SolidSense Edge Gateway

Experience The Edge of The Internet of Things

SolidSense BLE MQTT Gateway

MQTT interface definition

Version 1.0.4

SolidRun
Embedded Edge Computing

Embedded Edge Computing
www.Solid-Run.com

SolidRun Ltd., 7 HaMada st.,
2069201, Yokneam Illit, Israel.

Scope of the document

The document defines the MQTT interface for the SolidSense BLE MQTT gateway.

The document does not include the Bluetooth standard and is assumed to be known by the reader.

General structure of the MQTT interface

Identifiers

The gateways are directly connected on the MQTT Broker, so they do not appear in the topic sequence.

The BLE devices (tags/sensors/beacons/….) are uniquely identified by their MAC address in the form of 5 pairs of hexadecimal digits separated by colon.

Ex: 22:54:EF:6D:0C

BLE services and characteristics UUID are represented by their canonical hexadecimal strings.

By defaults the gateway ID is its hostname (serial number)

Dynamically typed values

The gateway is translating basic types from and into their Bluetooth transport representation. Here is the enumeration defining the types on the interface

Keyword	Numeric value	Associated type
BTRAW	0	Hexadecimal string representing binary values. Little endian on 16 bits words assumed. Length shall always be even.
INT	1	Integer, can be signed
FLOAT	2	Float, can be signed
STRING	3	UTF 8 string assumed
UUID	4	Bluetooth UUID as Hexadecimal string following the standard
BYTES	5	Binary data in hexadecimal string. This is not strictly equivalent to BTRAW as byte ordering can change

High level topics summary and MQTT structure

Topic name	Publish	Subscribe	Purpose
scan	N	Y	Control the gateway BLE scanner
filter	N	Y	Control the scan filters
scan_result	Y	N	Send back scanner status
advertisement	Y	N	Send advertisement from a device
gatt	N	Y	Top level topic for all GATT transactions from the broker
gatt_result	Y	N	Topic to send back GATT results

General syntax

/<high level topic>/<gateway ID>/<device address>/<sub-topic>

Sub-topics are optional and linked to a measurement or status that can be published individually like a temperature for instance. Device address is not used for the Scan and Filter topics. Sub-topics are optional (see corresponding paragraph)

Gateways will subscribe only for their own ID

Payload encoding

Payload is JSON encoded and the content and structure are explained for each high-level topic.

Scan topic

The role of the topic is to control the gateway behavior on the BLE interface. There is only a payload that define how the gateway will listen for BLE device and report the results.

These parameters can also be set via configuration file to allow the BLE gateway to start operating as soon as the system starts.

Keyword		**Type**	**Value**	**Action/signification**
command	M	String	start	Start listening on BLE and reporting advertising from the devices. The command is not time bound and will remain in effect until a stop command is received
			stop	Stop listening for device advertisement. This is occurring at the of the minimum listen period or after timeout for periodic scan. For indefinite scan started by “start” a scan_result is sent upon the actual end of scan.
			time_scan	Start listening for a time bound period if no timeout is defined the default timeout in the configuration is applied
The following fields are used to configure the scan what is sent at then of the scan (topic scan_result)				
timeout	O	Float		Scan time out in seconds (default=10s)
period	O	Float		Repeat period for timed scan. If 0 or omitted, no repeat. If superior to timeout then a pause time is set between scans. If inferior or equal to scan timeout, then scan restarts immediately
result	O	String	none	Nothing reported in timeout scan
			summary	Publish a summary report (see scan_result) – this is the default
			devices	Publish an array of devices detected and selected by filters during scan (see scan results)
The following define when and what will be published in the advertisement topic. Mode details to be given in the advertisement topic payload description,				
advertisement	O	String	none	No advertisement reported. Useful with time_scan to have the devices reported only at the end
			min	Minimum set of data (default)
			full	Full set of data of the advertisement frame
sub_topics		Boolean	false	No sub-topics (default)
			true	Beacons or Service data as separated sub-topics (see dedicated paragraph)
adv_interval	O	Float		Minimum reporting interval of advertising in seconds. Useful to avoid flooding the broker when some devices are advertising at high rate. If omitted each advertisement received will be pushed towards the broker.

Example JSON payload

{"command":"time_scan","timeout":20,"advertisement":"min","period":30,"sub_topics":true}

Start a periodic scan every 20 seconds for 20 seconds. Reports minimum info on advertisement and publish sub-topics when corresponding data are sent by the devices

Filter topic

This topic allows to set filters to avoid having advertisement messages for beacons or sensors that are not part of the application. Without filter, all BLE devices detected by the gateway are reported. Filters can be set statically via configuration file. They are always passed as a JSON array.

Keyword		Type	Value	Action/signification
type	M	String	rssi	Keep devices with a RSSI above or equal to the minimum defined (negative integer)
			white_list	Keep devices whose addresses are specified
			connectable	Keep devices that have the connectable flag (True or False) equal to the one specified
			starts_with	Keep the devices whose name starts with the string specified. Devices with no name are ignored
			mfg_id_eq	Keep the devices with the specifies Manufactured ID (4 hex digits). Devices with no manufacturer ID defined are ignored
			none	Do not create a filter. Used to clear the filter list and shall be the only item
min_rssi	M	Integer	Negative value between -30 and -99	Only for RSSI filter
match_string	M	String		Only for the starts_with filter
addresses	M	Array of string		Only for the white_list filter
connectable_flag	M	Boolean		Only for the connectable filter
mfg_id	M	Integer (4 hex digits)		Only for the Manufacturer ID filter

To combine filters, just pass an element to the array. In that case a AND condition will be applied.

Example JSON payload

[{"type":"rssi","min_rssi":-80},{"type":"connectable","connectable_flag":true}]

Select the devices with a RSSI above -80 that are connectable.

Note on RSSI filter: as the RSSI value can fluctuate rapidly, the filter is re-evaluated for each advertisement packet received and only when the threshold is reached a MQTT message is sent. However, if during the scan period the filter is validated once, then the device is kept as valid device and sent in the scan result message. It becomes therefore eligible for GATT transactions.

Scan result topic

This topic is published at the end of a timed scan when the Result keyword in the Scan parameters is not None

If the Summary is selected, then the following JSON structure is sent

Keyword		Type	Value	Action/signification
typestamp	M	Float		Time is seconds from the Epoch of scan end

error	M	Integer		Nonzero if an error occurred
dev_detected	M	Integer		Total number detected during the scan
dev_selected	M	Integer		Total number of devices selected after filtering

When the Devices option is selected the payload also includes an array of the following JSON structure for each selected device

Keyword		**Type**	**Value**	**Action/signification**
address	M	String		MAC address of the device
name	M	String		Local Name of the device, if no name is defined then a zero-length string is sent
rssi	M	Integer		Negative value corresponding to the Received Signal Strength Indicator

Advertisement topic

This topic is published by the gateway each time a advertisement frame is received and meet the following conditions:

a) The device is not filtered out
b) The advertisement frame is not within the advertisement interval set in the scan parameters

The device MAC address is part of the topic structure

/advertisement/<gateway ID>/<Device MAC>/

The payload structure has 2 options min or full. In the JSON structure table, the **min** fields are referred as Mandatory while those from the Full option only as marked as **full**, with a possible Optionality indicator meaning that the field is not present in the advertisement frame from the device

Keyword		**Type**	**Value**	**Action/signification**
local_name	M	String		Local Name of the device, if no name is defined then a zero-length string is sent
timestamp	M	Float		Time is seconds from the Epoch
rssi	M	Integer		Negative value corresponding to the Received Signal Strength Indicator
flags	M	Integer		Bitwise indicator (1 byte)
connectable	M	Boolean		True if the device is connectable
service_data	F/M	Integer		Number of service data values. If nonzero, then an array of service data value
service_data_array	F/O	Array		
service_uuid	M	Integer		16 bits UUID of the service
type	M	Enum		Value data type (see table)
value	M	Variable		Value of the service data, if type is Raw, then the raw hexadecimal byte string is sent
mfg_id	F/O	Integer		Manufacturer ID (16bits)
mfg_data	F/O	Hex_string		Hexadecimal string of the manufacturer data less the Manufacturer ID
tx_power	F/O	Integer		Value of Tx Power when sent from the device

GATT topic

The GATT topic allows to interact with the GATT protocol with devices that are GATT server. The gateway is always a GATT client.

Only the devices that have been selected during a scan can handle GATT request. If a request is sent to a device that has not been seen (and selected) during a scan, the request is rejected.

The topic is structured the following way:

/gatt/<gateway ID>/<device MAC> and the description of the request is in the payload

Keyword		**Type**	**Value**	**Action/signification**
command	M	String	read	Read a characteristic
			write	Write a characteristic
			allow_notifications	Allow receiving notifications for that characteristic. If a value is present in the request, it is written to the characteristic just after allowing the notifications for that device. (see details below)
			discover	Discover the device and send back the list of services and characteristics
transac_id	O	Integer		If set, that id will be present in the corresponding GATT result frame(s)
service	O	UUID		If present the request will look only for the service with that UUID, if not present this will raise an error an nothing is returned. For devices with many services this is speeding up the transactions
properties	O	Boolean		This parameter is used only for discover and if true, the properties of the characteristics will be reported as well.
bond	O	Boolean		If True, then a bond (pair) request is made after the connection
keep	O	Float		If present and non-zero, keep the connection open for the number of seconds after a message has been received from the device. A default value is applied for allow_notifications
The following fields needs to be passed for read/write/allow_notifications Single characteristic action or array are mutually exclusive				
characteristic	O	UUID		UUID of the Characteristic to be written or read or to allow notifications upon. UUID can be short form (4bytes hexadecimal string) or long form.
type	O	Enum		Type of the value to be written or expected for read. If omitted Raw Hex string is assumed
value	O			Value to be written
action_set	O	Array of tuple (char,type) or (char,type,value)		For optimization, an array of the following tuple can be passed and will

				be processed in one connection to the device

Note on allowing notifications.
To allow notifications, the process is writing the GATT Descriptor corresponding to the Characteristic.
If a type & value are indicated, then they are written to the Characteristic. To allow notifications on one Characteristic and start notifications by writing to same or another Characteristic, this is supported by using an action set. The first action shall contain on the Characteristic UUID of the one that shall send the notifications and the second shall include the full set= Characteristic UUID, type and value.

Example JSON payload

{"command":"read","keep":10.0,"action_set": [
{"characteristic":"2A00","type":3},
{"characteristic": "2A19","type":1},
{"characteristic":"F000AA01-0451-4000-B000-000000000000","type":5}
]}
Perform reading 3 characteristics and keep the connection alive for 10 seconds

{"command":"discover","keep":10.0,"properties":true,"service":"1800"}

Discover the service 0x1800 (Generic Access), return the characteristics and properties and keep the connection alive for 10 seconds

{"command":"allow_notifications","keep":10.0,"action_set": [
{"characteristic":"6e400003-b5a3-f393-e0a9-e50e24dcca9e","type":3},
{"characteristic": "6e400002-b5a3-f393-e0a9-e50e24dcca9e","type":3,"value":"LED_OFF"}

Trigger activity on Nordic serial service (6e400001-b5a3-f393-e0a9-e50e24dcca9e)
Allow notification on the first characteristic then write the string “LED_OFF” on the second.

GATT result topic

This topic is used by the gateway to publish the result of the corresponding GATT request

/gatt_result/<gateway ID>/<Device MAC>

The structure of the payload is depending from the nature of the request.

For read/write/notification

Keyword		**Type**	**Value**	**Action/signification**
command	M	String		Nature of the result (read/write/notification/discover)
error	M	Integer		If nonzero, then an error occurred
transac_id	O	Integer		Present if a transaction ID was set for the request
result	O	JSON structure		One of the following
For read/write/notify				
characteristic	M	UUID		For each valid write
For read and notify only				
type	M	Enum		
value	M			

For discover
An array of service descriptors is published with the GATT_Description tag.

Keyword		Type	Value	Action/signification
service_uuid	M	UUID		UUID of the service up to 128 bits
characteristics	M	Array of UUID		Set of characteristics UUID supported
If the properties parameter is set to true then characteristics are reported as a array of				
uuid	O	UUID		UUID of the characteristics
properties	O	String		List of properties in a single string separated by space

Advertisement sub-topics

In addition to the standard advertisement message and with the same rule the gateway can be configured to publish a specific message for the service data or beacons (Eddystone or iBeacon)

/advertisement/<gateway ID>/<MAC ADDRESS>/<Service data name or uuid>
Or
/advertisement/<gateway ID>/<MAC ADDRESS>/eddystone
Or
/advertisement/<gateway ID>/<MAC ADDRESS>/ibeacon

The service data name is the one from the Bluetooth standard. The value field will be converted according to standard. Not all service data Characteristics are implemented.

The payload for service data includes

Keyword		Type	Value	Action/signification
timestamp	M	float		Time in seconds of the Epoch
type	M	Enum		Type of the value
value	M			Converted following Bluetooth standard

For Eddystone

Keyword		Type	Value	Action/signification
timestamp	M	float		Time in seconds of the Epoch
type	M	Int		Type of Eddystone beacon
For UID Beacons				
txpower	M	int		
beacon_id	M	String		HEX digits
For URL Beacons				
txpower	M	Int		
url	M	String		
For other beacons				
frame	M	String		Hex digits

For iBeacon

Keyword		Type	Value	Action/signification
timestamp	M	float		Time in seconds of the Epoch
uuid	M	UUID		UUID of the Beacon
majmin	M	String		Major-Minor values
txpower	M	Int		

To avoid duplicate the advertisement messages, the standard advertisement messages can be turned off in the scan configuration command option.

General behavior and limitations

1. Scan and GATT operations are mutually exclusive. This is controlled by the gateway. All GATT connections are terminated before a Scan start. GATT transaction requests are refused while a Scan is running
2. In the current version (0.5), time_scan is more reliable and for permanent scanning it is preferred to use periodic scan instead of start/stop.
3. By default the service runs on hci0, but when several Bluetooth interface are existing this can be changed in the configuration file. Currently only one interface can be managed at a given time.

Configuration via Kura plugin

A configuration service in Kura (and accessible via Kapua) allows the configuration of the following parameters:

1. Activation/Deactivation of the BLE/MQTT gateway
2. Device ID (default is hostname)
3. MQTT broker URL and port
4. MQTT broker credentials
5. MQTT secure connection
6. Scan parameters
   To allow an autonomous operation the BLE scan can be configured from the screen, with all options defined in the scan payload definition available. If the scan is configured locally, then the gateway will start to operate after the configuration is applied and automatically upon gateway startup
7. Filters
   Filters can be defined statically in the configuration screen

Configuration files

All the gateway configuration files are in the /data/solidsense/ble_gateway directory. The file that is generated by the Kura plugin is *bleTransport.service.cfg*. When using the Kura plugin, it is not recommended to modify directly that file as it is overwritten by Kura.

There is also a general parameters file in a JSON file: parameters.json. This file is read by the service upon start only and include the following fields:
Interface: name of the interface to be used hci0/hc1/hci2
Notif_MTU: maximum length of a notification message
Trace: level of tracing (info by default) info/debug/error
Debug_bluez: allow low level tracing in the bluez interface and stack. Only for troubleshooting
Max_connect: maximum number of simultaneous GATT connection