AWS IoT AT Command Reference

Table of Contents

Introduction

The Anynet Click module responds to a small number of AT commands.

Overview

AT Command Interface

The Serial interface to the Click offers custom AT commands to allow data to be sent to the AWS IoT Platform, and extracted from it.

AnyNet Secure Firmware

The Eseye provided embedded firmware that runs on the Click module. This firmware handles loading the security material from the AnyNet SIM and establishing the MQTT / TLS links. Additionally it handles the link maintenance based upon the application requirements.

AnyNet Secure SIM

The Eseye AnyNet Secure SIM is soldered on the Anynet 2G Click module combines a multi profile global SIM solution (AnyNet) and the AWS security provisioning functionality.

Serial Interface

The Serial Interface operates at 9600 baud, 8n1

AnyNet Click AWS AT Commands

These are the AT command extensions that will be handled by the AnyNet Secure firmware to support easy communication with AWS IoT.

AT Commands

AT+QCCID	Query ICCID
Test Command **AT+QCCID=?**	Response **OK**
Read Command **AT+QCCID**	Response **ICCID of the SIM** **OK**

AT+GSN	Query IMEI
Test Command **AT+GSN=?**	Response **OK**
Read Command **AT+GSN**	Response **IMEI of the SIM** **OK**

AT+AWSVER	Query Version
Test Command **AT+AWSVER=?**	Response **OK**
Read Command **AT+AWSVER**	Response **VERSION info for the AnyNet AWS IoT code** **OK**

AT+AWSPUBOPEN	Open AWS Publish Topic
Test Command **AT+AWSPUBOPEN=?**	Response **+AWSPUBOPEN: (0-1),<awstopic>** **OK**
Read Command	Response

AT+AWSPUBOPEN?	list of configured publish topics in the format **<awspindex>: <awstopic>/<thingname>** **OK**
Write Command[1] **AT+AWSPUBOPEN=<awspindex>, "<awstopic>"**	Response If the format is correct the response will be **OK** Otherwise the response will be **ERROR** **The following URC will notify the result of the connection.** **+AWSPUBOPEN: <awspindex>,<aswconcode>**

Parameter	
<awspindex>	AWS publish socket index 0-1
<awstopic>	String Type, AWS publish topic path
<awsconcode>	0 Success -1 Error -2 Socket Already in use

<awstopic> will be combined with the "Thing Name" that is delivered to the Anynet secure SIM during device provisioning to form the complete subscribed resource name in the form <awstopic>/<thing_name>

e.g. temperature/thing_0001

The '/' and Thing name do not need to be present in the AWSSUBOPEN command as these will be added to the subscription automatically

AT+AWSPUBLISH	**AWS Publish**
Test Command **AT+AWSPUBLISH=?**	Response **AWSPUBLISH: (0-1), (1-240)[,(0-1)]** **OK**
Read Command **AT_AWSPUBLISH?**	Response **OK**
Write Command **AT+AWSPUBLISH=<awspindex>, <length>[,<qos>]**	Response

[1] Note that the aws topic must be contained within quotes

	If the topic connection is not open, if the device is unable to accept data or some other error occurs: **ERROR** Response **>** At the prompt input the data to be sent on this topic, data is accepted for the topic until <length> bytes are input If sending is successful: **SEND OK** If the process of sending data is blocked: **SEND FAIL**

Parameter	
<awspindex>	AWS publish socket index 0-1
<length>	Length of data to be delivered for this topic
<qos>	The QOS requirements on the delivery of this data. 1 Send, no acknowledgement required. SEND OK returns immediately 2 Send, acknowledgement required. SEND OK return on acknowledgement If not specified the default QOS is 1

AT+AWSSUBOPEN	Open AWS Subscribe Topic
Test Command **AT+AWSSUBOPEN=?**	Response **+AWSSUBOPEN: (0-1),<awstopic>** **OK**
Read Command **AT+AWSSUBOPEN?**	Response list of subscriber topics in the format **<awssindex>: <awstopic>/<thingname>** **OK**
Write Command[2] **AT+AWSSUBOPEN=<awssindex>, "<awstopic>"**	Response If the format is correct the response will be **OK** Otherwise the response will be **ERROR** **The following URC will notify the result of the connection.** **+AWSSUBOPEN: <awssindex>,<aswconcode>**

Parameter

<awssindex>	AWS subscribed socket index 0-7
<awstopic>	String Type, AWS publish topic path
<awsconcode>	0 Success -1 Error -2 Socket Already in use

<awstopic> will be combined with the "Thing Name" that is delivered to the Anynet secure SIM during device provisioning to form the complete subscribed resource name in the form <awstopic>/<thing_name>

e.g. resolution/thing_0001

The '/' and Thing name do not need to be present in the AWSSUBOPEN command as these will be added to the subscription automatically

AWS Notify of Subscribed Data	
+AWS:<awssindex>,<length> **<CR><LF><Binary Data>**	

[2] Note that the aws topic must be contained within quotes

Reference	This URC is to present subscribed data for a topic

Parameter

<awssindex>	AWS subscribed socket index 0-7
<length>	Length of data to be delivered for this topic

AT+AWSPUBCLOSE	**Close AWS Publish Topic**
Test Command **AT+AWSPUBCLOSE=?**	Response **+AWSPUBCLOSE: (0-7)** **OK**
Read Command **AT+AWSPUBCLOSE?**	Response **OK**
Write Command **AT+AWSPUBCLOSE=<awspindex>**	Response If the format is correct the response will be **OK** Otherwise the response will be **ERROR** **The following URC will notify the result of the connection.** **+AWSPUBCLOSE: <awspindex>, <aswconcode>**

Parameter

<awspindex>	AWS publish socket index 0-7
<awsconcode>	0 Success -1 Error -2 Socket Not Open

AT+AWSSUBCLOSE	**Close AWS Subscribed Topic**
Test Command **AT+AWSSUBCLOSE=?**	Response **+AWSSUBCLOSE: (0-7)** **OK**
Read Command **AT+AWSSUBCLOSE?**	Response **OK**
Write Command **AT+AWSSUBCLOSE=<awspindex>**	Response If the format is correct the response will be **OK**

	Otherwise the response will be **ERROR** **The following URC will notify the result of the connection.** **+AWSPUBCLOSE: <awspindex>, <aswconcode>**

Parameter

<awspindex>	AWS subscribed socket index 0-7
<awsconcode>	0 Success -1 Error -2 Socket Not Open

AT+AWSSTATE	**Check AWS State**
Test Command **AT+AWSSTATE=?**	Response **+AWSSTATE: (0-8)** **OK**
Read Command **AT+AWSSTATE?**	Response **+AWSSTATE: <state>** **OK**

Parameter

<state>	AWS system state 0 Idle 1 Waiting Keys 2 Connecting to network 3 Establishing SSL 4 SSL Connected 5 Connecting MQTT 6 Ready: MQTT Connected 7 Ready: Subscribed 8 Error

Operation

The act of opening a topic for publishing or subscribing will cause the modem to read the security information from the SIM card and establish a connection to the AWS IoT service.

Publishing will cause the binary data to be sent to AWS via MQTT and TLS.

Any data received on a subscribed topic will cause the modem to issue the +AWS URC to the application.

The AnyNet secure firmware will check for new security material upon an MQTT or TLS error.