------------

##### USAGE

CMD+MAC=03ef3b1200df<CR><LF>

This command sets the mac address of the module to 03:EF:3B:12:00:DF

Response	Description
RSP=<status><CR><LF>	@refer status codes

##### Note:

Reboot the module for this command to take effect

## 4.3. TCP/UDP/SSL commands set

### 4.3.1 CMD+SETSERVER

Command		Description
CMD+SETSERVER=<mode>,<local_port><CR><LF>		Command to set TCP/UDP/SSL server
Parameters		
<mode>	0 – TCP 1 – UDP 2 – SSL	
<local_port>	1 – 65535	

#### USAGE

CMD+SETSERVER=0,4000<CR><LF>

This commands sets a TCP server at port 4000

Response	Description
RSP=<status><CR><LF>	@refer status codes

### 4.3.2 CMD+SETCLIENT

Command		Description
CMD+SETCLIENT=<mode>,<remote_addr>,<remote_port>,<local_port><CR><LF>		Command to set TCP/UDP/SSL client
Parameters		
<mode>	0 – TCP 1 – UDP 2 – SSL	
<remote_addr>	IP address ot web address	
<remote_port>	1 – 65535	
<local_port> (optional parameter)	Local port to bind. Only valid for UDP	

#### USAGE

CMD+SETCLIENT=0,192.168.63.100,4000<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

### 4.3.3 CMD+CLOSESKT

Command	Description
CMD+CLOSESKT=<connection_id><CR><LF>	Command to close TCP/UDP/SSL connection
Parameters	
<connection_id>	0 – TCP 1 – UDP 2 – SSL

#### USAGE

CMD+CLOSESKT=conn\_id<CR><LF>

Response	Description
RSP=<status><CR><LF>	@refer status codes

### 4.3.4 CMD+SKTSENDDATA

Command	Description
CMD+SKTSENDDATA=<data_size>,<conn_id>,<dst_ip>,<dst_port>,<data>	Command to send data via TCP/UDP/SSL connections
Parameters	
<data_size>	Size of the data to be sent. Maximum size of the data that can be sent is 1600 bytes
<con_id>	1 – 9, 0 is reserved
<dst_id> (optional parameter)	Destination IP. Used only with UDP server
<dst_pot> (optional parameter)	Destination port. 1 – 65535, Used only with UDP server
<data>	Payload

#### USAGE

CMD+SKTSENDDATA=7,1,,Test123<CR><LF>

This command sends 'Test123' to device with connection id 1.

Response	Description
RSP=<status>,<con_id><CR><LF>	<status> @refer status codes <con_id> - Connection ID to which data was sent.



### 4.3.5 CMD+SKTRCVDATA

Command	Description
CMD+SKTRCVDATA=<con_id>,<size><CR><LF>	Command to get received data from TCP/UDP/SSL connections
Parameters	
<con_id>	Connection ID from the data is received. 1 – 9, 0 is reserved.
<size>	Size of the data received
<b>USAGE</b>	
On receiving 'EVT+SKTDATA' event, the Host controller can send the 'CMD+RECDATA' to get the data received.	
CMD+SKTRCVDATA=1, 25<CR><LF>	
This command receives 25 bytes of data from connect id 1.	
Response	Description
RSP=<status><CR><LF>	@refer status codes

### 4.3.6 CMD+SKTAUTORCV

Command	Description
CMD+SKTAUTORCV=<enable>	Command to enable/disable auto reception of data from TCP/UDP/SSL connections
Parameters	
<enable>	1 – Enable 2 – Disable
<b>USAGE</b>	
CMD+SKTAUTORCV=1<CR><LF>	
This command enables auto reception.	
Response	Description
RSP=<status><CR><LF>	@refer status codes

## 4.4. HTTP commands set

### 4.4.1 CMD+HTTP

Command	Description
CMD+HTTP=<[host]>,<[path]>,<[method]>,<[port]>,<[ssl_en]>,<[<-fields>],<[http_content_type]>,<[content_length]><CR><LF>	Command to push data to a web server using HTTP/HTTPs protocol

## Parameters

<[host]>	Host server address
<[path]>	Action and api key for server access (max length 3072 for GET method WE20D, Max length 1500 for WE10 for GET method , 500 for PUT and POST methods)
<[method]>	1 – Get 2 – Post 3 – Put
<[port]>	Port through which communication needs to happen (should be 44      3 for <[ssl_en]> 1)
<[ssl_en]>	0 – HTTP 1 – HTTPS
<[field]> (optional parameter)	HTTP data to be transferred (Max length 3072 for <b>WE20D</b> , Max length 1500 for <b>WE10</b> )
<[http_content_type]> (optional parameter)	max length 50
<[content_length]> (optional parameter)	total length of HTTP data to be transferred in current session, if data transferred through multiple POST and PUT requests

## USAGE

If you want to send data to link [https://api.thingspeak.com:80/update?api\\_key=55LNK1KBQKWC0EUX&field1=0](https://api.thingspeak.com:80/update?api_key=55LNK1KBQKWC0EUX&field1=0)

### GET method(without SSL)

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX&field1=28&field2=30],[1],[80],[0]<CR><LF>

### GET method with content type(without SSL)

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX&field1=28&field2=30],[1],[80],[0],[application/x-www-form-urlencoded]<CR><LF>

### GET method with SSL

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX&field1=28&field2=30],[1],[443],[1]<CR><LF>

### GET method with content type and SSL

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX&field1=28&field2=30],[1],[443],[1],[application/x-www-form-urlencoded]<CR><LF>

### POST method(without SSL)

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX],[2],[80],[0],[field1=28&field2=30]<CR><LF>

### POST method with content type(without SSL)

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX],[2],[80],[0],[field1=28&field2=30],[application/x-www-form-urlencoded]<CR><LF>

### POST method with SSL

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX],[2],[443],[1],[field1=28&field2=30]<CR><LF>

### POST method with content type and SSL

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX],[2],[443],[1],[field1=28&field2=30],[application/x-www-form-urlencoded]<CR><LF>

### PUT method with content type

CMD+HTTP=[httpbin.org],[/put],[3],[80],[0],[{"Create": "Dictionary", "latitude": 30.496346, "longitude": -87.640356}], [application/json]

### PUT method with content type and SSL

CMD+HTTP=[httpbin.org],[/put],[3],[443],[1],[{"Create": "Dictionary", "latitude": 30.496346, "longitude": -87.640356}], [application/json]

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

EVT+HTTPSTATUS=<http status from server>,<size of data received from server>,<data from server><CR><LF>

EVT+HTTPCONFAIL<CR><LF> - HTTP server could not be reached

Please refer CMD+HTTPDATA command for usage of <[http\_content\_length]>,  
this is used to transfer multiple POST and PUT requests in single HTTP session.

### 4.4.2 CMD+HTTPDATA

Command	Description
CMD+HTTPDATA=<[content_length]>,<[fields_value]>,<[path]>,<[http_content_type]>,<[method]><CR><LF>	Command to make multiple POST and PUT requests in single session
Parameters	
[content_length]	Length of HTTP data to be transferred in current request.
[fields_value]	HTTP data to be transferred, Max length 3072 for WE20D, Max length 1500 for WE10.
[path]	Action and api key for server access(optional parameter, if kept blank then [path] from CMD+HTTP will be used)
[type] (optional parameter, if kept blank then [http_content_type] from CMD+HTTP will be used)	<[http_content_type]> max length 50
[method] (optional parameter, if kept blank then [method] from CMD+HTTP will be used)	2 – POST 3 - PUT

#### USAGE

End data on 3 separate POST requests in same HTTP session link  
https://api.thingspeak.com:80/update?api\_key=55LNK1KBQKWC0EUX&field1=0

#### Total HTTP data to be sent 27

CMD+HTTP=[api.thingspeak.com],[/update?api\_key=55LNK1KBQKWC0EUX],[2],[80],[0],[],[application/x-www-form-urlencoded],[27]<CR><LF>

CMD+HTTPDATA=[9],[field2=30],[],[]<CR><LF>

CMD+HTTPDATA=[9],[field3=45],[],[]<CR><LF>

CMD+HTTPDATA=[9],[field1=45],[],[]<CR><LF>

NOTE: Each successive CMD+HTTPDATA request shall be placed after receiving EVT+HTTPDATA.

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### 4.4.3 CMD+HTTPCLOSE

Command	Description
CMD+HTTPCLOSE<CR><LF>	Command to close current HTTP session

Response	Description
RSP=<status><CR><LF>	<status> - Status codes. @refer status codes

## 4.5. MQTT commands

### 4.5.1. CMD+MQTTCONCFG

Command	Description
CMD+MQTTCONCFG=<version>,<client_ID>,<username>,<password>,<keepaliveinterval>,<cleansession>,<willflag>,<willtopic name>,<willtopicmsg>,<willqos>,<willretain><CR><LF>	Command to set MQTT connection configuration
Parameters	
<version>	3- mqtt version 3.1 4- mqtt version 3.1.1
<client_id>	MQTT client ID. Maximum length: 256 Bytes.
Optional Parameters	
<username>	MQTT versoin(currently only version 3 supported, value of the parameter ignored)
<password>	MQTT client ID. Maximum length: 256 Bytes.
< keepaliveinterval >	timeout of MQTT ping. Unit: second. Range [0, 7200]. The default value is 0 which will be force-changed to 120 s.
< cleansession >	set MQTT clean session. For more details about this parameter, please refer to the section Clean Session in MQTT Version 3.1.1.  0: enable clean session. 1: disable clean session.
<willflag>	flag to indicate if LWT is required
< willtopicname >	LWT (Last Will and Testament) message topic.
< willtopicmsg >	LWT message. Maximum length
< willqos >	LWT QoS, which can be set to 0, 1, or 2. Default: 0.
< willretain >	LWT retain, which can be set to 0 or 1. Default: 0
Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID

### USAGE

CMD+MQTTCONCFG=3,mqtt-celium,,,,,,,,,

CMD+MQTTCONCFG=3,mqtt-celium,celium,devices,,,,,,,,

NOTE: total length of the command should not exceed 1600 bytes.

#### 4.5.2 CMD+MQTTNETCFG

Command	Description
CMD+MQTTNETCFG=<host_address>,<port><CR><LF> command to set MQTT network configuration	
Parameters	
<host_address>	Address of the mqtt broker
<client_id>	Port for mqtt communication (currently only 1883 is supported)
Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID

#### USAGE

CMD+MQTTNETCFG=dev.rigtech.io,1883<CR><LF>

#### 4.5.3 CMD+MQTTPUB

Command	Description
CMD+MQTTPUB=<pub_topic>,<data><CR><LF>	
Command to publish data to broker	
Parameters	
<pub_topic>	Topic to which data is to be published (max size 50)
<data>	(max size 1024)

#### USAGE

CMD+MQTTPUB=base/state/humidity,35

Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID

#### Note:

EVT+MQTTPUB<CR><LF> is triggered on successful publish of MQTT data

#### 4.5.4 CMD+MQTTSUB

Command	Description
CMD+MQTTSUB=<sub_topic><CR><LF>	
Command to subscribe to MQTT topic	
Parameters	
<sub_topic>	Topic to which subscription is required

Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID
EVT+SUBDATA=<sub_topic>,<data><CR><LF>	

#### USAGE

CMD+MQTTSUB= base/state/led1

#### 4.5.5 CMD+MQTTUNSUB

Command	Description
CMD+MQTTUNSUB=<sub_topic><CR><LF>	Command to unsubscribe to mqtt topic

Parameters
<sub_topic>                      Topic to unsubscribe

Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID

#### USAGE

CMD+MQTTUNSUB= base/state/led1

#### 4.5.6 CMD+MQTTSTART

Command	Description
CMD+MQTTSTART=<param><CR><LF>	Command to start/stop to mqtt operation

Parameters
<param>                      1: Start MQTT communication 0: Stop MQTT communication

Response	Description
RSP=<status>,<length>,<data><CR><LF>	<status> - @refer status codes <value_handle> - Value handle of READ UUID

#### 4.5.7 CMD+MQTTSSL

Command		Description
CMD+MQTTSSL==<use ssl>,<use rootca>,<use clientca>,<use private key><CR><LF>		command to set MQTT SSL and CRT configuration
Parameters		
<use ssl>	0-Establish MQTT without SSL 1- Establish MQTT using SSL	
<use RootCA>	0-Establish MQTT without RootCA 1- Establish MQTT using RootCA	
<use RootCA>	0-Establish MQTT without ClientCA 1- Establish MQTT using ClientCA	
<use PrivateKey>	0-Establish MQTT without PrivateKey 1- Establish MQTT using PrivateKey	
Response		Description
RSP=<status><CR><LF>		<status> - @refer status codes

#### USAGE

CMD+MQTTSSL=1,1,1,1<CR><LF>



## 4.6. Bluetooth Low Energy (BLE) command set

### 4.6.1 CMD+BLEMODE

Command	Description
CMD+BLEMODE=<mode>CRLF	Command to select BLE Mode
Parameters	
<mode>	1 – Peripheral 2 – Central 3 - Central + Peripheral

#### USAGE

CMD+BLEMODE=1<CR><LF>

This command sets BLE module to peripheral mode.

Response	Description
RSP=<status><CR><LF>	@refer status codes

### 4.6.2 CMD+BLESCAN

Command	Description
CMD+BLESCAN=<mode><CR><LF>	Command to start/ stop BLE scan
Parameters	
<mode>	0 – Stop 1 – Start

#### USAGE

CMD+BLESCAN=1<CR><LF>

This commands starts to scan devices.

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

Below Event is generated on scanning a device

EVT+SCANINFO=<ADV\_TYPE>,<ADDRESS\_TYPE>,<MAC\_ADDRESS>,<RSSI>,[<ADV\_DATA/SCAN\_RSP\_DATA>]<CR><LF>

<ADV\_TYPE>

GAP\_ADV\_EVT\_TYPE\_UNDIRECTED = 0

GAP\_ADV\_EVT\_TYPE\_DIRECTED = 1

GAP\_ADV\_EVT\_TYPE\_SCANNABLE = 2

GAP\_ADV\_EVT\_TYPE\_NON\_CONNECTABLE=3

GAP\_ADV\_EVT\_TYPE\_SCAN\_RSP=4

<ADDRESS\_TYPE>

P - public address

R - random address

<MAC\_ADDRESS>

<ADV/SCAN\_RPS DATA>

#### 4.6.3 CMD+BLECON

Command	Description
CMD+BLECON=<ADDRESS_TYPE>,<MAC_ADDRESS><CR><LF>	Command to connect to BLE device
Parameters	
<ADDRESS_TYPE> (received in scan command)	P - public address - R - random address
<MAC_ADDRESS>	

#### USAGE

CMD+BLECON=P,665544778899<CR><LF>

This commands connects to BLE device with Public address and MAC ID 665544778899

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

Below event is generated On connecting to a device

EVT+BLECONNECT=<conn\_id><CR><LF>

<conn\_id> : connection ID of the connected device

#### 4.6.4 CMD+BLEDISCONN

Command	Description
CMD+BLEDISCONN=<conn_id><CR><LF>	Command to Disconnect from BLE device

#### USAGE

CMD+BLEDISCONN=1<CR><LF>

This commands disconnects from the BLE device with connection ID 1

#### Note:

On disconnecting from a device

EVT+BLEDISCONN=<conn\_id><CR><LF>

<conn\_id> : connection ID of the disconnected device

#### 4.6.5 CMD+BLESERVICE

Command	Description
CMD+BLESERVICE=<request type>,<conn_id>,<optional param 1>,<optional param 2><CR><LF>	Command to list out supported uuid and characteristic

Parameters	
<request type>	1 - List all primary service 2 - List all handles by UUID 3 - List all handles and character descriptors between 2 given service handles
<conn_id>	connection ID of the connection (received in command 3)
<optional parameter 1>	If request type is 2 - <UUID TYPE>
<UUID TYPE>	0 : BLE SIG/16 bit UUID 1 : USER defined/128bit UUID
<optional parameter 1>	If request type is 3 - <START HANDLE>
<START HANDLE>	0x00 to 0xffff
<optional parameter 2>	If request type is 2 - <UUID>
<optional parameter 2>	If request type is 3 - <END HANDLE>
<END HANDLE>	0x00 to 0xffff

#### USAGE

**Request type 1** , lists out all the primary services and corresponding start and end handles in the primay service.

CMD+BLESERVICE=1,0<CR><LF>

**Request type 2** , lists out all charactesrs in the given UUID , in the below case UUID is A00A.

CMD+BLESERVICE=2,0,1,A00A

**Request type 3** , lists out all character descriptors and service handles between any 2 handles.

BLE+SERVICE=3,0,0,ffff

Lists out all the service handles and CCCD supported by peripheral

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

Below event is generated.

1. EVT+BLESERVICE=1,[<UUID>],<start handle>,<end handle>
2. EVT+BLESERVICE=2,[<UUID>],<GATT CHAR PROP>,<handle value>

<[UUID]>  
 <GATT CHAR PROP>      GATT\_CHAR\_PROP\_READ 0x02  
                              GATT\_CHAR\_PROP\_WRITE\_NO\_RSP 0x04  
                              GATT\_CHAR\_PROP\_WRITE 0x08  
                              GATT\_CHAR\_PROP\_NOTIFY 0x10  
                              GATT\_CHAR\_PROP\_INDICATE 0x20

<handle value> handles to be used to access respective UUID

3. EVT+BLESERVICE=3,<[UUID]>,<handle>

<handle value> - handles to be used to access respective                      UUID

#### 4.6.6 CMD+BLECDATA

Command	Description
CMD+BLECDATA=<conn_id>,<type>,<value_handle>,<length>,<data><CR><LF>	Command to send data to connected peripheral
Parameters	
<conn_id>	Connection of ID of device to send data
<type>	1 –write request 2- write command
<value_handle>	Value handle of write UUID (received in command 5) (In case peripheral is celium we10/we20 device handle value is 11)
<length>	Length of data to be transmitted
<data>	

#### USAGE

CMD+BLECDATA=0,1,11,6,ABCDEF

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

Below event is generated on successful write

EVT+BLECWRITE<CR><LF>

#### 4.6.7 CMD+BLENTF

Command	Description
CMD+BLENTF=<conn_id>,<value handle>,<enable><CR><LF>	Command to enable disable notification from central

Parameters	
<conn_id>	Connection of ID of device to send data
<value_handle>	Value handle of write UUID (received in command 5) (In case peripheral is Celium WE10/WE20 device handle value is 14)
<enable>	1 - enable notification 0 - disable notification

## USAGE

CMD+BLENTF=0,14,1<CR><LF>

This command enables notification

Response	Description
RSP=<status><CR><LF>	@refer status codes

## Note:

Once notification is enabled below event is generated on data reception

EVT+BLECDATA=<data\_size>,<data><CR><LF>

## 4.6.8 CMD+BLEADV

Command	Description
CMD+BLEADV=<enable><CR><LF>	Command to start/stop BLE advertisement

Parameters	
<enable>	0 – Stop 1 – Start

## USAGE

CMD+BLEADV=1<CR><LF>

This command starts advertising

## Note:

EVT+BLEADVSTART<CR><LF> - indicates that device has started to broadcast BLE packet.

EVT+BLEADVSTOP<CR><LF> - indicates devices has stopped ADV broadcast

**On successful connection below event is generated.**

EVT+BLEPCON=<conn\_id><CR><LF>

**Once the central device changes notification status below event is generated**

EVT+BLENOTIFY=<mode><CR><LF>

<mode> - 1 : notification on

0 : notification off

**On receiving data from central below event is generated**

EVT+BLEPDATA=<data\_size>,<data><CR><LF>

#### 4.6.9 CMD+BLEPDATA

Command	Description
CMD+BLEPDATA=<data><CR><LF>	Command to send data to central device.

#### USAGE

CMD+BLEPDATA=abcdef

#### 4.6.10 CMD+BLENAME

Command	Description
CMD+BLENAME=<name><CR><LF>	command to set BLE ADV name Parameters

Parameters	Description
<name>	String of BLE name(should not exceed 20 bytes)

#### USAGE

CMD+BLENAME=CELIUMWE20<CR><LF>

Will set BLE ADV name to CELIUMWE20

## 4.7. OTA Commands

### 4.7.1 CMD+OTA

Command	Description
CMD+OTA=<host_server>,<port>,<file_name><CR><LF>	Command for over the air update via HTTP protocol

Parameters	
<host_server>	Address of the server where the new firmware is stored.
<port>	Port on which server can be accessed
<file_name>	Name of the .bin file containing new firmware(max 50)

#### USAGE

CMD+OTA=celiumDevices.com,80,firmware.bin

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

EVT+READY<CR><LF> is generated after OTA completion.

### 4.7.2 CMD+FIRMWARE

Command	Description
CMD+FIRMWARE=<image_select><CR><LF>	Command to select firmware image

Parameters	
<image_select>	0 – Old firmware 1 – New firmware

#### USAGE

CMD+FIRMWARE<CR><LF>=0

CMD+FIRMWARE<CR><LF>=1

Response	Description
RSP=<status><CR><LF>	@refer status codes

#### Note:

EVT+READY<CR><LF> after OTA image switch

## 4.8. Query Commands

### 4.8.1 CMD?VERSION

Command	Description
CMD?VERSION<CR><LF>	Command to get firmware version

#### USAGE

CMD?VERSION<CR><LF>

Response	Description
RSP=<status>,<version><CR><LF>	<status> - Status codes. @refer status codes <version> - Firmware version

### 4.8.2 CMD?UARTCONF

Command	Description
CMD?UARTCONF<CR><LF>	Command to get current UART configuration

#### USAGE

CMD?UARTCONF<CR><LF>

Response	Description
RSP=<status>,<baud>,<- databits>,<stopbits>,<parity>,<flow - control><CR><LF>	<status> - Status codes. @refer status codes <baud> - Baudrate @refer CMD+UARCONF <databits> - Data bits @refer CMD+UARCONF <stopbits> - Stop bits @refer CMD+UARCONF <parity> - Parity @refer CMD+UARCONF <flowcontrol> - Flow control @refer CMD+UARCONF

### 4.8.3 CMD?WIFI

Command	Description
CMD?WIFI<CR><LF>	Command to get wifi related details

#### USAGE

CMD?WIFI<CR><LF>

Response	Description
RSP=0x00, <mode>,<SSID>,<chl>,<sec>[,<key_ id>],<pwd>,<mac>,<ip>,<gw>,<num>,<BSSID> <CR><LF>	1. The information in order are wifi mode, SSID, channel, secu - rity mode, (key id for WEP), password, device mac, device IP, gateway.  2. In AP mode, additional parameters, number<num> and the BSSID<BSSID> of client will be added.



#### 4.8.4 CMD?BLENAME

Command	Description
CMD?BLENAME<CR><LF>	Command to read BLE NAME set in CMD+BLENAME

#### USAGE

CMD?BLENAME<CR><LF>

Response	Description
RSP=00,CELIUMWE20<CR><LF>	

#### 4.8.5 CMD?BLEMODE

Command	Description
CMD?BLEMODE<CR><LF>	command to read BLE mode set in CMD+BLEMODE

#### USAGE

CMD?BLEMODE<CR><LF>

Response	Description
RSP=00,<mode><CR><LF>	<mode> 1 - peripheral 2 - central 3 - Peripheral+central 255 - Mode not set

## 4.9. Web Socket Commands

**Note :** HTTP/HTTPS cannot be used simultaneously with websocket.

### 4.9.1 CMD+WSSSTART

Command	Description
CMD+WSSSTART=<[server url]>,<[ port]>,<[path]>,<[origin]><CR><LF>	Command to establish connection with web server

Parameters	Description
<[server url]>	URL of the web socket server [max 50 characters]
<[port]>	Port on which communication should happen(0 to 65535)
<[path]>	(optional) Action and api key for server access (max length 500)
<[origin]>	(optional) Client/self URL

### USAGE

#### To establish connection to

wss://demo.piesocket.com/v3/channel\_1?api\_key=oCdCMcMPQpbvNjUIzqtvF1d2X2okWpDQj4AwARJuAgtjhzKxVE - jQU6IdCjwm&notify\_self

#### Following is the command:

CMD+WSSSTART=[wss://demo.piesocket.com],[443],[/v3/channel\_1?api\_key=oCdCMcMPQpbvNjUIzqtvF1d2X2okW - pDQj4AwARJuAgtjhzKxVEjQU6IdCjwm&notify\_self],[ ]<CR><LF>

Response	Description
RSP=<status><CR><LF>	<status> - Status codes. @refer status codes

### Note

If port number is 0 then port 80 will be used for ws:// and port 443 will be used for wss://

Recommendation: Port number should never be 0 or blank.

### 4.9.2 CMD+WSSSEND

Command	Description
CMD+WSSSEND=<[data]>,<[data_Mask]><CR><LF>	Command to send data to connected web server

Parameters	Description
<[data]>	Data to be sent to web socket server[max 1500 bytes]
<[data_mask]>	0 - mask not used 1 - mask used

## USAGE

CMD+WSSSEND=[hello],[1]<CR><LF>

Will send “hello” to connected webserver

Response	Description
RSP=<status><CR><LF>	<status> - Status codes. @refer status codes

## Note

If the data mask is set incorrectly , server might terminate the connection

### 4.9.3 CMD+WSSCLOSE

Command	Description
CMD+WSSCLOSE<CR><LF>	Command to close web socket connection

## USAGE

CMD+WSSCLOSE<CR><LF>

Response	Description
RSP=<status><CR><LF>	<status> - Status codes. @refer status codes

## 4.10. Ping Commands

### 4.10.1 CMD+PING

Command	Description
CMD+PING=<ping IP><cr><lf>	command to request ping response from server

Parameters	Description
<ping IP>	Ip address of the server to ping

Response	Description
RSP=<status><CR><LF>	<status> - Status codes. @refer status codes

#### USAGE

To ping to Google server below is the command

CMD+PING=8.8.8.8<CR><LF>

#### Sequence of responses<CR><LF>

RSP=00<cr><lf> indicating command format and parameters are correct.

Followed by either EVT+PING and EVT+DETAILS if ping response received

Else EVT+PINGTIMEOUT and EVT+DETAILS if ping timeout.

## 4.11. User Flash Commands

### 4.11.1 CMD+USERFLASH

Command	Description
CMD+USERFLASH=<[host]>,<[path]>,<[user_space_type]>,<[port]>,<[ssl_en]><CR><LF>	command to download data to USER flash using HTTP

Parameters	Description
<host>	Host server address
<path>	Action and api key for server access (max length 500)
<[user_space_type]>	1 – MQTT Root CA 2 – MQTT Client CA 3 – MQTT Private key
<[port]>	Port through which communication needs to happen (should be 443 for <[ssl_en]> 1)
<[ssl_en]>	0 - HTTP 1 - HTTPS

Response	Description
RSP=<status>,<http data><CR><LF>	<status> - Status codes. @refer status codes <http data> - Data stream downloaded from HTTP(s) server

### USAGE

**e.g Download MQTT Root CA from server.**

```
CMD+USERFLASH=[celiumdevices.com],[/RootCA.pem],[1],[80],[0]<CR><LF>
```

**Download MQTT Client CA from server.**

```
CMD+USERFLASH=[celiumdevices.com],[/clientCA.pem],[2],[80],[0]<CR><LF>
```

**Download MQTT private key from server**

```
CMD+USERFLASH=[celiumdevices.com],[/Privatekey.pem],[3],[80],[0]<CR><LF>
```

#### 4.11.2 CMD+USERFLASHREAD

Command	Description
CMD+USERFLASHREAD=<[user_space_type]><CR><LF>	Command to read data stored in user flash

Parameters	Description
<[user_space_type]>	1 – MQTT Root CA 2 – MQTT Client CA 3 – MQTT Private key

Response	Description
RSP=<status>,<http data><CR><LF>	<status> - Status codes. @refer status codes <http data> - Data stream saved in flash using CMD+USERFLASH

#### USAGE

e.g Read MQTT Root CA stored.

CMD+USERFLASHREAD=1<CR><LF>

#### 4.11.3 CMD+USERFLASHDEL

Command	Description
CMD+USERFLASHDEL=<[user_space_type]><CR><LF>	Command to delete data stored in user flash

Parameters	Description
<[user_space_type]>	1 – MQTT Root CA 2 – MQTT Client CA 3 – MQTT Private key

Response	Description
RSP=<status>,<http data><CR><LF>	<status> - Status codes. @refer status codes

#### USAGE

e.g Delete MQTT Root CA stored.

CMD+USERFLASHDEL=1<CR><LF>

## 5. Conditional Commands

This section covers commands which need other commands as prerequisite.

### 5.1. Setting up static IP (CMD+STAIP)

To use the above command below is command sequence.

1. CMD+STAIP=192.168.1.150<CR><LF>
2. CMD+DHCP=2,2<CR><LF>
3. CMD+WIFIMODE=1<CR><LF>
4. CMD+CONTOAP=WIFIDEMO,WIFIPASS<CR><LF>

On executing above sequence, the module will connect to AP WIFIDEMO and can be accessed by using IP 192.168.1.150.

### 5.2. Setting up Gateway IP(CMD+GATEWAYIP)

To use the above command below is command sequence.

1. CMD+GATEWAYIP=192.168.99.100,192.168.99.102,192.168.99.1<CR><LF>
2. CMD+DHCP=1,1<CR><LF>
3. CMD+WIFIMODE=2<CR><LF>
4. CMD+SETAP=WIFIDEMO,WIFIPASS,,<CR><LF>

On executing above sequence, the module will act as an access point with SSID “WIFIDEMO” and password “WIFI - PASS” with IP 192.168.99.1 and connected devices will have IP 192.168.99.100 to 192.168.102

### 5.3 Establishing Connection to MQTT broker

To set up Connection to MQTT broker below command sequence is to be followed.

1. CMD+CONTOAP=WIFIDEMO,WIFIPASS<CR><LF>
2. CMD+MQTTCONCFG=3,mqtt-celium,,,,,,,,,<CR><LF>
3. CMD+MQTTNETCFG=dev.rightech.io,1883<CR><LF>
4. CMD+MQTTSTART=1<CR><LF>

Above command sequence will establish connection to broker mqtt-celium hosted on dev.rightech.io  
Once above sequence is completed CMD+MQTTPUB,CMD+MQTTSUB and CMD+MQTTUNSUB can be used as many times as required.

#### 5.4. Establishing connection to MQTT broker with certificates.

To set up Connection to MQTT broker below command sequence is to be followed.

1. CMD+WIFIMODE=1<CR><LF>
2. CMD+CONTOAP=WIFIDEMO,WIFIPASS<CR><LF>
3. CMD+USERFLASH=[celiumdevices.com],[/RootCA.pem],[1],[80],[0]<CR><LF>
4. CMD+USERFLASH=[celiumdevices.com],[/ClientCA.pem],[2],[80],[0]<CR><LF>
5. CMD+USERFLASH=[celiumdevices.com],[/Private.pem],[3],[80],[0]<CR><LF>
6. CMD+MQTTCONCFG=3,mqtt-celium,,,,,,,,,<CR><LF>
7. CMD+MQTTNETCFG=dev.rightech.io,1883<CR><LF>
8. CMD+MQTTSSL=1,1,1,1<CR><LF>
9. CMD+MQTTSTART=1<CR><LF>

Above command sequence will establish connection to broker mqtt-celium hosted on dev.rightech.io using certificates



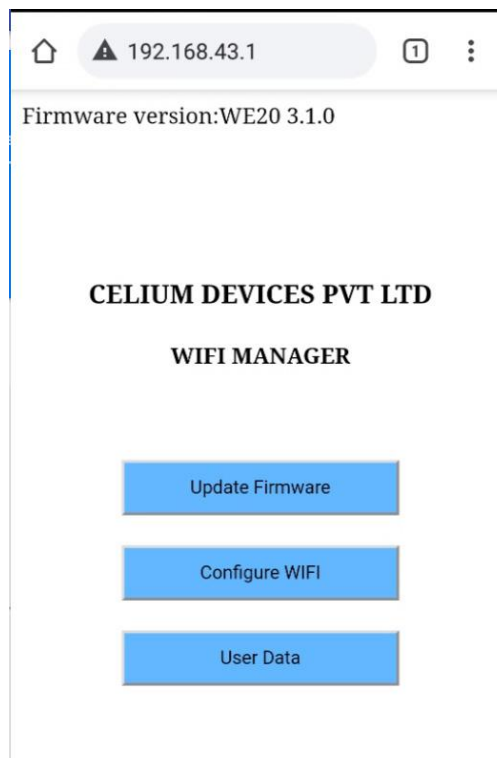
## 6. Homepage

### 6.1. Overview

Celium WIFI modules will be in AP mode on reset (unless CMD+AUTOCONAP is set). The AP SSID will be CELIUMWIFI and password will be CELIUMPASS. Once the device connects to module AP, the Homepage/Webpage hosted on the module can be accessed typing following IP 192.168.43.1 in browser address bar.

### 6.2. Index page

Following will be the index page on connection.



1. Firmware version will be displayed on top left corner.
2. OTA update can be done using UPDATE FIRWARE option
3. Router SSID and PASSWORD can be set using CONFIGURE WIFI option
4. User can update/send data to host controller using USER DATA option

### 6.3. UPDATE FIRMWARE

6.3.1 This feature needs active internet connection, If module is in AP mode below message will be displayed for UPDATE FIRMWARE



In this case, connect module to Router with internet access and try again.

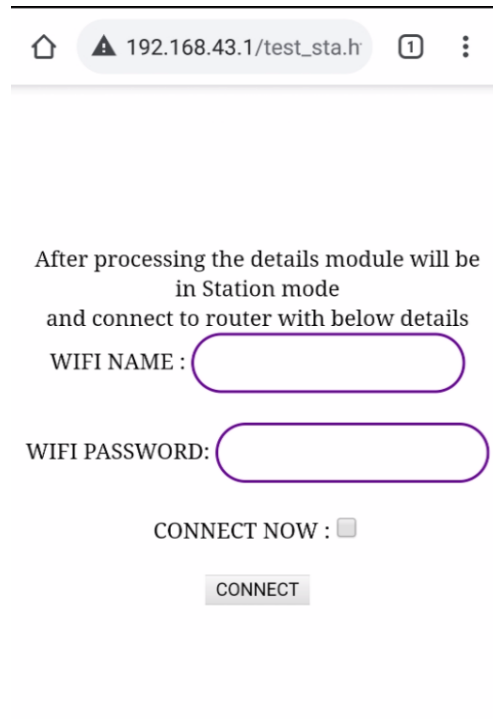
6.3.2.If module is connected to active internet connection,then below page is displayed on using option UPDATE FIRMWARE

A screenshot of a web browser interface showing a form for "OTA SERVER DETAILS". The address bar shows a home icon, a warning triangle icon, the URL "192.168.104.176/test\_p", a tab count of "1", and a menu icon. The status bar at the top shows signal strength, Wi-Fi, battery, and the time "9:00". The form contains three input fields: "Host Address :", "Firmware name :", and "Port :". Below the "Port :" field is a button labeled "UPDATE".

On providing Host Address, Firmware name and Port name , module will download ota file and update the Firmware. Before initiating OTA, EVT+HTMOTA will be triggered (details in section 7 )

#### 6.4. CONFIGURE WIFI

This feature is used to set ROUTER SSID and Password details. On selecting Option CONFIGURE WIFI below page will be displayed.



After processing the details module will be  
in Station mode  
and connect to router with below details

WIFI NAME :

WIFI PASSWORD:

CONNECT NOW : ☐

WIFI NAME and WIFI PASSWORD tabs are used to send router SSID and Password to WIFI Module.

If CONNECT NOW is NOT selected then only EVT+HTMSTA is triggered(details in section 7).

If CONNECT NOW is selected then EVT+HTMSTA is triggered and also the module tries to establish connection with the router using SSID and password provided.

#### 6.5. USER DATA

Using USER DATA option user can send data to host controller.

On selection USER DATA following webpage is displayed.

The screenshot shows a web browser interface. At the top, the address bar displays a home icon, a warning triangle, the URL "192.168.43.1/test\_ap.ht", a tab count of "1", and a menu icon. The main content area has a light gray background. It contains the text "The data updated will be sent to host controller" in bold. Below this is the label "USER DATA:" in bold. Underneath is a text input field with the placeholder text "USER DATA". At the bottom center of the form is a button labeled "UPDATE".

On entering user data in text box and pressing update EVT+HTMPAR is triggered (details in section 7).

## 7. Timing parameters and recommended command flow.

This section provides recommended code flow to be followed for ATCMD usage.

### 1. A delay of 2 seconds is to be met before sending a command after boot event EVT+READY<cr><lf> is received

Received EVT+READY<cr><lf>

Wait 2 Seconds

Send Commads

### 2.A delay of 2 seconds is to be met before sending a command after RSP=00<cr><lf> is received for CMD+SETAP

Send CMD+SETAP=wifidemo,wifipass,,,<cr><lf>

Receive RSP=00<cr><lf>

Wait 2 seconds

Send commands

### 3. Once CMD+CONTOAP=<parameters> is sent wait till EVT+CONTOAP or RSP=xx where xx is a non zero value

CMD+CONTOAP=<parameters>

Wait

RSP=00

EVT+CONTOAP=<parameters>

**send commands OR**

CMD+CONTOAP=<parameters>

wait

RSP=xx and xx is non zero

Send commands

## 8. Status Codes

Status codes are part of 'response' data set. The status codes are in ASCII format.

Sl. No	Status code	Description
1	RSP=00<CR><LF>	The command is accepted
2	RSP=01<CR><LF>	Invalid Command
3	RSP=02<CR><LF>	Invalid Parameter
4	RSP=03<CR><LF>	Invalid datasize
5	RSP=04<CR><LF>	Invalid state
6	RSP=05<CR><LF>	Blank space in command parameters
7	RSP=06<CR><LF>	Invalid parameter size
8	RSP=AA<CR><LF>	WiFi not connected
9	RSP=B1<CR><LF>	WiFi disconnect failed , try again
10	RSP=B2<CR><LF>	WiFi dissociation time out, try again
11	RSP=B3<CR><LF>	Memory not available , stop and start scan again
12	RSP=B4<CR><LF>	WiFi partial scan channel fail, stop and start scan again
13	RSP=B5<CR><LF>	WiFi mode error
14	RSP=B6<CR><LF>	wifi parameters errors
15	RSP=B7<CR><LF>	SSID empty
16	RSP=B8<CR><LF>	start AP failed, try again
17	RSP=B9<CR><LF>	start AP timeout, try again
18	RSP=BA<CR><LF>	wifi connect failed, try again
19	RSP=BB<CR><LF>	Can't get AP security mode and channel
20	RSP=BC<CR><LF>	AP security mode is open, no password is required
21	RSP=BD<CR><LF>	DHCP set address was not assigned
22	RSP=BE<CR><LF>	WIFI get setting failed for auto connect.
23	RSP=71<CR><LF>	MQTT client ID not provided
24	RSP=72<CR><LF>	MQTT unsub topic is not presetnt in sub list
25	RSP=73<CR><LF>	MQTT network cfg not set
26	RSP=74<CR><LF>	MQTT connetct cfg not set
27	RSP=75, <return code from MQTT server> <CR><LF>	MQTT pub failed
28	RSP=81<CR><LF>	Invalid node

Sl. No	Status code	Description
29	RSP=101<CR><LF>	Parameter number error
30	RSP=102<CR><LF>	Local port should be 1 65535
31	RSP=103<CR><LF>	Create con_id error
32	RSP=104<CR><LF>	Create server task error
33	RSP=105<CR><LF>	Create socket error
34	RSP=106<CR><LF>	Set socket option error
35	RSP=107<CR><LF>	Bind error
36	RSP=108<CR><LF>	Listen error
37	RSP=109<CR><LF>	TCP server already exists error
38	RSP=110<CR><LF>	Accept error
39	RSP=111<CR><LF>	Create con_id for seed error
40	RSP=112<CR><LF>	UDP server already exists error
41	RSP=113<CR><LF>	Server can't start under TT(transparent transmission) mode
42	RSP=114<CR><LF>	Connection type is unknown (SSL isn't supported)
43	RSP=115<CR><LF>	Listening socket on bind_ip:port failed for ssl server
44	RSP=116<CR><LF>	Malloc failed for server certificate
45	RSP=117<CR><LF>	Malloc failed for server key
46	RSP=118<CR><LF>	x509_crt_parse failed for server certificate
47	RSP=119<CR><LF>	x509_crt_parse failed for server ca list
48	RSP=120<CR><LF>	pk_parse_key failed for server key
49	RSP=121<CR><LF>	Hang node failed for ssl server
50	RSP=122<CR><LF>	Accept error for ssl server
51	RSP=123<CR><LF>	Malloc failed for ssl seed
52	RSP=124<CR><LF>	Initialization failed for ssl context
53	RSP=125<CR><LF>	ssl_set_own_cert error
54	RSP=126<CR><LF>	ssl handshake failed for ssl seed
55	RSP=127<CR><LF>	Create node failed for ssl seed

Sl. No	Status code	Description
56	RSP=201<CR><LF>	Parameter number error
57	RSP=202<CR><LF>	Remote IP format or host unfound error
58	RSP=203<CR><LF>	Remote port should be 1 65535 error
59	RSP=204<CR><LF>	Create con_id error (none available)
60	RSP=205<CR><LF>	Create client task error
61	RSP=206<CR><LF>	inet_ntoa_r remote address error
62	RSP=207<CR><LF>	Create socket error
63	RSP=208<CR><LF>	Hang node error for tcp client
64	RSP=209<CR><LF>	Connect error for tcp client
65	RSP=210<CR><LF>	Hang node error for udp client
66	RSP=211<CR><LF>	Local port should be 1 65535
67	RSP=212<CR><LF>	Bind local port error
68	RSP=213<CR><LF>	Connection already exists for TT(transparent transmission) mode
69	RSP=214<CR><LF>	Set broadcast on socket failed
70	RSP=215<CR><LF>	Set multicast add membership on socket failed
71	RSP=216<CR><LF>	Set multicast interface failed
72	RSP=217<CR><LF>	Connection type is unknown (SSL isn't supported)
73	RSP=218<CR><LF>	Initiate a TCP connection with host:port failed for ssl client
74	RSP=219<CR><LF>	Memory allocation failed for ssl context structure
75	RSP=220<CR><LF>	SSL context initialization failed
76	RSP=221<CR><LF>	SSL handshake failed
77	RSP=222<CR><LF>	Hang node failed for ssl client
78	RSP=301<CR><LF>	Command format error
79	RSP=302<CR><LF>	<Buffer Size> error (should be 1 MAX_BUFFER(default 1600))
80	RSP=303<CR><LF>	<con_id> is not found
81	RSP=304<CR><LF>	recvfrom() error for udp server
82	RSP=305<CR><LF>	recvfrom() error for udp client/seed
83	RSP=306<CR><LF>	TCP server should receive from seed
84	RSP=307<CR><LF>	Connection lost
85	RSP=308<CR><LF>	read( ) error for tcp con_idfatpk



Sl. No	Status code	Description
86	RSP=501<CR><LF>	Parameter number error
87	RSP=502<CR><LF>	<Buffer Size> exceeds ATPT send buffer size
88	RSP=503<CR><LF>	con_id is not found
89	RSP=504<CR><LF>	<UDP Client IP> or <UDP Client Port> error for udp server case
90	RSP=505<CR><LF>	sendto( ) error for udp server
91	RSP=506<CR><LF>	sendto( ) error for udp server
92	RSP=507<CR><LF>	TCP server should send data to the seed
93	RSP=508<CR><LF>	Write error for tcp client/server
94	RSP=91<CR><LF>	Memory alloc failed, try again
95	RSP=92<CR><LF>	Server nit found
96	RSP=93<CR><LF>	Invaild address in new image
97	RSP=94<CR><LF>	Send HTTP req failed, check wifi connection
98	RSP=95<CR><LF>	Read sockedt failed,cheeck port settings
99	RSP=96<CR><LF>	New firmware size is 0
100	RSP=97<CR><LF>	Unable to erase , try again
101	RSP=98<CR><LF>	Flash write failed, try again
102	RSP=99<CR><LF>	Signature invalid
103	RSP=31<CR><LF>	Unable to send data, try again
104	RSP=34,1<CR><LF>	INVALID STATE, Central mode off
105	RSP=34,2<CR><LF>	INVALID STATE, connection already established
106	RSP=34,3<CR><LF>	INVALID STATE, connection not established
107	RSP=34,4<CR><LF>	INVALID STATE, Peripheral mode off
108	RSP=34,5<CR><LF>	INVALID STATE, Notification not enabled
109	RSP=35,1<CR><LF>	No primary services discovered
110	RSP=35,2<CR><LF>	Invalid UUID or UUID type
111	RSP=35,3<CR><LF>	NO service handle present in given range
112	RSP=C1<CR><LF>	HTTP session running , close the session to make new request
113	RSP=C2<CR><LF>	NO HTTP session running.
114	RSP=D1<CR><LF>	Web socket server connection running, close and try again
115	RSP=D2<CR><LF>	Web socket server not connected,connect and try again

## 9. Events

Sl. No	Events	Description
1	EVT+READY<CR><LF>	This event is generate upon module reset and indicates that the module is ready to receive commands from the host
2	EVT+CONTOAP=<ip_addr><CR><LF>	<p>This event is generated when the module in station mode connects to an AP.</p> <p><b>Parameters:</b>            &lt;ip_addr&gt; : IP address of the connected AP.</p> <p>@refer CMD+CONTOAP command</p>
3	EVT+DISCON<CR><LF>	This event is generated when the module disconnects from the AP/STA
4	EVT+STACON=<ip_addr><CR><LF>	<p>This event is generated when the module is in AP mode and a device connects to it.</p> <p><b>Parameters:</b>            &lt;ip_addr&gt; : IP address of the connected device.</p> <p>@refer CMD+WIFIMODE and CMD+SETAP commands</p>
5	EVT+SKTCON=<con_id>,<ip_addr>,<port>,<socket><CR><LF>	<p>This event is generated when a TCP/UDP socket connection is established between devices.</p> <p><b>Parameters:</b>            &lt;con_id&gt; : Connection ID.            &lt;ip_addr&gt; : IP address of the connected device.            &lt;port&gt; : Port number            &lt;socket&gt; : Socket number</p> <p>@refer CMD+SETSERVER and CMD+SETCLIENT commands</p>
6	EVT+SKTDATA=<data_length>,<con_id>,<data><CR><LF>	<p>This event is generated when the device receives data over TCP/UDP</p> <p><b>Parameters:</b>            &lt;data_length&gt; : Length of the data received.            &lt;con_id&gt; : Connection ID of the peer            &lt;data&gt; : Data received</p>
7	EVT+SCANAP=<num>,<SSID>,<channel>,<security>,<RSSI>,<MAC><CR><LF>	<p>This event is triggered if APs are detected while scanning</p> <p><b>Parameters:</b></p> <p>&lt;num&gt; : serial number of AP            &lt;SSID&gt; : name of AP/WIFI            &lt;channel&gt; : channel on which AP was detected            &lt;security&gt; : 0 - open            1 - wep            2 - wpa tkip            3 - wpa aes            4 - wpa2 aes            5 - wpa2 tkip            6 - wpa2 mixed            7 - wpa/wpa2 aes            8 - wpa2 enterprise            9 - wpa/wpa2 enterprise            10 - wp3-sae aes            11 - unknown</p>

Sl. No	Events	Description
		<RSSI> : RSSI Value <MAC> : MAC ID
8	EVT+SCANAPDONE<CR><LF>	This event is triggered when all the AP in the vicinity are scanned.
9	EVT+HTTPSTATUS=<http status from server>,<size of data received from server>,<data from server><CR><LF>	This even is triggered when the server responds with a status.
10	EVT+HTTPCONFFAIL<CR><LF>	HTTP server could not be reached.
11	EVT+MQTTPUB<CR><LF>	This event is triggered on successful publish of MQTT data.
12	EVT+SUBDATA=<subscribe_topic>,<data_from_MQTT_broker><CR><LF>	This event is triggered when the device receives data on a subscribed topic.
13	EVT+HTTPCONNCLOSE<CR><LF>	Indicates HTTP session closed
14	EVT+HTTPHEADERDAIL<CR><LF>	Indicates invalid header data
15	EVT+HTTPDATA,<data_size><CR><LF>	This event informs <data_size> amount of data has to be sent to server in current session
16	EVT+BLES SCAN=<ADV_TYPE>,<ADDRESS_TYPE>,<MAC_ADDRESS>,<RSSI>,<[ADV_DATA/SCAN_RSP_DATA]><CR><LF>	This event is generated on scanning a device
17	EVT+BLECCON=<conn_id><CR><LF>	This event is generated when BLE connection is established in central role
18	EVT+BLEDISCONN=<conn_id><CR><LF>	This event is generated on disconnecting from a device
19	EVT+BLESERVICE=1,<[UUID]>,<start handle>,<end handle>	After discovery of UUID and Characteristics
20	EVT+BLESERVICE=2,<[UUID]>,<GATT CHAR PROP>,<handle value>	After discovery of UUID and Characteristics
21	EVT+BLESERVICE=3,<[UUID]>,<handle>	After discovery of UUID and Characteristics
22	EVT+BLECWRITE<CR><LF>	This event is generated on successful write
23	EVT+BLECDATA=<data_size>,<data><CR><LF>	Once notification is enabled, this event is generated on data reception
24	EVT+BLEADVSTART<CR><LF>	Indicates that device has started to broadcast BLE packet
25	EVT+BLEADVSTOP<CR><LF>	Indicates devices has stopped ADV broadcast
26	EVT+BLEPCON=<conn_id><CR><LF>	This event is generated when BLE connection is established in peripheral role
27	EVT+BLENOTIFY=<mode><CR><LF>	Once the central device changes notification status below event is generated
28	EVT+BLEPDATA=<data_size>,<data><CR><LF>	On receiving data from central below event is generated
29	EVT+HTMOTA=<host_address>,<file_name>,<service_port><CR><LF>	Event to indicate OTA is triggered from homepage  <b>&lt; host_address &gt;</b> : Address of the server where the new firmware is stored(max size 50) <b>&lt;file_name&gt;</b> : Name of the .bin file containing new firmware(max size 50) <b>&lt; service_port&gt;</b> : Port on which server can be accessed

Sl. No	Events	Description
30	EVT+HTMSTA=<ap_name>,<ap_password>,<request_type><CR><LF>	<p>Event to indicate Router details are updated through homepage.</p> <p>&lt;ap_name&gt; : SSID as provided in Homepage          &lt;ap_password&gt; : Password as provided in Homepage          &lt;request_type&gt; : 0 - send details to HOST controller          1 - Send details to HOST controller and try to connect to AP.</p> <p>NOTE:if request_type is 1 then EVT+HTMSTA will be followed by EVT+CONTOAP if connection established else will return a Response code</p>
31	EVT+HTMPAR=<user_data><CR><LF>	<p>Event to indicate user sent data using Homepage ( max length 300 bytes)</p> <p>&lt;user_data&gt; : user data</p>
32	EVT+WSSCONSET<CR><LF>	Web socket connection established
33	EVT+WSSCONNCLOSE<CR><LF>	Web socket connection closed
34	EVT+WSSCONNFALL<CR><LF>	Web socket connection failed, invalid url or path.
35	EVT+WSSCONTFALL<CR><LF>	Web socket context failed, try again
36	EVT+WSSDATATX<CR><LF>	Data sent to connected web socket server
37	EVT+WSSDATARX=<data_lenght>,<rx_data><CR><LF>	<p>Data received from web socket server</p> <p>&lt;data_lenght&gt; : length of data received from web socket server          &lt;data&gt; : data received from web socket server</p>
38	EVT+MQTTCONFAIL<CR><LF>	MQTT connection failed ,retry with correct credentials
39	EVT+MQTTRUNNING<CR><LF>	MQTT connection Established