

# DashIO Dashboard Protocol

Revision Date: 23 July 2021

## 1. Description

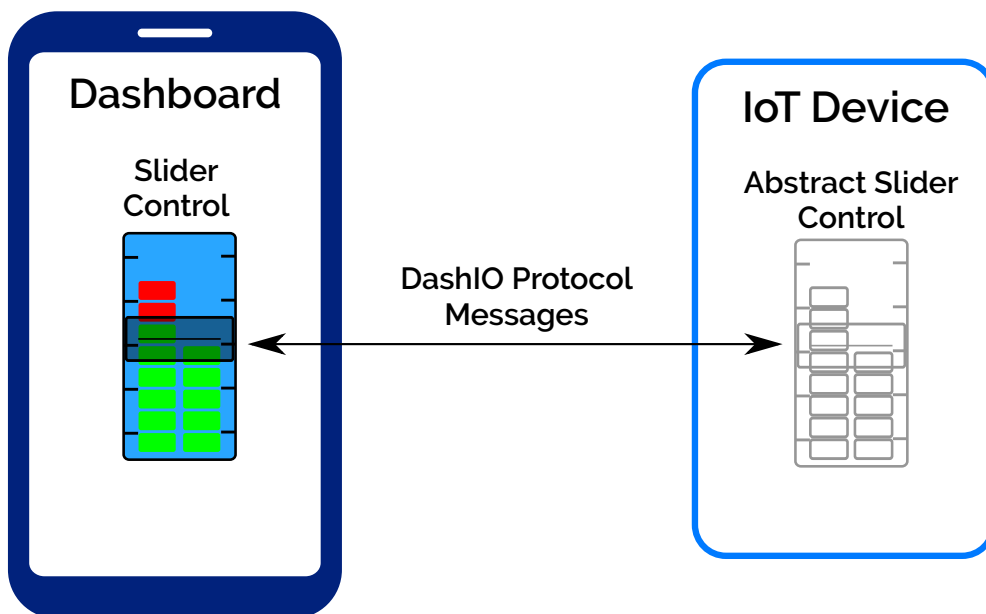
**DashIO Dashboard Protocol** is a lightweight data-exchange protocol for asynchronous networking between an **DashIO Dashboard** and **IoT devices**. The **DashIO Dashboard** may be deployed on a mobile device or computer to monitor or control one or more IoT devices. An IoT device may be any networked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

The **DashIO Dashboard 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 **DashIO 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.

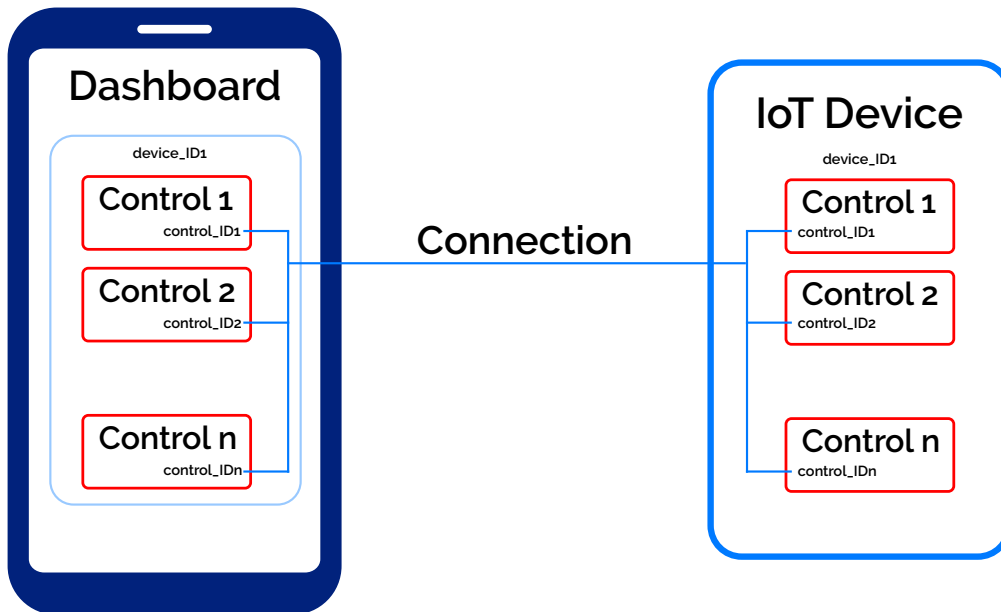
Messages are used to send and receive data between **controls** on the DashIO Dashboard and a corresponding **abstract controls** within the IoT device, as shown in the figure below:



System Messages are also used for purposes other than controls messaging, for example to setup connections and deliver configuration and alarm messages.

## 1.1. DashIO Dashboard

The **DashIO Dashboard** displays to the user the information from the **controls** in the IoT device. The controls in the IoT device are modelled within the DashIO Dashboard and information is transferred between each control in the IoT device and its counterpart in the DashIO 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 DashIO 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 **DashIO 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

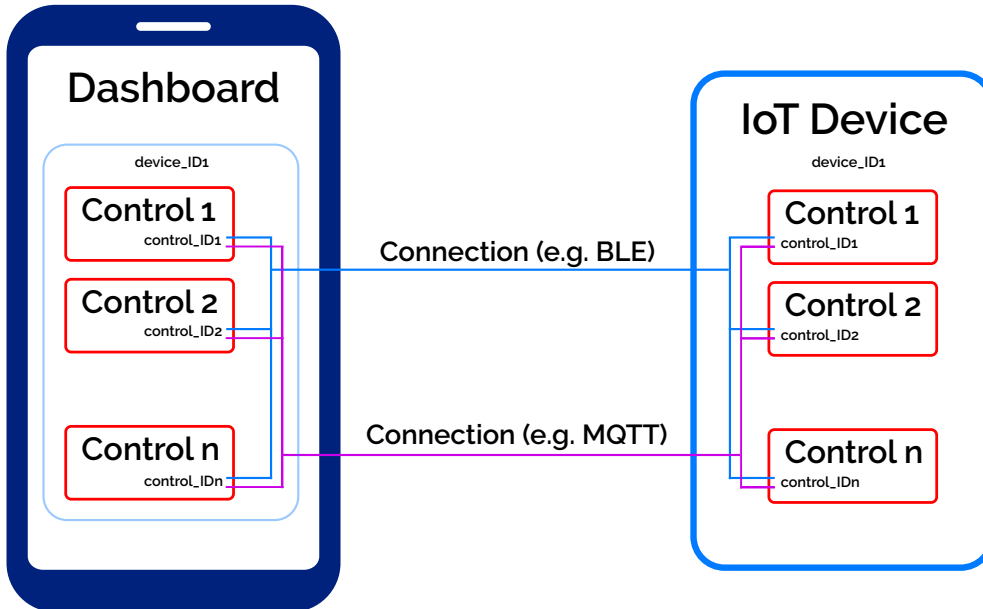
An IoT device must be uniquely identified to the DashIO Dashboard and this identifier is called the **device\_ID**. The **device\_ID** may be obtained from the **mac** address of the BLE, WiFi or Ethernet peripheral within the IoT device. Alternatively, a UUID may be used.

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 DashIO 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 DashIO Dashboard has established multiple connections to a single IoT device, it will only continue communicate over one connection with the following priority:

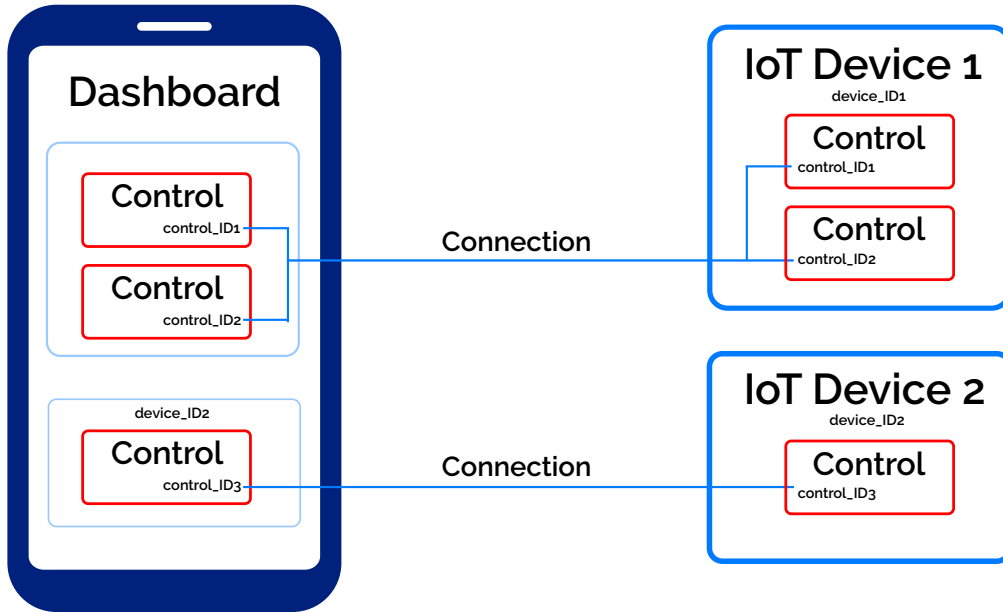
1. BLE
2. TCP
3. MQTT

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 DashIO Dashboard will switch communication to the next online connection from the priority schedule. Similarly, if a higher priority connection comes online, the DashIO 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 DashIO Dashboard can communicate with many IoT devices through multiple connections simultaneously. Communication between the DashIO 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
<b>Device_ID</b>	Unique device identifier (typically the mac address)
<b>Message_Type</b>	<p>ASCII String that uniquely defines the type of message. It may be either a system <b>System_Message_Type</b> or a <b>Control_Type</b>.</p> <p>A <b>System_Message_Type</b> message is used to send networking and configuration information between the DashIO Dashboard and the IoT device.</p> <p>A <b>Control_Type</b> message is used to send information about a specific control between the DashIO Dashboard and the IoT device.</p>
<b>Identifier</b>	<p>The <b>Identifier</b> field is an ASCII String identifier for the specific <b>Message_Type</b>.</p> <p>If the <b>Message_Type</b> is a <b>Control_Type</b>, the <b>Identifier</b> is the control identifier (<b>Control_ID</b>)</p> <p>If the <b>Message_Type</b> is a <b>System_Message_Type</b>, the <b>Identifier</b> field is NOT required and the message becomes:  <b>Message</b> = “\t <b>Message_Type</b> \t <b>Data</b> \n”</p>
<b>Data</b>	The <b>Data</b> 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 <b>Data</b> field). The number of sub-fields varies according to the <b>Message_Type</b> and <b>Identifier</b> .
\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
<b>System_Message_Type</b>	<p>“WHO” Announce <b>device_ID</b>, <b>device_type</b> and <b>device_name</b>, message</p> <p>“CONNECT” Connection validation</p> <p>“STATUS” Request current value for all controls</p> <p>“CFG” Configuration data</p> <p>“NAME” Device Name</p> <p>“WIFI” WiFi Country Code, SSID and Password</p> <p>“TCP” TCP port</p> <p>“DASHIO” DashIO server MQTT Username and Password</p> <p>“MQTT” MQTT URL, port, SSL (enabled), Username, and Password</p> <p>“MSSG” Popup message on the dashboard</p>
<p><b>System_Message_Type</b></p> <p>Specific to MQTT connections</p>	<p>“ONLINE” Online message for MQTT (for MQTT <b>Data</b> topic)</p> <p>“OFFLINE” Offline message Last Will and Testament (for MQTT <b>Data</b> topic)</p> <p>“WHO” Announce <b>device_ID</b>, <b>device_name</b> (for MQTT <b>Announce</b> topic)</p> <p>“ALM” Alarm message (push notification for MQTT <b>Alarm</b> topic)</p>
<p><b>Control_Type</b></p> <p>For controls displayed on the DashIO Dashboard</p>	<p>“BTTN” Button control</p> <p>“TEXT” Text display and Input</p> <p>“SLDR” Slider control</p> <p>“BAR” Bar graph display under slider</p> <p>“KNOB” Knob control</p> <p>“KBDL” Knob display surrounded by dial</p> <p>“DIAL” Dial display</p> <p>“DIR” Direction display</p> <p>“GRPH” Graph display</p> <p>“TGPH” Graph display with time on the X axis</p> <p>“SLCTR” Selector control</p> <p>“MENU” Menu control</p> <p>“BTGP” Button Group control</p> <p>“LOG” Event Log display</p> <p>“LBL” Label Box display</p>

## 2.2. Data

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

**Data** = “*Data*<sub>1</sub> \t *Data*<sub>2</sub> \t ...*Data*<sub>N</sub>” (where N >= 0)

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

## 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 DashIO 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 DashIO Dashboard.	STATUS	N/A	N/A
Configuration	Configuration request and response messages. The response message contains the details of all <b>controls</b> and <b>alarms</b> available on an IoT device may be sent to the DashIO Dashboard. This helps simplify setup of the Dashboard. Configuration messages are used to send this information to the DashIO Dashboard.	CFG	Configuration request message contains the display dimensions in mm.	Basic information for all available Controls.
Device Name	Update the Device Name	NAME	New name of the device	Name of the device
WiFi Setup	Update the WiFi SSID and password	WIFI	New WiFi Country Code, SSID and password	N/A
TCP Setup	Update the TCP port	TCP	New TCP port	N/A
DashIO Setup	Update the DashIO 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 DashIO Dashboard. The DashIO 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 <b>online</b> and able to communicate. Used for MQTT connections to speed up reconnection.	ONLINE	N/A	N/A
Offline	Device is <b>offline</b> and unable to communicate. Used for MQTT connections when setup a the Last Will and Testament (LWT) message.	OFFLINE	N/A	N/A
Popup Message	Popup message on the dashboard	MSSG	N/A	Title, header and caption of the popup message.

\* Please note that “**WHO**” and “**CONNECT**” messages and their reply 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 DashIO Dashboard. All other messages are optional.




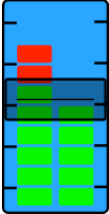


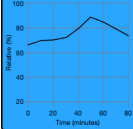
## 2.4. Controls Messaging

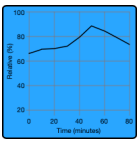


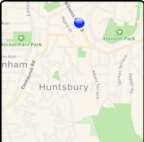



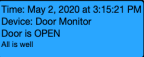
Controls messages contain fields having the following meaning:

- **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 DashIO Dashboard's perspective, defined as either **Send\_Data** or **Receive\_Data**.

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

DashIO 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		<b>Button</b> 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		<b>Slider</b> 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		<b>Knob</b> 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 <b>Text</b> messages	TEXT	Text message	Message to display
Graph Display		<b>Graph</b> that can show zero or more graph lines or bars for one or more IoT devices	GRPH	N/A	Graph line information (Y axis values)

Dashboard Control	Description	Controls Message Fields			
		Message_Type	Data		
		Control_Type	Send_Data	Receive_Data	
Time Graph Display		<b>Time Graph</b> 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		<b>Dial</b> or gauge display to show a numeric value	DIAL	N/A	Dial position value
Direction Display		Compass style <b>Direction</b> rose	DIR	N/A	Compass direction (Decimal Degrees) and secondary data
Map Display		<b>Map</b> for displaying GPS data for one or more IoT devices	MAP	N/A	Position data
Selector		<b>Selector</b> 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

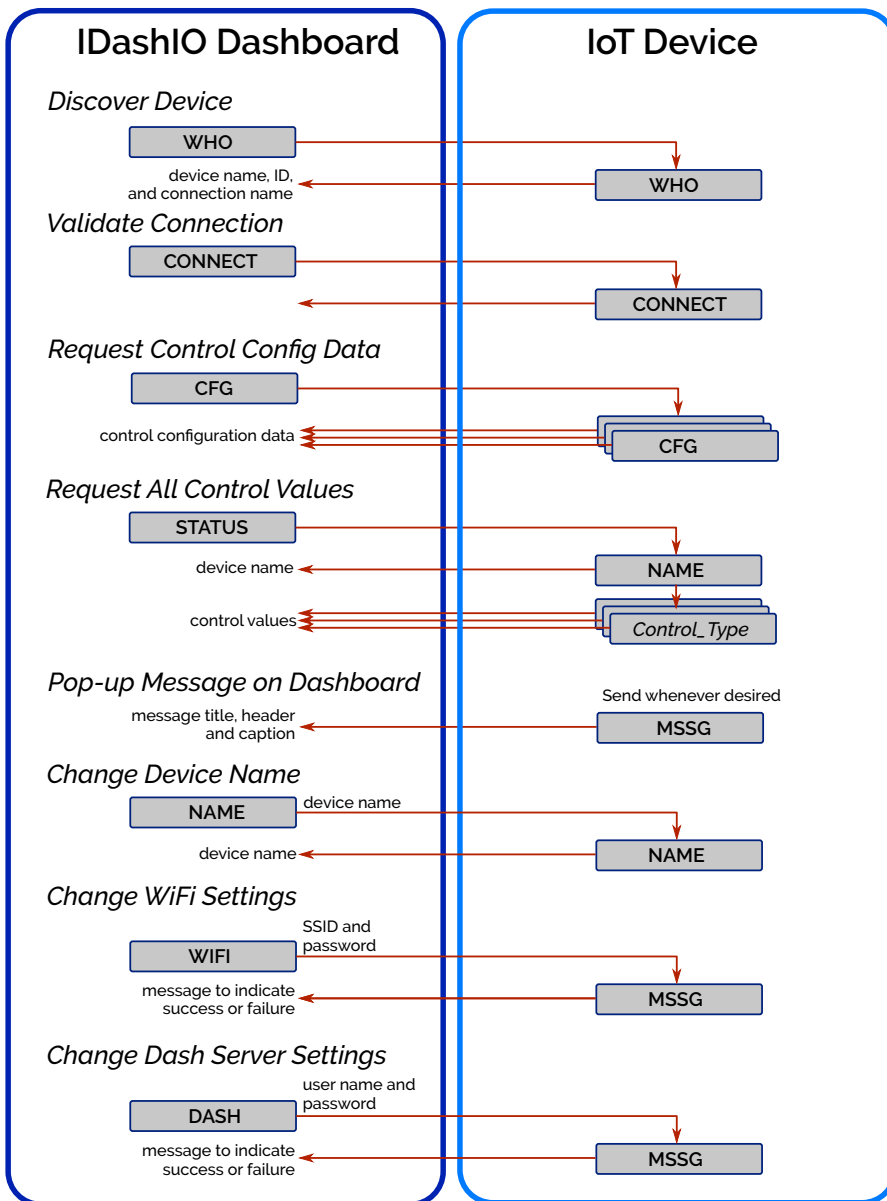
For details of each control, please refer to the document titled “DashIO Controls”.

### 3. Message Request-Reply Overview

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

#### 3.1. System Messaging

All but one system message group is generated as a request message from the DashIO Dashboard. The Popup Message (MSSG) being the only exception.



## 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 DashIO Dashboard:

### ONLINE

The DashIO 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

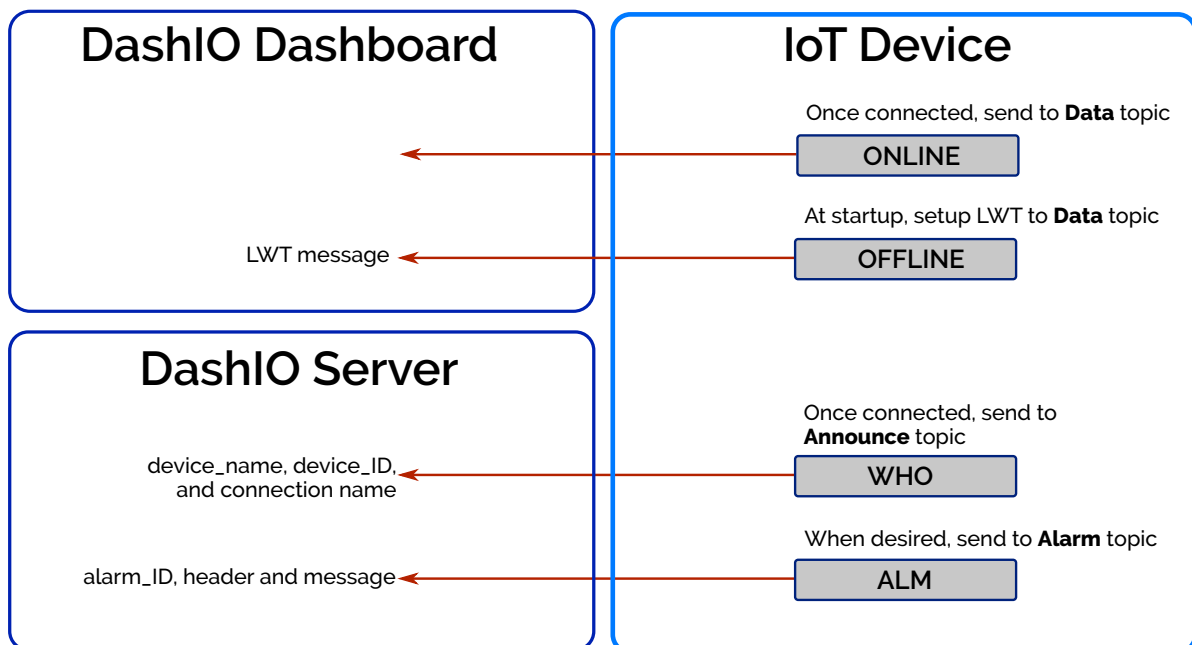
The Last Will and Testament (LWT) should be setup for the MQTT **Data** topic with the OFFLINE message.

### WHO

Once the IoT device is connected to the **DashIO** server it should send the WHO message to the MQTT **Announce** topic. This enables the DashIO server to inform the DashIO 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 DashIO server whenever the IoT device needs to send an alarm. The DashIO server will forward this message to the Apple or Google push notification service, which will then send the notifications to the DashIO Dashboards that are connected to the IoT device.

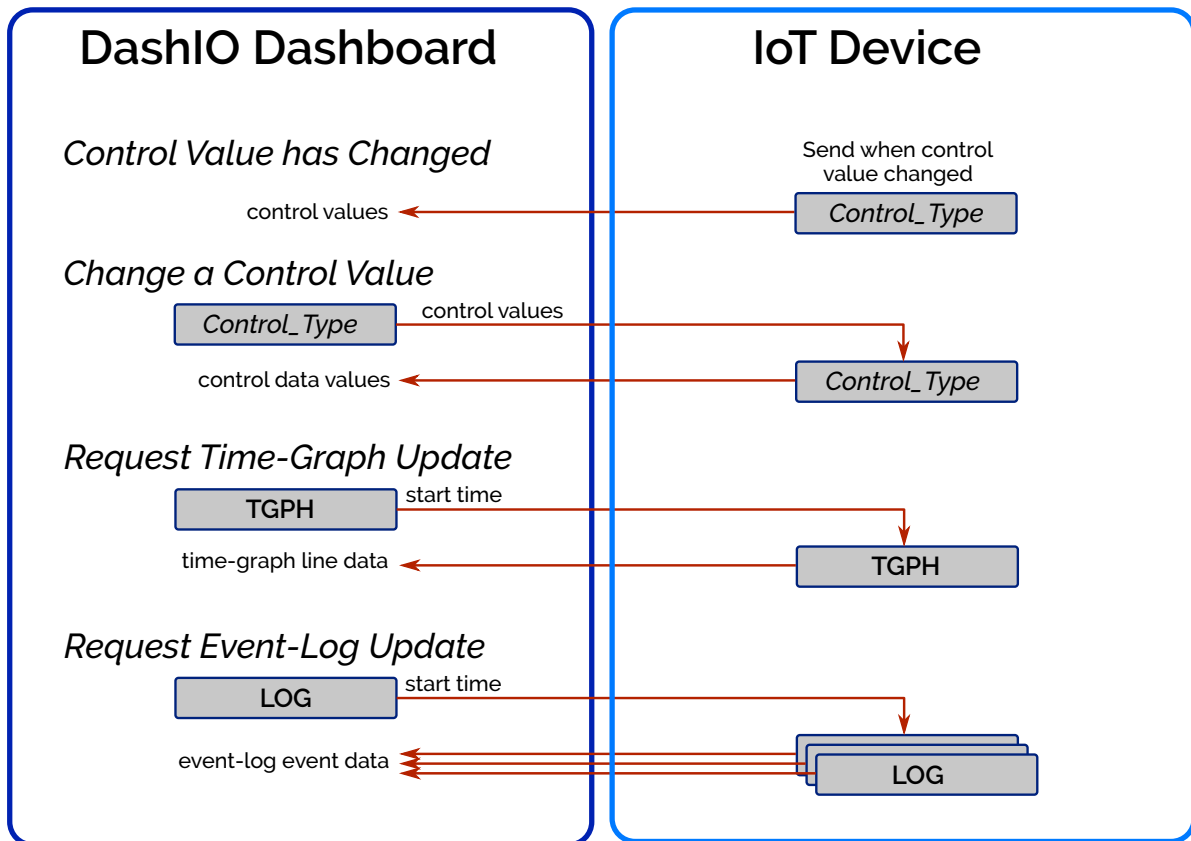


### 3.3. Control Messaging

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

The IoT device can send **Control\_Type** messages to the DashIO Dashboard whenever the control value changes or at regular intervals. Similarly, the DashIO Dashboard can send a value to a control when the user changes the value in the DashIO Dashboard. However, when the IoT device receives a **Control\_Type** message from the DashIO Dashboard, it is good practice to reply with another **Control\_Type** message to confirm to the DashIO 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 DashIO Dashboard. For example, a button **Control\_Type** message (BTTN) will be sent by the DashIO 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 (TGPH) and Event-Log (LOG) **Control\_Type** request messages from the DashIO Dashboard are used to request updated Time-Graph and Event-Log information respectively. These request messages include a **start time**, which indicates to the IoT device that the DashIO 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 DashIO Dashboard to IoT device

Action	Message	Notes
Request Device Connection Information	“\t WHO \n”	The WHO message must be replied too.
Validate Connection	“\t <b>Device_ID</b> \t CONNECT \n”	The CONNECT message must be replied too.
Request control initial conditions	“\t <b>Device_ID</b> \t STATUS \n”	Reply with the values for the controls, the set their initial conditions on the DashIO Dashboard after a connection has been established and validated.
Request Config Data	“\t <b>Device_ID</b> \t CFG \t <b>Width</b> \n”	The IoT device use respond with all configuration messages. Width is the DashIO dashboard screen with number of columns.
Update Device Name	“\t <b>Device_ID</b> \t NAME \t <b>Device_Name</b> \n”	The <b>Device_Name</b> is the name of the IoT device which may be changed from the DashIO Dashboard. If <b>Device_Name</b> 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 <b>Device_ID</b> \t WIFI \t <b>Country_Code</b> \t <b>SSID</b> \t <b>Password</b> \n”	The IoT device must reply to this message.
Update TCP Settings	“\t <b>Device_ID</b> \t TCP \t <b>Port</b> \n”	The IoT device must reply to this message.
Update DashIO Settings	“\t <b>Device_ID</b> \t DASHIO \t <b>User_Name</b> \t <b>password</b> \n”	The IoT device must reply to this message.
Update MQTT Settings	“\t <b>Device_ID</b> \t MQTT \t <b>URL</b> \t <b>port</b> \t <b>SSL</b> \t <b>User_Name</b> \t <b>Password</b> \n”	URL is the MQTT host URL. SSL is “true” when enabled and “false” when disabled. The IoT device must reply to this message.
Button Press	“\t <b>Device_ID</b> \t BTTN \t <b>Control_ID</b> \n”	
Send Text	“\t <b>Device_ID</b> \t TEXT \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is any ASCII text
Slider Moved	“\t <b>Device_ID</b> \t SLDR \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> , with or without decimal point
Menu Press	“\t <b>Device_ID</b> \t MENU \t <b>Control_ID</b> \n”	
Selector Selected	“\t <b>Device_ID</b> \t SLCTR \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is the index of the selected item.

Action	Message	Notes
Knob Turned	"\t <b>Device_ID</b> \t KNOB \t <b>Control_ID</b> \t <b>Data</b> \n"	<b>Data</b> , with or without decimal point
Request Time Graph Data	"\t <b>Device_ID</b> \t TGPH \t <b>Control_ID</b> \t <b>Time</b> \n"	The IoT device will then download the graph line data for all datapoints recorded on or after this <b>Time</b> . The time is UTC.
Request Event Log Data	"\t <b>Device_ID</b> \t LOG \t <b>Control_ID</b> \t <b>Time</b> \n"	The IoT device will then download all events after this <b>Time</b> . The time is UTC.
Online	"\t <b>Device_ID</b> \t ONLINE \n"	Useful for improving efficiency of MQTT connections.
Offline	"\t <b>Device_ID</b> \t OFFLINE \n"	Used for LWT for MQTT

### 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"

"\tFE:ED:CA:FE:BA:BE\tBTTN\tOnOff\n"

=> send a button press message to a **Control\_Type** "BTTN" with **Control\_ID** "OnOff"

"\tFE:ED:CA:FE:BA:BE\tMENU\tm01\n"

=> send a request update menu message to a **Control\_Type** "MENU" with **Control\_ID** "tm01"

## 4.2. Messages From IoT device to DashIO MQTT Broker

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

Action	Message	Notes
Notify Alarm	"\t <b>Device_ID</b> \t ALM \t <b>Alarm_ID</b> \t <b>Header</b> \t <b>Message</b> \n"	<b>Alarm_ID</b> is the identifier of the alarm. <b>Header</b> is the notification message heading and <b>Message</b> is the notification message body. Both <b>Header</b> and <b>Message</b> are ASCII text.  Alarm messages must be sent on the <b>alarm</b> topic.

Action	Message	Notes
Announce Connection	"\t <b>Device_ID</b> \t WHO \t <b>Device_Type</b> \t <b>Device_Name</b> \n"	Announce connection messages must be sent on the <b>announce</b> topic once the DashIO MQTT server has been connected too.  The announce connection messages should not be confused with other WHO messages that are part of standard messaging.

"\tFE: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 IoT device to DashIO Dashboard

Action	Message	Notes
WHO reply with identification information	"\t <b>Device_ID</b> \t WHO \t <b>Device_Type</b> \t <b>Device_Name</b> \n"	Send the <b>Device_ID</b> , <b>Device_Type</b> and <b>Device_Name</b> used to identify the IoT device.
Acknowledge CONNECT message	"\t <b>Device_ID</b> \t CONNECT \n"	Sent after receiving a CONNECT message. This message is required to complete the creation of a connection.
Updated Device Name	"\t <b>Device_ID</b> \t NAME \t <b>Device_Name</b> \n"	<b>Device_Name</b> successfully changed
Updated WiFi Settings	"\t <b>Device_ID</b> \t WIFI \n"	WiFi credentials successfully changed
Updated TCP Settings	"\t <b>Device_ID</b> \t TCP \n"	<b>Port</b> successfully changed
Updated DashIO Settings	"\t <b>Device_ID</b> \t DASHIO \n"	DashIO <b>User_Name</b> successfully changed
Updated MQTT Settings	"\t <b>Device_ID</b> \t MQTT \n"	MQTT settings successfully changed
Update Button State	"\t <b>Device_ID</b> \t BTTN \t <b>Control_ID</b> \t <b>State</b> \t <b>IconName</b> \t <b>Text</b> \n"	Changes the button icon or indicator color. <b>State</b> is an ASCII Boolean represented as "ON" or "OFF". <b>IconName</b> and <b>Text</b> are optional and can be used to set the icon name and text of the button. Subsequent messages that do NOT contain the <b>IconName</b> or <b>Text</b> will set the icon name and text of the button back to their default configuration values.
Update Slider Position	"\t <b>Device_ID</b> \t SLDR \t <b>Control_ID</b> \t <b>Data</b> \n"	<b>Data</b> is and ASCII Integer or Real Number.



Action	Message	Notes
Update Selector Index	“\t <b>Device_ID</b> \t SLCTR \t <b>Control_ID</b> \t <b>Selector_Data</b> \n”	<b>Selector_Data</b> is the index and group of text items used in the selector
Update Single Bar Graph	“\t <b>Device_ID</b> \t BAR \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is and ASCII Integer or Real Number.
Update Double Bar Graph	“\t <b>Device_ID</b> \t BAR \t <b>Control_ID</b> \t <b>Data</b> <sub>1</sub> \t <b>Data</b> <sub>2</sub> \n”	<b>Data</b> <sub>1</sub> and <b>Data</b> <sub>2</sub> are and ASCII Integer or Real Numbers. <b>Data</b> <sub>2</sub> is optional.
Update Text Display	“\t <b>Device_ID</b> \t TEXT \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is any ASCII text string
Update Knob	“\t <b>Device_ID</b> \t KNOB \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is any ASCII Integer or Real Number.
Update Knob Dial	“\t <b>Device_ID</b> \t KBDL \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is any ASCII Integer or Real Number.
Update Dial Display	“\t <b>Device_ID</b> \t DIAL \t <b>Control_ID</b> \t <b>Data</b> \n”	<b>Data</b> is any ASCII Integer or Real Number.
Update Direction Display	“\t <b>Device_ID</b> \t DIR \t <b>Control_ID</b> \t <b>Data</b> <sub>1</sub> \t <b>Data</b> <sub>2</sub> \n”	<b>Data</b> <sub>1</sub> is any ASCII Integer or Real Number representing Decimal Degrees. Secondary data <b>Data</b> <sub>2</sub> is any ASCII text string. <b>Data</b> <sub>2</sub> is optional.
Update Line for Graph Display	“\t <b>Device_ID</b> \t GRPH \t <b>Control_ID</b> \t <b>Line_Data</b> \n”	<b>Line_Data</b> is the data for a single graph line as defined below
Update Line for Time Graph Display	“\t <b>Device_ID</b> \t TGPH \t <b>Control_ID</b> \t <b>T_Line_Data</b> \n”	<b>T_Line_Data</b> is the data for a single graph line as defined below
Send Event	“\t <b>Device_ID</b> \t LOG \t <b>Control_ID</b> \t <b>Event_Data</b> \n”	<b>Event_Data</b> is the data for a single event line as defined below
Update Map Display	“\t <b>Device_ID</b> \t MAP \t <b>Control_ID</b> \t <b>Map_Data</b> \n”	<b>Map_Data</b> is the position information.
Configuration	“\t <b>Device_ID</b> \t CFG \t <b>Config_ID</b> \t <b>Config_Data</b> \n”	<b>Config_ID</b> defines the type of configuration data as defined later in this document. <b>Config_Data</b> is the configuration data, the format of which is dependent upon the <b>Config_ID</b> .
Popup Message	“\t <b>Device_ID</b> \t MSSG \t <b>Title</b> \t <b>Header</b> \t <b>Caption</b> \n” “\t <b>Device_ID</b> \t MSSG \t <b>Title</b> \t <b>Header</b> \n” “\t <b>Device_ID</b> \t MSSG \t <b>Title</b> \n”	<b>Title</b> , <b>Header</b> and <b>Caption</b> of the pop message on the dashboard.

### 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”

“\tFE: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.

"\tFE: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

## 4.4. Selector\_Data for Selector Control

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

**Selector\_Data** = "**Index** \t **Item**<sub>1</sub> \t **Item**<sub>2</sub> \t ...**Item**<sub>N</sub>"

Where **Index** is the index of **Item** that the selector is set too. The selection items of the selector are set to the **Items** in the message. For  $N > 0$ ,  $N$  is the number of selector text items in the selector. If  $N = 0$ , only the selector index is 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).

## 4.5. Line\_Data for Graph Control

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

**Line\_Data** = “**Line\_ID** \t **Line\_Name** \t **Line\_Type** \t **Color** \t **Y<sub>1</sub>** \t **Y<sub>2</sub>** \t ...**Y<sub>N</sub>**”

where  $N > 0$  is the number of points on the graph line

<b>Line_Data Fields</b>	<b>Description</b>
<b>Line_ID</b>	ASCII String containing a unique identifier for the Graph Line of the Graph display with <b>Control_ID</b> . DashIO Dashboard will add a new graph line for each new <b>Line_ID</b> .
<b>Line_Name</b>	Arbitrary name of the graph line, to be shown to the user on the Graph display on the DashIO Dashboard.
<b>Line_Type</b>	Shows how the Graph Line should be presented and may be one of the following: <ul style="list-style-type: none"> <li>• “<b>LINE</b>” Draws a line between each point <b>Y<sub>n</sub></b></li> <li>• “<b>BAR</b>” Draws a bar up to each point <b>Y<sub>n</sub></b></li> <li>• “<b>SEGBAR</b>” Draws a bar that is broken into segments up to each point <b>Y<sub>n</sub></b></li> <li>• “<b>PEAKBAR</b>” Draws a single bar segment at each point <b>Y<sub>n</sub></b></li> <li>• “<b>DEL</b>” Delete the line from the graph</li> </ul> If the <b>Line_Type</b> field does not contain a value or has an incorrect value, it will default to “ <b>LINE</b> ”.
<b>Color</b>	May be either the color name or an index of the color to be used from within the colors defined within <b>DashIO Dashboard</b> .  If the <b>Color</b> field does not contain a value or has an unknown value, it will default to the color black.
<b>Y<sub>n</sub></b>	The <b>Y</b> axis value of each point in the graph (where $n = 1$ to $N$ and $N$ is the number of graph points).  The <b>Y</b> values are the Y axis graph values and they may be integers or real numbers. The graph values are evenly spaced across the X axis.

### Receive Graph Line Message Example:

“\tFE:ED:CA:FE:BA:BE\tGRPH\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** “GRPH” with **Control\_ID** “OutsideTemp”, **Line\_ID** “L1”, **Line\_Type** “SEGBAR”, **Color\_Index** “9” and graph Y axis line values of “23.4”, “24.6”, “27”, “27.5”, “26.6”, “26.2”, “26”, “25.8”

## 4.6. Line\_Data for Time Graph Control

Time Graph controls can show many graph lines simultaneously. Each graph line is created with a message as follows:

**Line\_Data =**

**“Line\_ID \t Line\_Name \t Line\_Type \t Color \t T<sub>1</sub>,Y<sub>1</sub> \t T<sub>2</sub>,Y<sub>2</sub> \t ...T<sub>N</sub>,Y<sub>N</sub>”**

where  $N > 0$  is the number of points on the graph line and a comma (“,”) is used to separate the time and the Y value.

<b>Line_Data Fields</b>	<b>Description</b>
<b>Line_ID</b>	ASCII String containing a unique identifier for the Graph Line of the Graph display with <b>Control_ID</b> . DashIO Dashboard will add a new graph line for each new <b>Line_ID</b> .
<b>Line_Name</b>	Arbitrary name of the graph line, to be shown to the user on the Graph display on the DashIO Dashboard.
<b>Line_Type</b>	Shows how the Graph Line should be presented and may be one of the following: <ul style="list-style-type: none"> <li>• <b>“LINE”</b> Draws a line between each point <math>T_n, Y_n</math></li> <li>• <b>“BAR”</b> Draws a bar, starting at time <math>T_n</math> and ending at time <math>T_{n+1}</math>, filled to the value <math>Y_n</math></li> <li>• <b>“BOOL”</b>, Draws a shaded area (Alpha = 0.5) on the graph. When <math>Y_n</math> is <b>“true”</b>, <b>“t”</b>, <b>“high”</b> or <b>“h”</b>, the graph is shaded from time <math>T_n</math> until the time of a data point where <math>Y = \text{“false”}</math>, <b>“f”</b>, <b>“low”</b> or <b>“l”</b> is received. These values are NOT case sensitive.</li> <li>• <b>“DEL”</b> Delete the line from the graph</li> </ul> <p>If the <b>Line_Type</b> field does not contain a value or has an incorrect value, it will default to <b>“LINE”</b>.</p>
<b>Color</b>	May be either the color name or an index of the color to be used from within the colors defined within <b>DashIO Dashboard</b> .  If the <b>Color</b> field does not contain a value or has an unknown value, it will default to the color black.
<b>T<sub>n</sub></b>	The UTC time <b>T</b> of the data point <b>Y</b> . The following IOS 8601 based formats are supported: <ol style="list-style-type: none"> <li>1) Local time: “yyyy-MM-ddTHH:mm:ss”</li> <li>2) UTC: “yyyy-MM-ddTHH:mm:ssZ” (no need to manage timezone or daylight savings)</li> <li>3) Timezones: “yyyy-MM-ddTHH:mm:ss±hh:mm” (or ±hhmm or ±hh)</li> <li>4) <b>T<sub>1</sub></b> is either 1), 2) or 3) above and <b>T<sub>2</sub></b> to <b>T<sub>n</sub></b> are the number of minutes since <b>T<sub>1</sub></b>. The number of minutes may be an integer or float, where the value after the decimal point is the decimal fraction of a minute.</li> </ol>

<b>Line_Data Fields</b>	<b>Description</b>
<b>Y<sub>n</sub></b>	<p>The <b>Y</b> axis value of the point in the graph corresponding to the time <b>T</b>. (where <math>n = 1</math> to <math>N</math> and <math>N</math> is the number of graph points).</p> <p>The <b>Y</b> values are the Y axis graph values. They may be integers, real numbers, or text for when the Line_Type is “<b>BOOL</b>”.</p> <p>If the Line_Type is “<b>LINE</b>”, a break in the line may be added by setting the <b>Y</b> value of a point to “<b>b</b>” or “<b>break</b>”. The break should normally be used as the first data point in the line.</p>

## 4.7. Event\_Data for Event Log Control

Event Log controls can list many events from divergent IoT devices in a single list. Each event is created with a message as follows:

**Event\_Data = “Time \t Color \t Text<sub>1</sub> \t Text<sub>2</sub> \t ...Text<sub>N</sub>”**

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

<b>Line_Data Fields</b>	<b>Description</b>
<b>Time</b>	<p>The UTC time <b>T</b> of the data point <b>Y</b>. The following IOS 8601 based formats are supported:</p> <ol style="list-style-type: none"> <li>1) Local time: “yyyy-MM-ddTHH:mm:ss”</li> <li>2) UTC: “yyyy-MM-ddTHH:mm:ssZ” (no need to manage timezone or daylight savings)</li> <li>3) Timezones: “yyyy-MM-ddTHH:mm:ss±hh:mm” (or ±hhmm or ±hh)</li> <li>4) <b>T<sub>1</sub></b> is either 1), 2) or 3) above and <b>T<sub>2</sub></b> to <b>T<sub>n</sub></b> are the number of minutes since <b>T<sub>1</sub></b>. The number of minutes may be an integer or float, where the value after the decimal point is the decimal fraction of a minute.</li> </ol>
<b>Color</b>	<p>May be either the color name or an index of the color to be used from within the colors defined within <b>DashIO Dashboard</b>.</p> <p>If the <b>Color</b> field does not contain a value or has an unknown value, it will default to the color black.</p>
<b>Text<sub>n</sub></b>	Row of text to be displayed on the Event Log

## 4.8. Map\_Data for Map Control

There are two formats available for the map data and they are identified by the DashIO Dashboard by the number of fields in the data. The map data is displayed on the DashIO Dashboard when the pin on the map of the DashIO Dashboard is tapped.

### 4.9.1 JSON Map Data Payload

The **Map\_Data** is a single field describing the GPS information with 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	yyyy-MM-dd'T'HH:mm:ssZ (refer to ISO 8601)
"latitude"	number	Latitude in decimal degrees
"longitude"	number	Longitude in decimal degrees
"avgeSpeed"	number	Average speed since the last message in meters/second
"peakSpeed"	number	Maximum speed since the last message in meters/second
"course"	number	Course direction in decimal degrees. A negative value indicates an unknown heading
"altitude"	number	Altitude in meters
"message"	string	Displayed with the pin on the map of the DashIO Dashboard
"distance"	number	Accumulated distance since the last message in meters

### 4.9.2. Simple Map Data Payload

The **Map\_Data** has three fields:

**Map\_Data = "Latitude \t Longitude \t Message"**

Where the Latitude and Longitude are in decimal degrees.

## 4.9. Config\_ID and Config\_Data for Configuration Messages

Configuration messages provide information about each control in the IoT device to the DashIO Dashboard. This allows the user to easily setup the DashIO 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:

<b>Config_ID</b>	<b>Config_Data</b>
"NA"	Configuration data is not available for this device and the <b>Config_Data</b> is ignored by the DashIO Dashboard. The Config_Data does not need to be included in the message.
"BAS"	The configuration data ( <b>Config_Data</b> ) contains only basic configuration information for controls and as defined below.
<b>Control_Type</b> or <b>Connection_Type</b>	<p>The <b>Config_Data</b> is JSON formatted text describing each parameter and its associated value for the particular <b>Control_Type</b></p> <p>or</p> <p>Additional layout <b>Control_Types</b></p> <ul style="list-style-type: none"> <li>• "DVCE" Device properties</li> <li>• "DVVW" Device View (config messages only)</li> </ul> <p>or</p> <p><b>Connection_Type</b> for communication connections configuration messages</p> <ul style="list-style-type: none"> <li>• "MQTT" MQTT Connection</li> <li>• "BLE" BLE Connection</li> <li>• "TCP" TCP Connection</li> </ul>



#### 4.10.1. Basic Control Configuration

Basic control configuration only provides information to the DashIO Dashboard that the control exists and can be easily identified. The Basic control configuration data (where **Config\_ID** = "BAS") is as follows:

**Config\_Data** = "**Control\_Data**<sub>1</sub> \t **Control\_Data**<sub>2</sub> \t ... **Control\_Data**<sub>N</sub>"

and

**Control\_Data**<sub>n</sub> = "**Control\_Type**<sub>n</sub> \t **Control\_ID**<sub>n</sub> \t **Control\_Title**<sub>n</sub>"

where n = 1 to N and N is the number of controls

<b>Config_Data Fields</b>	<b>Description</b>
<b>Control_Type</b>	As defined earlier in this document. Only the following control types may be used in a Basic configuration message:  "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 "GRPH" Graph display "TGPH" Graph display with time on the X axis "SLCTR" Selector control "LOG" Event Log display "LBL" Label Box display
<b>Control_ID</b>	As defined earlier in this document.
<b>Control_Title</b>	The title of the control to be displayed on the DashIO Dashboard control

When the DashIO Dashboard receives a Basic configuration message, it creates a new **Device View** if one doesn't exist for the deviceID of the connection.

## 4.10.2. 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 DashIO 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:

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

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 DashIO 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 **DashIO 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. "name, wifi, tcp, dashio, mqtt" where "name" = can change change the device name "wifi" = can change the wifi Country Code, SSID and password "tcp" = can change the TCP port "dashio" = can change the DashIO server MQTT username and password "mqtt" = can change the MQTT URL, port, SSL (enabled), username, and password

### Device View

Element	Value Data Type	Purpose
"controllID"	string	Identifier of the device view
"title"	string	Text to be displayed in the title box of the control
"iconName"	string	Name of the icon from the icons in DashIO Dashboard. Displayed on the device view selector.
"color"	number	Color of device view background

Element	Value Data Type	Purpose
"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"	<b>true or false</b>	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"
"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

Element	Value Data Type	Purpose
"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"	<b>true or false</b>	Enable the button to respond to touch and send messages
"iconName"	string	Name of the icon from the icons in DashIO 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”
“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”	<b>true or false</b>	Send message only when the the knob is released
“dialFollowsKnob”	<b>true or false</b>	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”	<b>true or false</b>	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, or "intvl" format ISO8601 text as hours, minutes and seconds.
"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 hexadecimal
"closeKbdOnSend"	<b>true or false</b>	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"
"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"	<b>true or false</b>	Send message only when the the knob is released
"sliderEnabled"	<b>true or false</b>	Show the slider
"knobColor"	string	Color of the knob
"sendOnlyOnRelease"	<b>true or false</b>	Send message only when the the knob is released
"barFollowsSlider"	<b>true or false</b>	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



## 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"
"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 side
"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
"yAxisMin"	number	Value of the vertical axis bottom
"yAxisMax"	number	Value of the vertical axis top
"yAxisNumBars"	number	Positive integer. Number of vertical grid lines, including the top and bottom axes

## 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
"yAxisMin"	number	Value of the vertical axis bottom
"yAxisMax"	number	Value of the vertical axis top
"yAxisNumBars"	number	Positive integer. Number of vertical grid lines, including the top and bottom axes

## 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
“controllID”	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”
“iconName”	string	Name of the icon from the icons in DashIO Dashboard. Displayed on the menu
“text”	string	Text displayed on the menu

## Button Group

Element	Value Data Type	Purpose
“controllID”	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”
“iconName”	string	Name of the icon from the icons in DashIO Dashboard. Displayed on the menu
“text”	string	Text displayed on the menu
“gridView”	<b>true or false</b>	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”

## Connection - MQTT

Element	Value Data Type	Purpose
“userName”	string	Username for the MQTT host
“hostURL”	string	URL for the MQTT host

## Connection - BLE

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

### Connection - TCP

Element	Value Data Type	Purpose
“ipAddress”	string	IP address for the TCP connection of the IoT device. ??? Not required for mDNS, but speeds things up.
“port”	number	Port for the TCP connection of the IoT device. ??? Not required for mDNS.

### Alarm - Push Notification

Element	Value Data Type	Purpose
“controllID”	string	Identifier of the alarm on the IoT device.
“description”	string	Description of the alarm.
“soundName”	string	The name of the alarm sound. If this is left blank, the default sound will be used.

## 4.10. MQTT Communications Considerations

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

<b>Topic Name</b>	<b>Message</b>	<b>Full Topic required for the MQTT Broker</b>
Control	<b>Send_Data</b> messages from the DashIO Dashboard to the IoT device to control the IoT device.	“ <b>User_Name/deviceID/control</b> ”
Data	<b>Receive_Data</b> messages from the IoT device to display on a control on the DashIO Dashboard	“ <b>User_Name/deviceID/data</b> ”
Alarm	<b>Alarm</b> message sent from the IoT device to the MQTT broker to initiate a push notification alarm to the device the DashIO Dashboard is installed on.	“ <b>User_Name/deviceID/alarm</b> ”
Announce	<b>Announce</b> 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.	“ <b>User_Name/deviceID/announce</b> ”

Where:

- The **User\_Name** is the user name of the DashIO 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 **DashIO MQTT broker connection must use SSL.**

## 4.11. Device Discovery

The IoT device discovery process within the DashIO 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 DashIO Dashboard sends a WHO message and receives a valid who response.

### 4.11.1. BLE Discovery

**BLE** device discovery is achieved with the **DashIO 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 **DashIO Dashboard** will always attempt to connect to IoT devices with TCP connections over the local network that the **DashIO 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 **DashIO 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 **DashIO Cloud Services** MQTT broker is used (i.e. dash.dashio.io on port 8883 using SSL).

Device discovery is an integral part of the **DashIO Cloud Services** and is initiated by the IoT device sending a **DashIO 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 **DashIO** MQTT broker and stored as an available device for the particular user. The **DashIO Dashboard** retrieves the available IoT devices from the **DashIO Cloud Services** database.