# Updating Services over Cellular Networks

# Updating Services over Cellular Networks

## Contents


Introduction ..... 2

Prerequisites ..... 2

Sending and Receiving SMS messages ..... 3

Sending SMS using AT Commands (7670 & 7600) ..... 3

Receiving SMS using AT Commands (7670 &7600) ..... 3

Updating “Key-Pair” Services ..... 4

Updating ThingSpeak using “HTTP” AT Commands (7670) ..... 4

Updating WAMP/LAMP/RAMP using “HTTP” AT Commands (7670 & 7600) ..... 6

MQTT Brokers ..... 8

Publish to ThingSpeak MQTT using AT Commands (7670) ..... 8

Subscribe to ThingSpeak MQTT using AT Commands (7670) ..... 10

Publish to a Mosquitto Broker using AT Commands (7670) ..... 12

Subscribe to a Mosquitto Broker using AT Commands (7670) ..... 13

AT Commands 7600 (v2.xx) ..... 16

Updating  ThingSpeak using “HTTP” AT Commands (7600) ..... 16

Publish to MQTT (7600) ..... 16

Subscribe to MQTT (7600) ..... 16

# Updating Services over Cellular Networks

## Introduction

The deployment of the Internet of Things has fuelled rapid growth in cellular connectivity. With "IOT-only" SIM cards providing data for pennies-per-month the cost to implement remote sensors is well within the reach of all. Popular modules include devices by SIMcom (and clones) such as the SIM800 and SIM76xx series.

Devices such as the SIM800 work over the 2G and 3G networks whilst the SIM76xx series work over 4G/LTE. With much of the world having already retired their 2G and 3G networks there is a clear need to migrate to newer technology. Incidentally the United Kingdom although having already retired its 3G network is committed to providing 2G until the global retirement date of 2033.

Unfortunately, it isn't as easy as just swapping out the modules for something newer. The AT Command set has also evolved, deprecating certain instructions as well as adding in new features. Code that worked before cannot be guaranteed to still work.

Flowcode made it quite easy to include the SIM800 module in designs by having a component for it, and by using the UART(RS232) component you could also control the module directly by issuing whatever AT Command you wished and receiving replies. At time of writing there is no component for the 76xx series, so direct control is necessary utilising the AT Commands over UART.

It is the intent of this document, and accompanying Flowcharts, to demonstrate how you can use AT Commands to instruct SIM76xx modules to interact with popular services such as those using "key-pairs", an example being ThingSpeak, MQTT Brokers such as mosquitto and also basic messaging via SMS.

To test, I created a RAMP server on my network and gave associated ports external visibility. This is documented in the Flowcode forum for those interested.

The AT Commands provided in this document are just the commands you *could* incorporate in your own projects to interact. Nothing more than very basic working examples of "how to".

## Prerequisites

You will need a SIM76xx module. I used the 7600E and 7670E for testing. Note that my modules were not running the same AT version. My 7600E appears to be running 76xx v2.xx whilst the 7670E appears to be using 767x v1.01. The 767x v1.01 commands appear very similar to 76xx v3.

Code here uses the 7670 module, with the necessary 7600 changes shown later where needed.

If you intend to interact with MQTT Brokers you will need to be running AT v2.0 or newer.

Pay attention to power requirements as the modules can draw up to two amps when transmitting. Also, the Tx and Rx pins are most likely 3.3v and not 5v tolerant.

If you are connecting via a USB-TTL adapter for testing, then ensure the adapter is based on the FT232 chip not CH340 as you can have issues using the Header pins with the CH340. This is more so with SIM7600 modules.

# Updating Services over Cellular Networks

## Sending and Receiving SMS messages

Sending and receiving SMS messages is quite easy to do. We will check we can “talk” to the module, set the SMS mode to “text” and check we are registered on the Network. After that we will send and receive messages.

### Sending SMS using AT Commands (7670 & 7600)

Code in blue is entered, with expected / typical reply in black

AT
OK

AT+CMGF=1 // 1 = Text Mode
OK

AT+CREG? // Query Network Status
+CREG: 0,1 // 0 = disabled unsolicited registration result codes. 1 = status (1 = registered Home)
OK

AT+CMGS="0123456789" // Recipient number

> Test // message to send. After entering message send CTRL+Z (hex1A / Dec 026)
+CMGS: 14 // the number is the total number of messages sent
OK

### Receiving SMS using AT Commands (7670 &7600)

When an SMS arrives the module will send out

+CMTI: "SM",1 // where “1” is the memory location where message is stored

AT+CMGRD=1 // Read and Delete message in above location

+CMGRD: "REC UNREAD","0123456789","","25/04/17,22:28:21+4" //Message Info
Reply // actual message received

OK

Flowcode can easily look for the incoming “+CMTI: "SM",1” message and if present branch to a receive routine that extracts the actual message or acts upon it.

# Updating Services over Cellular Networks

## Updating “Key-Pair” Services

A “Key-Pair” service or Key-Value Pair to give it its proper title is one in which information is contained in pairs. For example, you may have a number of sensors gathering data and to send we would structure the information with a unique sensor identifier and its value (e.g. sensor1=22.5). This is how services such as ThingSpeak accept data from you and it may be sent as part of an address or perhaps json encoded.

### Updating ThingSpeak using “HTTP” AT Commands (7670)

From your ThingSpeak account you will need to obtain your unique API-Key, used to identify you, and this will be incorporated into the string to be sent.

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"        // change to your APN
OK

AT+CSQ?
+CSQ 17;99                            // Query rssi and BER
OK

AT+CREG?                              // Query network registration
+CREG: 0,1                            // Unsolicited Response Code, Status
OK

AT+CGREG?                             // Query GPRS network registration
+CGREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CEREG?                             // Query EPS network registration
+CEREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+NETOPEN                            // Start TCP/IP services
OK
+NETOPEN: 0
```

# Updating Services over Cellular Networks

```
AT+HTTPINIT                          // Start HTTP services
OK

AT+HTTPPARA="URL",”https://api.thingspeak.com/update?api_key=<your Key>&field1=27”
                                     // Sets server path and parameters, enter
                                     your key and field values etc
OK

AT+HTTPACTION=0                      // Perform the HTTP action (0=GET)
OK
+HTTPACTION: 0,200,2                 //  method, server reply code, number of returned bytes

AT+HTTPTERM                          // Stop HTTP services
OK

AT+NETCLOSE                          // Stop TCP/IP services
OK
+NETCLOSE: 0
```

Using the SIM7670 module the above code will update your ThingSpeak account. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

By modifying the AT+HTTPPARA command you can easily update other services such as PushingBox or your own WAMP/LAMP/RAMP server.

# Updating Services over Cellular Networks

## Updating WAMP/LAMP/RAMP using “HTTP” AT Commands (7670 & 7600)

You will need to obtain your username, password and URL and this will be incorporated into the string to be sent. The example below uses the target script and key-pairs documented elsewhere in the forum for my RAMP install.

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"        // change to your APN
OK

AT+CSQ?
+CSQ 17;99                            // Query rssi and BER
OK

AT+CREG?                              // Query network registration
+CREG: 0,1                            // Unsolicited Response Code, Status
OK

AT+CGREG?                             // Query GPRS network registration
+CGREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CEREG?                             // Query EPS network registration
+CEREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+NETOPEN                            // Start TCP/IP services
OK
+NETOPEN: 0

AT+HTTPINIT                           // Start HTTP services
OK

AT+HTTPPARA="URL",”http://aaa.bbb.ccc.ddd:port/update.php?Sensor=1&Value=17”
                                      // Sets server path and parameters, enter
                                      your key and field values etc
OK

AT+HTTPACTION=0                       // Perform the HTTP action (0=GET)
OK
+HTTPACTION: 0,200,2                  //  method, server reply code, number of returned bytes
```

# Updating Services over Cellular Networks

```
AT+HTTPTERM                    // Stop HTTP services
OK

AT+NETCLOSE                    // Stop TCP/IP services
OK
+NETCLOSE: 0
```

Using the SIM7670 module the above code will update your WAMP/LAMP/RAMP server. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

# Updating Services over Cellular Networks

## MQTT Brokers

Many deployed IOT devices rely on MQTT to pass information via Brokers. These Brokers act as a central hub allowing clients to Publish (send) and Subscribe (receive) to Topics.

If you have a ThingSpeak account, then with just a few clicks you can configure it as a Broker or you may wish to host your own Broker using mosquitto or similar.

Depending on AT Command version installed in your module, it can greatly simplify connecting to a Broker. For the 76xx module, version 2.0 onward features MQTT support. The 7670 also supports.

Irrespective of service provider, you will need some basic information to allow clients to connect and the following is typical

| | |
|---|---|
| Host: | The target IP address (or URL if over internet) |
| Port: | 1883 which is the standard port number |
| Client ID: | Unique ID to identify your client (e.g. My_Client1 ). Note some services will provide you with an ID to use. |
| Username: | Username to access your service. Note some services will provide you with a username to use. |
| Password: | Password to access your service. Note some services will provide you with a password to use. |

### Publish to ThingSpeak MQTT using AT Commands (7670)

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"        // change to your APN
OK

AT+CSQ?
+CSQ 17;99                            // Query rssi and BER
OK

AT+CREG?                              // Query network registration
+CREG: 0,1                            // Unsolicited Response Code, Status
OK

AT+CGREG?                             // Query GPRS network registration
+CGREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CEREG?                             // Query EPS network registration
+CEREG: 0,1                           // Unsolicited Response Code, Status
OK
```

# Updating Services over Cellular Networks

```
AT+CMQTTSTART                          // Start MQTT services
OK
+CMQTTSTART: 0

AT+CMQTTACCQ=0,"<Client ID>"           // Set unique client ID
OK

AT+CMQTTCONNECT=0,"tcp://mqtt3.thingspeak.com:1883",60,1,"<username>","<password>"
                                       // Connect to Broker using username&password
OK
+CMQTTCONNECT: 0,0

AT+CMQTTTOPIC=0,xx                     // xx =  length of Topic
><Topic>                               // Topic to publish to *
OK

AT+CMQTTPAYLOAD=0,xx                   // xx = length of message / payload to be sent
><Message / Payload                    // Actual message / payload to be sent
OK

AT+CMQTTPUB=0,0,20                     // Publish message to server (index, QOS, timeout)
OK
+CMQTTPUB: 0,0                         // index, status (0=success)

AT+CMQTTDISC=0                         // Disconnect from Broker
OK
+CMQTTDISC: 0,0

AT+CMQTTREL=0                          // Release current client
OK

AT+CMQTTSTOP                           // Stop MQTT service
OK
+CMQTTSTOP: 0
```

* To publish to a ThingSpeak channel the topic will be channels/<channel-ID>/publish/fields/fieldx where <channel-ID> is your channel ID obtained from when you created your channel, and x is the channel field number you wish to publish to.

Using the SIM7670 module the above code will update your ThingSpeak account. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

# Updating Services over Cellular Networks

By modifying the AT+ CMQTTACCQ, the AT+CMQTTCONNECT, the AT+CMQTTTOPIC and AT+CMQTTPAYLOAD commands you can easily update other services such as your own mosquitto broker.

## Subscribe to ThingSpeak MQTT using AT Commands (7670)

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"        // change to your APN
OK

AT+CSQ                                // Query rssi and BER
+CSQ: 15,99
OK

AT+CREG?                              // Query network registration
+CREG: 0,1                            // Unsolicited Response Code, Status
OK

AT+CGREG?                             // Query GPRS network registration
+CGREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CEREG?                             // Query EPS network registration
+CEREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CMQTTSTART                         // Start MQTT services
OK
+CMQTTSTART: 0

AT+CMQTTACCQ=0,"<Client ID>"          // Set unique client ID
OK

AT+CMQTTCONNECT=0,"tcp://mqtt3.thingspeak.com:1883",60,1,"<username>","<password>"
                                      // Connect to Broker using username&password
OK
+CMQTTCONNECT: 0,0

AT+CMQTTSUBTOPIC=0,xx,0               // xx =  length of Topic
><Topic>                              // Topic to subscribe to *
OK
```

# Updating Services over Cellular Networks

AT+CMQTTSUB=0 // Subscribe to the topic
OK
+CMQTTSUB: 0,0

*At this point you are now Subscribed*

When a client publishes a message to the Broker, you will automatically receive:-

+CMQTTRXSTART: 0,40,2 // Start with length of topic and message
+CMQTTRXTOPIC: 0,40 // Length of Topic to be received
channels/1737480/subscribe/fields/field1 // Topic
+CMQTTRXPAYLOAD: 0,2 // Length of message / payload to be received
17 // Actual message / payload
+CMQTTRXEND: 0 // End of message

AT+CMQTTUNSUB=0,0 // Unsubscribe from Broker
OK
+CMQTTUNSUB: 0,18

AT+CMQTTDISC=0 // Disconnect from Broker
OK
+CMQTTDISC: 0,0

AT+CMQTTREL=0 // Release current client
OK

AT+CMQTTSTOP // Stop MQTT service
OK
+CMQTTSTOP: 0

* To subscribe to a ThingSpeak channel the topic will be channels/<channel-ID>/subscribe/fields/fieldx where <channel-ID> is your channel ID obtained from when you created your channel, and x is the channel field number you wish to subscribe to.

If you omit /fields/fieldx then *any* data arriving for *any* field will send a message containing *ALL* fields and data.

Using the SIM7670 module the above code will subscribe to your ThingSpeak account. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

By modifying the AT+ CMQTTACCQ, the AT+CMQTTCONNECT and the AT+CMQTTSUBTOPIC commands you can easily update other services such as your own mosquitto broker.

# Updating Services over Cellular Networks

## Publish to a Mosquitto Broker using AT Commands (7670)

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"        // change to your APN
OK

AT+CSQ?
+CSQ 17;99                            // Query rssi and BER
OK

AT+CREG?                              // Query network registration
+CREG: 0,1                            // Unsolicited Response Code, Status
OK

AT+CGREG?                             // Query GPRS network registration
+CGREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CEREG?                             // Query EPS network registration
+CEREG: 0,1                           // Unsolicited Response Code, Status
OK

AT+CMQTTSTART                         // Start MQTT services
OK
+CMQTTSTART: 0

AT+CMQTTACCQ=0,"<Client ID>"          // Set unique client ID
OK

AT+CMQTTCONNECT=0,"tcp://<IP-address or URL>:1883",60,1,"<username>","<password>"
                                      // Connect to Broker using username&password
OK
+CMQTTCONNECT: 0,0

AT+CMQTTTOPIC=0,xx                    // xx =  length of Topic
><Topic>                              // Topic to publish to *
OK

AT+CMQTTPAYLOAD=0,xx                  // xx = length of message / payload to be sent
><Message / Payload                   // Actual message / payload to be sent
OK
```

# Updating Services over Cellular Networks

```
AT+CMQTTPUB=0,0,20                  // Publish message to server (index, QOS, timeout)
OK
+CMQTTPUB: 0,0                      // index, status (0=success)

AT+CMQTTDISC=0                      // Disconnect from Broker
OK
+CMQTTDISC: 0,0

AT+CMQTTREL=0                       // Release current client
OK

AT+CMQTTSTOP                        // Stop MQTT service
OK
+CMQTTSTOP: 0
```

* To publish to mosquitto you need to select a Topic. This can either be an existing Topic or created dynamically.

Using the SIM7670 module the above code will update your mosquitto broker. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

## Subscribe to a Mosquitto Broker using AT Commands (7670)

Code in blue is entered, with expected / typical reply in black (query requests are not necessary but recommended).

```
AT
OK

AT+CGDCONT=1,"IP","<your APN>"      // change to your APN
OK

AT+CSQ                              // Query rssi and BER
+CSQ: 15,99
OK

AT+CREG?                            // Query network registration
+CREG: 0,1                          // Unsolicited Response Code, Status
OK

AT+CGREG?                           // Query GPRS network registration
+CGREG: 0,1                         // Unsolicited Response Code, Status
OK
```

# Updating Services over Cellular Networks

```
AT+CEREG?                                   // Query EPS network registration
+CEREG: 0,1                                 // Unsolicited Response Code, Status
OK

AT+CMQTTSTART                               // Start MQTT services
OK
+CMQTTSTART: 0

AT+CMQTTACCQ=0,"<Client ID>"                // Set unique client ID
OK

AT+CMQTTCONNECT=0,"tcp://<IP-address or URL>:1883",60,1,"<username>","<password>"
                                            // Connect to Broker using username&password
OK
+CMQTTCONNECT: 0,0

AT+CMQTTSUBTOPIC=0,xx,0                     // xx =  length of Topic
><Topic>                                    // Topic to subscribe to *
OK

AT+CMQTTSUB=0                               // Subscribe to the topic
OK
+CMQTTSUB: 0,0
```

*At this point you are now Subscribed*

When a client publishes a message to the Broker, you will automatically receive:-

```
+CMQTTRXSTART: 0,40,2                       // Start with length of topic and message
+CMQTTRXTOPIC: 0,40                         // Length of Topic to be received
<topic subscribed to>                       // Topic
+CMQTTRXPAYLOAD: 0,2                        // Length of message / payload to be received
17                                          // Actual message / payload
+CMQTTRXEND: 0                              // End of message


AT+CMQTTUNSUB=0,0                           // Unsubscribe from Broker
OK
+CMQTTUNSUB: 0,18

AT+CMQTTDISC=0                              // Disconnect from Broker
OK
+CMQTTDISC: 0,0
```

# Updating Services over Cellular Networks

```
AT+CMQTTREL=0                    // Release current client
OK

AT+CMQTTSTOP                     // Stop MQTT service
OK
+CMQTTSTOP: 0
```

* To subscribe to mosquitto you need to select a Topic. This can either be an existing Topic or created dynamically.

Using the SIM7670 module the above code will subscribe to your Mosqutitto Broker. If incorporating into Flowcode it is recommended to look for the correct responses and act before proceeding further.

# Updating Services over Cellular Networks

## AT Commands 7600 (v2.xx)

If using a 7600 module with AT Commands v2.xx then the code in the previous pages may or may not work dependent on which AT version is actually installed.

These commands / changes worked for me and may help guide you.

### Updating ThingSpeak using "HTTP" AT Commands (7600)

After the AT+HTTPPARA command is sent, issue the following two commands to set additional headers before proceeding.

```
AT+HTTPPARA="UA","SIM7600_MODULE"
OK

AT+HTTPPARA="CONTENT","application/x-www-form-urlencoded"
OK
```

### Publish to MQTT (7600)

Add time out value to CMQTTPUB and CMQTTDISC

```
AT+CMQTTPUB=0,0,60
OK
+CMQTTPUB: 0,0

AT+CMQTTDISC=0,60
OK
+CMQTTDISC: 0,0
```

### Subscribe to MQTT (7600)

Change AT+CMQTTUNSUB and AT+CMQTTDISC to:-

```
AT+CMQTTUNSUBTOPIC=0,xx,0        // xx =  length of Topic
><Topic>                         // Topic to unsubscribe from
OK

AT+CMQTTDISC=0,60
OK
+CMQTTDISC: 0,0
```