Dash Protocol

Revision Date: 12 December 2023

1. Description

Dash Protocol is a lightweight data-exchange protocol for asynchronous networking between an **Dash IoT Dashboard** and **IoT devices**. The **Dash IoT Dashboard** may be deployed on a mobile device or computer to monitor or control one or more IoT devices. An IoT device may be any networked or connected device that is desired to be monitored or controlled (e.g. an Arduino, Raspberry Pi, Beaglebone, PC, Mac etc).

The **Dash Protocol** has the following features:

- ASCII text messages (Human readable)
- Easy to generate and parse
- Designed to be transported by any network
- Variable length messages
- Asynchronous communication
- Routable
- Extensible

The **Dash IoT Dashboard** enables a number of user configurable **controls** to be placed on the dashboard. The *controls* model the behaviour of the IoT device. Each *control* is able to display information received from the IoT device or to send control information to the IoT device.

Messages are used to send and receive data between *controls* on the Dash IoT Dashboard and a corresponding *abstract controls* within the IoT device, as shown in the figure below:
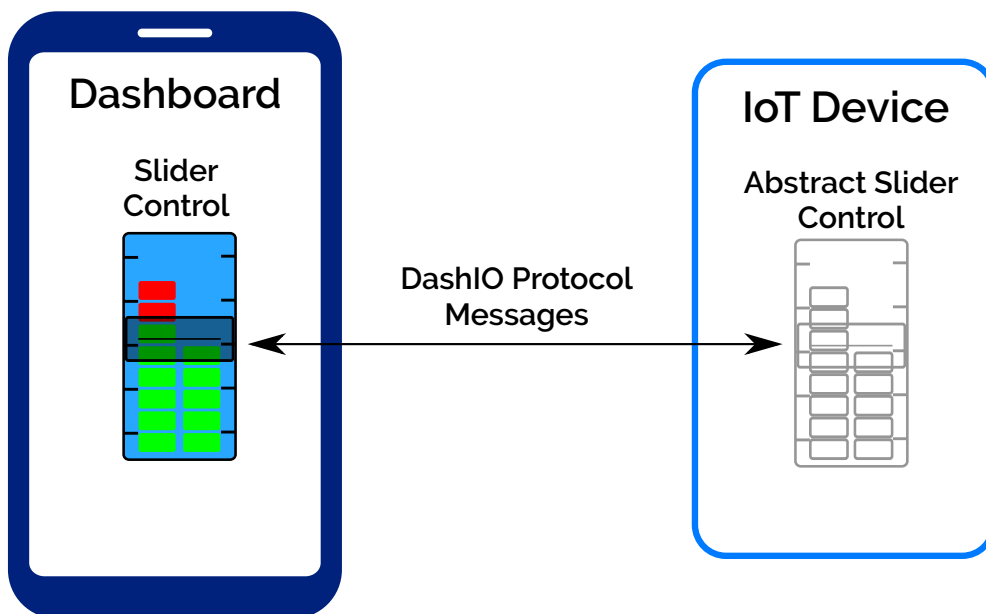
System Messages are also used for purposes other than controls messaging, for example to setup connections and deliver configuration and alarm messages.

1.1. Dash IoT Dashboard

The **Dash IoT Dashboard** displays to the user the information from the *controls* in the IoT device. The controls in the IoT device are modelled within the Dash IoT Dashboard and information is transferred between each control in the IoT device and its counterpart in the Dash IoT Dashboard through one or more *connections*. A connection may be **Bluetooth Low Energy (BLE)**, **TCP** or **MQTT** and the following figure illustrates the connection based messaging:
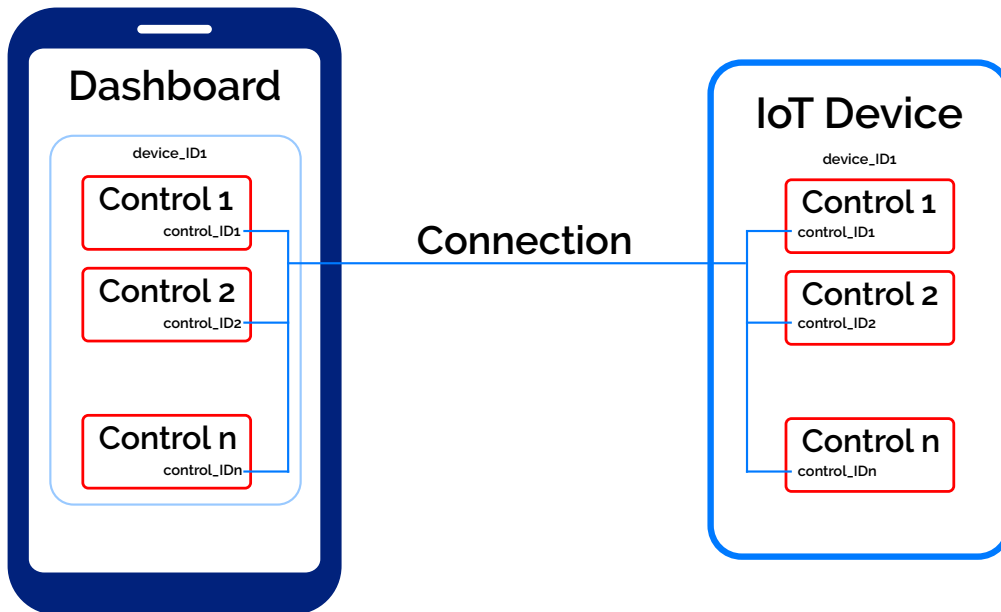
Controls for a device are displayed on the Dash IoT Dashboard on a graphical feature called a **Device View**. There may be one or more Device Views for an IoT Device. The device view contains a "theme" of layout features and colors that are common to all controls displayed on the Device View.

Two identifiers, the *control_ID* and *device_ID*, are required to uniquely identify each **control** on an **IoT device** to the **Dash IoT Dashboard**.

1.2. control_ID

Each control on an IoT device requires an identifier, the *control_ID*, which uniquely identifies the control in the IoT device. The *control_ID* does NOT need to be unique across all devices, so a UUID is NOT required. It can be a short and meaningful text string to make it human readable and to minimise message lengths.

1.3. device_ID and device_name

An IoT device must be uniquely identified to the Dash IoT Dashboard and this identifier is called the *device_ID*. The *device_ID* may be obtained from the **mac** address of **the** BLE, WiFi or Ethernet peripheral within the IoT device. Alternatively, a UUID may be used.

The IoT device also requires a *device_name*, which is used for convenience and should be human readable. It does not need to be unique and may be changed within the IoT device at any time.

1.4. Multiple Connections to a Device

The Dash IoT Dashboard may connect to an IoT device through more than one connection. Each connection is of a different technology (BLE, TCP or MQTT).

An double connection is shown in the following figure:
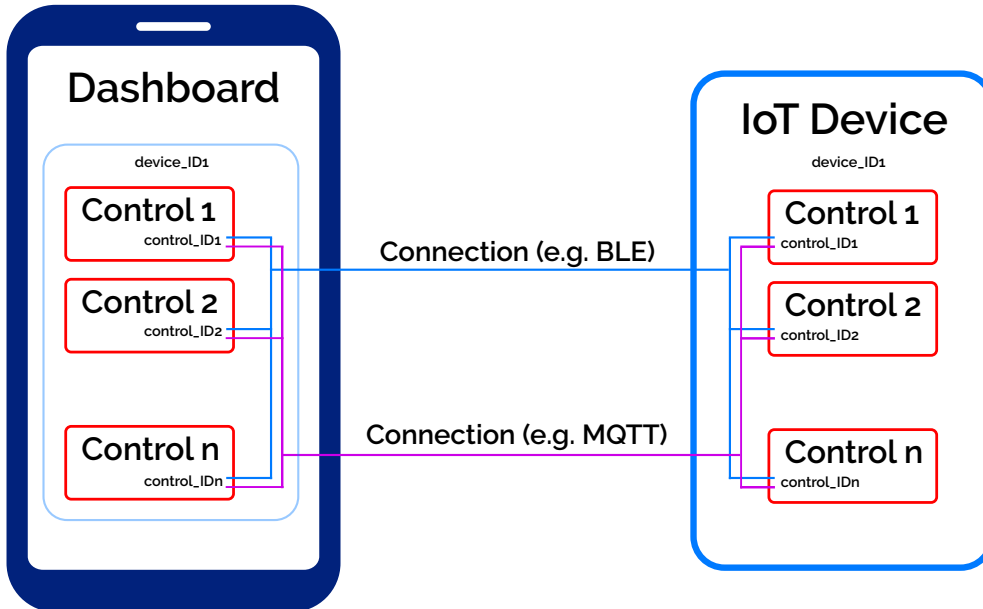
Once the Dash IoT Dashboard has established multiple connections to a single IoT device, it will only continue communicate over one connection with the following priority:

1. BLE
2. TCP
3. MQTT

Therefore, a local connection is selected in preference to a more remote connection.

If the selected connection goes offline (e.g. a BLE connection goes out of range), the Dash IoT Dashboard will switch communication to the next online connection from the priority schedule. Similarly, if a higher priority connection comes online, the Dash IoT Dashboard will switch to use the higher priority connection.

1.5. Connection To Many Devices

By using the **device_ID** and **control_ID** addressing scheme, the Dash IoT Dashboard can communicate with many IoT devices thorough multiple connections simultaneously. Communication between the Dash IoT Dashboard and two IoT devices is shown in the following figure:
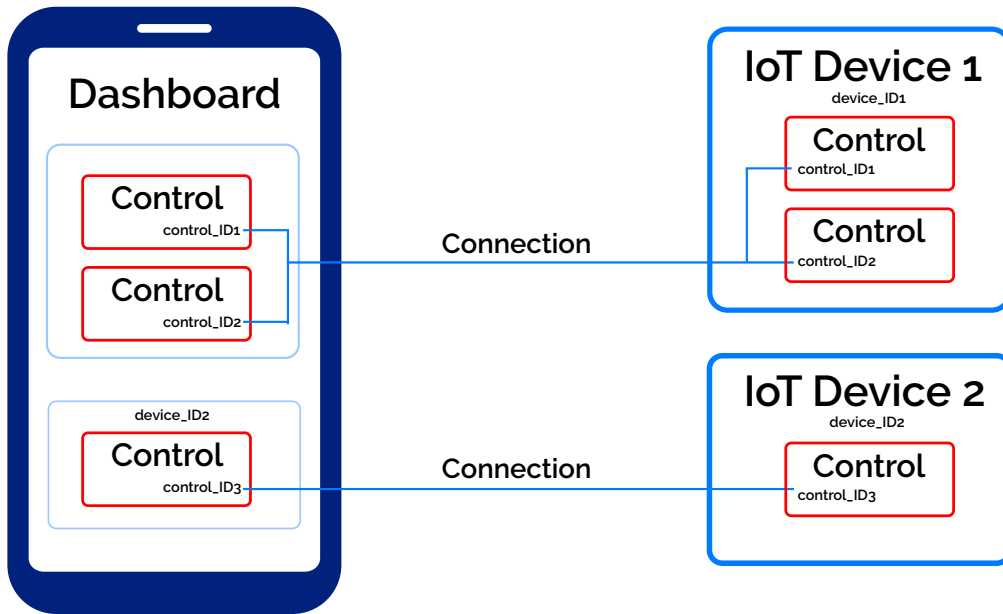
2. Message Structure

Each message is an ASCII string and is shown enclosed within quotations marks with the following message structure:

Message = "\t *Device_ID* \t *Message_Type* \t *Identifier* \t *Data* \n"

The quotation marks are not part of the message. Spaces in the structure are included for clarity but are not part of the message. However, spaces may be used as characters within any text string.

Message Field	Description
\t	Horizontal tab (ASCII TAB) character for delimiting message fields
Device_ID	Unique device identifier (typically the mac address)
Message_Type	ASCII String that uniquely defines the type of message. It may be either a system *System_Message_Type* or a *Control_Type*. A *System_Message_Type* message is used to send networking and configuration information between the Dash IoT Dashboard and the IoT device. A **Control_Type** message is used to send information about a specific control between the Dash IoT Dashboard and the IoT device.
Identifier	The *Identifier* field is an ASCII String identifier for the specific *Message_Type*. If the *Message_Type* is a *Control_Type*, the *Identifier is the control identifier (Control_ID)* If the *Message_Type* is a *System_Message_Type*, the *Identifier* field is NOT required and the message becomes: *Message* = "\t *Message_Type* \t *Data* \n"
Data	The *Data* field is the information sent from, or received by, a control and is defined in the following section. Data may be comprised of zero, one or more sub-fields, each separated by a horizontal tab character "\t". (i.e. some messages do not contain the *Data* field). The number of sub-fields varies according to the *Message_Type* and *Identifier*.
\n	New Line (ASCII LF) character to indicate the end of the message. Messages can be concatenated and the New Line character delimits each message.

2.1. Message_Type

The *message_types* are defined in the following table:

Message_Type	Message_Type values
System_Message_Type	"**WHO**" Announce **device_ID**, **device_type** and **device_name**, message "**CONNECT**" Connection validation "**STATUS**" Request current value for all controls "**CFG**" Configuration data "**NAME**" Device Name "**WIFI**" WiFi Country Code, SSID and Password "**TCP**" TCP port "**DASHIO**" Dash server MQTT Username and Password
System_Message_Type Specific to MQTT connections	"**ONLINE**" Online message for MQTT (for MQTT **Data** topic) "**OFFLINE**" Offline message Last Will and Testament (for MQTT **Data** topic) "**WHO**" Announce **device_ID**, **device_name** (for MQTT **Announce** topic) "**ALM**" Alarm message (push notification for MQTT **Alarm** topic) "**STE**" Data storage enable (for MQTT **Announce** topic)
Control_Type For controls displayed on the Dash IoT Dashboard	"**BTTN**" Button control "**TEXT**" Text display and Input "**SLDR**" Slider control "**BAR**" Bar graph display under slider "**KNOB**" Knob control "**KBDL**" Knob display surrounded by dial "**DIAL**" Dial display "**DIR**" Direction display "**CHRT**" Chart display "**TGRPH**" Graph display with time on the X axis "**SLCTR**" Selector control "**MENU**" Menu control "**BTGP**" Button Group control "**LOG**" Event Log display "**CLR**" Color Picker control "**AVD**" Audio Visual display "**LBL**" Label Box display "**ACTN**" Action and Task processing control

2.2. Data

The **Data** element of the basic message is defined as:

Data = "**Data**$_1$ \t **Data**$_2$ \t …**Data**$_N$" (where N >= 0)

Data Field	Description
Data$_n$ where n = 0 to N	ASCII String providing data to, or from, a control (where n = 1 to N and N is the number of data elements). The **Data**$_n$ may take the form of one of the following: **Text** (e.g. "Go now!", "ON", "OFF") **Integer** (e.g. "-342" or "12") **Real Number** (e.g. "32.543" or "-0.034467")

2.3. System Messaging

System Messages are independent of Controls and contain fields having the following meaning:
- **Message_Type** - defines the type of system message and is referred to as **System_Message_Type**.
- **Data** - The send or receive information of the message, from the Dash IoT Dashboard's perspective, defined as either **Send_Data** or **Receive_Data**.

System Message	Description	Message Fields		
		Message_Type	Data	
			Send_Data	Receive_Data
Who*	Advertise unique device identifier and device name. Also used for searching for TCP devices.	WHO	N/A	Device unique identifier, name of the device, and device type
Connect*	Handshake when connection is made to confirm the IoT device is connected.	CONNECT	N/A	N/A
Status	Control initial condition request message. The IoT device shall use this message as a trigger to update the values of the controls on the Dash IoT Dashboard.	STATUS	N/A	N/A
Configuration	Configuration request and response messages. The response message contains the details of all **controls** and **connections** available on an IoT device to be sent to the Dash IoT Dashboard. This helps simplify setup of the Dashboard.	CFG	Configuration request message contains the display dimensions in mm.	Basic information for all available Controls and Connections.
Device Name	Update the Device Name	NAME	New name of the device	Name of the device
WiFi Setup	Update the WiFi Country Code, SSID and password. Not all IoT Devices will require the Country Code.	WIFI	New WiFI Country Code, SSID and password	N/A
TCP Setup	Update the TCP port	TCP	New TCP port	N/A
Dash MQTT Setup	Update the Dash MQTT username and password	DASHIO	New MQTT username and password	N/A

MQTT Setup	Update the MQTT username and password	MQTT	New MQTT URL, port, SSL (SSL enabled = true), username, and password	N/A
Alarm	Alarm notification message, used to create push notifications for the Dash IoT Dashboard. The Dash IoT Dashboard does not receive these messages directly. They are processed by the MQTT server to create push notification messages for either Apple or Android devices.	ALM	N/A	Alarm message
Online	Device is **online** and able to communicate. Used for MQTT connections to speed up reconnection.	ONLINE	N/A	N/A
Offline	Device is **offline** and unable to communicate. Used for MQTT connections when setup a the Last Will and Testament (LWT) message.	OFFLINE	N/A	N/A

* Please note that "**WHO**" and "**CONNECT**" messages and their reply messages are compulsory as they are required to establish communication with the IoT device and initiate configuration messages or to update the control values on the Dash IoT Dashboard. All other messages are optional.
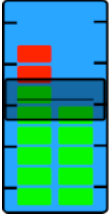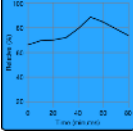
2.4. Controls Messaging
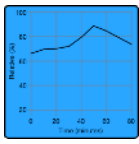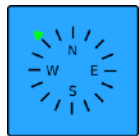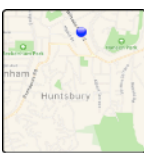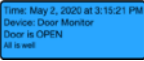
Controls messages contain fields having the following meaning:
- *Message_Type* - defines the type of control the message refers to and is defined as *Control_Type*.
- *Identifier* - is an identification string for the specific control the message is communicating with and is defined as *Control_ID* (not shown in the table below). It may only be comprised of uppercase and lowercase characters and numbers. The Control_ID is only required to be unique to the IoT device and may be a short, human readable string.
- *Data* - The send or receive information for the control, from the Dash IoT Dashboard's perspective, defined as either *Send_Data* or *Receive_Data*.

As well as sending messages to the IoT device, controls in the Dash IoT Dashboard are capable of receiving data from their counterparts in the IoT device and displaying the data. This ensures a feedback path to the Dash IoT Dashboard when the user changes the value of a control on the Dash IoT Dashboard.

Dash IoT Dashboard user configurable controls are shown in the following table:

Dashboard Control		Description	Controls Message Fields		
			Message_Type	Data	
			Control_Type	Send_Data	Receive_Data
Button		**Button** with icon, text, or "LED". A Button may also be used as an indicator for alarm monitoring and resetting.	BTTN	Empty	Button icon or indicator set to "**ON**" or "**OFF**"
Slider With Bar Graph		**Slider** or Fader control over a single or double bar graph behind.	SLDR	Slider value when slider position changed	Slider position value
			BAR	N/A	Bar graph value(s)
Knob With Dial Display		**Knob** control with a colored dial surrounding the knob	KNOB	Knob value when knob rotation changed	Knob position value
			KBDL	N/A	Dial position value
Text Box		Display or send **Text** messages	TEXT	Text message	Message to display
Chart Display		**Chart** that can show zero or more chart lines or bars for one or more IoT devices	CHRT	N/A	Chart line information (Y axis values)

Dashboard Control		Description	Controls Message Fields		
			Message_Type	Data	
			Control_Type	Send_Data	Receive_Data
Time Graph Display		**Time Graph** that can show zero or more time series graph lines for one or more IoT devices.	TGRPH	Request update data, starting from a the last data point time	Graph line information (Time and Y axis value pairs)
Dial Display		**Dial** or gauge display to show a numeric value	DIAL	N/A	Dial position value
Direction Display		Compass style **Direction** rose	DIR	N/A	Compass direction (Decimal Degrees) and secondary data
Map Display		**Map** for displaying GPS data for one or more IoT devices	MAP	Request update of track waypoints, starting from a the last waypoint time	Position data
Selector		**Selector** is used to select on of many text options.	SLCTR	Index of the newly selected option	Index of the selected option
Menu		Presents a popup menu of editable controls, suitable for parameter setup.	MENU	Empty	N/A
Button Group		Presents a popup menu of Button controls.	BTGP	Empty	N/A
Event Log		Shows the most recent event and presents a popup of all events.	LOG	Request update of events, starting from a the last event time	Event data
Color Picker		Pick a colour from either a spectrum or color wheel.	CLR	Color as a hexadecimal string (e.g. "#24F5A4"	Color as a hexadecimal string (e.g. "#24F5A4"
Audio Visual		Play video, audio or show images	AVD	N/A	URL string

For details of each control, please refer to the document titled "Dash Controls".

3. Message Request-Reply Overview

The majority of messages are a request-response message sequence initiated by the Dash IoT Dashboard. There are also some instances where the IoT device may send a message without a specific request message from the Dash IoT Dashboard.

3.1. System Messaging

All but one system message group is generated as a request message from the Dash IoT Dashboard.
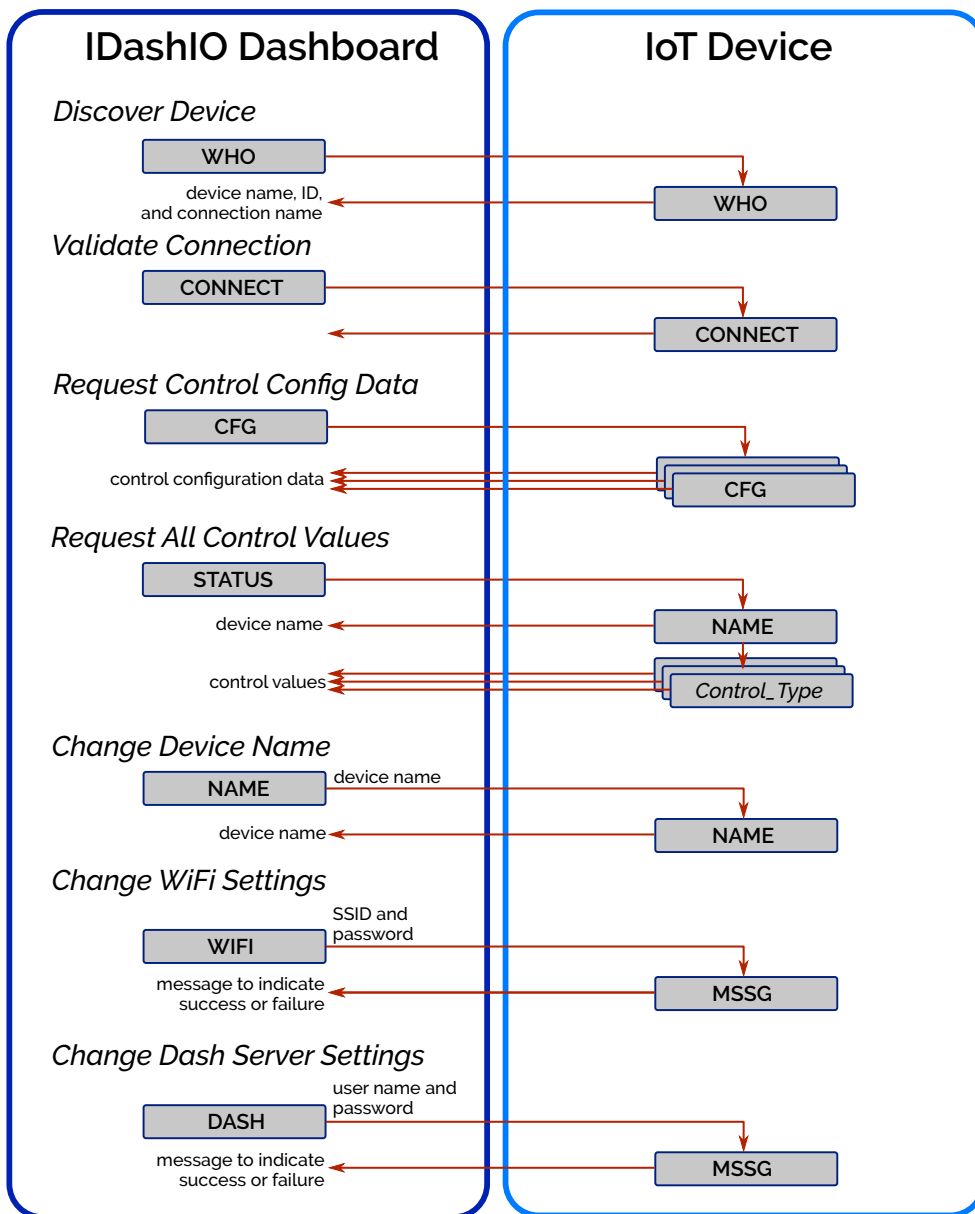
3.2. MQTT Specific System Messaging

For MQTT connections there are four message that are sent from The IoT device without requiring a request from the Dash IoT Dashboard:

ONLINE
> The Dash IoT Dashboard monitors the MQTT **Data** topic for ONLINE messages to speed up reconnection when an IoT device has gone offline and comes back online.

OFFLINE
> The Last Will and Testament (LWT) should be setup for the MQTT **Data** topic with the OFFLINE message.

WHO
> Once the IoT device is connected to the **Dash** server it should send the WHO message to the MQTT **Announce** topic. The enables the **Dash** server to inform the **Dash IoT Dashboard** that the IoT device is available to be connected too.

ALM
> The IoT device may send an ALM (Alarm or push notification) request to the **Dash** server whenever the IoT device needs to send an alarm. The **Dash** server will forward this message to the Apple or Google push notification service, which will intern sent the notifications to the Dash IoT Dashboards that are connected to the IoT device.

DashIO Dashboard	IoT Device
	Once connected, send to **Data** topic
	ONLINE
	At startup, setup LWT to **Data** topic
LWT message	OFFLINE

DashIO Server	IoT Device
	Once connected, send to **Announce** topic
device_name, device_ID, and connection name	WHO
	When desired, send to **Alarm** topic
alarm_ID, header and message	ALM
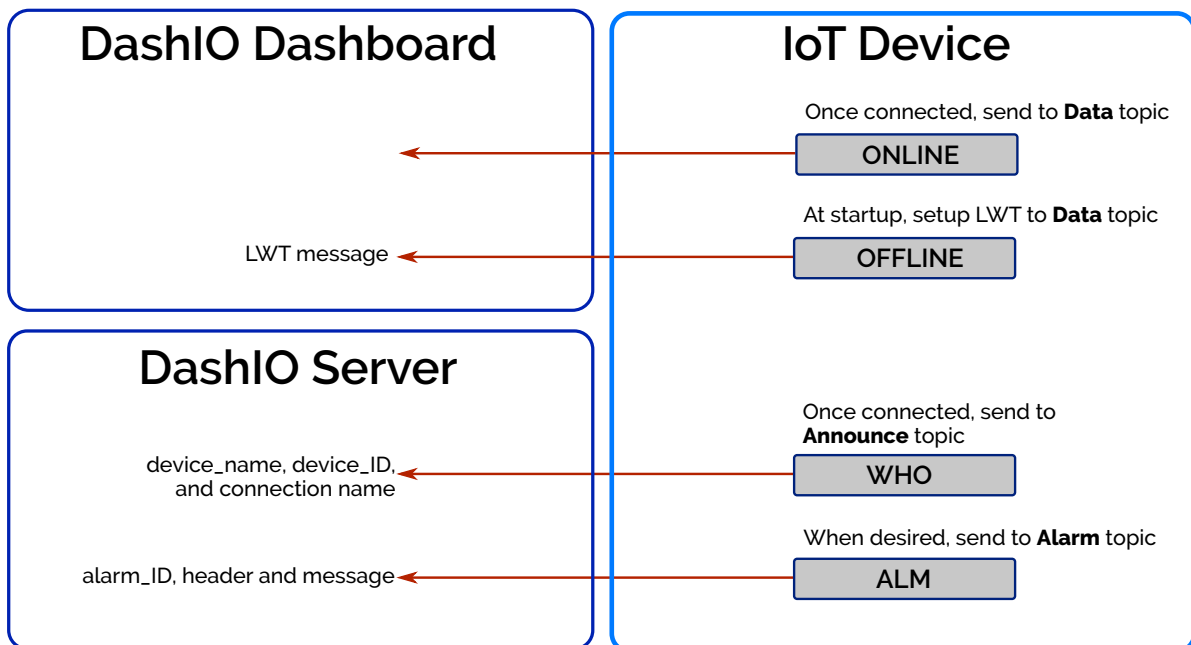
3.3.Control Messaging

Control messages are the core messages that enable to Dash IoT Dashboard and IoT device to transfer IoT information. These messages are based on the **Control_Type** (e.g. BTTN, CHRT, DIAL etc) and pass the value of the control.

The IoT device can send **Control_Type** messages to the Dash IoT Dashboard whenever the control value changes or at regular intervals. Similarly, the Dash IoT Dashboard can send a value to a control when the user changes the value in the Dash IoT Dashboard. However, when the IoT device receives a **Control_Type** message from the Dash IoT Dashboard, it is good practice to reply with another **Control_Type** message to confirm to the Dash IoT Dashboard user that a control value has changed or an action has occurred.

The **Control_Type** in the IoT device response message does not need to match the **Control_Type** in the message received from the Dash IoT Dashboard. For example, a button **Control_Type** message (BTTN) will be sent by the Dash IoT Dashboard when the button is pressed, but the IoT device may reply with a dial control (DIAL) message, to show that a value of the dial has changed as a result of the button press.
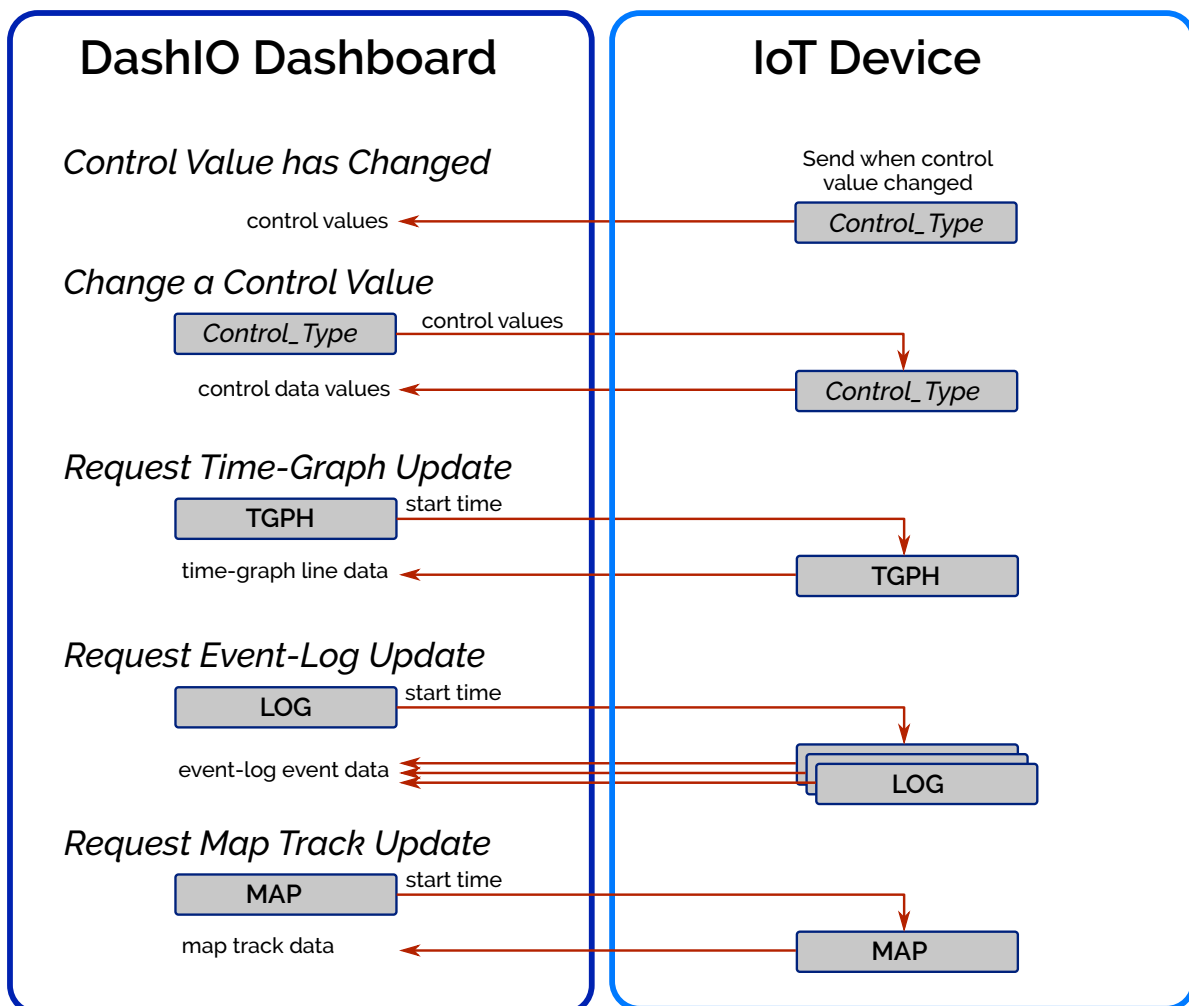
The Time-Graph (TGRPH), Event-Log (LOG) and Map (MAP) **Control_Type** request messages from the Dash IoT Dashboard are used to request updated Time-Graph, Event-Log and Map information respectively. These request messages include a **start time**, which indicates to the IoT device that the Dash IoT Dashboard only requires information from AFTER the start time.

4. Message Details

Messages for each **Message_Type** are listed in the following subsections:

4.1. Messages From Dash IoT Dashboard to IoT device

Action	Message	Notes
Request Device Connection Information	"\t WHO \n"	The WHO message must be replied too.
Validate Connection	"\t **Device_ID** \t CONNECT \n"	The CONNECT message must be replied too.
Request control initial conditions	"\t **Device_ID** \t STATUS \n"	Reply with the values for the controls, the set their initial conditions on the Dash IoT Dashboard after a connection has been established and validated.
Request Config Data	"\t **Device_ID** \t CFG \t **Dashboard_ID** \t **Width** \n"	The IoT device use respond with all configuration messages. The **Dashboard_ID** is a UUID that identifies the requesting dashboard. Width is the Dash IoT Dashboard screen with number of columns.
Update Device Name	"\t **Device_ID** \t NAME \t **Device_Name** \n"	The **Device_Name** is the name of the IoT device which may be changed from the Dash IoT Dashboard. If **Device_Name** is an empty string, this message is simply requesting the current device name from the IoT device. The IoT device must reply to this message.
Update WiFi Settings	"\t **Device_ID** \t WIFI \t **Country_Code** \t **SSID** \t **Password** \n"	Update the WiFi credentials. The IoT device must reply to this message.
Update TCP Settings	"\t **Device_ID** \t TCP \t **Port** \n"	Change the TCP port. The IoT device must reply to this message.
Update DashIO Settings	"\t **Device_ID** \t DASHIO \t **User_Name** \t **password** \n"	Change the credentials for the DashIO server. The IoT device must reply to this message.
Reset Device	"\t **Device_ID** \t RST \n"	The IoT device must reply to this message before resetting.
Button Press	"\t **Device_ID** \t BTTN \t **Control_ID** \n"	
Send Text	"\t **Device_ID** \t TEXT \t **Control_ID** \t **Data** \n" "\t **Device_ID** \t TEXT \t **Control_ID** \t **Data** \t **Color** \n"	**Data** is any ASCII text **Color** is an optional field defining the text color. It is either the color name from colors in Dash IoT Dashboard (e.g. "blue") or a hexadecimal string (e.g. "#23A89F")

Action	Message	Notes
Slider Moved	"\t **Device_ID** \t SLDR \t **Control_ID** \t **Data** \n"	**Data**, with or without decimal point
Menu Press	"\t **Device_ID** \t MENU \t **Control_ID** \n"	
Button Group Press	"\t **Device_ID** \t BTGP \t **Control_ID** \n"	
Selector Selected	"\t **Device_ID** \t SLCTR \t **Control_ID** \t **Data** \n"	**Data** is the index of the selected item.
Knob Turned	"\t **Device_ID** \t KNOB \t **Control_ID** \t **Data** \n"	**Data**, with or without decimal point
Color Picked	"\t **Device_ID** \t CLR \t **Control_ID** \t **Color** \n"	Chosen **Color** as a hexadecimal string (e.g. "#23A89F")
Request Time Graph Data	"\t **Device_ID** \t TGRPH \t **Control_ID** \t **Dashboard_ID** \t **Time** \n" "\t **Device_ID** \t TGRPH \t **Control_ID** \t **Dashboard_ID** \t **Time** \t **Line_ID** \n"	The IoT device will then download the graph line data for all datapoints recorded after this **Time***. The time is UTC. If the **Line_ID** field is NOT included or is blank, the IoT device will download the data for ALL graph lines. The **Dashboard_ID** is a UUID that identifies the requesting dashboard.
Request Event Log Data	"\t **Device_ID** \t LOG \t **Control_ID** \t **Dashboard_ID** \t **Time** \n"	The IoT device will then download all events after this **Time***. The time is UTC. The **Dashboard_ID** is a UUID that identifies the requesting dashboard.
Request Map Data	"\t **Device_ID** \t MAP \t **Control_ID** \t **Dashboard_ID** \t **Time** \n"	The IoT device will then download all map track after this **Time***. The time is UTC. The **Dashboard_ID** is a UUID that identifies the requesting dashboard.
Request Action Control data	"\t **Device_ID** \t ACTN \t **Action_Request** \t **Action_Payload** \n"	**Action_Request** specifies the request made to an **Action**. **Action_Payload** is and optional field that is only required and varies according the the **Action_Request**. See below for details.
Online	"\t **Device_ID** \t ONLINE \n"	Useful for improving efficiency of MQTT connections.
Offline	"\t **Device_ID** \t OFFLINE \n"	Used for LWT for MQTT

* If the **Time** is not a valid IOS 8601 based time format, but is a valid number (**n**), the dashboard is requesting the most recent **n** data points. This may occur when the dashboard is requesting the maximum number of data points it can display.

Send Message Examples:

"\tWHO\n"
=> send a device connection information request message to an IoT device

"\tFE:ED:CA:FE:BA:BE\tTEXT\tD1\t13\n"

=> send a message to a **Control_Type** "TEXT" with **Control_ID** "D1" and a value of "13"

"\t*FE:ED:CA:FE:BA:BE*\t*BTTN\tOnOff\n"
=> send a button press message to a **Control_Type** "BTTN" with **Control_ID** "OnOff"

"\t*FE:ED:CA:FE:BA:BE*\t*MENU\tm01\n"
=> send a request update menu message to a **Control_Type** "MENU" with **Control_ID** "tm01"

4.2. Messages From IoT device to Dash MQTT Broker

IoT devices connected to the Dash MQTT broker can take advantage of messages that are interpreted specifically by the MQTT broker and are NOT published to the Dash IoT Dashboard. These messages are described in the following table:

Action	Message	Notes
Notify Alarm	"\t *Device_ID* \t ALM \t *Alarm_ID* \t *Header* \t *Message* \n"	*Alarm_ID* is the identifier of the alarm. *Header* is the notification message heading and *Message* is the notification message body. Both *Header* and *Message* are ASCII text. Alarm messages must be sent on the **alarm** topic.
Announce Connection	"\t *Device_ID* \t WHO \t *Device_Type* \t *Device_Name* \n" or "\t *Device_ID* \t WHO \t *Device_Type* \t *Device_Name* \t *Config_Rev* \n"	*Config_Rev* is an optional whole number that informs the **Dash IoT Dashboard** what revision of the configuration (layout) is available on the IoT device. If *Config_Rev* is not included, the default value of 0 will be used. Announce connection messages must be sent on the **announce** topic once the Dash MQTT server has been connected too. The announce connection messages should not be confused with other WHO messages that are part of standard messaging.
Request Dash Server Time.	"\t *Device_ID* \t CLK \n"	Request the time from the Dash server. Request server time messages must be sent on the **announce** topic once the Dash MQTT server has been connected too.
Enable Data Store	"\t *Device_ID* \t STE \t *Control_Type* \t *Control_ID* \n"	Enable or disable data storage on the **dash** server. Enable Data Store messages must be sent on the **announce** topic once the Dash MQTT server has been connected too.

"\t*FE:ED:CA:FE:BA:BE*\tALM\tAL1\tOver Temperature\tExcessive booster chamber temperature\n"
=> receive a message for a notify alarm *Message_Type* "ALM" for *Control_ID* "AL1" and alarm notification header "Over Temperature" and message "Excessive booster chamber temperature".

4.3. Messages From Dash MQTT Broker to IoT device

Action	Message	Notes
Dash Server Time	"\t ***Device_ID*** \t CLK \t svr_clk \t ***time*** \n"	After a **Request Dash Server Time** message has been sent to the Dash server, the server will respond with the **Dash Server Time** message on the **Control** topic. The ***time*** is Unix time (UTC seconds since the 1st of January 1970).

4.4. Messages From IoT device to Dash IoT Dashboard

Action	Message	Notes
WHO reply with identification information	"\t **Device_ID** \t WHO \t **Device_Type** \t **Device_Name** \n" or "\t **Device_ID** \t WHO \t **Device_Type** \t **Device_Name** \t **Config_Rev** \n"	Send the **Device_ID**, **Device_Type** and **Device_Name** used to identify the IoT device. **Config_Rev** is an optional whole number that informs the **Dash IoT Dashboard** what revision of the configuration (layout) is available on the IoT device. If **Config_Rev** is not included, the default value of 0 will be used.
Acknowledge CONNECT message	"\t **Device_ID** \t CONNECT \n"	*Sent after receiving a CONNECT message. This message is required to complete the creation of a connection.
Updated Device Name	"\t **Device_ID** \t NAME \t **Device_Name** \n"	**Device_Name** successfully changed
Updated WiFi Settings	"\t **Device_ID** \t WIFI \n"	WiFi setup successfully changed
Updated TCP Settings	"\t **Device_ID** \t TCP \n"	**Port** successfully changed
Updated DashIO Settings	"\t **Device_ID** \t DASHIO \n"	Dash **User_Name** successfully changed
Will Reset Device	"\t **Device_ID** \t RST \n"	The IoT device will reset
Update Button State	"\t **Device_ID** \t BTTN \t **Control_ID** \t **State** \t **IconName** \t **Text** \n"	Changes the button icon or indicator color. **State** is an ASCII Boolean represented as "**ON**" or "**OFF**". **IconName** and **Text** are optional and can be used to set the icon name and text of the button. Subsequent messages that do NOT contain the **IconName** or **Text** will set the icon name and text of the button back to their default configuration values.
Update Slider Position	"\t **Device_ID** \t SLDR \t **Control_ID** \t **Data** \n"	**Data** is and ASCII Integer or Real Number.
Update Selector Index	"\t **Device_ID** \t SLCTR \t **Control_ID** \t **Selector_Data** \n"	**Selector_Data** is the index and group of text items used in the selector
Update Single Bar Graph	"\t **Device_ID** \t BAR \t **Control_ID** \t **Data** \n"	**Data** is and ASCII Integer or Real Number.
Update Double Bar Graph	"\t **Device_ID** \t BAR \t **Control_ID** \t $Data_1$ \t $Data_2$ \n"	$Data_1$ and $Data_2$ are and ASCII Integer or Real Numbers. $Data_2$ is optional.

Action	Message	Notes
Update Text Display	"\t **Device_ID** \t TEXT \t **Control_ID** \t **Data** \n"	**Data** is any ASCII text string
Update Knob	"\t **Device_ID** \t KNOB \t **Control_ID** \t **Data** \n"	**Data** is any ASCII Integer or Real Number.
Update Knob Dial	"\t **Device_ID** \t KBDL \t **Control_ID** \t **Data** \n"	**Data** is any ASCII Integer or Real Number.
Update Dial Display	"\t **Device_ID** \t DIAL \t **Control_ID** \t **Data** \n"	**Data** is any ASCII Integer or Real Number.
Update Direction Display	"\t **Device_ID** \t DIR \t **Control_ID** \t $Data_1$ \t $Data_2$ \n"	$Data_1$ is any ASCII Integer or Real Number representing Decimal Degrees. Secondary data $Data_2$ is any ASCII text string. $Data_2$ is optional.
Send Current Colour	"\t **Device_ID** \t CLR \t **Control_ID** \t **Color** \n"	**Color** as a hexadecimal string (e.g. "#23A89F")
Send URL for Audio Visual Display	"\t **Device_ID** \t AVD \t **Control_ID** \t **URL** \n"	**URL** of the audio visual component to be displayed. Must be secure.
Update Line for Chart Display	"\t **Device_ID** \t CHRT \t **Control_ID** \t **Line_Data** \n"	**Line_Data** is the data for a single chart line as defined below
Update Line for Time Graph Display	"\t **Device_ID** \t TGRPH \t **Control_ID** \t **T_Line_Data** \n"	**T_Line_Data** is the data for a single graph line as defined below
Send Event	"\t **Device_ID** \t LOG \t **Control_ID** \t **Event_Data** \n"	**Event_Data** is the data for a single event line as defined below
Update Map Display	"\t **Device_ID** \t MAP \t **Control_ID** \t **Map_Data** \n"	**Map_Data** is the position information.
Configuration	"\t **Device_ID** \t CFG \t **Dashboard_ID** \t **Config_ID** \t **Config_Data** \n"	**Config_ID** defines the type of configuration data as defined later in this document. **Config_Data** is the configuration data, the format of which is dependent upon the **Config_ID**.
Send Action Control Data	"\t **Device_ID** \t ACTN \t **Action_Request** \t **Action_Payload** \n"	This message is always a response to a Request Action Control data message that was sent to the IoT device. **Action_Request** is the has the same value that was used in the Request Action Control data message. **Action_Payload** is and optional field that is only required and varies according the the **Action_Request**. See below for details.

Receive Message Examples:

"\tFE:ED:CA:FE:BA:BE\tBTTN\tONOFF\tON\n"

=> receive a message for a **Control_Type** "BTTN" with **Control_ID** *"ONOFF"* and a value of "ON"

"\\t*FE:ED:CA:FE:BA:BE*\\tBTTN\\tONOFF\\tOFF\\tTRASH\\tStop\\n"
=> receive a message for a **Control_Type** "BTTN" with **Control_ID** *"ONOFF"* and a value of "**OFF**". The icon is changed to the **Trash** icon and the text "**Stop**" is displayed.

"\\t*FE:ED:CA:FE:BA:BE*\\tBAR\\tVolume\\t9\\t72\\n"
=> receive a message for **Control_Type** "BAR" with **Control_ID** "Volume" and a value of "9" for the left hand bar and "72" for the right hand bar

4.5. Selector_Data for Selector Control

Selector controls can be dynamically setup with the **Selector_Data** as follows:

Selector_Data = "**Index** \\t **Item**$_1$ \\t **Item**$_2$ \\t …**Item**$_N$"

Where **Index** is the index of **Item** that the selector is set too. The selection items of the selector are set to the **Item**s in the message. For N > 0, N is the number of selector text items in the selector. If N = 0, only the selector index if sent in the message and the selector items are not changed. Therefore, you only need to send the items when sending the initial conditions after a **WHO** message is received or if you want to update the selector).

4.6. Line_Data for Chart Control

Chart controls can show many chart lines simultaneously. Each chart line is created with a message as follows:

Line_Data = "*Line_ID* \t *Line_Name* \t *Line_Type* \t *Color* \t *Y_Axis* \t Y_1 \t Y_2 \t …Y_N"

where N > 0 is the number of points on the chart line

Line_Data Fields	Description
Line_ID	ASCII String containing a unique identifier for the Chart Line of the Chart display with *Control_ID*. Dash IoT Dashboard will add a new chart line for each new *Line_ID*.
Line_Name	Arbitrary name of the chart line, to be shown to the user on the Chart display on the Dash IoT Dashboard.
Line_Type	Shows how the Chart Line should be presented and may be one of the following: • "**LINE**" Draws a line between each point Y_n • "**BAR**" Draws a bar up to each point Y_n • "**SEGBAR**" Draws a bar that is broken into segments up to each point Y_n • "**PEAKBAR**" Draws a single bar segment at each point Y_n • "**DEL**" Delete the line from the chart If the *Line_Type* field does not contain a value or has an incorrect value, it will default to "**LINE**".
Color	May be either the color name or an index of the color to be used from within the colors defined within **Dash IoT Dashboard**. If the **Color** field does not contain a value or has an unknown value, it will default to the color black.
Y_Axis	Select the left of right Y axis to scale the: • "**LEFT**" or "**L**" to use the left side Y axis • "**RIGHT**" or "**R**" to use the right side Y axis
Y_n	The *Y* axis value of each point in the chart (where n = 1 to N and N is the number of chart points). The *Y* values are the Y axis chart values and they may be integers or real numbers. The chart values are evenly spaced across the X axis.

Receive Chart Line Message Example:

"**\t*FE:ED:CA:FE:BA:BE*\tCHRT\tOutsideTemp\tL1\tSEGBAR\t8\t23.4\t24.6\t27\t27.5\t26.6\t26.2\t26\t25.8\n**"
=> receive a message for *Control_Type* "CHRT" with *Control_ID* "OutsideTemp", *Line_ID* "L1", *Line_Type* "SEGBAR", *Color_Index* "9" and chart Y axis line values of "23.4", "24.6","27", "27.5" "26.6", "26.2", "26", "25.8"

4.7. Line_Data for Time Graph Control

Time Graph controls can show many graph lines simultaneously. There are two formats available for the Time Graph data and they are identified by the Dash IoT Dashboard by the number of fields in the data.

4.6.1 Long Format Time Graph Data

The long format Time Graph line data includes both meta data for the line (line_name, color and line_type.) and optionally line data points.

Line_Data =
"***Dashboard_ID*** \t ***Line_ID*** \t ***Line_Name*** \t ***Line_Type*** \t ***Color*** \t ***Y_Axis*** \t T_1,Y_1 \t T_2,Y_2 \t …T_N,Y_N"

where $N \cong 0$ is the number of points on the graph line and a comma (",") is used to seperate the time and the Y value.

Line_Data Fields	Description
Dashboard_ID	The ***Dashboard_ID*** is a UUID that identifies the dashboard that requested the data and can be obtained from the requesting message.
Line_ID	ASCII String containing a unique identifier for the Graph Line of the Graph display with ***Control_ID***. Dash IoT Dashboard will add a new graph line for each new ***Line_ID***.
Line_Name	Arbitrary name of the graph line, to be shown to the user on the Graph display on the Dash IoT Dashboard.
Line_Type	Shows how the Graph Line should be presented and may be one of the following: • "**LINE**" Draws a line between each point T_n,Y_n • "**BAR**" Draws a bar, starting at time T_n and ending at time T_{n+1}, filled to the value Y_n • "**BOOL**", Draws a shaded area (Alpha = 0.5) on the graph. When Y_n is "**true**", "**t**", "**high**" or "**h**", the graph is shaded from time T_n until the time of a data point where Y = "**false**", "**f**", "**low**" or "**l**" is received. These values are NOT case sensitive. • "**DEL**" Delete the line from the graph If the ***Line_Type*** field does not contain a value or has an incorrect value, it will default to "**LINE**".
Color	May be either the color name or an index of the color to be used from within the colors defined within **Dash IoT Dashboard**. If the **Color** field does not contain a value or has an unknown value, it will default to the color black.
Y_Axis	Select the left of right Y axis to scale the: • "**LEFT**" or "**L**" to use the left side Y axis • "**RIGHT**" or "**R**" to use the right side Y axis

Line_Data Fields	Description
T_n	The UTC time T of the data point Y. The following IOS 8601 based formats are supported: 1) Local time: "yyyy-MM-ddTHH:mm:ss" 2) UTC: "yyyy-MM-ddTHH:mm:ssZ" (no need to manage timezone or daylight savings) 3) Timezones: "yyyy-MM-ddTHH:mm:ss±hh:mm" (or ±hhmm or ±hh) 4) T_1 is either 1), 2) or 3) above and T_2 to T_n are the number of minutes since T_1. The number of minutes may be an integer or float, where the value after the decimal point is the decimal fraction of a minute.
Y_n	The Y axis value of the point in the graph corresponding to the time T. (where n = 1 to N and N is the number of graph points). The Y values are the Y axis graph values. They may be integers, real numbers, or text for when the Line_Type is "**BOOL**". If the Line_Type is "**LINE**", a break in the line may be added by setting the Y value of a point to "**b**" or "**break**". The break should normally be used as the first data point in the line.

4.6.2 Short Format Time Graph Data

Line_Data = "**Line_ID** \t **T**,**Y**"

where a comma (",") is used to seperate the time and the Y value.

or

Line_Data = "**Line_ID** \t **Y**"

Line_Data Fields	Description
Line_ID	ASCII String containing a unique identifier for the Graph Line of the Graph display with **Control_ID**. Dash IoT Dashboard will add a new graph line for each new **Line_ID**.
T	The UTC time T of the data point Y. The following IOS 8601 based formats are supported: 1) Local time: "yyyy-MM-ddTHH:mm:ss" 2) UTC: "yyyy-MM-ddTHH:mm:ssZ" (no need to manage timezone or daylight savings) 3) Timezones: "yyyy-MM-ddTHH:mm:ss±hh:mm" (or ±hhmm or ±hh) If the time T is not included in the message, the receiver of the message will insert the time that the message is received.
Y	The Y axis value of the point in the graph corresponding to the time T. The Y values are the Y axis graph values. They may be integers, real numbers, or text for when the Line_Type is "**BOOL**".

When using the short format line date, the line meta data will default to line_color = "***black***" and line_type = **LINE**. However, a long format line data message (with or without data points) may be

sent at any time to set the meta data for the line. An ideal time to do this is with the STATUS (initial condition) messages.

4.8. Event_Data for Event Log Control

Event Log controls can list many events from an IoT device in a single list. There are two formats available for the Event Log data and they are identified by the Dash IoT Dashboard by the type of data in the fields.

4.7.1 Long Format Event Log Data

***Event_Data* = "*Dashboard_ID* \t *Event*$_1$ \t *Event*$_2$ \t …*Event*$_N$"**

where $N \geq 0$ is the number of events in the message.

The ***Dashboard_ID*** is a UUID that identifies the dashboard that requested the data and can be obtained from the requesting message.

The ***Event*** is a field describing the event information using a simple JSON object string where each event element is a key : value pair:

```
{
        "element₁" :  value₁,
        "element₂" :  value₂,
                …
        "elementₙ" :  valueₙ,
}
```

Element	Value Data Type	Purpose
"time"	string	The UTC time T of the data point Y. The following IOS 8601 based formats are supported: 1) Local time: "yyyy-MM-ddTHH:mm:ss" 2) UTC: "yyyy-MM-ddTHH:mm:ssZ" (no need to manage timezone or daylight savings) 3) Timezones: "yyyy-MM-ddTHH:mm:ss±hh:mm" (or ±hhmm or ±hh)
"color"	string	May be either the color name or an index of the color to be used from within the colors defined within **Dash IoT Dashboard**.
"lines"	[string]	Array containing rows of text to be displayed on the Event Log.

4.7.2 Short Format Event Log Data

Each event is created with a message as follows:

***Event_Data* = "*Time* \t *Color* \t *Text*$_1$ \t *Text*$_2$ \t …*Text*$_N$"**

where $N \geq 0$ is the number of rows of text to be displayed on the event log.

Line_Data Fields	Description
Time	The UTC time *T* of the data point *Y*. The following IOS 8601 based formats are supported: 1) Local time: "yyyy-MM-ddTHH:mm:ss" 2) UTC: "yyyy-MM-ddTHH:mm:ssZ" (no need to manage timezone or daylight savings) 3) Timezones: "yyyy-MM-ddTHH:mm:ss±hh:mm" (or ±hhmm or ±hh) If the time *T* is an empty string (""), the receiver of the message will insert the time that the message is received.
Color	May be either the color name or an index of the color to be used from within the colors defined within **Dash IoT Dashboard**. If the **Color** field does not contain a value or has an unknown value, it will default to the color black.
Text$_n$	Row of text to be displayed on the Event Log.

4.9. Map_Data for Map Control

The **Map_Data** can be used to show either a **point** location or one or more **tracks** on the **Map** control on the Dash IoT Dashboard app. The Map_Data contains the field **Track_ID**, which is used to identify individual tracks from an IoT device. Subsequent messages with the same **Track_ID** are accumulated into a single track. However, if the **Track_ID** field contains empty text (e.g. ""), then only the most recent point will be displayed on the Map control instead of a track.

The map data is displayed on the Dash IoT Dashboard when the pin on the map of the Dash IoT Dashboard is tapped.

There are two formats available for the map data.

4.8.1 Long Format Map Track Data

The long format Map track data includes both meta data for the track (track_name and color.) and optionally track waypoints.

$$\textit{Map\_Data} = \textit{"Dashboard\_ID} \text{ \textbackslash t } \textit{Track\_ID} \text{ \textbackslash t } \textit{Text} \text{ \textbackslash t } \textit{Color} \text{ \textbackslash t } \textit{WP}_1 \text{ \textbackslash t } \textit{WP}_2 \text{ \textbackslash t } \ldots \textit{WP}_N\text{"}$$

where $N \geq 0$ is the number of points for the track included in this message.

The **Dashboard_ID** is a UUID that identifies the dashboard that requested the data and can be obtained from the requesting message.

The **Text** field is displayed with the pin on the map of the Dash IoT Dashboard.

The **Color** field sets the color of the track.

The **WP** is a field describing the GPS waypoint information using a simple JSON object string where each GPS element is a key : value pair:

```
{
        "element₁" :  value₁,
        "element₂" :  value₂,
                …
        "elementₙ" :  valueₙ,
}
```

Element	Value Data Type	Purpose
"time"	string	The UTC time T of the data point Y. The following IOS 8601 based formats are supported: 1) Local time: "yyyy-MM-ddTHH:mm:ss" 2) UTC: "yyyy-MM-ddTHH:mm:ssZ" (no need to manage timezone or daylight savings) 3) Timezones: "yyyy-MM-ddTHH:mm:ss±hh:mm" (or ±hhmm or ±hh)
"latitude"	string	Latitude in decimal degrees
"longitude"	string	Longitude in decimal degrees
"avgeSpeed"	string	Average speed since the last message in meters/second

Element	Value Data Type	Purpose
"peakSpeed"	string	Maximum speed since the last message in meters/second
"course"	string	Course direction in decimal degrees. A negative value indicates an unknown heading
"altitude"	string	Altitude in meters
"distance"	string	Accumulated distance since the last message in meters

The latitude and longitude elements are required, and all other elements are optional.

All data types are string to match common GPS data.

4.8.2. Short Format Map Data

The short format *Map_Data* has two fields:

Map_Data = "*Track_ID* \t *Latitude* , *Longitude*"

Where the *Latitude* and *Longitude* are in decimal degrees and are separated by a comma (",").

4.10. Action_Request and Action_Payload for Action Controls

The *Action_Request* and **Action_Payload** are used for the **Action Station** dash functionality and is beyond the scope of this document. Please contact DashIO Connect Limited if you are interested in using this functionality: info@dashio.io

4.11. Config_ID and Config_Data for Configuration Messages

Configuration messages provide information about each control in the IoT device to the Dash IoT Dashboard. This allows the user to easily setup the Dash IoT Dashboard without having to know about each control in the IoT device.

The **Config_ID** defines the type of configurations message and the contents of the **Config_Data** as follows:

Config_ID	Config_Data
"NA"	Configuration data is not available for this device and the **Config_Data** is ignored by the Dash IoT Dashboard. The Config_Data does not need to be included in the message.
"C64"	The configuration data (**Config_Data**) contains **JSON** data for the complete configuration of the IoT device, which has been **zlib** compressed and **Base64** encoded. The C64 configuration data can be generated by the **Dash IoT Dashboard** and copied directly into the IoT device code as text. This is by far the easiest way to perform configuration of the Dash IoT Dashboard from the IoT device.
Control_Type or **Connection_Type**	The **Config_Data** is JSON formatted text describing each parameter and its associated value for the particular **Control_Type** or Additional layout **Control_Types** • "**DVCE**" Device properties • "**DVVW**" Device View (config messages only) or **Connection_Type** for communication connections configuration messages • "**MQTT**" MQTT Connection • "**BLE**" BLE Connection • "**TCP**" TCP Connection

4.9.1. Controls JSON Configuration

The control JSON configuration data provides default values for most parameters of each control. This enables the control to be easily setup by the Dash IoT Dashboard user.

The **Config_Data** is a single field describing the configuration information with a simple JSON object string where each configuration parameter is a key : value pair:

{

        "element$_1$" : value$_1$,
        "element$_2$" : value$_2$,
            …
        "element$_n$" : value$_n$,

}

Note that colors are a string that may be one of the following:
- RGB color in hexadecimal e.g. "#12F84C"
- Color name from colors in Dash IoT Dashboard e.g. "blue"
- Index to the color e.g. "4"

Configuration Message Order
    Once the IoT device has received a configuration data request message, the IoT device must immediately reply with the configuration response messages. Always send the **device** configuration message first and the **device views** configuration messages last. The **device** message contains the number of **device views**, therefore, the **Dash IoT Dashboard** will know that all configuration message have been received once the correct number of **device views** configuration messages have been received.

Device

Element	Value Data Type	Purpose
"numDeviceViews"	number	Total number of Device Views for the IoT device
"deviceSetup"	string	Text containing the comma separated device setup options e.g. "**ble, tcp, mqtt, name, wifi, tcp_port, dash**" where: "**ble**" = BLE connection available on this device "**tcp**" = TCP connection available on this device "**mqtt**" = MQTT connection available on this device (currently not available) "**name**" = can change change the device name "**wifi**" = can change the WiFi Country Code, SSID and password. "**tcp_port**" = can change the TCP port. "**dash**" or "**dashio**" = can change the **dash** MQTT server username and password. The username is set to the currently logged in username of the mobile device. "**rst**" = the device will accept a device reset request.
"cfgRev"	number	Whole number for the configuration (layout) revision

Device View

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the device view

Element	Value Data Type	Purpose
"title"	string	Text to be displayed in the title box of the control
"gridColumns"	number	Number of grid columns in the DeviceView (used for Snap-To-Grid)
"gridRows"	number	Number of grid rows in the DeviceView (used for Snap-To-Grid)
"iconName"	string	Name of the icon from the icons in Dash IoT Dashboard. Displayed on the device view selector.
"color"	number	Color of device view background
"ctrlMaxFontSize"	number	Maximum size of the the font displayed on the controls on the device view. The font size may be reduced for each control to enable the text to fit on the control
"ctrlBorderOn"	**true** or **false**	Enable the border for the controls on the device view
"ctrlBorderColor"	number	Color of the border and title of controls on the device view
"ctrlColor"	number	Color of miscellaneous graphics and icons of the controls on the device view
"ctrlBkgndColor"	string	Background color of the controls on the device view
"ctrlBkgndTransparency"	number	Transparency of the background color of the controls on the device view, (0 to 100)
"ctrlTitleFontSize"	number	Some of the font of the title of the controls on the device view
"ctrlTitleBoxColor"	string	Color of the title box of the controls on the device view
"ctrlTitleBoxTransparency"	number	Transparency of the title box color of the controls on the device view (0 to 100)
"ctrlTitlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Label

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Element	Value Data Type	Purpose
"style"	number	Style of the label. May be "basic" for simple text, "border" to add a border to the basic style, or "group" for a label with a boarder and title that can bet used to surround a group of controls
"color"	string	Color of the displayed text.

Button

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"buttonEnabled"	**true** or **false**	Enable the button to respond to touch and send messages
"style"	string	Style of the Button control. May be "basic" for simple icon and text that is set to the on or off color or "highlight" to add a highlighted area behind the button which is set to the on or off color.
"iconName"	string	Name of the icon from the icons in Dash IoT Dashboard. Displayed on the button
"text"	string	Text displayed on the button
"offColor"	string	Color of text or icon when the button is in the OFF state.
"onColor"	string	Color of text or icon when the button is in the ON state.

Knob with Dial Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"titlePosition"	string	Text to be displayed in the title box of the control
"ctrlTitlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"style"	string	Presentation style of the knob: "normal" or "pan"
"min"	string	Minimum value of the dial behind the knob
"max"	string	Maximum value of the dial behind the knob
"redValue"	number	Between min and max. When the dial value is above this number it will be colored red
"showMinMax"	**true** *or* **false**	Send message only when the the knob is released
"dialFollowsKnob"	**true** *or* **false**	The dial behind the knob shows the same value as the knob
"dialColor"	string	Color of the dial

Dial Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"titlePosition"	string	Text to be displayed in the title box of the control
"ctrlTitlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"min"	string	Minimum value of the dial behind the knob
"max"	string	Maximum value of the dial behind the knob
"redValue"	number	Between min and max. When the dial value is above this number it will be colored red
"dialFillColor"	string	Dial fill color
"pointerColor"	string	Color of the pointer of the dial
"style"	string	Presentation style of the dial: "bar", "pie" or "pieinv"
"numberPosition"	string	Position of the display of the value: "off", "left", "right" or "centre"
"showMinMax"	**true** *or* **false**	Send message only when the the knob is released
"precision"	number	Numeric precision of the data - number of characters excluding the decimal point. 0 = "Off", 1... 6

Direction Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Element	Value Data Type	Purpose
"style"	string	Presentation style of the dial: "NSEW", "DEG" or "DEGPS"
"pointerColor"	string	Color of the pointer
"calAngle"	number	Correction offset in degrees for the direction pointer
"units"	string	To be displayed below the text of the secondary data
"precision"	number	Numeric precision of the secondary data (if it is numerical) - number of characters excluding the decimal point. 0 = "Off", 1... 6

Text Box

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"maxFontSize"	number	The maximum size of the text font. The font size will never be larger than this value
"format"	string	Allows the displayed text to be formatted to a different style. • "none", "num" format as a number and apply the precision parameter and units • "datetime" format ISO8601 text as a date and time • "dtlong" format ISO8601 text as a date and time and include the timezone • "intvl" format ISO8601 text as hours, minutes and seconds • "Log" displays the unformatted text in multi-line format to show sequential messages.
"textAlign"	string	"center", "left" or "right"
"units"	string	To be displayed after the text in the Text Box
"precision"	number	Numeric precision of the data (if it is numerical) - number of characters excluding the decimal point. 0 = "Off", 1... 6
"kbdType"	string	"none", "all" for all characters, "num" for numeric decimal, "int" for numeric integer, "datetime" for date and time selection, "date" for date selection, "time" for time selection, "intvl" for time interval selection, or "hex" for hexidecimal
"closeKbdOnSend"	**true** *or* **false**	Hide the keyboard once the message has been sent

Slider with Bar Graph

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"min"	string	Minimum value of the dial behind the knob
"max"	string	Maximum value of the dial behind the knob
"redValue"	number	Between min and max. When the dial value is above this number it will be colored red
"showMinMax"	**true** *or* **false**	Send message only when the the knob is released
"sliderEnabled"	**true** *or* **false**	Show the slider
"knobColor"	string	Color of the knob
"sendOnlyOnRelease"	**true** *or* **false**	Send message only when the the knob is released
"barFollowsSlider"	**true** *or* **false**	The bar behind the slider shows the same value as the knob
"barColor"	string	Color of the bar
"barStyle"	string	"solid" or "seg" for a bar divided into segments

Chart Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"xAxisLabel"	string	Title of the horizontal axis
"xAxisMin"	number	Value of the horizontal axis left hand side
"xAxisMax"	number	Value of the horizontal axis right hand sid
"xAxisNumBars"	number	Positive integer. Number of horizontal grid lines, including the left and right axis
"xAxisLabelsStyle"	string	"on" for on the grid lines, or "between" for between the grid lines
"yAxisLabel"	string	Title of the vertical axis (left side)
"yAxisMin"	number	Value of the vertical axis bottom (left side)
"yAxisMax"	number	Value of the vertical axis top (left side)
"yAxisNumBars"	number	Positive integer. Number of vertical grid lines, including the top and bottom axes
"yAxisLabelRt"	string	Title of the second vertical axis (right side)
"yAxisMinRt"	number	Value of the second vertical axis bottom (right side)
"yAxisMaxRt"	number	Value of the second vertical axis top (right side)

Time Graph Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"yAxisLabel"	string	Title of the vertical axis (left side)
"yAxisMin"	number	Value of the vertical axis bottom (left side)
"yAxisMax"	number	Value of the vertical axis top (left side)
"yAxisNumBars"	number	Positive integer. Number of vertical grid lines, including the top and bottom axes
"yAxisLabelRt"	string	Title of the second vertical axis (right side)
"yAxisMinRt"	number	Value of the second vertical axis bottom (right side)
"yAxisMaxRt"	number	Value of the second vertical axis top (right side)

Selector

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Menu

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"style"	string	Style of the Menu control. May be "basic" for simple icon and text or "highlight" to add an encircling border.
"iconName"	string	Name of the icon from the icons in Dash IoT Dashboard. Displayed on the menu
"text"	string	Text displayed on the menu

Button Group

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"style"	string	Style of the Button Group control. May be "basic" for simple icon and text or "highlight" to add an encircling border.
"iconName"	string	Name of the icon from the icons in Dash IoT Dashboard. Displayed on the menu
"text"	string	Text displayed on the menu
"gridView"	**true** *or* **false**	Show as a grid of Button controls.

Event Log

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Map Display

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Color Picker

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"
"pickerStyle"	string	"wheel" for wheel or "spec" spectrum
"sendOnlyOnRelease"	**true** *or* **false**	Send message only when the the knob is released

Audio Visual

Element	Value Data Type	Purpose
"controlID"	string	Identifier of the control
"parentID"	string	Identifier of the parent of the control (device view or menu)
"xPositionRatio"	number	Position of the left side of the control as a ratio of the screen width (0.0 to 1.0)
"yPositionRatio"	number	Position of the top side of the control as a ratio of the screen height (0.0 to 1.0)
"widthRatio"	number	Width of the control as a ratio of the screen width (0.0 to 1.0)
"heightRatio"	number	Height of the control as a ratio of the screen height (0.0 to 1.0)
"title"	string	Text to be displayed in the title box of the control
"titlePosition"	string	Position of the title box of the controls on the device view. May be "top", "bottom", or "none"

Connection - BLE

Element	Value Data Type	Purpose
"serviceUUID"	string	Service UUID
"readUUID"	string	Characteristic UUID for read
"writeUUID"	string	Characteristic UUID for write

4.12. MQTT Communications Considerations

MQTT communications is bases on publishing and subscribing to *topics* on the MQTT broker. The topics use for communication between each IoT device and the MQTT broker are defined as follows:

Topic Name	Message	Full Topic required for the MQTT Broker
Control	*Send_Data* messages from the Dash IoT Dashboard to the IoT device to control the IoT device.	"*User_Name*/*deviceID*/control"
Data	*Receive_Data* messages from the IoT device to display on a control on the Dash IoT Dashboard	"*User_Name*/*deviceID*/data"
Alarm	**Alarm** message sent from the IoT device to the MQTT broker to initiate a push notification alarm to the device the Dash IoT Dashboard is installed on.	"*User_Name*/*deviceID*/alarm"
Announce	**Announce** message sent from the IoT device to the MQTT broker to allow the broker to include the IoT device in the broker's list of connections. This is optional, but is used by the message broker to manage shared connections.	"*User_Name*/*deviceID*/announce"

Where:
- The *User_Name* is the user name of the **Dash** account.
- The *deviceID* is a unique device identifier for the IoT device. The Mac address works well for the deviceID. Alternatively, a UUID may be used.
- The **Dash** *MQTT broker connection must use SSL.*

4.13.Device Discovery

The IoT device discovery process within the Dash IoT Dashboard is important to make it simple for users to discover and setup their IoT devices. The discovery process varies according to each connection type. However, regardless of the connection type, the final step in device discovery is when the Dash IoT Dashboard sends a WHO message and receives a valid who response.

4.11.1. BLE Discovery

BLE device discovery is achieved with the **Dash IoT Dashboard** scanning for **BLE** enabled IoT devices which must be setup as follows:

- The IoT device is setup as a BLE **peripheral**.
- The peripheral name (or local name) should be set to "**DashIO**" for the first 6 characters and appended with whatever text you wish.
- The **Service UUID** must be **advertised** and the peripheral must be **connectable**. No additional or secondary advertised data is required. Some BLE peripherals advertise the Service UUID by default, while others must be specifically setup to advertise the Service UUID.
- The advertised service must have **notify** and **write** characteristics. It may be a single characteristic containing both **notify** and **write** properties.

Secure (encrypted) BLE connections may be used. Please note that Bluetooth libraries can be resource intense, such as ESP32 BLE libraries.

4.11.2. TCP Discovery

TCP connections require an *IP address* and *port*. The **Dash IoT Dashboard** will always attempt to connect to IoT devices with TCP connections over the local network that the **Dash IoT Dashboard** is operating on. Therefore, if the IoT device is not on the same local network as the dashboard, they will not be able to connect.

Device discovery is achieved with **mDNS**. Using mDNS allows the **Dash IoT Dashboard** to discover IoT devices on any port on the local network having the **mDNS** service "**_DashIO._tcp.**"

4.11.3. MQTT Discovery

MQTT device discovery can ONLY occur when the **Dash Cloud Services** MQTT broker is used (i.e. dash.dashio.io on port 8883 using SSL).

Device discovery is an integral part of the **Dash Cloud Services** and is initiated by the IoT device sending a **Dash Protocol WHO** message to the **announce** MQTT topic. The WHO message contains the device identifier, device type and device name which are retrieved by the **Dash** MQTT broker and stored as an available device for the particular user. The **Dash IoT Dashboard** retrieves the available IoT devices from the **Dash Cloud Services** database.