

Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice

- [Overview](#)
- [Getting Started: Using the Firefly Ice](#)
- [Getting Started: Using the Parts Kit](#)
- [Getting Started: Using the Development Kit](#)
- [Getting Started: Firmware Development](#)
- [Getting Started: Using the API](#)
- [Block Diagram](#)
- [Electronics](#)
- [FCC](#)

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Search This Site

[Search](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Newsletter

[Subscribe](#)



Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Electronics

The schematic and board design files are in Eagle 6 format in the [firefly-ice-electronics](#) repository on github. The bill of materials is generated from the schematic file using the BOM tool in the [firefly-production-tools](#) repository. For a quick preview, here are PDF versions of some of the design information:

- [Schematic \(PDF\)](#)
- [Bill of Materials \(PDF\)](#)
- [Top Placement Diagram \(PDF\)](#)
- [Bottom Placement Diagram \(PDF\)](#)

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Search This Site

[Search](#)

Newsletter

[Subscribe](#)



Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Getting Started: Firmware Development

Compiling the Firmware

The Firefly firmware uses freely available libraries and tools. To get setup for building the Firefly Ice firmware you will need the following:

- GNU Tools for ARM Embedded Processors
- EnergyMicro CMSIS Software
- Firefly version of the EnergyMicro CMSIS USB Software
- Firefly Ice Firmware

The following commands can be used in the Mac OS X terminal to download and extract all of the above:

```
curl -O https://launchpadlibrarian.net/135590305/gcc-arm-none-eabi-4_7-2013q1-20130313-mac.tar.bz2
tar jxf gcc-arm-none-eabi-4_7-2013q1-20130313-mac.tar.bz2
export PATH=`pwd`/gcc-arm-none-eabi-4_7-2013q1/bin:$PATH
```

```
curl -O http://cdn.energymicro.com/dl/packages/EM_CMSIS_3.0.2.zip
unzip -d energymicro EM_CMSIS_3.0.2.zip
git clone https://github.com/denisbohm/energymicro-usb.git
git clone https://github.com/denisbohm/firefly-ice-firmware.git
cd firefly-ice-firmware
mkdir obj bin
```

Finally, from within the firefly-ice-firmware directory you can build the firmware using make:

```
make
```

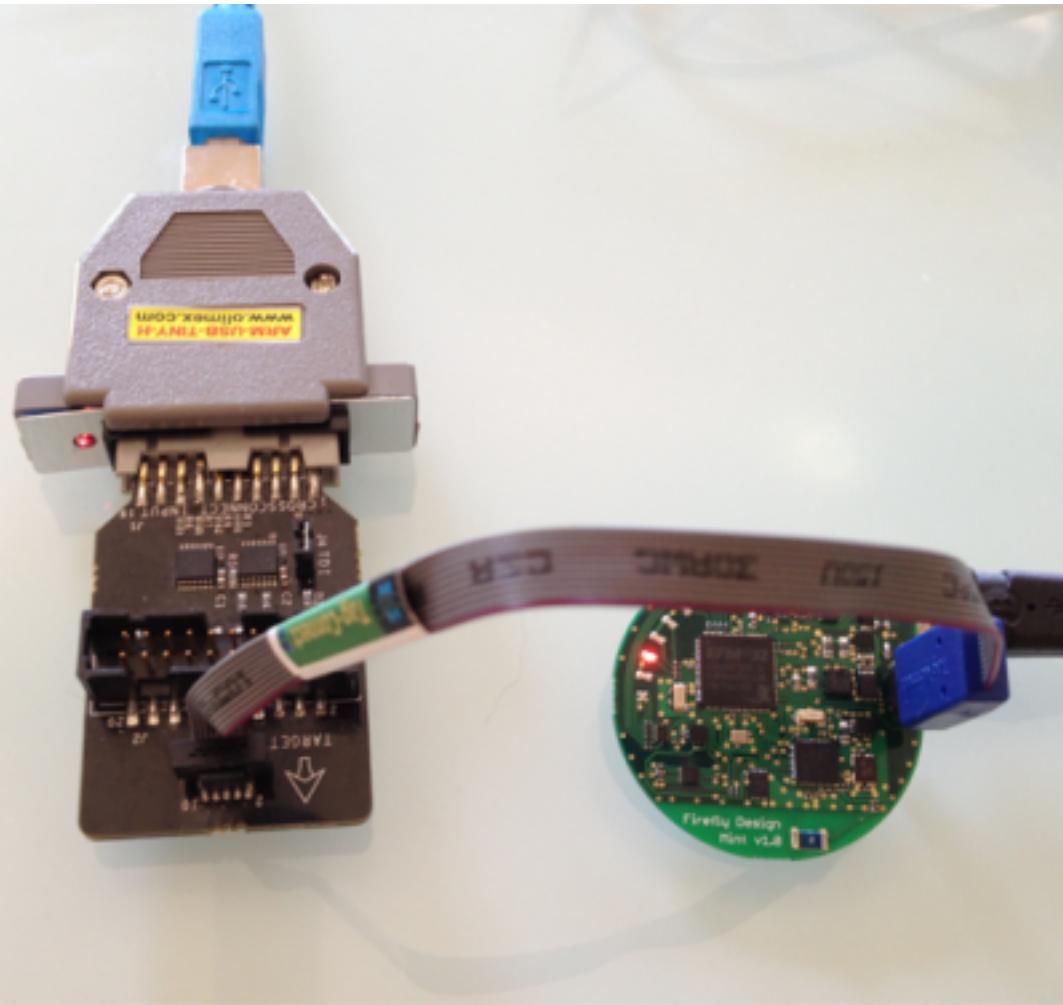
The resulting FireflyIce.elf binary file can be found in the bin directory.

Loading the Firmware

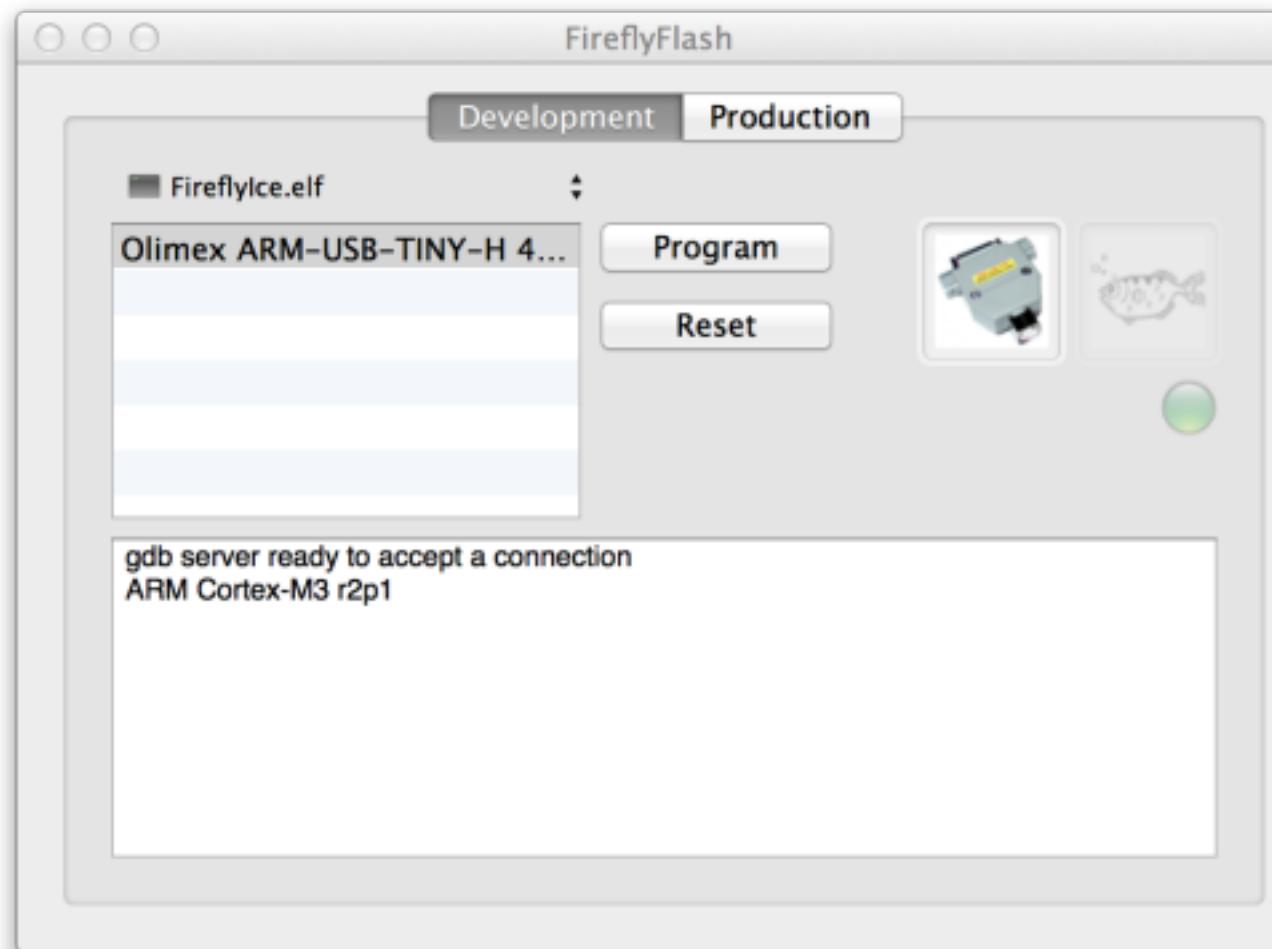
It's time to get the hardware setup. In addition to a Firefly Ice PCBA, you will need the following:

- Olimex ARM-USB-TINY-H JTAG (olimex.com, mouser.com, digikey.com)
- USB Cable A Male to B Male (digikey.com)
- Rowley ARM SWD Adapter (mouser.com)
- Tag-Connect TC2030-CTX-NL Spring Pin Cable (tag-connect.com)
- Tag-Connect TC2030-CLIP (tag-connect.com, digikey.com)
- USB Cable A Male to B Micro Male (digikey.com)

Connect everything together, plug the USB cable from the ARM-USB-TINY-H into your Mac, and plug the USB cable from the Firefly Ice into any USB port for power.



The Firefly Flash production software will be used to load the compiled Fireflyice.elf binary into the Firefly Ice PCBA. Download and run [FireflyFlash.zip](#) (it should automatically unzip when downloaded by Safari).



Click on "Select firmware ELF file..." and select the Fireflyice.elf file that you compiled in the steps above. Click "Program" and in a few seconds the indicators on the Firefly Ice should indicate that the firmware is running.

Debugging the Firmware

The Firefly Flash software acts as a gdb remote target for debugging. To start debugging with gdb:

arm-none-eabi-gdb

then enter the following gdb commands:

```
target extended-remote tcp:127.0.0.1:9000
```

```
set architecture armv3m
```

Now you can use [gdb](#) commands for debugging.

Alternatives

The process above is one way to build, run, and debug Firefly Ice. But it isn't the only option. For example, a project file for the [Rowley Associates CrossWorks for ARM](#) development system is included in the Firefly Ice firmware repository as well. CrossWorks runs on Windows, Linux, and the Mac.

Any [ARM Cortex-M3](#) gcc distribution could be used for compiling.

Any JTAG with SWD capability and the associated tool chain could be used for running and debugging. For example, the Olimex [ARM-JTAG-SWD](#) in combination with the Olimex [ARM-JTAG-20-10](#) can be used instead of the Rowley ARM SWD Adapter with the same tool chain as above.

References

- [GNU Tools for ARM Embedded Processors](#)
- [Energy Micro Software Downloads](#)
- [Firefly Repositories](#)

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Search This Site

[Search](#)

Newsletter

[Subscribe](#)



Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Getting Started: Using the API

The Firefly Ice API is used by apps to communicate with the Firefly Ice device. Apps can communicate wirelessly via Bluetooth 4.0 Low Energy and/or USB.

The first implementation of the Firefly Ice API is for iOS & Mac OS X. This document covers that API.

Windows and Android support are planned for future releases. These libraries will be conceptually similar, but will be implemented using native code on their respective platforms.

Quick Start

The API has a quick way to get started. This can be used to start working with the Firefly Ice device without needing in depth knowledge of the device or the Core Bluetooth Framework.

The quick start approach is to simply create a Firefly Ice Manager object when your app starts then connect/disconnect to devices as desired:

```
#import "ZZHardwareId.h"
#import <FireflyDevice/FDFireflyIceManager.h>

@interface MyClass : NSObject <FDFireflyIceManagerDelegate>
@property FDFireflyIceManager *fireflyIceManager;
@end

@implementation MyClass

(void)initialize
{
    _fireflyIceManager = [FDFireflyIceManager managerWithDelegate:self];
}

(void)fireflyIceManager:(FDFireflyIceManager *)manager discovered:(FDFireflyIce *)fireflyIce
{
    // if/when the app is ready to connect the device:
    [_fireflyIceManager connectBLE:fireflyIce];
    FDFireflyIceChannelBLE *channelBLE = (FDFireflyIceChannelBLE *)fireflyIce.channels[@"BLE"];
    CBPeripheral *peripheral = channelBLE.peripheral;
    // if peripheral.identifier is not nil we may have connected to the device before
}

(void)fireflyIceManager:(FDFireflyIceManager *)manager openedBLE:(FDFireflyIce *)fireflyIce
```

```
{
}

(void)fireflyIceManager:(FDFireflyIceManager *)manager identified:(FDFireflyIce *)fireflyIce
{
    // the hardware id can now be used to query the web site for a registered user
    NSString *hardwareId = [ZZHardwareId hardwareId:fireflyIce.hardwareId.unique];
}

(void)fireflyIceManager:(FDFireflyIceManager *)manager closedBLE:(FDFireflyIce *)fireflyIce
{
}
```

The delegate will be notified on discovery, open, close, and when the device has been identified (the firmware version and hardware id are received after opening). When a device is open the Firefly Ice Manager will manage all device communication: setting the time, updating the firmware, transferring activity data to the web site, etc.

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on !\[\]\(8d0f0e0fe25b320c33272c52aec1fbca_img.jpg\) LinkedIn](#)

Search This Site

[Search](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Newsletter

[Subscribe](#)



© 2014 Firefly Design LLC. All rights reserved [Website hosting by Shopify](#).

Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Getting Started: Using the Development Kit

Firefly Ice Development Kit Contents

The Firefly Ice development kit includes:

- Firefly Ice Printed Circuit Board Assembly
- 80 mAh Lithium Polymer Battery
- Plastic Parts (black top and bottom pieces; plus 1 each of green, pink, blue, and black shell pieces)
- Firefly Ice USB Cable (for charging and communication)
- SWD Cable
- Retaining Clip for SWD Cable
- SWD Adapter
- JTAG Adapter
- JTAG USB Cable
- PCBA Expansion Connector(s)



Powering the Firefly Ice

The Firefly Ice is shipped to you without the battery connected. You can test out the Firefly Ice plug PCBA without the battery by attaching a USB cable plugged into a power source, such as a laptop or wall charger. Once this is done the device will startup and apps can connect to the device using Bluetooth and/or USB.

Using the Firefly Ice Utility App

Clone the firefly-sync repository and open the Firefly Utility Xcode project:

```
git clone https://github.com/denisbohm/firefly-sync.git  
open firefly-sync/FireflyUtility/FireflyUtility.xcodeproj
```

Run the project using your iPhone 4s, 5, 5s, or 5c. The first screen shows a list of all the Firefly Ice devices that have been discovered. Touch one of the devices to move to the Overview panel for that device. Touch "Connect" at the top right to open a connection to the device. On a successful connection the LED on the device will indicate that the connection is open and the information on the overview panel will update in the app.

Firmware Development Setup

The Firefly Ice PCB has a Cortex Serial Wire Debug (SWD) JTAG port connection that can be used for development. The included hardware is used to make the SWD connection. Connect the supplied parts in the following order:

1. JTAG Adapter to JTAG USB Cable.
2. SWD Adapter to JTAG Adapter.
3. SWD Cable to SWD Adapter.
4. SWD Cable to Firefly Ice PCBA.
5. Retaining Clip to SWD Cable.



You are now ready to get started with [firmware development](#).

Hardware Development Setup

An expansion socket and header are provided for expanding the hardware. Solder the PCBA expansion socket to the Firefly Ice PCBA. The expansion header is provided for use on your custom expansion PCBA.

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on **LinkedIn**](#)

Search This Site

Search

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Newsletter

[your@email.com](#)

Subscribe



© 2014 Firefly Design LLC. All rights reserved [Website hosting by Shopify](#).

Firefly Design LLC

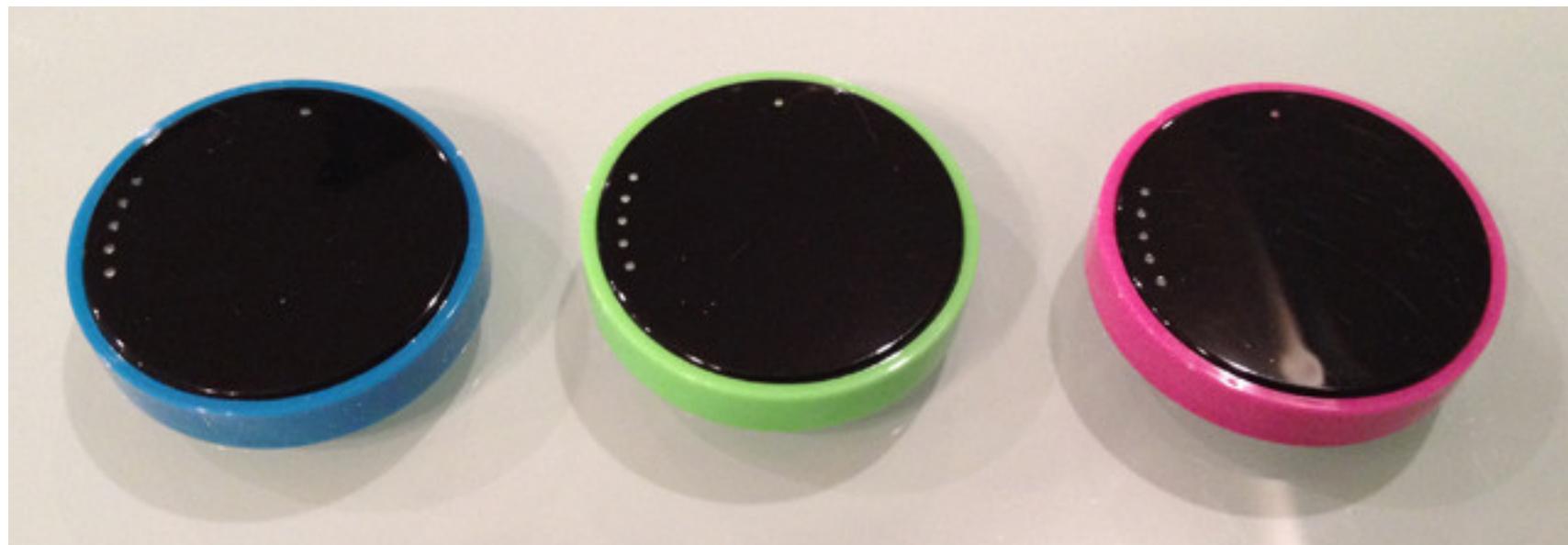


Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Getting Started: Using the Firefly Ice

Firefly Ice

The assembled Firefly Ice is ready to be used with any app that supports it.



Waking the Firefly Ice

The Firefly Ice is shipped to you in storage mode. In this mode the device is in a very low power state to preserve the battery charge. The device can be stored in this way for up to 6 months. To wake up the Firefly Ice simply plug the USB charging cable into the device and into a power source, such as a laptop or wall charger. This will wake up the device so that apps can connect using Bluetooth or USB.

Using the Firefly Ice Utility App

Clone the firefly-sync repository from github. Open the FireflyUtility Xcode project. Run the project using your iOS device. On the first screen in the app the Firefly Ice devices that have been discovered are listed. Touch one of the devices to initiate a connection. On a successful connection the device will indicate the connection and the general information will update in the app.

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Search This Site

Search

210 Lavaca Street #2303

Austin, TX 78701

Newsletter

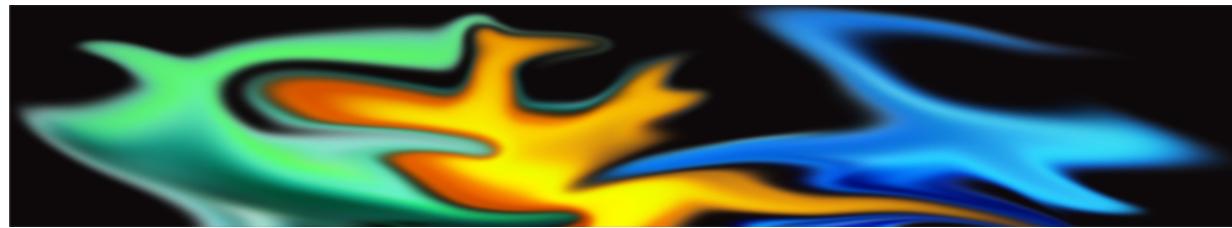
your@email.com

Subscribe



© 2014 Firefly Design LLC. All rights reserved [Website hosting by Shopify](#).

Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Getting Started: Using the Parts Kit

Firefly Ice Development Kit Contents

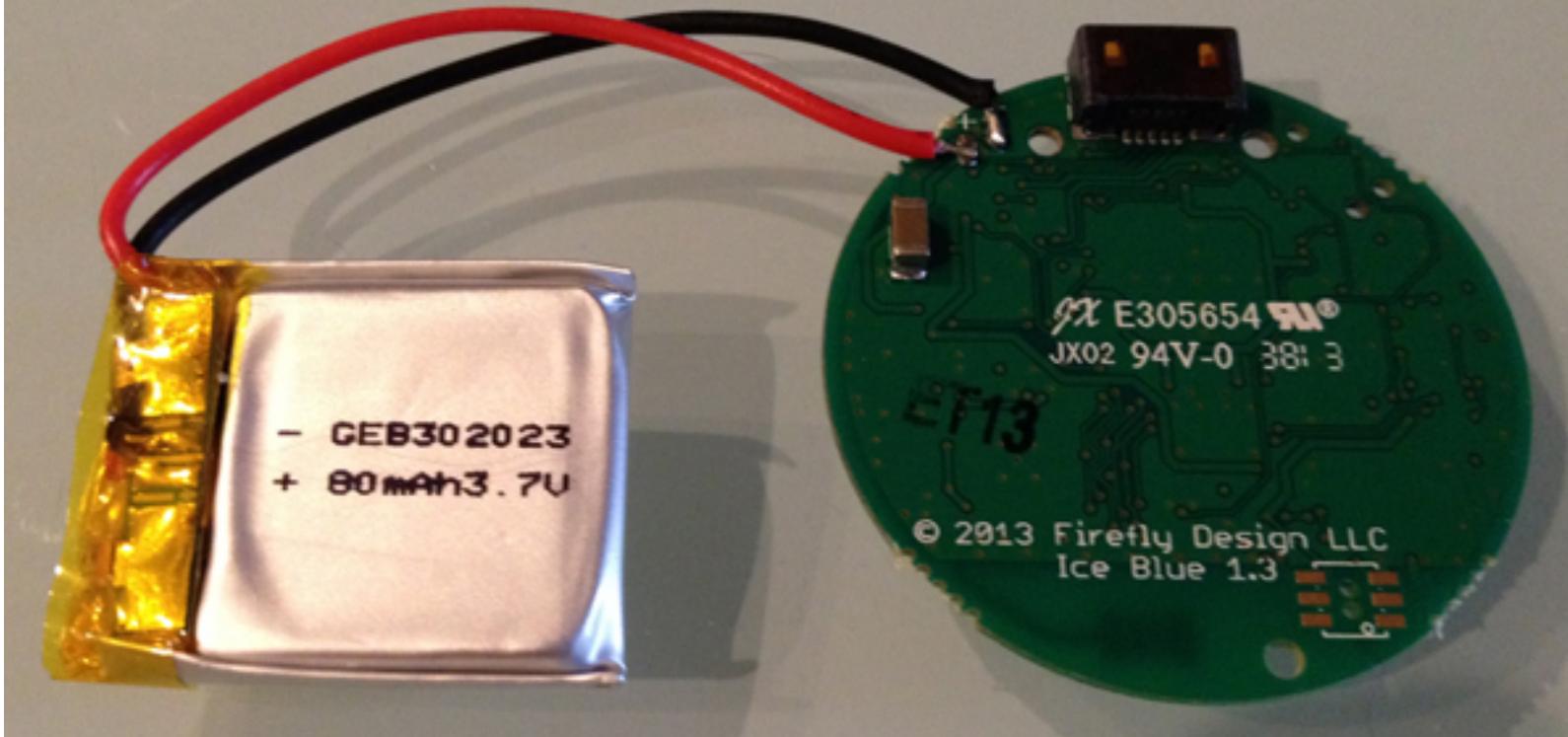
The Firefly Ice parts kit includes:

- Firefly Ice printed circuit board assembly.
- 80 mAh Lithium Polymer battery.
- Plastic parts: black top and bottom pieces; plus 1 each of green, pink, blue, and black shell pieces.
- PCBA Expansion Connector(s).



Powering the Firefly Ice

The Firefly Ice is shipped to you without the battery connected. You can test out the Firefly Ice plug PCBA without the battery by attaching a USB cable plugged into a power source, such as a laptop or wall charger. Once this is done the device will startup and apps can connect to the device using Bluetooth and/or USB. To use the device with the supplied battery simply solder the battery leads to the battery terminals as shown here:



Using the Firefly Ice Utility App

Clone the firefly-sync repository and open the Firefly Utility Xcode project:

```
git clone https://github.com/denisbohm/firefly-sync.git
open firefly-sync/FireflyUtility/FireflyUtility.xcodeproj
```

Run the project using your iPhone 4s, 5, 5s, or 5c. The first screen shows a list of all the Firefly Ice devices that have been discovered. Touch one of the devices to move to the Overview panel for that device. Touch "Connect" at the top right to open a connection to the device. On a successful connection the LED on the device will indicate that the connection is open and the information on the overview panel will update in the app.

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Search This Site

[Search](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Newsletter

[Subscribe](#)



Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: Overview

The currently available device is the Firefly Ice.

Electronics

- ARM Cortex-M3 Processor with 256KB Flash & 32KB RAM
- Bluetooth 4.0 Low Energy Radio
- Full Speed USB
- 3-Axis Accelerometer
- 3-Axis Magnetometer
- Temperature Sensor
- Rechargeable Lithium Polymer Prismatic Battery
- LED Indicators (2 RGB + 2 Red + 1 Amber/Green)
- 2MB Nonvolatile Storage
- Real Time Clock
- Very Low Power Sleep & Active Modes

Firmware

- Calculates & Stores Activity Metric
- Commands & Data Transfer Via Bluetooth & USB
- Connection Status Indication
- Charging Status Indication
- Diagnostics Log

API

- iOS Library (iOS 7)
- Mac OS X Framework (10.6-10.9)
- Discovery Support
- Commands & Data Transfer Via Bluetooth & USB
- Diagnostics Log

Mechanical

- 3 Polycarbonate Parts
- Simple production assembly process
- Parts can be press fit during development

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on !\[\]\(115eff7009a76771e6b7adb966005e4c_img.jpg\) LinkedIn](#)

Search This Site

[Search](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

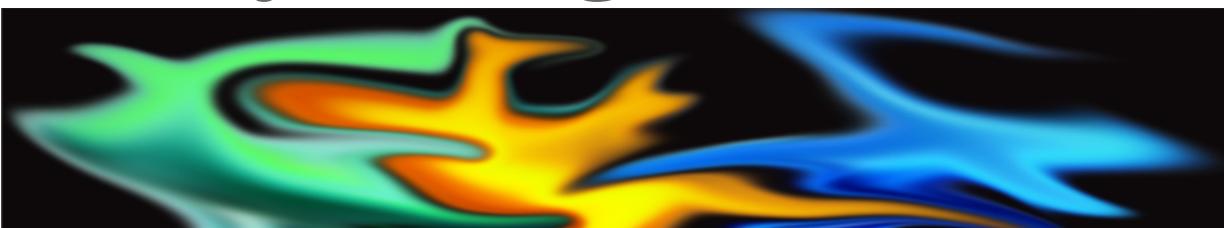
Newsletter

[Subscribe](#)



© 2014 Firefly Design LLC. All rights reserved [Website hosting by Shopify](#).

Firefly Design LLC



Firefly Ice: An Open Source Wearable / Internet of Things Platform

Firefly Ice: FCC

The Firefly Ice has been FCC tested.

The FCC ID U49-ICE can be found on the bottom of the device:



Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

NOTE: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

- Consult the dealer or an experienced radio/ TV technician for help.

About Us

Firefly Design has been around for 20 years helping others create great products.

Contact Us

denis@fireflydesign.com

[View my profile on LinkedIn](#)

Search This Site

[Search](#)

Firefly Design LLC
210 Lavaca Street #2303
Austin, TX 78701

Newsletter

[Subscribe](#)



© 2014 Firefly Design LLC. All rights reserved [Website hosting by Shopify](#).