bMyQ Driver

bMyQ is a driver that allows your connected home to control and retrieve the status of your MyQ-enabled motorized devices.

This driver is a known as a "UI Button" driver. This class of drivers does not use Control4 's standard template as an interface but instead can use an icon or series of icons to communicate the driver's status and send commands.

Setup

1) Within your Control4 project, install an instance of the bMyQ driver for each device you wish to control
2) Enter your MyQ username and password
3) Run the “Get Devices” Action to retrieve all devices associated with your account
4) Select the device to control in the “Devices” property
5) Select Device Status Source. When using Contact Sensor, link the appropriate CONTACT_SENSOR input in Composer Connections. When using MyQ, configure Polling settings. *See Polling Notes for discussion
6) Configure Confirm Select and Ignore Sensor Double Trigger as appropriate. *See respective notes sections for discussion
7) Configure/add each bMyQ driver instance to the desired room(s) navigator experience menu
8) Refresh navigators

Properties

Driver Setup

Upgrade Mode	BNet Solutions drivers can automatically update themselves. Options are “Automatic”, “Minor Only” and “Upgrade Now”. See “Upgrade Mode” section below
Admin Server Enabled	Enables / Disables the driver’s Admin Server. See “Admin Server” section below
Admin Port	Listening port assigned to Admin Server
Admin Token	Token required to access the driver’s Admin Server

License Setup

License Key	Key used to permanently enable driver functionality
License Status	Display current state of driver’s license
Driver Version	Installed driver version
MAC Address	Unique network interface identifier for the Controller

MyQ Setup

Username	MyQ account username
Password	MyQ account password
Devices	List of devices associated with your MyQ account
Device Status Source	Selects data source to determine device status. Choices are “MyQ Polling” and “Contact Sensor”. *See Device Status Source Notes for discussion

Device Status	Device's currently known state
Polling Enabled	Enable/Disable polling. *See Polling Notes section for discussion
Polling Interval	Interval (in minutes) that bMyQ will update device status
Device Travel Time	Duration (in seconds) of a device's travel time excluding closing warning alarm (beeps)
Confirm Select	Enabled/Disable confirmation click before executing device action
Ignore Sensor Double Trigger	Enable/Disable driver's ability to Ignore sensor double trigger or false toggle notifications. See Ignore Sensor Double Trigger Notes section for discussion

Debug Settings

Debug Mode	Sets where the driver outputs debug information. See "Troubleshooting" section below for more information
Debug Level	Sets how much debug detail the driver outputs

Actions

Get Devices	Retrieve devices associated with MyQ account
Update Device State	Retrieves current state for device selected in "Devices" property
Open Device	Send "Open" command to device selected in "Devices" property
Close Device	Send "Open" command to device selected in "Devices" property

Commands

Open Device	Send "Open" command to device selected in "Devices" property
Close Device	Send "Open" command to device selected in "Devices" property

Events

Open	Fired when device enters "open" state
Opening	Fired after "open" command is successfully sent to MyQ
Closed	Fired when device enters "closed" state
Closing	Fired after "closed" command is successfully sent to MyQ
Error	Fired after 3 failed MyQ service re-query attempts

Variables

The following variables are available and updated after each "New Data Received" event. They are useful for constructing messages (Push or Email) as well as constructing logic flow within Composer.

Status	Driver activity status
Device Name	MyQ device name
Device State	Device's current state – Open/Opening/Closed/Closing

Conditionals

Device Open	True if device state is "Open", False if device state is "Closed"
Device Closed	False if device state is "Open", True if device state is "Closed"

Device Status Source Notes

bMyQ can either use a Contact Sensor or MyQ Polling service as the device's status source.

When using MyQ Polling, the driver will, in accordance with the Polling Enabled and Polling Interval settings, query MyQ for a device's status. The MyQ service will determine the timing for

all device events – Opening / Open / Closing / Closed.

When using Contact Sensor, the driver will use the CONTACT_SENSOR input, as configured in Composer Connections, to determine the timing for the Closed and Opening events. Open and Closing events, because they happen at the opposite end of device's travel from the typical contact sensor location, are determined by the MyQ service (Open) or generated by the driver (Closing after sending Close command).

Polling Notes

The Polling Interval is configurable within a range of 1 minute to 10 minutes and defaults to every 5 minutes. Polling is a balance between keeping the driver's device status up to date vs overloading the service provider's API - MyQ in this case.

"Update" polling is suspended after a command is successfully sent and is replaced with "Device Travel" polling until the device successfully completes the command – see Device Travel Time Notes for details. Afterword, "Update" polling is resumed (if enabled) at the configured interval. The design intention is to allow longer "Update" polling interval but keep the driver more informed about the device's state after a command is issued.

"Device Travel" polling is an independent function from "Update" polling and operates even when "Update" polling is disabled and when Device Status Source is set to Contact Sensor.

Polling Enabled and Polling Interval settings are ignored when Device Status Source is set to Contact Sensor.

Device Travel Time Notes

The Device Travel Time controls how long bMyQ will wait, after issuing a command, until it checks the MyQ service for a device's status. The time, in seconds, should be set to the device's typical complete travel time (without the closing warning beeps for close travel). If the status is not what was expected ("Open" for open command or "Closed" for close command), bMyQ will wait 5 seconds and re-query the status. An Error event is triggered if the expected state is not received after 3 re-queries.

Confirm Select Notes

When Confirm Select is enabled, the driver will require a second click/select before sending a command to the device. The UI will display the "Confirm Open" / "Confirm Close" icon for 2 seconds during which the "Confirm" click/select must be received to initiate the device command.

Ignore Sensor Double Trigger Notes

My garage sensors have an interesting feature whereby they send several contact signals as a door opens or closes. Because they all exhibit the same behavior and appear to be installed correctly, it seems to be a "feature" of this particular hardware. In Control4, my "Open" sequence is expressed as OPENED / CLOSED / OPENED and my "Closed" sequence is CLOSED / OPENED / CLOSED.

If you see this behavior, enabling Ignore Sensor Double Trigger (triple in my case) will instruct the driver to act on the first OPENED/CLOSED notification but ignore, for 2 seconds, additional OPENED and CLOSED notifications.

Device Icons

The below table describes bMyQ's icon set:

	MyQ device is open
	MyQ device is opening
	MyQ device is closed
	MyQ device is closing
	Confirm Open Command
	Confirm Close Command

UI Button

As mentioned above, bMyQ is a driver of type "UI Button". Clicking on the bMyQ icon will send an "Open" command to a device with a "Closed" state and will send a "Close" command to a device with an "Open" state. Commands, via the driver's icon or via Actions, will only execute if the device's state is "Open" or "Closed" and are not queued for execution. That is, commands send while the device is "Opening" or "Closing" will be dropped.

Sending commands via the driver icon works reliably on all connected Navigators (T3s, EA1, etc.) and on the Control4 app when the current view is refreshed (not to be confused with a navigator refresh). That is, unless the app is opened directly to a view that contains the driver, Control4 does not always register a click. Instead, you may need to back out, then navigate back into the view containing the driver icon.

To help discern when Control4 successfully registers a click, the driver's icon changes to the 'dot-dot-dot' icon when it receives a click event and then shows the updated device status after the click window ends.

Admin Server

BNet Solutions drivers' have a built-in webserver that looks and behaves like the Properties, Actions and Lua Output tabs for the driver in Composer. The Admin Server's default port for the bMyQ driver is 41000 and is configurable in the driver's properties. Using a web browser, navigate to http://[controller ip]:41000 where "controller ip" is the IP Address of your Control4 Director (EA5, EA3, EA1, etc). For example, http://192.168.1.100:41000. The Admin Server is protected by a challenge page that requires a token to continue. By default, the token is "bMyQAdmin". Once authenticated, the token is stored in a cookie (technically hashed, then

stored) so you won't need to log in every time. The token is configurable via the driver's property page. The Admin Server is enabled by default but can be disabled entirely via the driver's property page.

Upgrade Mode

BNet Solutions drivers can automatically update themselves. New driver functionality or capability is typically packaged as an incremental "Major" version (v3, v4 etc). "Minor" versions (v2.3, v2.4) are typically maintenance releases that update underlying libraries, address a specific issue or usability concern.

Automatic	When "Automatic" is selected, the driver will upgrade/update itself when a new version is available. "Automatic" Upgrade Mode is strongly suggested.
Minor Only	Restricts the driver from upgrading between major versions but allows updates between minor versions.
Update Now	Checks for and upgrades to any newer Major or Minor version.

Known Issues and Limitations

- **MyQ does NOT provide support or documentation for their Cloud service and may, at any time, decide to change their API implementation.** If a change impacts bMyQ, I will make every effort to adopt the updated API set – this is how I control my garage doors as well!
- After the driver is updated via "Update Now", dynamically populated drop-down boxes may appear blank. Refresh the project (File/Refresh) or simply change focus away from then back to this driver to repopulate. This seems to be a Composer issue.

Troubleshooting

All BNet Solutions products have an additional 'Submit' Debug Mode. With this mode selected, the driver creates a unique log file to capture the Lua output based on the selected Debug Level (usually set to "5 - Debug"). Once 'Submit' Debug Mode is deselected, either manually or when the Debug Timer expires, the Submit Debug Log is uploaded to the BNet Solutions Server for analysis.

The server notifies me when Submit files are uploaded but if you have not purchased a license, I will have no way to reach back out to you for troubleshooting so please email me your contact information.

Trial and Purchase

All BNet Solutions drivers are fully functional for a 7-day Trial Period. To continue usage after the Trial Period, you will need to purchase a license. To do so, install the bLicense driver from https://bnet4solutions.com and follow the documented purchase steps. There are no refunds so please review the documentation for any limitations or known issues and determine acceptance during the Trial Period.

Legal

By using this driver, you are indicating that you have read and agree with the Policies and Terms that govern its usage as published here.

My Contact Information

You can reach me at blucas@bnet4solutions.com for comments or questions.

bMyQ Driver

Change Log

v1 - 11/19	Initial Release
v2 - 11/19	Added driver auto-update functionality
v3 - 04/20	Added Device Status Source (Contact Sensor or MyQ Polling)
	Added configurable Device Travel Time
	Added Confirm Select functionality
	Added Ignore Sensor Double Trigger functionality