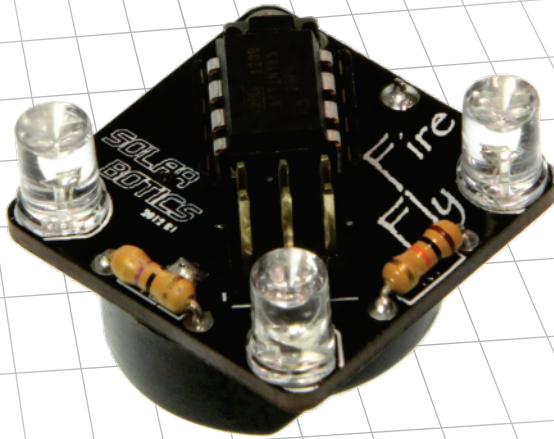


SB-FireFly

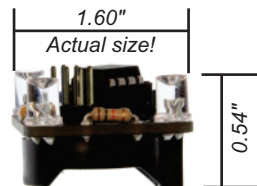
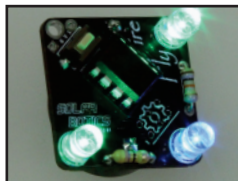
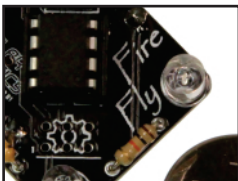
Arduino™-compatible Microcontroller

What if the Tiny Cylon and Arduino Uno had a baby, what would it look like? The SB-FireFly of course!



Build Time: 10mins
Skill Level: Beginner (1/5)

- Arduino-compatible with minor additions
- ATtiny85V development board
- Complete with CR2032 battery
- Only 12 parts!
- ISP header for easy reprogramming
- Wearable as a necklace, keychain, or badge.
- Hackable into remote-control reader / rebroadcaster (req's additional parts)



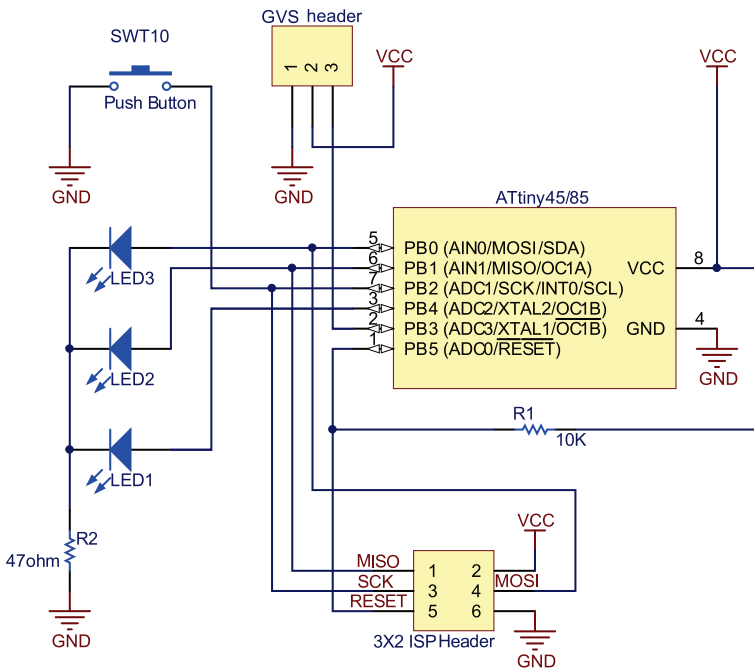
INTRODUCTION

BLIFNAR. Blinky. LED-thingy. These all describe the SB-FireFly. This coin-cell powered microcontroller runs three LEDs through selectable light blinking sequences with smooth transitions.

The SB-FireFly features an Arduino-accommodating ATtiny85 microcontroller, so with a minor adjustment to your Arduino programming IDE, you can reprogram it to suit your whim. It also features a cleverly hidden 3-pin expansion port usable for hacking.

We've designed the SB-FireFly so when not in use, it eats only 0.0002mA - a very small amount of current, so it will be ready to wake and blink an LED when you need it to!

SB-FireFly 1.0 Schematic



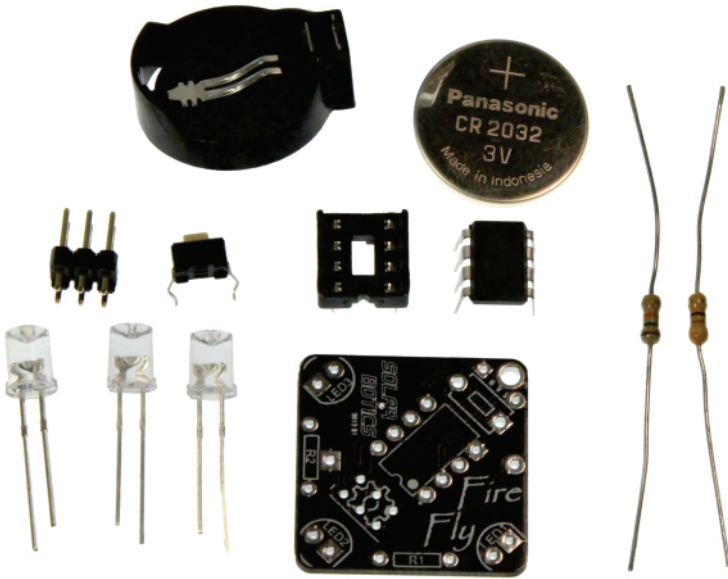
“Arduino” is a trademark of the Arduino Team (www.arduino.cc). The SB-FireFly is licensed under the Creative Commons A-SA 3.0 license. Get full design files from our website!

Disclaimer of Liability

Solarbotics Ltd. is not responsible for any special, incidental, or consequential damages resulting from any breach of warranty, or under any legal theory, including lost profits, downtime, good-will, damage to or replacement of equipment or property, and any costs or recovering of any material or goods associated with the assembly or use of this product. Solarbotics Ltd. reserves the right to make substitutions and changes to this product without prior notice. Sorry - we hate legalese too, but “don’t pick on us” doesn’t impress the lawyers.

PARTS LIST

- 1 x SB-FireFly printed circuit board (PCB)
- 1 x Pre-programmed Attiny85 with 8Mhz internal oscillator
- 2 x Green SBLED
- 1 x Blue SBLED
- 1 x SWT10 switch (mode/power pushbutton)
- 1 x 47ohm resistor (yellow / violet / black) for LED current limiter
- 1 x 10kohm resistor (brown / black / orange) for reset pull-up
- 1 x 8-pin DIP carrier
- 1 x ISP 3x2 male header (programming)
- 1 x 2032 coin cell battery
- 1 x 2032 coin cell battery holder



We strongly suggest you count the parts in your kit to make sure you have all the parts listed (c'mon - there's barely a handful of parts, so count them!). If anything is missing, contact Solarbotics Ltd. for replacement parts information.

Tools & Materials Needed (The 4 'S's):

- Soldering iron
- Solder
- Safety glasses
- Side/flush cutters

CONSTRUCTION!

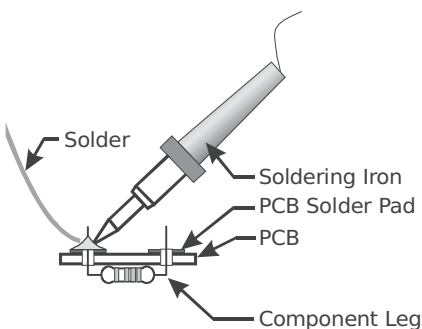
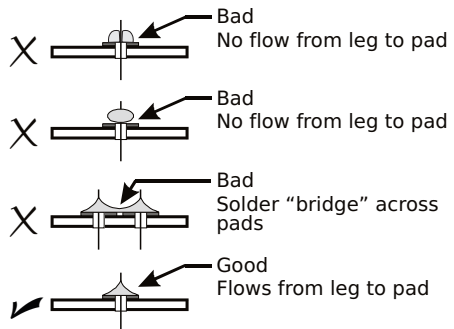
Soldering - The Essentials:

The most important skill needed to successfully construct your device is soldering. Soldering is melting a special metal (called, um..., "solder") between two components to make an electrical connection. We can also use solder like glue, to build things out of metals. Please note: You must make sure to use *electrical* solder, and not plumbers solder, which is used for piping and really isn't good for electronics.

Much like you, solder likes to go where it's the warmest (this is why Florida is so popular). The trick to successful soldering is to make the *parts* hot, and the melting solder will run there. If you don't heat up the parts first, the solder will find the hottest thing around - your soldering iron, and not your parts! Do *not* melt solder to the tip of your iron and try to smear it onto the parts, as it just won't work. You're an electronics hobbyist, not a painter!

Successful soldering is a 4 step process:

- 1) Wipe the hot tip of the soldering iron on a sponge so the tip is clean and shiny.
- 2) Jam the soldering iron into the hole where the component leg comes through the soldering pad.
- 3) Count to 5 (which lets the soldering iron make the pad and component leg nice and hot).
- 4) From opposite the iron, push solder into the corner between the leg and pad, melting the solder until there's enough to make the solder joint look like a tiny volcano.

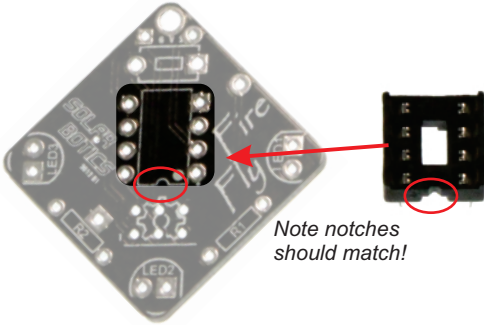
**Bad & Good Solder Joints**

SB-FIREFLY MANUAL

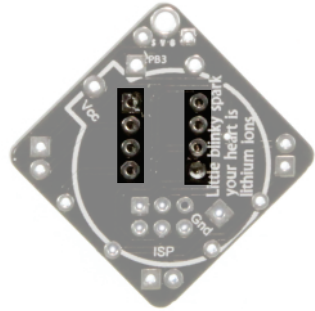
CONSTRUCTION!

Now that you know how to solder and you've inventoried all the parts, it's time to put this little blinky bug together!

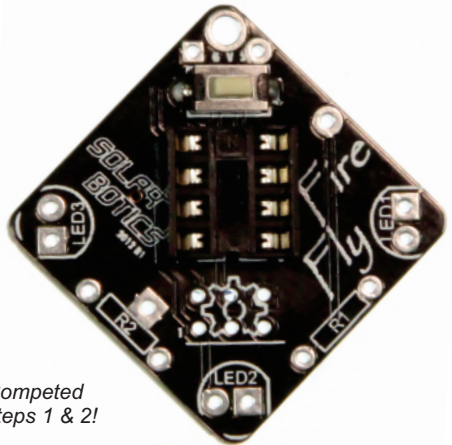
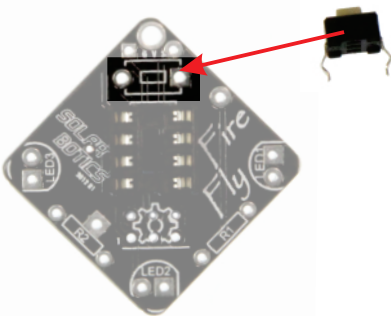
Step 1 - DIP Socket: The DIP socket is just a carrier for the microcontroller that makes it easy to remove than desoldering every pin. Install it with the half-moon notch matching the one printed on the PCB.



Flip it over, and bend a few pins over with your thumbnail so the DIP socket stays put while you finish soldering it in from the back.



Step 2 - Mode Button: Install the push button just above the DIP carrier. It's designed to snap into place with a little force, so push it in, and solder it down.



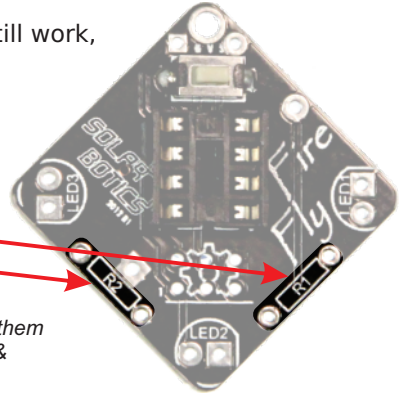
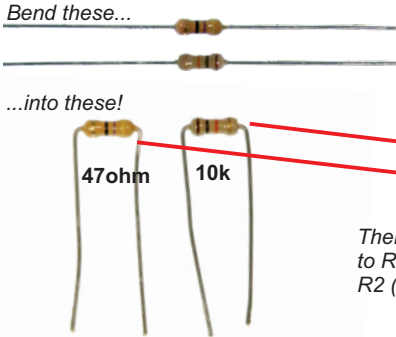
Completed steps 1 & 2!

CONSTRUCTION!

Step 3 - Resistors: First thing to do is to bend the leads right over 90° close to the resistor body, so they fit into their spot on the PCB. Next, install the 10k resistor (colors: Brown / Black / Orange) to location 'R1', and solder it in.

Do the same for the 47 ohm resistor (colors: yellow / purple / black), installing to location 'R2'.

Don't mix these up! Your SB-FireFly will still work, but barely!



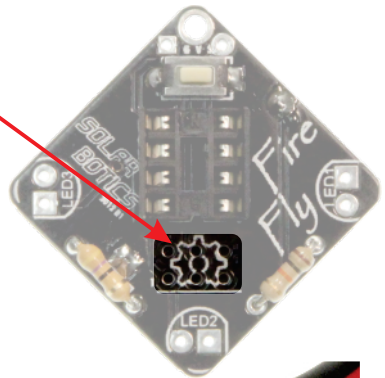
Then install them to R1 (10k) & R2 (47ohm)

Step 4 - ICP Decision time! Do you like to *watch* blinkies, or bend blinkies to your will? This step adds simple programming to your SB-FireFly.

If you think you might want to modify your SB-FireFly code, install the 3x2 male pin header in the ISP location, under the Open Hardware logo.



Installation of the *optional 2x3 programming header*



It makes it look a bit more technogeeky rather than clean & blinky. Your choice.

Step 5 - Trimming! Yes, we have to *specifically* tell you to do this. If you don't trim the sticky-outy bits on the bottom, the battery holder *will not* fit very well.

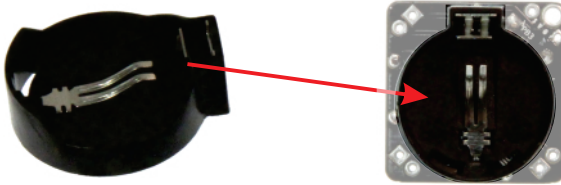
Trim all pins/leads down, nice and tidy.



Trim all leads & pins on the PCB underside

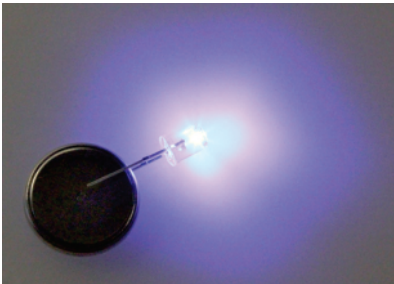
CONSTRUCTION!

Step 6 - The Battery Holder: If you manage to get this wrong, we'll be impressed. Not amazed, but impressed. Just install it on the backside so it matches the outline, and solder it in.

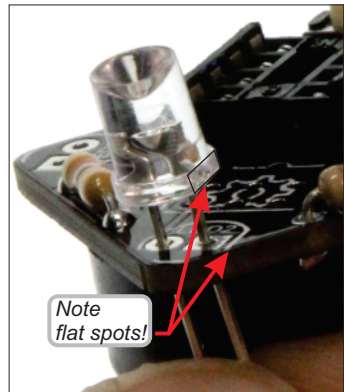


Step 7 - LEDs: Blinky parts! First thing to do is figure out which is which, and how to install them. LEDs don't work in reverse, so pay attention to the flat spot on the side of the clear lens, and install it so it matches the flat spot printed on the circuit board.

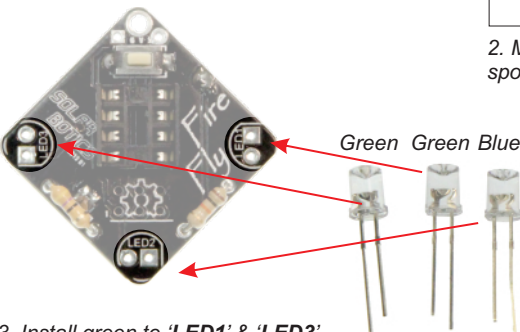
We're going to put the green LEDs in positions '**LED1**' and '**LED3**', and the blue in '**LED2**'. Not sure which LED is which? Use the coin cell to test the LEDs by wedging it in between the LED leads. If it doesn't light up, flip the coin cell around.



1. Test for LED color with the coin cell
blue installs at '**LED2**'



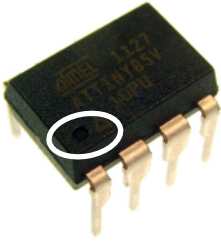
2. Make sure you match the flat spot on the LED to the PCB



3. Install green to '**LED1**' & '**LED3**'.
Install blue to '**LED2**'

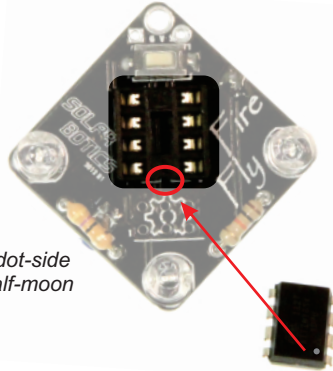
CONSTRUCTION!

Step 8 - Installing the IC: Braaaains... Yes, we're using a fairly powerful tiny microcontroller to blink lights (but we'll leverage that into more cool applications later!).



See the dot on the corner of the ATtiny85? Match it with the one semi-hidden under the DIP carrier - it's on the same side as the half-moon notch. Pinch the pins in a bit, and insert the ATtiny into the DIP carrier. Done!

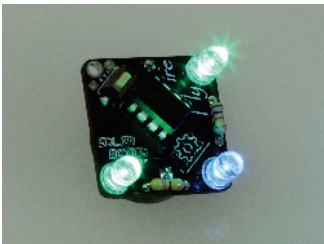
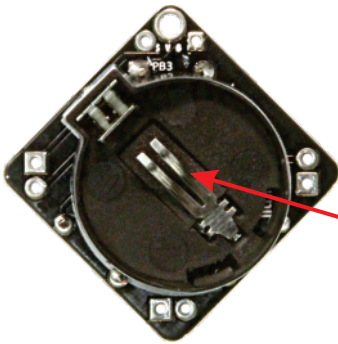
See the dot? Dots are important. Dots keep you from putting the IC in backwards. We like dots...



Insert IC, dot-side nearest half-moon notch

Step 9 - The Battery: Remember that coin-cell battery you used to test the LEDs? Yup, we're using that again. Snap it into the battery holder, and the LEDs should start doing their thing.

Press the push button to cycle through the different modes and speeds. When they turn off, the SB-Firefly is in sleep/off mode.



Fully-functional SB-FireFly!

Battery installed. Make sure it snaps in!

HACKING IT!

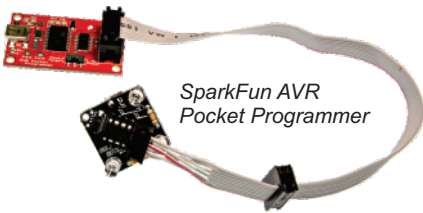
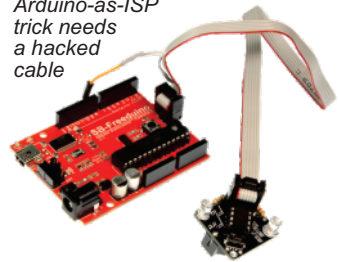
Before we think about what you can do with a microcontroller, 3 LED points, and a spare set of pads, *how* do you hack it? The geniuses at the MIT Media lab High-Low tech site have step-by-step instructions programming the ATtiny microcontroller using the popular Arduino 1.0 IDE. The hack requires that it runs on the 8Mhz internal oscillator (which it already does).

If you want to use an Arduino-as-ISP trick, you modify a \$3.00 6-pin ribbon cable by soldering a single pin to the second line in and hook it up to pin 10 on the 'duino board. See the details on the MIT website. Remember to remove the battery when programming this way!

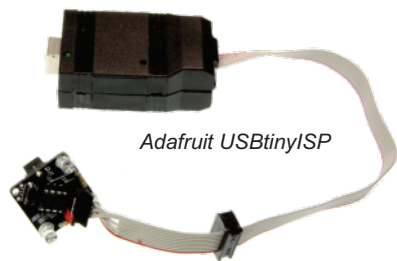
You can keep the battery in when using ther Arduino-supported ISP programmers from SparkFun and Adafruit, but remove the 'USB Power' jumper on the Adafruit USBtinyISP, or set the 'No Power' switch on the SparkFun AVR pocket progammer. Use the "USBtinyISP" programming in the Arduino IDE.



Arduino-as-ISP trick needs a hacked cable



SparkFun AVR Pocket Programmer



Adafruit USBtinyISP

So what kinds of SB-FireFly hacks are there? We're working on:

- Tucking them into translucent jars to float down the river for a night-time art project.
- Tucking them into balloons for an evening-at-the-park festival.
- Replace some of the LEDs with IR transmitting LEDs, then using a TSSOP4038 38kHz IR sensor to the spare GVS input, that reads remote controls, so we can build...
- A remote Wii sensor lightbar replacement, that turns on when it senses the TV "on" command.
- An IR remote "macro" bug, that waits for TV "on" commands, and also turns on the surround sound audio system via IR.
- A miniature "TV-B-Gone" (another great OSHW project) pendant, that transmits global "TV-off" signals.
- An IR remote "learner" that can record and playback IR signals.

The SB-FireFly started as a fun in-house project that was turned into a neat after-dark art project in Calgary. Then it was used at a local science center for an evening buildfest.

The included CR2032 coin cell battery has been tested to run a SB-FireFly continuously for 24 hours before the blue fades out, and over 36 hours for the green. When not running, it sleeps, drawing only 0.0002mA of current!

There is even a spare pin available for additional hacking. Add a CdS cell for automatic dark-activation, or a IR signal decoder for remote-control operation. It's a handy little hacking board for the AVR ATtiny45/85 microcontroller, programmable by Arduino!

Standard SB-FireFly modes:

- Blink All LEDs mode (slow, reg, fast)
Sleep
- Metronome mode (slow, reg, fast)
Sleep
- Disco mode (slow, reg, fast)
Sleep
- Fading Eyes mode (slow, reg, fast)
Sleep
- Firefly mode (slow, reg, fast)
Sleep

Note: All modes and speeds toggled with the on-board push button.



Visit us online for more info and cool stuff:

www.solarbotics.com

Solarbotics Ltd.
3740D - 11A Street NE, Suite 101
Calgary, Alberta T2E 6M6
Canada



Toll Free: 1-866-276-2687
International: +1 (403) 232-6268



Fax: +1 (403) 226-3741



Made in Canada