



b6Launch™ Guide for Arduino Yun

Overview

Install

- Step 1. Arduino Yun Configuration
 - Installing the Arduino IDE
 - Powering on the Arduino Yun
 - Setting COM Port, WiFi or Ethernet
 - Customizing .Hex File Output
- Step 2. Basic6 IoT Web Console
 - Initial Login to Console
 - Device Token Registration
- Step 3. Basic6 Technology Install
 - Basic6 Technology Sketch
 - Credential Entry
 - Serial Monitor
 - Arduino Yun Shield Install

Website Navigation

- Headers
- Device Container/Groups
- Creating a Group

Account

- Plan
 - Plan Update
 - Downgrade a Plan
- My Profile
- Payment Methods
 - Adding a Payment Method
 - Adding an Additional Card Payment
 - Setting an Additional Payment Method
 - Delete a Payment Method
- Billing
- Device Token
 - Replacing a Device Token
- Change Password

Delete Account
Timeout

Controls/Functionality

Device Controls
Latest Status
Main Display Window
Push Program
 Adding a Sketch
 Uploading a .Hex File
 Uploading a Sketch
 Edit a Program
 Pushing a Program
Program History
Accessing the Program History
Execute Command

References and Resources

OVERVIEW

This document outlines a step by step procedure to successfully install and configure the Arduino Yun within the B6Launch platform. Once configured the Arduino Yun will be accessible with full management capabilities.

Necessary hardware:

Arduino Yun microcontroller/computer board

Micro USB cable

Tools:

Computer with USB port, running Arduino IDE Version Arduino IDE 1.5.4 or later

Free download from Arduino.cc

INSTALL

STEP 1. ARDUINO YUN CONFIGURATION

This section will walk the user through the necessary out of the box Arduino Yun configuration. If you are familiar with the Arduino Yun and the initial setup process, please proceed to Step 2 of this manual.

Installing the Arduino IDE

1. This section of the guide will walk you through the downloading and installing of the Arduino software (also known as the Arduino IDE - short for Integrated Development Environment). To start the initial configuration process, follow the provided link to the Arduino IDE installation package:
<http://arduino.cc/en/main/software>
2. The currently available Arduino 1.6.3 IDE software can be used with any Arduino board, including the Yun. Simply select your operating system from the provided listing on the right of the screen and download the package.



ARDUINO 1.6.3

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

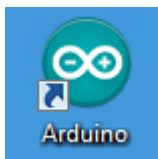
Windows Installer
Windows ZIP file for non admin install

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits

[Release Notes](#)
[Source Code](#)
[Checksums](#)

3. A desktop icon will be created upon successful IDE installation.



Powering on the Arduino Yun

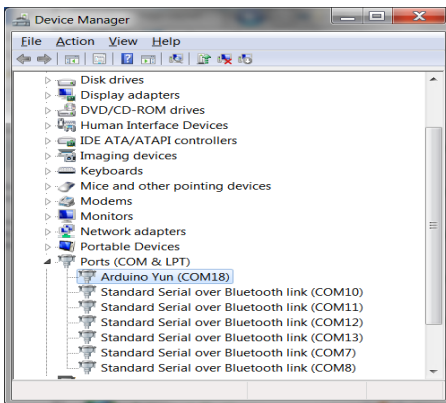
1. Connect the Arduino Yun Board to Your Host (Laptop/ Computer) using a **Micro USB Data Cable** as shown in the figure below.



The normal USB Port should be connected to the Computer and the smaller end (Micro USB) should connect to the Yun Board.

(Please Note: Arduino Yun and in general Arduino boards operate with a maximum of 5 Volts. If the input exceeds 5Volts, the Arduino board will burn. Powering your Yun from your computer with a Mini USB cable will ensure that the proper voltage is used.)

2. Once the Board is powered, you should see an LED lit on the Yun Board. Any necessary additional drivers should be automatically installed at this time and you should see the Yun connection through COM Port in your Device Manager (Windows) and WiFi Connection from Yun in your WiFi Network.



For Device Manager: Click **Start**→**Control Panel**→**Device Manager**.



For WiFi Network: Click on the Network Icon or **Start**→**Control Panel**→**Networking Sharing Center**

If you are using a Mac, click on the WiFi icon in the top right of your screen (near the volume icon). Here, you should see a new WiFi network emitted by the Yun. It should look something like "Arduino Yun-90A2DAF00BF0". Choose this network. If you do not see this network, hold the restart button on the Yun near the USB port for over 5 seconds to restart its WiFi emission.

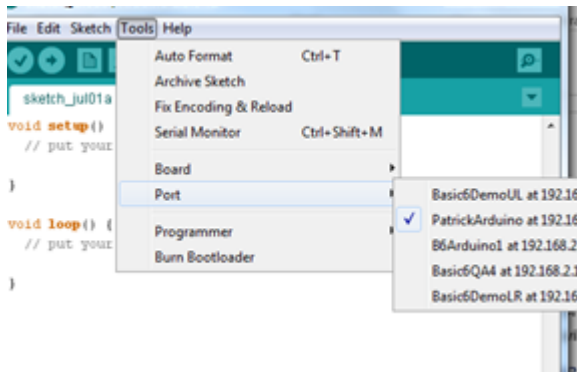
Please note, that if you connect your computer to the Arduino's WiFi in order to configure it, and if your computer was connected to the internet via your WiFi router, you will now lose

internet connectivity. To restore internet connectivity, simply reconnect your computer to your router WiFi.

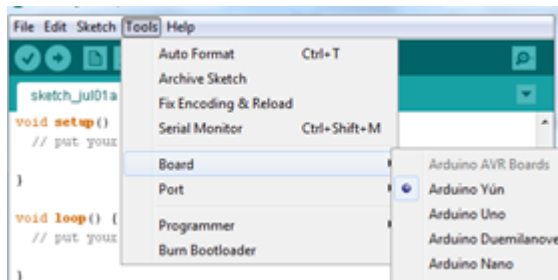
Setting COM Port, WiFi or Ethernet

1. From the desktop, launch the IDE via the previously created desktop icon and navigate within the interface to **Tools → Port → COM Ports**.

You will see list of available COM Ports. Check the Port which was connected/as shown by Device Manager. (This will establish serial communication between your computer and the Arduino)



2. Navigate to **Tools → Board → Set the board to Arduino Yun**



This will establish the correct protocol to IDE to use Arduino Yun Board.

3. Click on the below link <http://arduino.cc/en/Guide/ArduinoYun#toc13> and follow the Arduino supplied instructions to connect to the WiFi.

Once completed, you should see the IP Address listed below the COM Ports drop down menu selection.

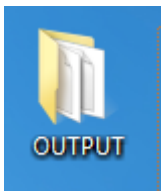
(Please take note to enter the word: **arduino** as the initial Password entry in the Welcome Message screen)

4. Click on the link <http://arduino.cc/en/Guide/ArduinoYun#toc15> and follow instructions to connect the Yun device to Ethernet.

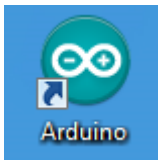
Customizing .Hex File Output

An important part of the configuration process is the creation of a .HEX file Output folder, the use of which will be discussed later in the guide within the Push Sketch/Program section. When Arduino compiles your sketches, it produces an output file called .HEX file. This file is usually stored in some remote location usually under user/AppData. To streamline the process of accessing created .HEX files, Basic6 recommends the creation of a readily available repository.

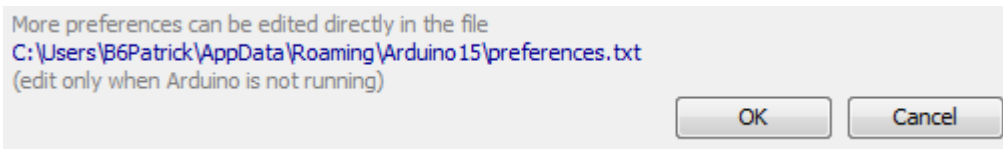
1. To begin, Create a New Folder on your desktop or within selected file system that will act as .HEX file repository




2. Click on desktop icon to open and access the Arduino IDE.



3. Within the Menu selection Click On **File > Preferences**
4. Scroll down to access the provided link between the “More preferences can be edited directly in the file”; “edit only when Arduino is not running” messages



5. This will provide access to the Preferences file document within a text editor

Name	Date modified	Type	Size
 preferences	7/1/2014 9:48 AM	Text Document	3 KB

6. Within the opened preferences text document customize the location of .HEX file location by entering the build path that was created in the previous Output File Creation step. In the following example a Folder named OUTPUT was created on the user desktop.

build.path=C:\Users\B6Patrick\Desktop\OUTPUT

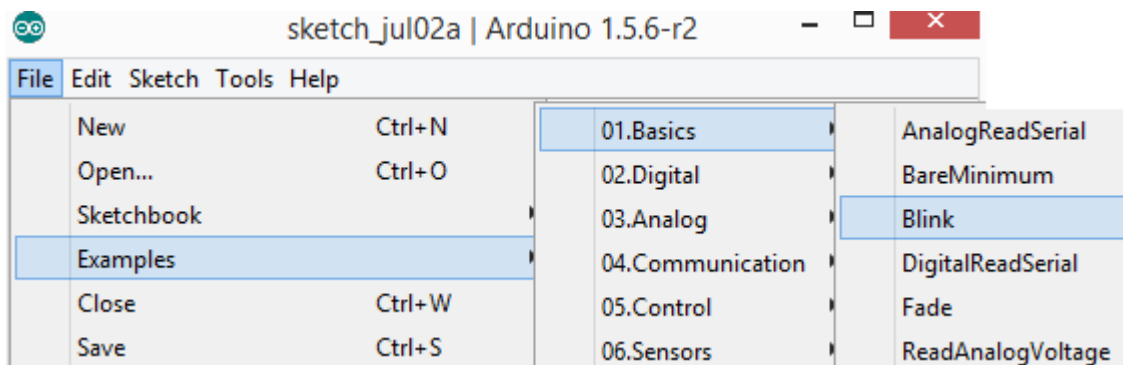
7. Enter the line for the build path as the third line of the preferences file via the Edit function of the Toolbar, Clicking Save when complete. The Arduino should be restarted at this time to register any changes.

```

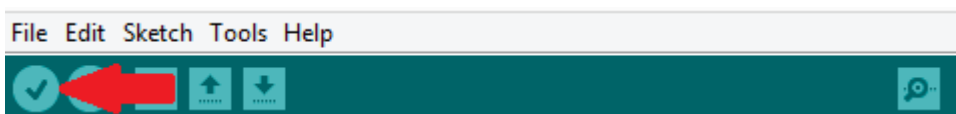
board=yun
browser.linux=mozilla
build.path=C:\Users\B6Patrick\Desktop\OUTPUT
build.warn_data_percentage=75
console=true
console.auto_clear=true
console.error.file=stderr.txt
console.length=500
console.lines=4

```

8. To test the created .HEX file repository, within the IDE interface access `File>Examples>01.Basics>Blink` which enables the creation of a quick sketch that can be pushed out to the Arduino.



9. Once selected, click the Verify button (check mark) in the upper left of the Arduino IDE to compile the **Blink** Sketch. Upon successful completion, this will cause the built-in LED lights on the Arduino Yun to blink intermittently.



10. The output of the sketch should be captured and viewable within the designated .HEX file folder.

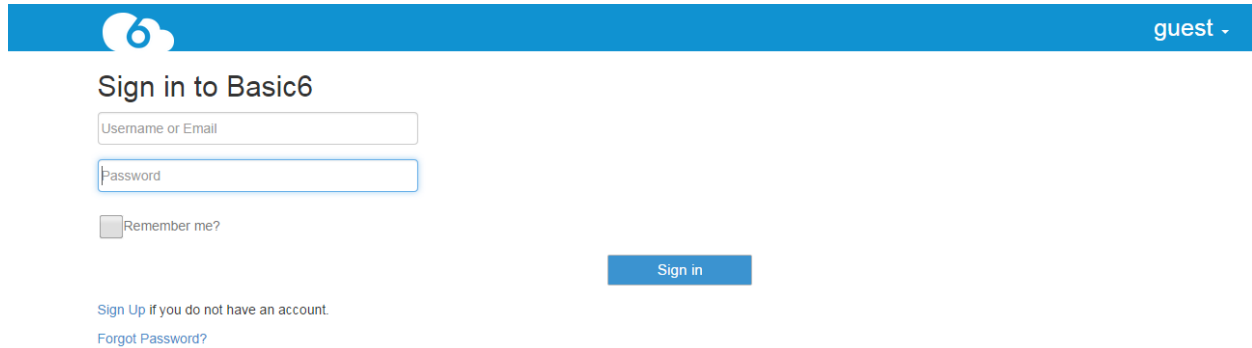
Step 2: Basic6 IoT Web Console

Initial Login to the Basic6 Web Console

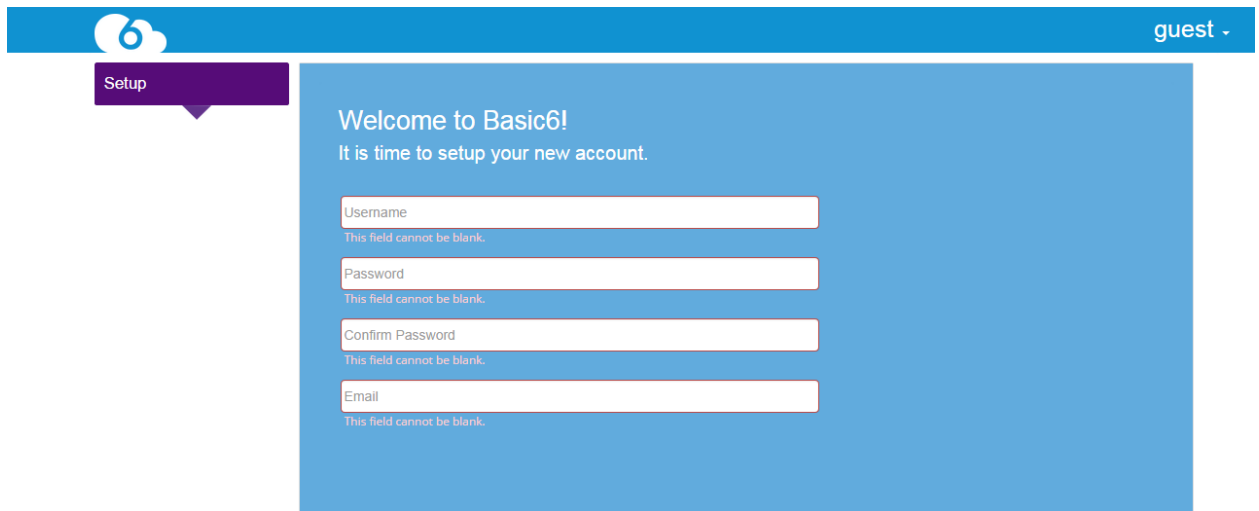
This step involves the initial setup of credentials along with the delivery of a device token.

1. Begin by navigating to the Basic6 IoT login page: <http://iot.basic6.com/>

Select the **Sign Up (if you do not have an account)** Action located in the bottom left of the screen.



2. As seen below, select and enter a unique User Name and Password which will be confirmed to you via email.

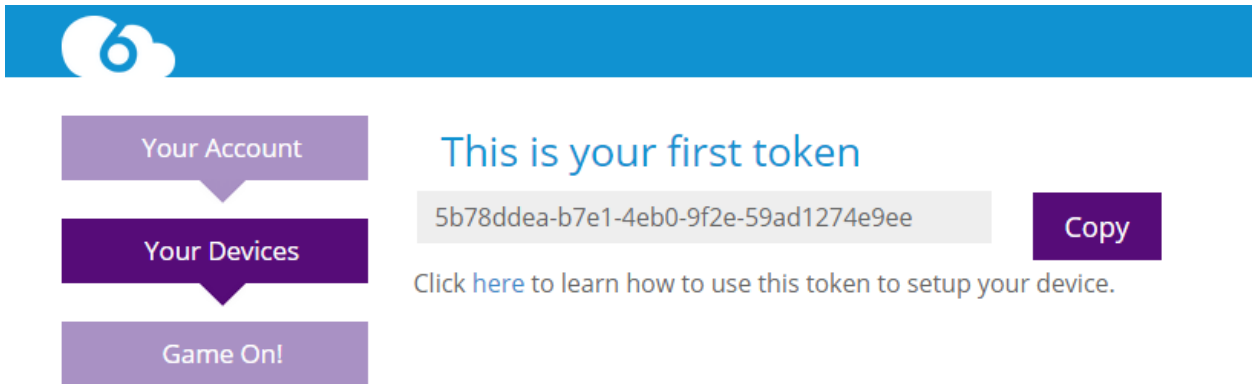


3. After email confirmation, click on the provided link and enter credentials to access the Basic6 website. You will be presented with a few screens, which include the entry of Account information (not required), along with a video walk through. The most important of the screens will be the presentation of the Device Token.

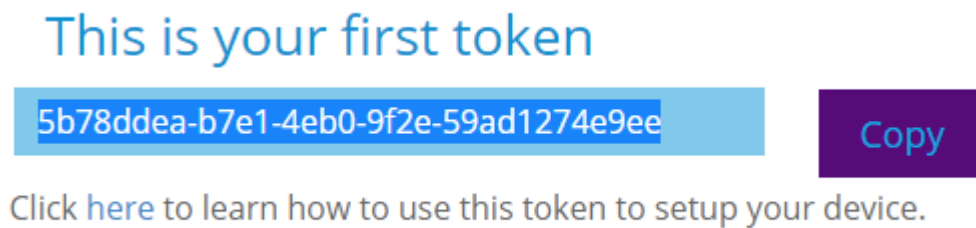
Device Token Registration

- The second screen of the Basic6 Account Sign-up process supplies every Basic6 user account with a single, unique *Device Token* that is used for authentication across all connected devices. This device token can be quickly changed if there is any fear of compromise.

A device is initially registered via a combination of Basic6 Account credentials and device token. Once the process has completed, the device token is removed in favor of a device id, with no trace of the Device Token remaining on the individual devices.



This unique token, which can be viewed only once, can be copied at this time (via the supplied Copy button) and will be used to register devices, correlating them to your Basic6 account. It is recommended that the supplied token be saved in a safe, retrievable repository.



- After clicking through all of the Welcome/Sign-up screens, you will come to the main B6Launch screen which will display all of your connected devices.

Welcome to Basic6!

It looks like you have not registered a device yet. Click [here](#) to get started.

Step 3: Basic6 Technology

Basic6 Technology Sketch

This section of the document outlines the step-by-step procedure necessary to install and configure the Basic6 technology in the Arduino Yun's Linux environment. This process is predicated on the creation of a Basic6 user account. If you have not already created an account with Basic6 (Step 2), (<https://iot.basic6.com/>) do so now, then come back here.

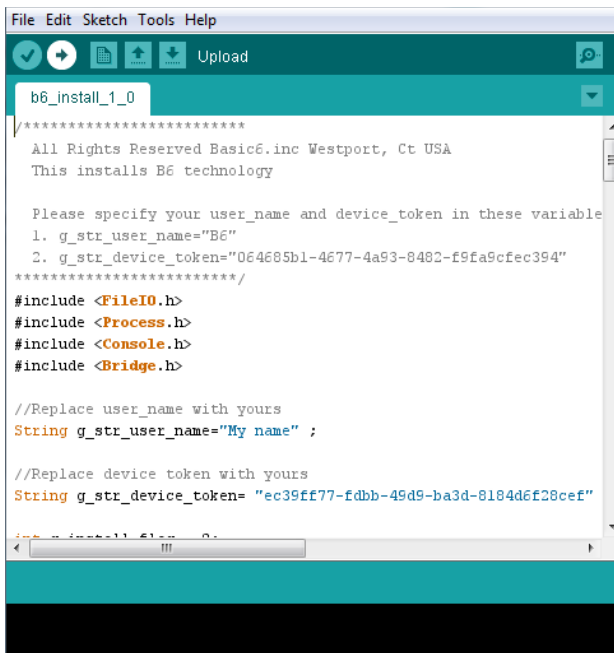
In these actions, you will download a provided Launch sketch which will automatically open within the Arduino IDE. Within this sketch you will adjust the values for username and device token to match your B6 provided credentials.

Basic6 technology is a python based daemon service which runs in a Linux environment. It requires the python runtime 2.7 (which comes by default with the Arduino Yun) and a Websocket Client module (which will be installed as a pre-requisite). The installation requires root administrative privileges on the device in order to setup the daemon service which requires changes to the crontab (cron tables), init.d, and rc.local files.

1. Basic6 provides an installation sketch for you to run that configures your device with Basic6 technology: <https://www.basic6.com/b6launch/getting-started/quickwalkthrough> (Step 2: installation sketch link). Please download the provided .zip file within the link.

Name	Size	Packed	Type	Modified	CRC32
..			File folder		
basic6_yun_install_sketch_1_2.ino	5,035	1,471	Arduino Source C...	3/25/2015 12:1...	33C1D215

2. The Basic6 Installation Sketch will open the Arduino IDE with the configured sketch ready for pushing:



```
File Edit Sketch Tools Help
Upload
b6_install_1_0
/*****
All Rights Reserved Basic6, Inc Westport, Ct USA
This installs B6 technology

Please specify your user_name and device_token in these variable
1. g_str_user_name="B6"
2. g_str_device_token="064e85b1-4e77-4a93-8482-f9fa9cfec394"
*****/
#include <FileIO.h>
#include <Process.h>
#include <Console.h>
#include <Bridge.h>

//Replace user_name with yours
String g_str_user_name="My name" ;

//Replace device token with yours
String g_str_device_token= "ec39ff77-fdbb-49d9-ba3d-8184d6f28cef"
```

Credential Entry

1. Before sketch *Upload*, adjust the values for **user_name** and **device_token** to match your B6 provided credentials. The device token will be registered against all your connected devices and for security reasons is removed in favor of a device id that will be linked to your user account. If the Basic6 supplied device token is misplaced, a replacement can be regenerated by accessing the Device Token tab within the user's Basic6 Account page.

```
b6_install_agent_1_0 | Arduino 1.5.6-r2
File Edit Sketch Tools Help
b6_install_agent_1_0 $
#include <FileIO.h>
#include <Process.h>
#include <Console.h>
#include <Bridge.h>

//Replace user_name with yours
String g_str_user_name="MyName" ;

//Replace device token with yours
String g_str_device_token= "ABCDEF0c-c9d4-4e83-956f-f66eb438a8ea" ;

int g_install_flag = 0;

void setup() {
  Bridge.begin();
  Console.begin();
  while (!Console);
  Console.println("Installing Agent...");
  FileSystem.begin();
  delay(1000);
}

void loop()
{
```

Serial Monitor

When you are ready for the sketch to continue, open the IDE's Serial Monitor. (The Serial Monitor is available in the Tools menu or within the IDE by clicking on the looking glass icon).



The Serial Monitor output will inform you as to whether the install was Successful.

Go back to your Basic6 account (www.iot.basic6.com), refresh; your device should be available within the device container display on the left side of the screen.

To install B6 technology on an Arduino Yun Shield:

Dragino Yun Shield

In order to upload a program to the Yun shield, it must be compiled specifically for the Yun shield. If you try to push any of the existing sketches from our UI onto the Yun shield, the sketch will not work properly.

The sketches must be recompiled with the board type of "Arduinio Uno — Dragino Yun" instead of Arduino Uno or Arduino Yun. This is done by verifying the source code using the Arduino IDE and uploading the new hex file (made specifically for the Yun shield) to the Basic6 Web Console. Please see page 11 of the following guide (for instructions on how to add the "Arduino Uno — Dragino Yun" board type to the Arduino IDE.)

<http://www.instructables.com/files/orig/F73/ZMFZ/HXHR1BL5/F73ZMFZHXHR1BL5.pdf>

http://wiki.dragino.com/index.php?title=Yun_Shield#USE_WITH_ARDUINO_BOARDS

Website Navigation

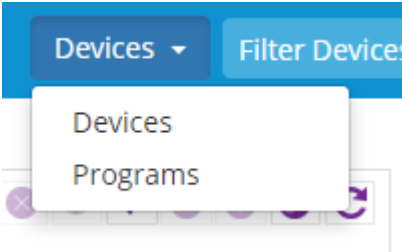
Headers

1. Within the Basic6 website (<https://iot.basic6.com>) you should see the following header on the home page after a successful login.

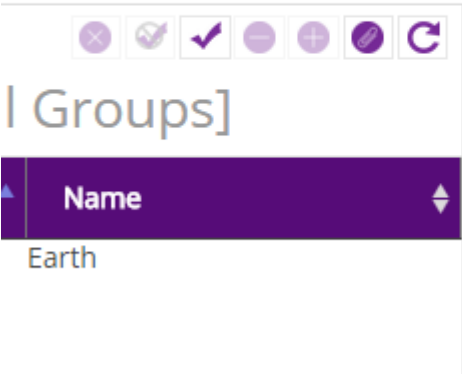


The center portion of the home page header is broken down into three sections, *View*, *Filter Devices*, and *Groups*:

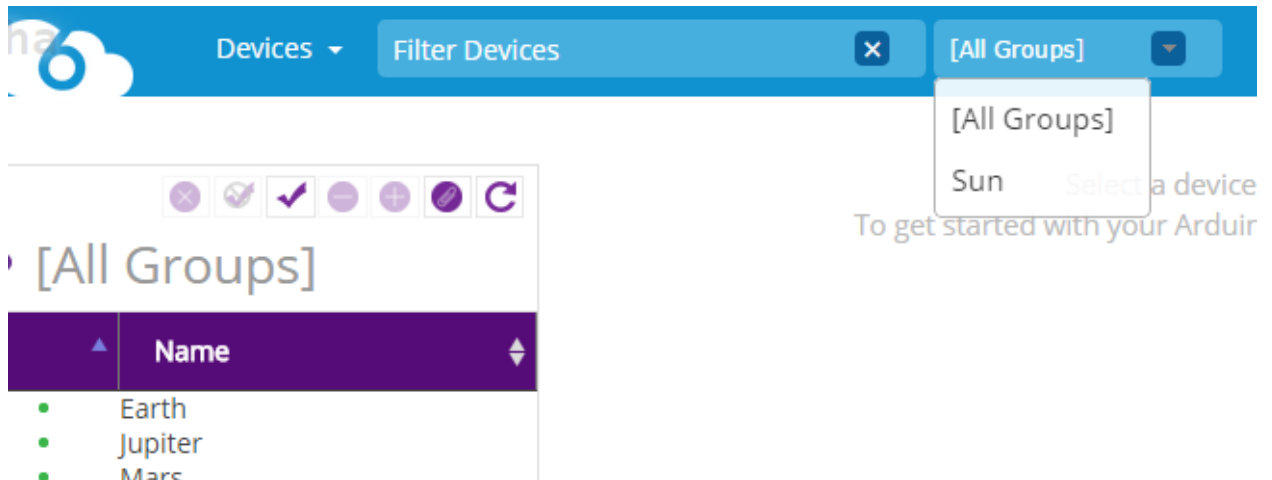
2. The left 'View' dropdown allows you to change between the "Devices" or the "Programs" view. By default, the Devices view is selected. The Programs view allows you to see your library of saved programs that you can use to push to your devices.



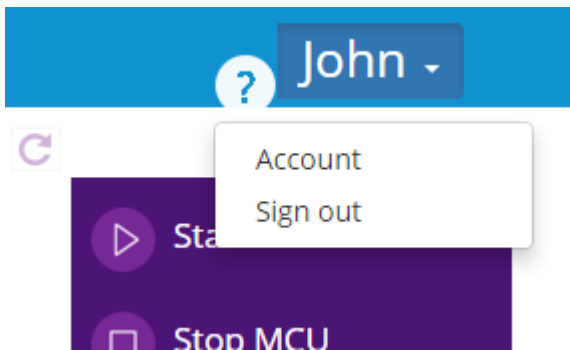
3. Next to the view dropdown is the "Filter Devices" search box. This filter box allows you to quickly find the devices you are searching for by name (in your current selected group). If you enter your filter criteria in the input area, you should see your device list update automatically on the device list to the lower left. See picture below:



- To the right of the Filter Devices input is a Combo box. You can use this combo box to quickly find a group that you have created. Selecting your new group will automatically change your group selection in the device container. Please note that the bracketed "[All Groups]" selection is the default selected group. As depicted below, this corresponds with the device container on the lower left.



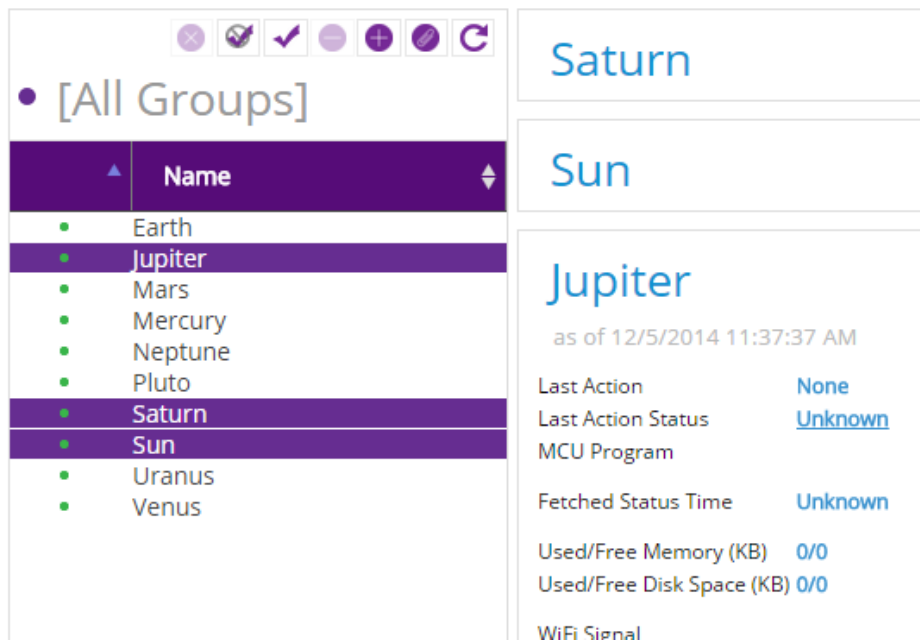
- Finally to the right of the header, you will see your name. Click on your name and you will be presented with a couple of options where you can either make updates to your account (which is discussed in further detail in the Accounts section of this manual) or Sign Out of your account.



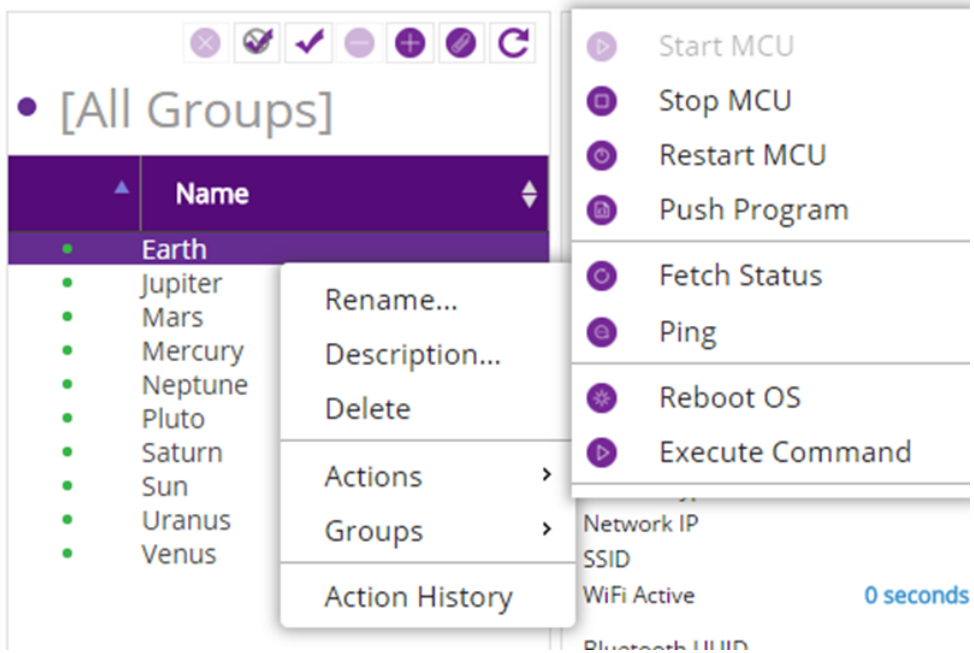
Device Container / Groups

When you log into Basic6, you will see your device list on the left hand side. This area is the Device Container. Your devices are managed by putting those devices in a "group". The icons on the top of the container allow you to manage your groups and allows you to add and remove both devices and groups. By default, all your devices are stored in '[All Groups]'. This is not a group but a list of all your device (You cannot delete this.) See below for details on how to create a group.

- Whenever you select any of your devices, your device will appear in the center display panel. Clicking on subsequent devices will cause the previous device panel to minimize. See below.



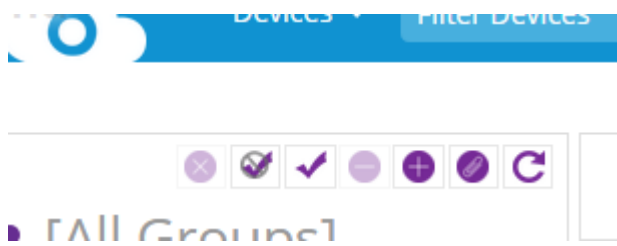
2. The device container also has context menus that allow you to operate on individual devices by right clicking on that device.
 - *Rename* allows you to rename your device as displayed within the B6Launch console.
 - *Description* allows you to attach descriptions/notes to your device.
 - *Delete* allows you to delete your device. (Please note that your device will be deleted from the B6Launch Platform if this action is selected).
 - *Actions* are shortcuts to controls. This is explained in the next section.
 - *Groups* allow you add or remove a device from the group.
 - *Action History* allows you to view the status history for the device.



3. On the top of the Device Container are buttons that allow you to manage your devices or groups. From left to right, the buttons are:

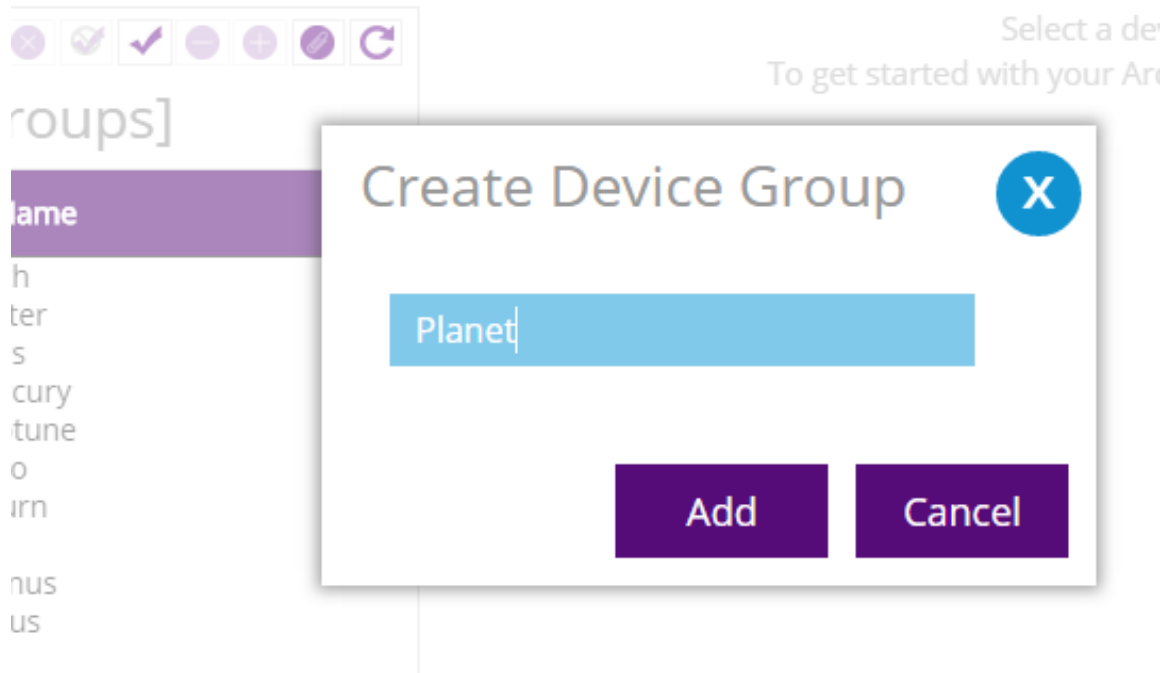
- Delete Group
- Deselect All - (devices)
- Select All - (devices)
- Remove Device(s) from Group
- Add Device(s) to Group
- Create Device Group (Create a new group)
- Refresh to find new devices

Note that removing groups or devices from groups will not delete devices from your account.

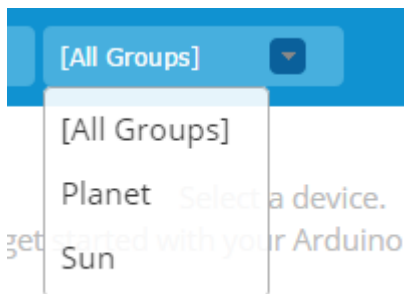


Creating a Group

1. To create a group, click on the paper clip icon. You should see a popup that will ask for the new group name.



2. Once you have created the group, you can add devices to that group using the '+' icon. To navigate to the new group, use the Groups combo box at the header.



3. After you have selected your group, you will see the list of devices associated with that group. You can then select the devices from the group to control, by simply clicking on them. You can also select the Select All 'check mark' icon to control all devices in the same manner.

Controls allow you to perform an action on any number of selected devices. For example, in the picture below, there are four devices selected, Earth, Mars, Jupiter, and Mercury. On the right hand side of the selected devices are the controls. You can click on one of the controls and that action will be performed on all of the selected devices in the center panel. You can carry out this action on an unlimited number of connected devices.

The screenshot shows a web console interface. On the left, there is a panel titled "Planet" with a toolbar containing icons for delete, refresh, check, uncheck, add, edit, and refresh. Below the toolbar is a table with a "Name" column and a list of planets. The first four planets (Earth, Jupiter, Mars, Mercury) are selected, indicated by a green dot in a purple bar. The details panel on the right shows the selected planet "Neptune" and its status information.

Name
• Earth
• Jupiter
• Mars
• Mercury
• Neptune
• Pluto
• Saturn
• Uranus
• Venus

Jupiter

Earth

Mars

Neptune

as of 12/5/2014 12:09:51 PM

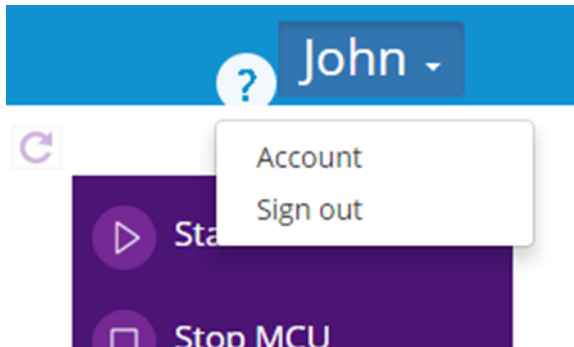
Last Action **None**

Last Action Status **Unknown**

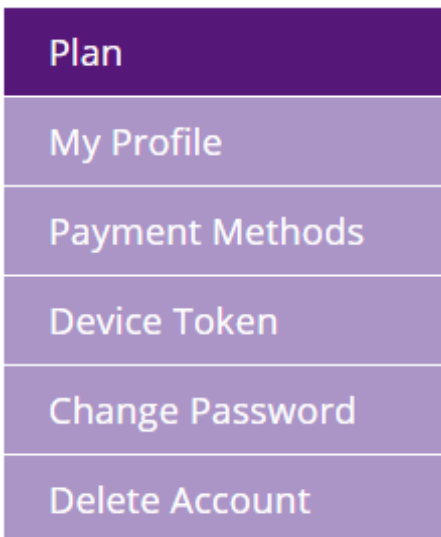
MCU Program

Account

The 'Account' section of the Basic6 website allows you to manage details pertaining to your user account. As discussed earlier, you can access this section of the site by clicking on your name on the upper right hand side, select 'Account' from the drop down listing, as seen below.



Once in the Account section, a user will have access to several new action items related to their account, these are displayed on the left side of the screen, and include access to *Plan*, *My Profile*, *Payment Methods*, *Device Token*, *Change Password*, *Delete Account*.



Plan

By default, all new user accounts to the B6Launch platform start with the Basic6 Maker plan. This entails the ability to connect up to 5 devices with up to 1 GB of data transfer. This Plan will be listed at the top of the screen under 'My Current Monthly Plan' heading. You will also see that this pre-selected plan is 'grayed out' in the available plan offerings. The Basic6 Maker Plan does not require any payments.

My Current Monthly Plan

MAKER

- 5 Devices
- 1GB Data Transfer

Plan update

If you find that you need to add more devices or data transfer than allowed in your current plan, you can upgrade the Maximum devices in the plan section of your account. To "Switch" plans, simply select another plan from the available listing and select 'Upgrade' and follow the steps for payment. The Plans spell out cost per month data usage and allowed # of managed devices. Initial upgrade will require entry of credit card information and you will be notified via email of the successful Plan change.

<p>MAKER</p> <p>Free</p> <p>\$0.00</p> <p>per month</p> <ul style="list-style-type: none">• 5 Devices• 16GB Data Transfer	<p>SMALL</p> <p>Small</p> <p>\$7.99</p> <p>per month</p> <ul style="list-style-type: none">• 30 Devices• 36GB Data Transfer	<p>CUSTOM</p> <p>Contact us</p> <p>?</p> <p>per month</p> <ul style="list-style-type: none">• ? Devices• ?GB Data Transfer
<p>SMALL BUSINESS</p> <p>Small Business</p> <p>\$24.99</p> <p>per month</p> <ul style="list-style-type: none">• 55 Devices• 116GB Data Transfer	<p>BUSINESS</p> <p>Business</p> <p>\$99.99</p> <p>per month</p> <ul style="list-style-type: none">• 225 Devices• 456GB Data Transfer	

Downgrade a Plan

Within the Plans section, select the lesser Plan followed by a 'Downgrade' action. The Cancel Current Plan button in the lower right of the screen will become active. You will be notified via email of the successful Plan change.

If a user downgrades a plan to Maker, they will ONLY have access to the first five (5) devices that were configured to the account.

My Profile

The 'My Profile' section allows for the editing of any and all user account field entries. These fields are set during the initial account signup.

Payment Methods

Adding a Payment Method

After the selection of an upgraded device plan, a user will be presented with a *New Payment Method* button. Simply follow the steps to register your card. Basic6 does not store any card information as card payments are managed through authorize.net

Adding an Additional Card Payment

Within the Payment Methods tab, simply click on the Add button on the lower right side of the screen and follow the steps for card entry.

Setting Default Payment

Within the listing of all available credit cards, Simply click on the card you would like to assign default status to and select the *Make Default* button.


Deleting a Payment Method

Within the listing of all available credit cards, Simply click on the card you would like to delete – after selection, select the Delete button.

Billing

If you are on the Maker Plan you will not be billed. All other plans are billed monthly on the first day of the month. All billing is on a prorated basis – meaning that if you were to sign up June 15th – you would only be billed for the 15 remaining days in the month.

Payment Methods

Default	Card	Expiration
	XXXX-1000	

[Make Default](#) [Add](#) [Delete](#)

Device token

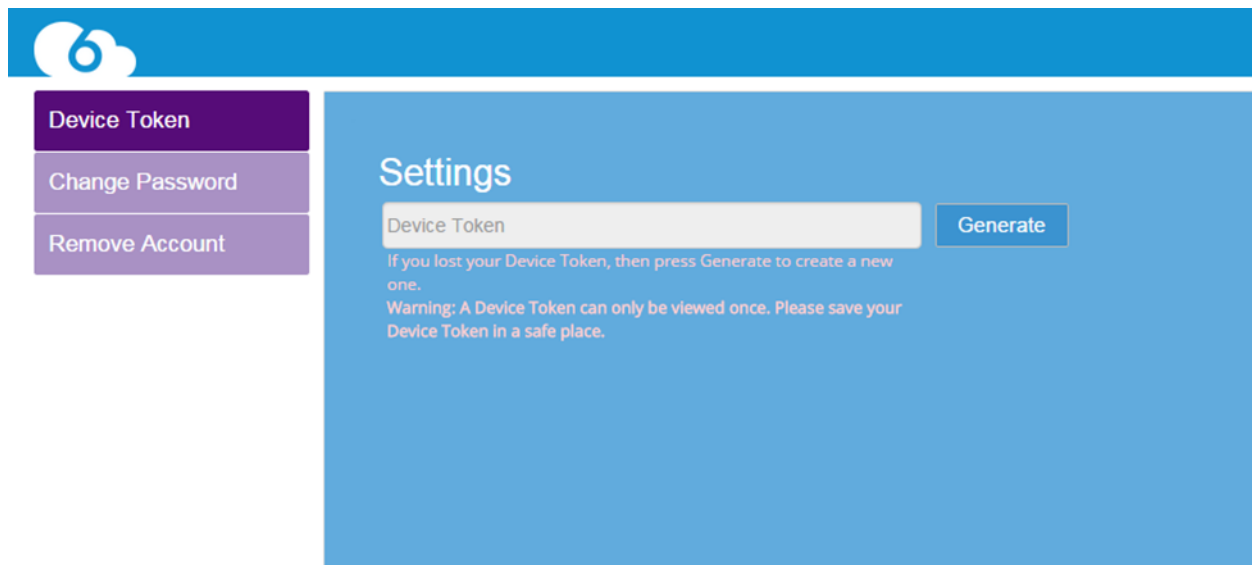
After signup, every Basic6 user account receives a single, unique *Device Token* that is used for authentication across all connected devices. A device is initially registered via a combination of Basic6 Account credentials and device token. Once the process has completed, the device token is removed in favor of a device id, with no trace of the Device Token remaining on the individual devices.

The device token can be quickly changed if there is any fear of compromise.

How do I replace it?

If the need arises, a user can easily generate a unique replacement device token. Within the 'Device Token' tab, selecting the 'Generate' button on the lower right side of the screen will enable the creation of a new unique Device Token.

Successfully connected devices DO NOT require device token update(s).



Change Password

The Change Password action enables a user to quickly change/edit their B6Launch Platform login credentials. Simply enter current (Old) password, followed by entry of new and confirming Passwords. Click Save to process.

Change Password



The form consists of three input fields stacked vertically. Each field has a placeholder text and a red error message below it. To the right of the fields are two buttons: 'Clear' and 'Save'.

Old Password
This field cannot be blank.

New Password
This field cannot be blank.

Confirm Password
This field cannot be blank.

Clear Save

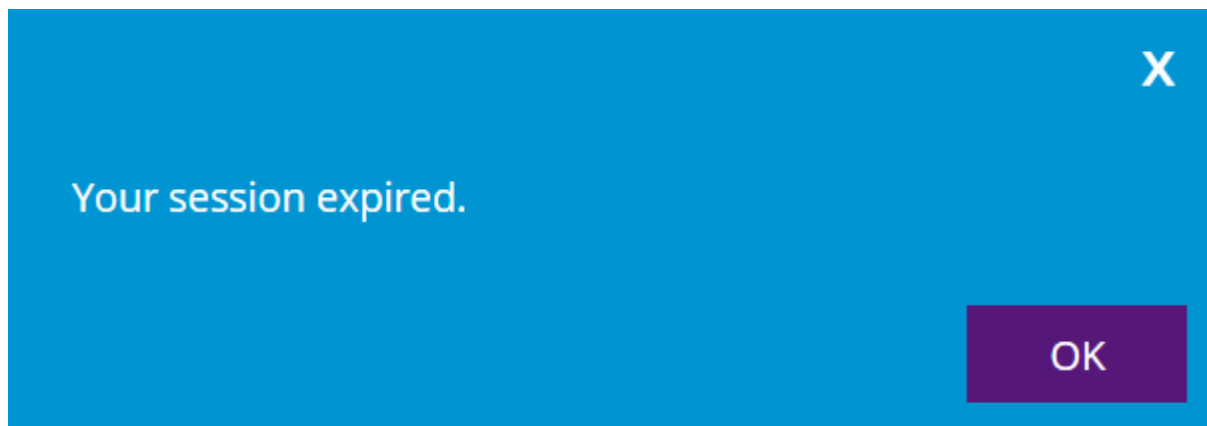
Delete Account

On the left side of the Account screen select the *Delete Account* action. Click Yes to the right of “Are you sure you want to delete your account?” message. You will be prompted twice for security purposes.

Please note that all device connectivity and login credentialing will be deleted from the B6Launch Platform. To re-enable connectivity to the platform – a user must enter new credentialing (user id and password).

Timeout

After a period of 30 minutes of account inactivity, a user will be prompted if they wish for their account to remain active. If the user does not respond to this messaging, the session will automatically be shut down.

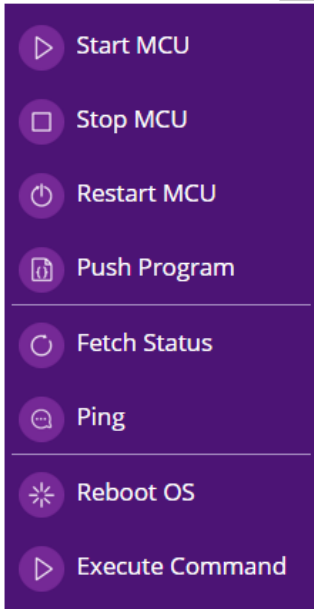


Controls/Functionality

When you install the Basic6 agent, your device makes a persistent connection to the Basic6 network. This allows you to remotely control your device using the B6Launch Platform at all times. The B6Launch platform, however, does not make a direct connection to your device. Instead, the platform connects to our infrastructure and queues commands that are then forwarded to your device. As a result, when you send a command to your device, you will need to refresh to get the latest status of your device.

List of Device Controls

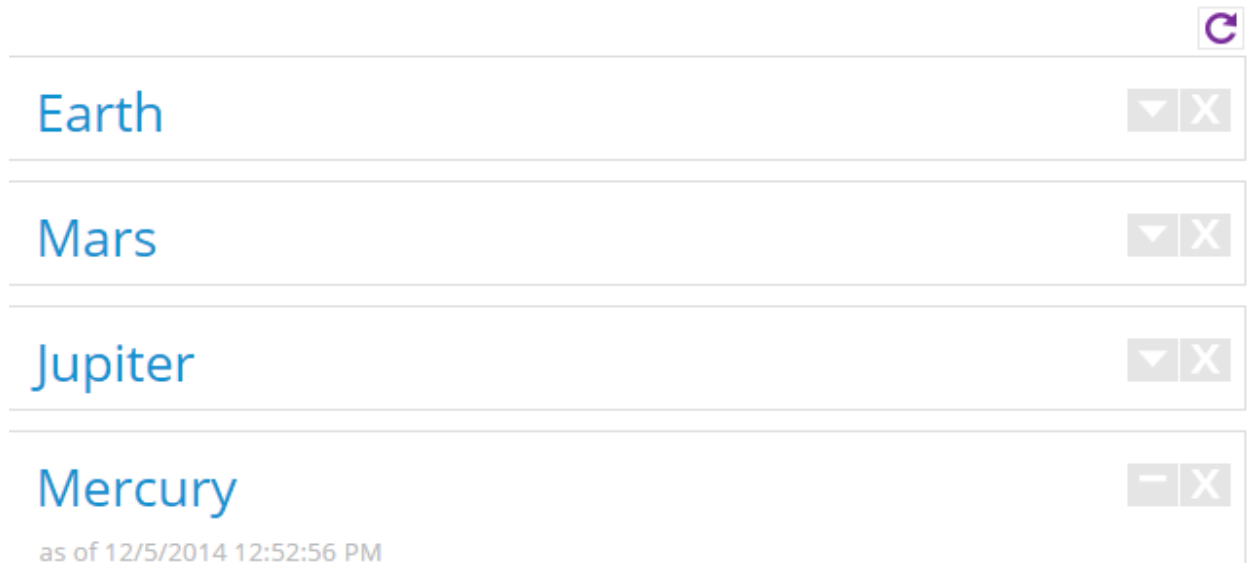
- Start MCU - send a Start Micro-Controller command to the device.
- Stop MCU - send a Stop Micro-Controller command to the device.
- Restart MCU - Send a Restart Micro-Controller command to the device.
- Push Program - Push a sketch/computer program to the device.
- Fetch Status - Get the current state of the devices from the device.
- Ping - Ping the device to ensure device is active.
- Reboot OS - Send reboot command to the device.
- Execute Command - Send your own custom command to the device.



This functionality is found in one of two places, on the right side of the main display screen, or as a function of 'Right-Clicking' on an individual device and selecting 'Actions'.

Latest Status

To get the latest status of your devices after you have sent your command, click on the refresh icon on the upper right of your selected device.



To view the status of your device, click on the caret to the right of the device name (See above). This will expand the device window to show you the details for that device.

Main Display Window

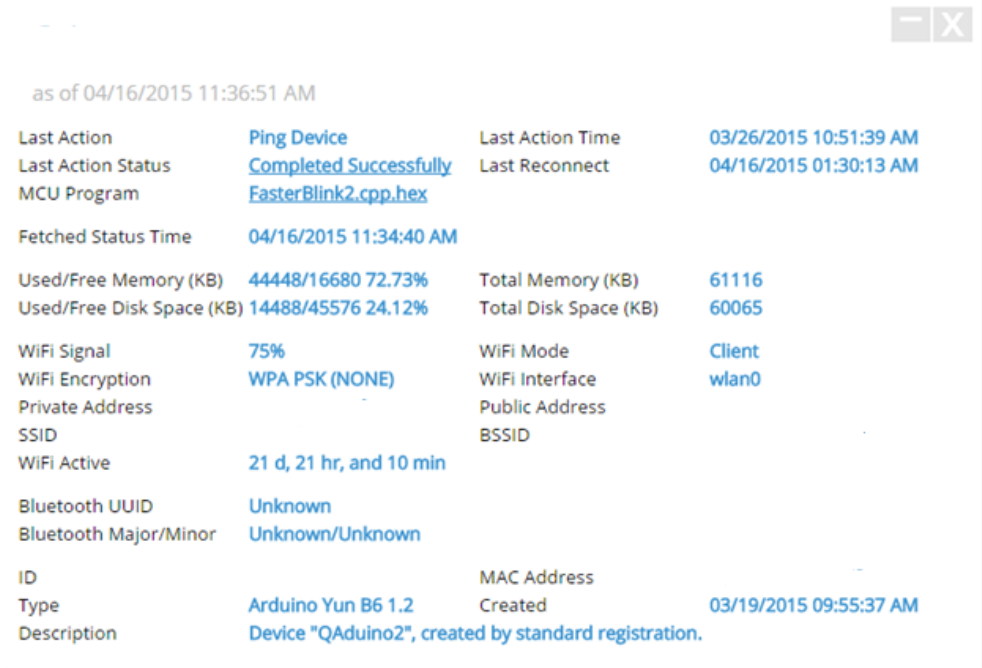
Below is an edited depiction of an individual device's display. Within the main screen, a user will be able to see:

Last Actions, Last Action Time, Last Action Status (Success/Fail), Last Reconnect (to the B6 Platform), and last MCU Program run. Clicking on the provided link next to 'Last Action Status', will result in a popup showing a log for the device.

Usage and status checks are run on Memory and Disk Space

WiFi and network connectivity is captured and displayed

B6 displays the device's ID, MAC address, Type, Creation date, and Description.



Push Program

The "Push Program" control allows you to upload and send microcontroller programs (or "sketches") to your selected devices. When you press the "Push Program" control, you will be presented with a popup dialog that will display your library of programs. You can add or delete programs from your library by using the "Add" and "Delete" buttons.

When opened, the Programs display presents the user with two pre-installed sketch options. These sketches can be executed across any connected device(s).

Name	Date Created	Public	Select a Program
Blink.cpp.hex	01/01/2015 12:00:00 AM	✓	
Pink_Panther_Tune.cpp.hex	01/01/2015 12:00:00 AM	✓	

Adding a Sketch

If you click on the "Add" button, another dialog will show that will allow you to add additional .Hex files or scripts to your library.

Uploading a Hex File

Begin by selecting the Hex File button within the left most display.



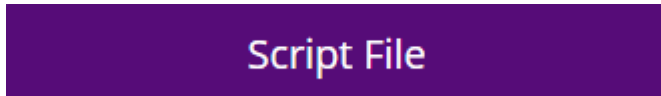
This will open a file browser. Choose the target .hex file. (.Hex file folder creation is discussed in STEP 1. ARDUINO YUN CONFIGURATION)

The selected .hex file will appear in the Program display. The light blue box within the interface allows for the entry of a Program description (this can be edited later).

Uploading a Sketch

At this time you can also add the corresponding Script file (.ino) which will help you keep track of which sketch is running on what device.

Select the Script button.



Continue the process following the same steps used previously in the .hex file configuration.

Once completed, the corresponding Script file will appear in the right side of the display, Click Upload to process.

The completed Program/Sketch is now saved in your account's Library with a Date Created timestamp.

Please note that clicking on the "Upload" button will not push your script to your devices. Instead, the program will be saved to your account Library.

Edit a Program

As discussed earlier, the Edit button allows for changing of a Program's description. Click "Save" when any edits are completed. A Sketch/Program cannot be edited.

Pushing a Program

After you have uploaded your scripts, you will need to go back to the device container display. Select the device(s) that you wish to execute against, select the Push Program action, pick a script from your library and then click the "Push" button. This will send your program to all your selected devices.

Back in the display, the target device(s) Last Action will show that a Push Program has run, the Last Action Status will show the action's status, and the MCU Program will show the selected sketch. The Last Action Time will display time stamp of the action.

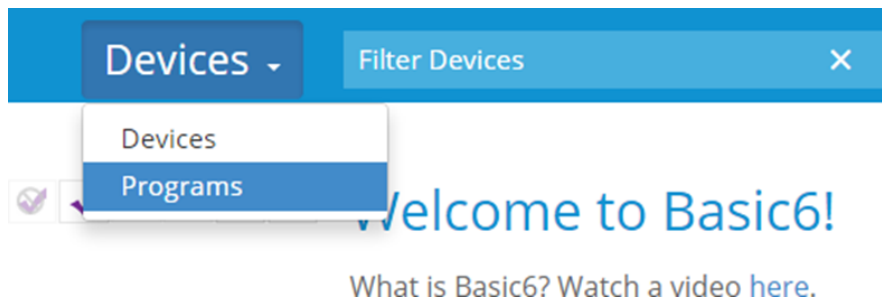
Clicking on the provided link in the Action History field you will see a columned listing of all executed actions with Status and time stamp information. An individual action can be clicked upon to show more information.

Program History

You can see the output of issued Programs and commands by "right clicking" on a device and selecting Action History. You can also click on the provided "Last Action Status" link provided in the device details display.

Accessing the Program Library

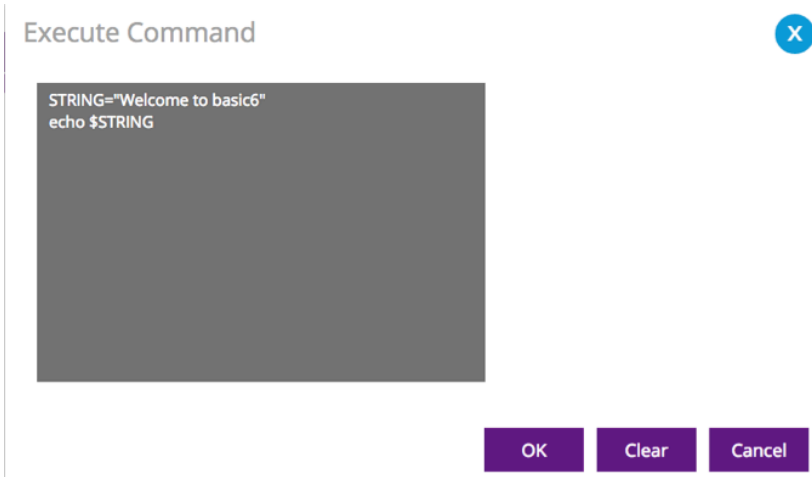
Once a Sketch has been created and saved, you can quickly access your library of programs via a shortcut link. On the header, next to the Filter Devices search box, there is a drop-down that allows you to change between the current Device view or Programs view. Simply select one or the other to navigate between the two views.



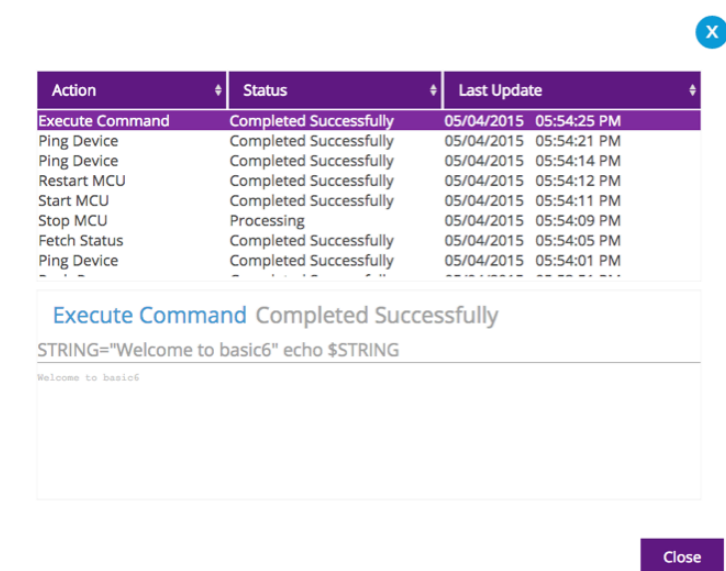
Execute Command

The Execute Command control allows you to execute Linux commands directly on your devices, and is particularly useful in most deployment scenarios when it will simply not be possible to establish an SSH connection. Using this feature, any number of devices can be managed.

Simply type in your intended command and then click the 'OK' button.



Simply type in your command and then click the 'OK' button. All output is captured and readable within each device's Action History.



References and Resources

<http://arduino.cc/en/Tutorial/HomePage> - Guides on taking your first steps with programming Arduinos

<http://playground.arduino.cc/> - More advanced ideas and guides on programming Arduinos

<http://arduino.cc/en/Reference/HomePage> - An overview of how to use commands in Arduino Programming

<http://arduino.cc/en/Guide/Environment> - How to use the Arduino IDE

Support@basic6.com – Email for any Web Interface related issues.

1. Tips and Tricks

- a. Start small, while learning how to create Sketches it is best to start with a single LED or sensor and move up; it is easier to develop an understanding of how an Arduino works from simple code.
- b. Don't reinvent the wheel. Sketches exist for many Arduino sensors; consider adapting published code, especially for simple operations.
- c. Use Temboo (<http://temboo.com>) or a similar service to create sketches without having to create your own code
- d. In testing, push code from the Arduino IDE before uploading to Basic6. It is a huge time saver if you are working with only one Arduino and are unsure if you sketch actually works. Bring in Basic6 when it is time to mass deploy your code to all of your Arduinos.
- e. During testing have an LED on the device tied to any data collection the device is doing - even if you are monitoring the results on your computer. It's always nice to have visual feedback.