

Lab 2: The Full IoT Pathway

Building Sensor-Controlled Mood Lights

Overview

This lab walks through the full IoT pathway while also providing exposure to key pieces of an IoT product such as the hardware application (sensor, actuators, processing, and transceivers), the mobile application, and the cloud application (frontend display, data processing and storage).

In this lab you will build 3 devices of escalating complexity:

1. You will start by build a smart device – a device that uses embedded logic to respond dynamically to its environment
2. Then you will build a connected device – a device that can be controlled via a remote software interface
3. You will end by building an IoT device – a physical device that is directly connected to the internet

Specifically, in this lab you will be building a mood light that has its color and brightness controlled by its surrounding environment.

Hardware

- Arduino Mega 2560 with Mega-B Bread™Board shield
- 3x Breadware LED modules
- 1x Breadware Accelerometer Module
- 1x Breadware Brightness Module
- 1x Breadware Temperature Module
- USB 2.0 cable Type A Male to Type B Male
- An iOS device to use the Bread™Connect app
- A computer to use the online Breadware IDE and the Breadware Flasher

Breadware IDE

- Visit <https://dev.breadware.com>
- Register for a free student Breadware account
- Log in to create, save, and edit your projects as well as view your dashboards.

Download the Bread™Connect App Software

Currently only supported on iOS. You can find the download here.

<https://itunes.apple.com/us/app/bread-connect/id1172516219?mt=8>

Log into the Bread™Connect App with the same account information you used to log into the Breadware IDE.

Download the Breadware Flasher

If the Breadware Flasher is not already installed on your computer, you will need to download it from the Breadware site.

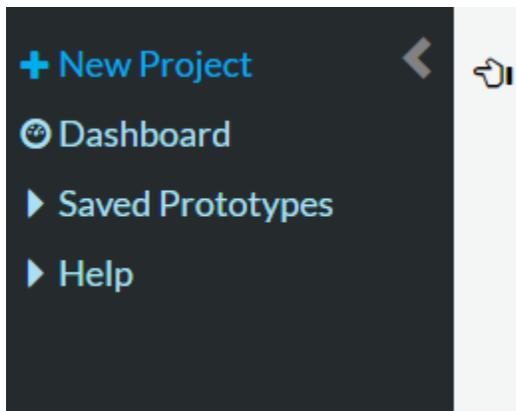
- Visit <https://docs.breadware.com/>
 - Click on “Downloads” in the directory on the left-hand side of the page
 - Download and install the Breadware Flasher for either Windows or macOS
 - Log into the Breadware Flasher with the same account information you used to log into the Breadware IDE.
-

Part One: Building the hardware application

This section covers assembling and programming the hardware for your sensor-controlled mood light. At the end of this section you will have a smart device.

1a. Create a New Project

Open the Breadware IDE at <https://dev.breadware.com> and sign in/register with your account if you have not already done so. Start a new project by expanding the left side navigation bar and clicking “New Project” button.



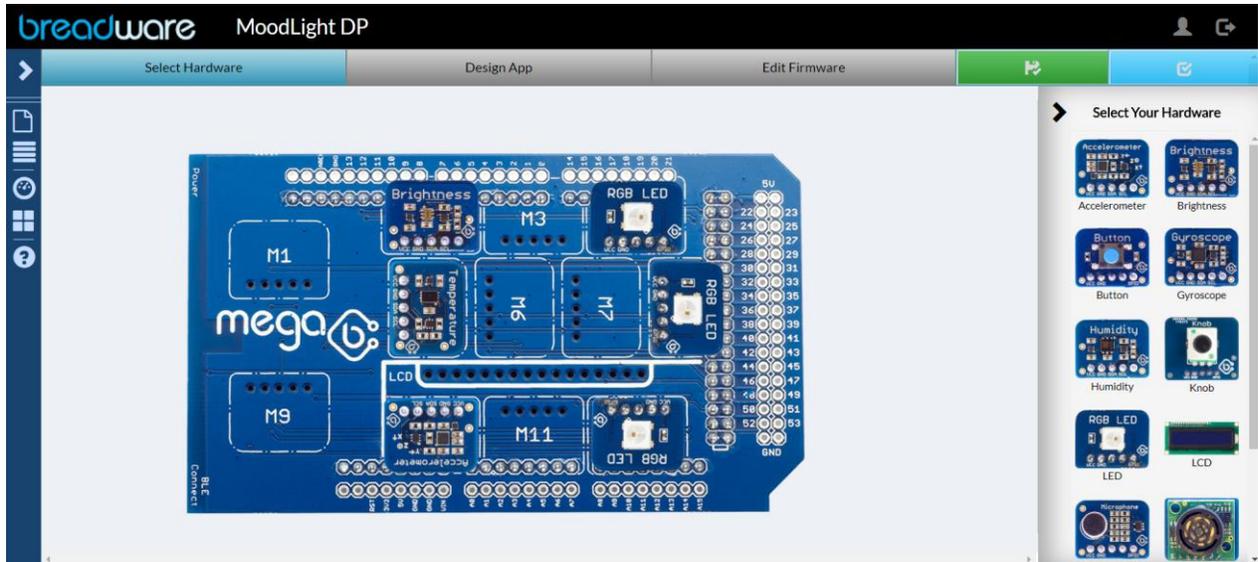
1. Begin by clicking “NewProject” and give your project a name. A name like “MoodLight [your initials]” is recommended to avoid confusion arising from multiple teams having their Mood Light broadcasting with the same project name.
2. Select the type of Bread™Board you are using – for this lab you are using the Mega-B.
3. Select a Blank template to get started

1b. Adding Hardware Using the Hardware Builder

Navigate to your project’s hardware builder by clicking “Select Hardware” in the navigation bar at the top of the page (under your project’s name)

1. Open the hardware module menu on the right side of the page
2. Add the following modules to your project (placement does not matter as long as the locations of your physical modules match the locations that you placed the modules in the Breadware IDE)
 - LED: Named “LED1”
 - LED: Named “LED2”
 - LED: Named “LED3”
 - Microphone: Named “Mic”
 - Brightness: Named “Lux”
 - Temperature: Named “Tmp”

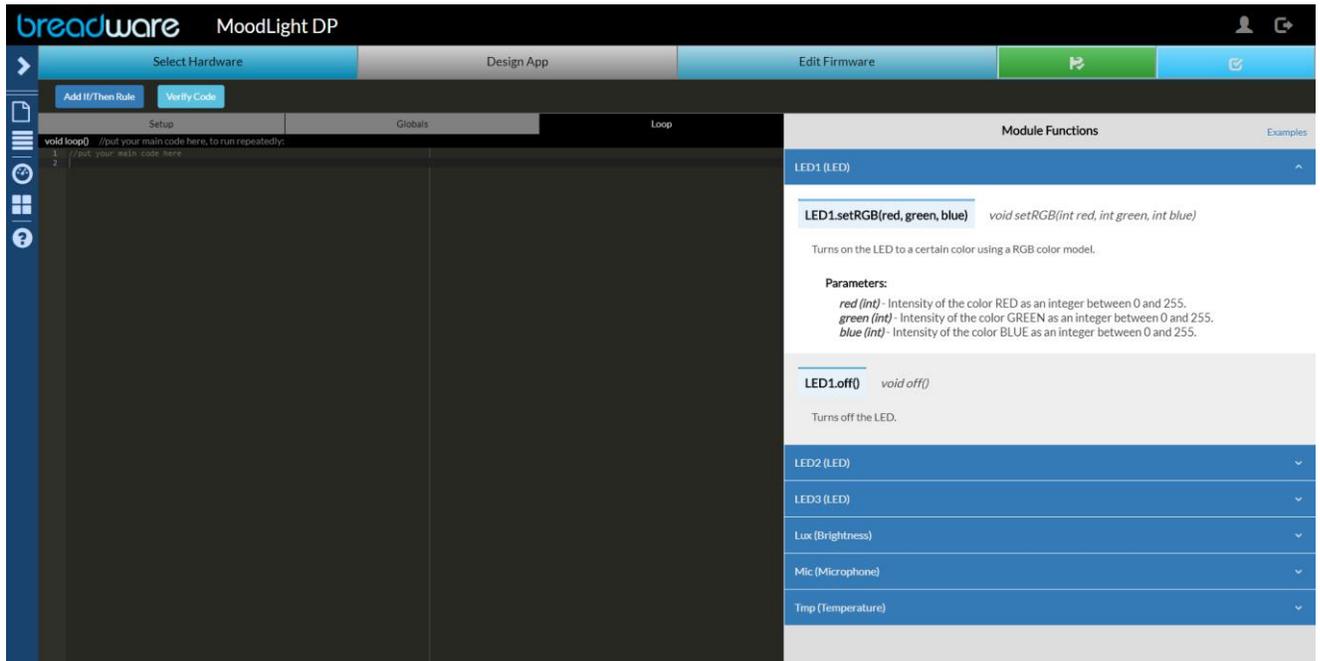
The below image shows the Mega-B in the Breadware IDE implemented with all of the modules required for this project.



1c. Writing Code Using the Firmware Editor

Navigate to your project's firmware editor by clicking "Write Firmware" in the navigation bar at the top of the page. This is where you will begin to write the program that will run on your Bread™ Board.

Here we are going to write the logic that controls the intensity and the color of the 3 LEDs based on the readings from the sensors



Program Logic

The temperature sensor will control the 'red' intensity of the mood light; the brightness sensor will control the 'blue' intensity; and the accelerometer will control the 'green' intensity. The LEDs are controlled by an RGB triplet where each index can range from 0 to 255.

Specifically, we will start with the following logic:

- $\text{Tmp} = 23\text{ }^{\circ}\text{C} \rightarrow \text{R} = 0$ ----- $\text{Tmp} = 33\text{ }^{\circ}\text{C} \rightarrow \text{R} = 255$
- $\text{Acc} = +10\text{ m/s}^2 \rightarrow \text{G} = 0$ ----- $\text{Mic} = -10\text{ m/s}^2 \rightarrow \text{G} = 255$
- $\text{Lux} = 10 \rightarrow \text{B} = 0$ ----- $\text{Lux} = 10,000 \rightarrow \text{B} = 255$

Start by copying and pasting the following logic into the code editor

Note Ensure that the module names are set according to the naming shown section **1B**, if the modules have different names you will receive errors when you try to compile/verify the code

Copy the following code snippet into the "Globals" tab

```
// These floats will store the sensor readings
float TmpReading;
float LuxReading;
float AccReading;

// These ints will set the LED RGB values
int R;
int G;
int B;
```

Copy the following code snippet into the "Loop" tab

```
//put your main code here
TmpReading = Tmp.read();
LuxReading = Lux.read();
AccReading = Acc.read_z();

// Convert temperature sensor reading to value
between 0 and 255
R = (int) ((TmpReading-23)*(255.0/(33.0-23.0)));
R = max(0,R);
R = min(255,R);

// Convert Microphone sensor reading to a value
between 0 and 255
G = (int) ((-AccReading+10)*(255.0/(10+10)));
G = max(0,G);
G = min(255,G);

// Convert brightness sensor reading to a value
between 0 and 255
B = (int) ((LuxReading-10)*(255.0/(5000.0-10.0)));
B = max(0,B);
B = min(255,B);

LED1.setRGB(R,G,B);
LED2.setRGB(R,G,B);
LED3.setRGB(R,G,B);
```

1d. Flashing Code onto a Bread™Board

Flash your project onto your Mega-B Bread™Board. You will need to have the Breadware Flasher installed on your computer. For a step-by-step on how to flash code see https://docs.breadware.com/flashing_firmware/index.html.

The steps for flashing the code are as follows:

1. First save your project
2. Hit verify code to run a compiler verification of your code (you will need to fix any errors that arise before you can flash the project)
3. Open the Breadware Flasher desktop application and log in with your account credentials
4. Connect the Mega-B Bread™Board to a USB port on your machine
5. Select the project you want to flash
6. Click “Flash Firmware”. This process will take a few seconds

You have now built a smart mood light that has its brightness and color regulated by its environment. Experiment with your mood light by varying the inputs to the sensors (e.g. applying heat to the temperature sensor by touching it with your finger, increasing/decreasing the light incident upon the brightness sensor, tilting the device to adjust the accelerometer reading).

The next step will be adding connectivity to your mood light to add greater functionality.

Part Two: Mobile App Controller for the Mood Light

This section covers creating and programming a mobile app to add more functionality to your mood light. At the end of this section you will have a connected device.

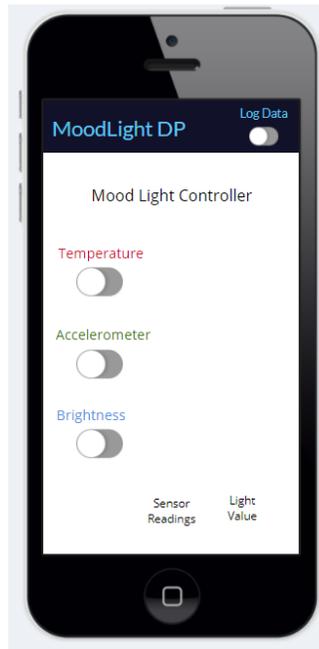
2a. Using the Mobile App Builder

Navigate to your project’s mobile app builder by clicking “Design App” in the navigation bar at the top of the page (under your project’s name)

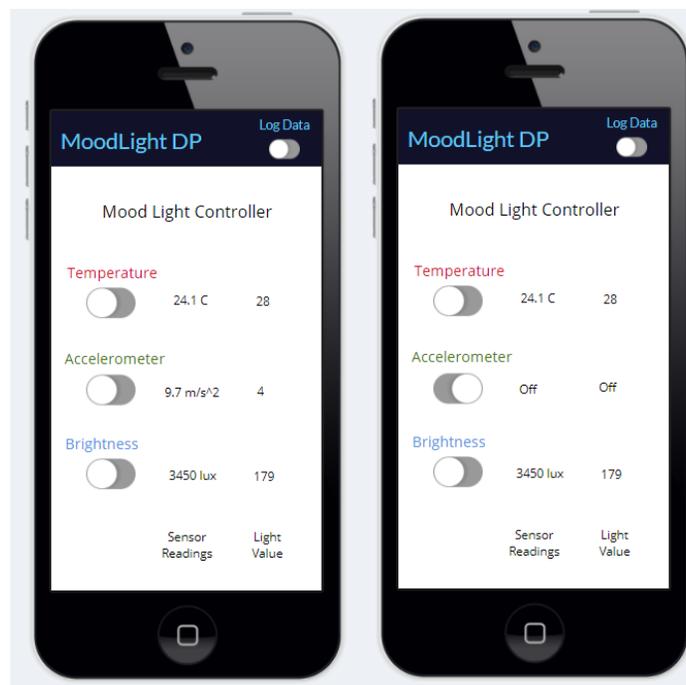
1. Open the mobile module menu on the right side of the page. Notice that there are 3 tabs covering 3 categories of elements that you may add to your mobile app – “General”, “Interactive”, and “Alerts”. In this section we will be building a simple app with only the following types of elements
 - App Text
 - App Data Display
 - App Switch

If you desire to add more aesthetic design and functionality to your app, you can do it after this section

2. Add static text to the mobile app screen as shown in the image below



3. Next add the following App Switch modules as shown below:
 - App Switch: Named "TmpEnable" (This will be the uppermost switch)
 - App Switch: Named "AccEnable" (This will be the middle switch)
 - App Switch: Named "LuxEnable" (This will be the lowermost switch)
4. Next add the following App Data Display modules as shown below:
 - App Data Display: Named "TmpValue" (Upper left)
 - App Data Display: Named "RedValue" (Upper right)
 - App Data Display: Named "AccValue" (Middle left)
 - App Data Display: Named "GreenValue" (Middle Right)
 - App Data Display: Named "LuxValue" (Bottom left)
 - App Data Display: Named "BlueValue" (Bottom Right)



2a. Adding the Mobile Control Functionality in the Code Editor

You have two goals with your mobile app controller for your mood light

1. Control:

Toggleing any of the switches will enable/disable certain sensors and therefore certain colors in your mood light.

- (a) a switch in the left position enables the corresponding sensor
- (b) a switch in the right position disables the corresponding sensor

2. Data Acquisition:

For all enabled sensors, your app should display both:

- (a) the real-time sensor reading (with appropriate units)
- (b) the real-time color setting (a number ranging between 0-255)

For all disabled sensors, the app should display:

- (a) "Off" in both data displays of the corresponding sensor

Implement the mobile app logic in accordance with the two goals above.

Sending data to mobile app

To write data to the data displays you can use commands similar to the below example for the Lux Data Display

```
String luxString = String(LuxReading) + " lux";  
LuxValue.write(luxString);
```

Note The function calls available for all of the modules used in your project are shown in the drop-down list on the right side of the code editor. For even more detail go to <https://docs.breadware.com/>

Part Three: Web Dashboard and IoT for your Mood Light

This section covers creating a web dashboard to log the data from your Mood Light to a database. This section also starts a discussion of how the collected data may be used to turn your connected device (from Part Two) into an IoT device.

3a. Building a Web Dashboard for Your Mood Light

Breadware dashboards are built from widgets that receive data directly from Breadware Modules. These widgets provide a graphical view of the data readings from one or multiple Breadware devices and also store the data for future analysis.

Note The Mega-B Bread™Board on which you have built your mood light only has BLE (Bluetooth Low Energy), so the data from your Mood Light will reach the web dashboard by travelling through the mobile app as a proxy.

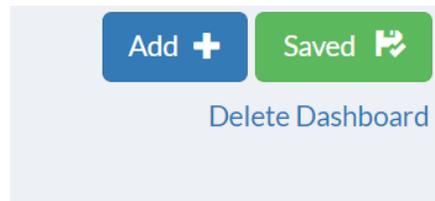
Let's start by creating a dashboard

The left side navigation menu in the Breadware IDE has a dashboard section where you can choose to start a new dashboard. Click on this and name your dashboard "Mood Light Dashboard"

Next let's add some widgets

You are going to add 6 widgets – a real-time data display as well as a graphic display for each of the 3 sensors.

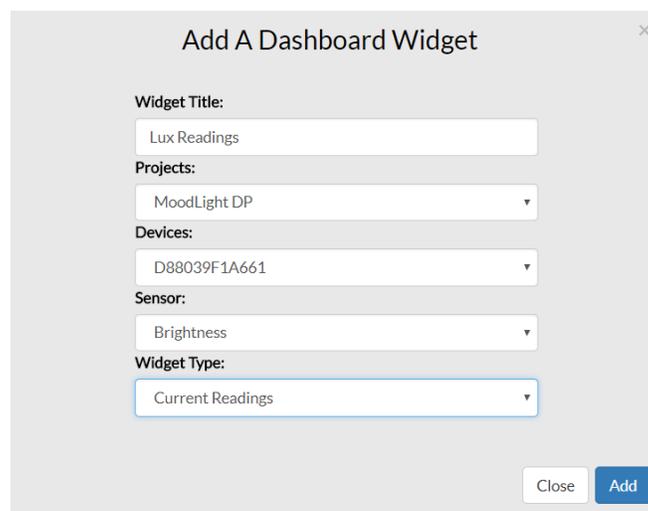
1. To begin adding a widget, click the "Add+" button in the upper right hand corner of the dashboard.



2. Give your new widget a title
3. Select which device you will be collecting data from. You can select device ID directly from the dropdown, or can filter available devices by first selecting the project you want to collect data from

Note In order for a Breadware device to show up in the list, you must first have connected it to the Breadware Connect app. Further, you must log into the Breadware Connect App with the same credentials that you used to log into the Breadware IDE to access the dashboard.

4. Next select what module you want this widget to collect data from
5. Finally, you will need to select how you want this widget to display the data on the dashboard (for this lab you will be using both "Current Readings" and "Spline").

A screenshot of a dialog box titled "Add A Dashboard Widget". The dialog contains several dropdown menus: "Widget Title" with the text "Lux Readings", "Projects" with "MoodLight DP", "Devices" with "D88039F1A661", "Sensor" with "Brightness", and "Widget Type" with "Current Readings". At the bottom right, there are two buttons: "Close" and "Add".

Repeat the above process 6 times to create 2 widgets for each of the 3 sensors on your mood light.

Positioning and Resizing Widgets

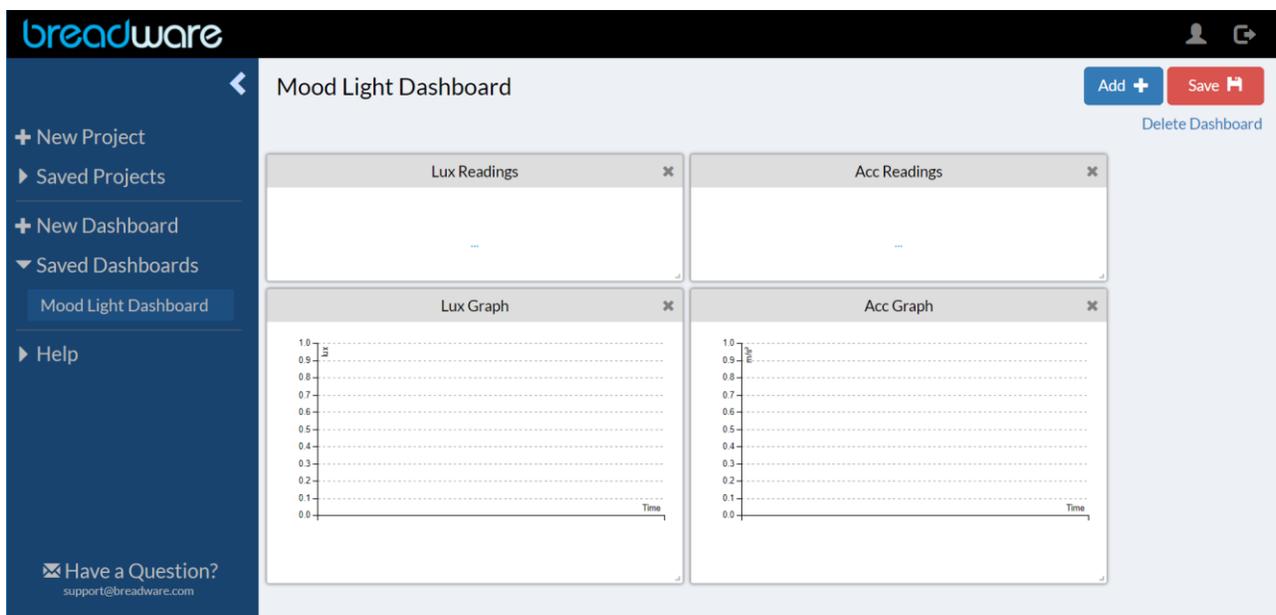
The Dashboard allows for some basic customization and styling of the widgets.

Positioning: Click and hold down the mouse to move widgets around

Resizing: Click and hold down the bottom right hand corner to resize widgets

Saving Dashboard

You can save you dashboard by clicking the save icon in the upper right corner.



3b. Run and Experiment and Then Download the Data

Now that you have your dashboard hooked up it is time to run an experiment.

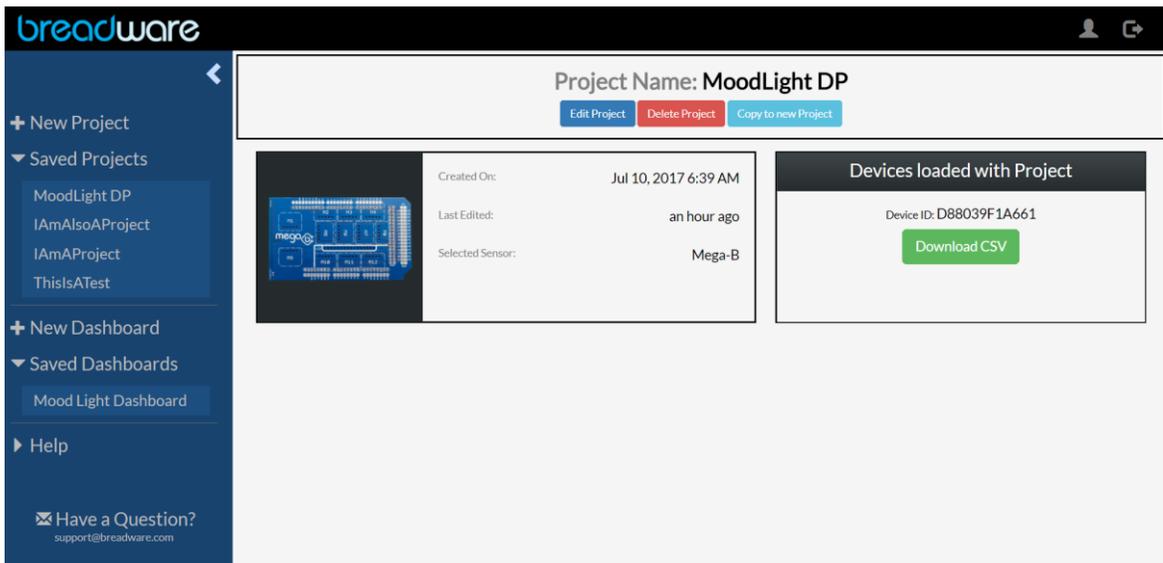
Note the time that you are starting the experiment. We will run an ~60 second experiment.

To start the data collection, connect to your mood light with your Bread™Connect app.

During the 60 seconds manipulate the mood light in a manner such that there will be a recognizable characteristic to your experiment (e.g. cover and uncover the brightness sensor every 5 seconds for the 60 second period)

Downloading Data

After you have recorded data to your dashboard you can download that data into a CSV for further data analysis. To do this follow the below steps:



1. Use the left-hand menu to navigate back to “Saved Projects” and click on your Mood Light project.
2. In the cover page of the Mood Light project you will notice a box near the right displaying devices IDs loaded with this project. Click on the device from which you have been collecting data to download a CSV of that data.
3. A pop up window will appear allowing you to set the time window over which to download data. Set the start and end of the time window appropriately and then hit “download”.
4. In the downloads folder of your PC you can now open your file as a CSV. Open the file and investigate the fields. Make sure you can identify what each of the columns of information mean.
5. Graph the temperature / accelerometer / and brightness data and make sure that the data you see shows the characteristics that you would expect to see based on what you did in the experiment.

Download CSV ✕

Name your CSV File:

Filename:

MoodLight_DP-D88039F1A661

Select a Time Frame:

*We currently only store 7 day's worth of data.

Start:

Mon, Jul 10, 2017 1:10 PM

07/10/2017

▲ ▲ ▲

1 : 10 PM

▼ ▼ ▼

End:

Mon, Jul 10, 2017 1:11 PM

< July 2017 >

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

1:11 PM

Close
Download

3c. Expanding Functionality of the IoT Mood Light – a Discussion

The Internet of Things is just the internet... accessed by a physical device. Just as the internet gives us access to a wealth of information, documentation, and databases at the tips of our fingers, so it can too for a device. This allows for a small device with limited processing power and memory to leverage the full information and computing power of the internet – accomplishing tasks that it never could in isolation.

Finish this project by thinking through and discussing some of the functionalities that could be enabled in your device now that we have implemented internet connectivity.

A few examples could be:

- Instead of the mood light varying based on the environment directly around the mood light, the mood light could be configured to represent information taken from different cities around the planet.
- The mood light could read in weather data and respond predictively to what the conditions are projected to be 1 day in the future.
- The mood light can be hooked into your social media feed and utilize a learning algorithm to try and respond to your mood directly.