

Bargraph Breakout Kit

2011.12.27

Overview

This kit contains a PCB and all the parts needed to build a fun, 30-LED bargraph that can be driven by an Arduino or other microcontroller.

This kit uses 74HC595 shift registers and three, 10-LED bargraph modules to create the display. We supply one green, one yellow and one red module for "safe / caution / DANGER!" displays, or you can use other 10-LED bargraphs for different colors (not included). The display is easily driven by any microcontroller with an SPI interface, and you have individual control over every LED.

Specifications

Power

- 5V at up to 250mA (all LEDs on)
(Will operate at 3.3V at reduced brightness)

Interface

- Serial Peripheral Interface (SPI)
- Clock (CLK), Data (SIN) and Latch (LAT)
- Up to 25MHz CLK speed

Dimensions

- Overall: 2" x 3" (1/16" board thickness)
- Hole pattern: 1.8" x 2.8" (4 x 0.125" hole, 4-40)

Assembly

If you've never soldered before, it's easy! Visit our tutorial at www.sparkfun.com/tutorials/106 to learn the basics.

Tools needed:

- Soldering iron
- Solder
- Flush cutters
- Safety glasses
- Flux pen (optional*)

*The flux pen makes solder flow easier, but you can get by without it. If you do have one, use it to wipe some flux on the areas you'll be soldering.

Getting ready

All the parts go on the top side of the board, (the side with the white outlines and text, called silkscreen), and you'll solder on the opposite (bottom) side of the board. You may have to bend or straighten the leads on the parts to make them fit in the holes.

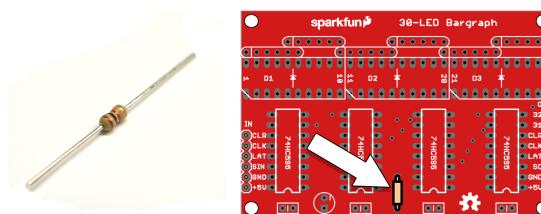
Read through ALL the assembly instructions before starting, to make sure you know exactly where all the parts go and what direction to put them in. If anything is unclear, contact us *before* you solder – we'll be glad to help!

ALWAYS WEAR SAFETY GLASSES when soldering to prevent splattering solder or flying cut leads from getting in your eyes!

We'll solder the parts to the board in order of shortest to tallest. This way you can always lay the board on its face to keep the part you're working on flush to the board.

1. 1K resistor R1

This is the small, tan, cylindrical part with brown / black / red color bands. It's not polarized, so it doesn't matter which direction you put it in. Insert it into the rectangular outline in the center of the board marked "1K", flip the board over, and solder the leads. Once they're soldered, clip the leads close to the board.



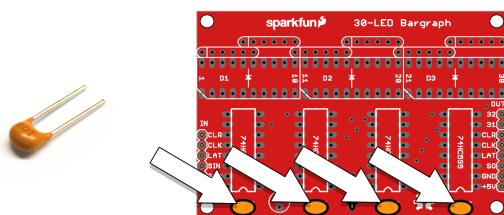
TIP: It may help to gently bend the leads outwards after inserting the part into the board, to help hold it flush to the board while you're soldering. This goes for all two-lead parts.

Bargraph Breakout Kit

2011.12.27

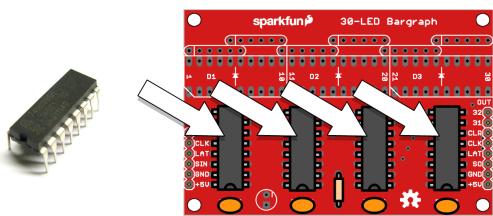
2. 0.1uF capacitors C1, C2, C3, C4

These are the small, yellowish parts with two leads. They go in the four small white boxes on the bottom edge of the board. These parts aren't polarized, so you can insert them in either direction. Place and solder one part at a time, and clip the leads when you're done.

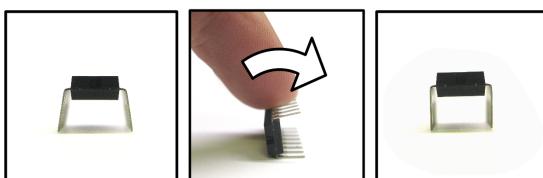


3. 74HC595 integrated circuits (AKA "IC" or "chip") IC1, IC2, IC3, IC4

These are the four boxy black parts with lots of legs. *These parts are polarized, and must be inserted in the proper direction to function!* Look for a "notch" on the narrow end of the part, nearest to pin 1. When inserting the IC into the board, ensure that the notch on the part matches the notch in the silkscreen.



TIP: ICs typically come from the factory with their leads a bit too far apart for the holes. You can carefully roll them against a hard, flat surface to *slightly* bend all the leads back in towards the center to get them to fit better. Do this a little at a time and check as you go so you don't bend them too far.



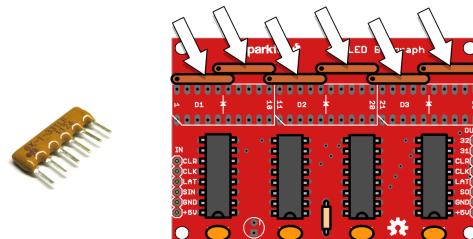
Solder one IC at a time. Ensure that it's seated all the way into the holes, flip the board over, and solder each pin.

TIP: Hold the IC tight to the board while soldering the first few pins. The furthest two diagonal pins are a good choice. Once it's secured by a few solder joints, the IC won't move for the rest of the soldering job.

You do *not* have to clip off all the little pins when you're done (unless you really want to).

4. 330 Ohm resistor packs R2, R3, R4, R5, R6, R7

These are the 6-pin yellowish parts. *These parts are polarized, and need to be inserted in the proper direction to function!* These parts go into the oval-shaped patterns on the board. Each oval pattern has one square hole marked with an extra line in the silkscreen; this is pin 1. Each of the parts will have a large brown dot near pin 1. Ensure that pin 1 on the part, goes into hole 1 on the board.



Solder one part at a time. Insert a part into the board, flip the board over, and solder one of the pins. Check to see if the part is still straight. If it isn't, re-melt the joint, and simultaneously straighten up the part. Once it's straight, solder the rest of the pins. You don't have to clip these pins unless you want to.

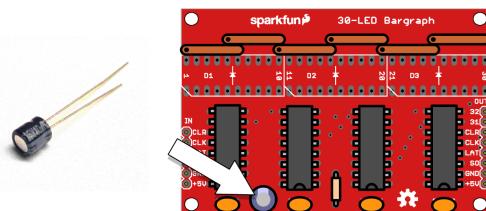
5. 10uF capacitor C5

This is the blue, cylindrical part with two leads on one end. *This part is polarized and must be inserted in the proper direction to function!* It goes in the silkscreened circle on the board. Look closely at the board: note that there's a minus sign ("−") next to one of the two holes. Now look closely at the part. There will be a band with minus symbols near one of the two leads.

Bargraph Breakout Kit

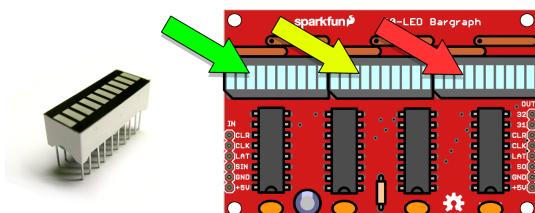
2011.12.27

Insert the part so that the minus lead goes into the minus hole. Turn the board over, ensure the part is flush against the board (you might bend the leads out a bit to hold it), solder the leads and cut off the excess.



6. LED bargraph modules LED1,LED2,LED3

These parts are polarized, so it's important to insert them correctly! Each of these parts has a very slightly beveled corner on ONE of the vertical edges, indicating pin 1. It can be hard to see, so look at all four edges. Three of them will be "sharp" looking, and the fourth will look slightly beveled or rounded. Insert the part so the beveled edge matches up with the beveled corner in the white silkscreen on the board. (The silkscreen bevel is hard to see; it's on the side closest to the ICs).



The bargraphs should have a red, green, or yellow ink spot on them to tell you their color. We figure most people would want to put green in LED1, yellow in LED2, and red in LED3 to have the LEDs light up in the order: "safe / caution / DANGER!", but you can put them in any order you wish, or replace them with other colors available from www.sparkfun.com.

Insert one LED bargraph at a time. It can be hard to get all those pins to go in all those holes, so make

sure the pins are straight first. Once it's in, ensure that it's flush to the board, turn the board over, and solder the pins. Like the ICs, if you solder the two farthest diagonal pins first, they will hold the bargraph for the rest of the soldering. When you're done, clip the leads.

7. Congratulations, you're done soldering all the electronic components!

Now you'll want to decide how to connect the board to your Arduino or other microcomputer system. This is done through the header area marked "IN" on the left-hand side of the board (the "OUT" header is used to daisy-chain more bargraphs if you wish.)

If you're going to wire it directly to your microcontroller, you can solder wires directly to the header. If you'd rather use jumper wires, you can solder a 6-pin header to the board (not included, use SparkFun part number PRT-00116).

TIP: If you'll be putting your bargraph into a case where it will be mounted to the inside of the front panel, you may want to connect your wires or header to the back side of the PCB.

Once you're done with all the soldering, you may want to peel the protective plastic film from the front of the LED modules. You may also want to clean the board to get rid of flux (the clear residue left after soldering). It's not strictly necessary to clean off the flux, but it makes the final product look nicer. If you're using SparkFun solder with water-soluble flux, you can clean it in warm water using an old toothbrush. If you're using other types of solder, a bit of rubbing alcohol will dissolve the flux. The parts can withstand being wet, but **be sure to let the board dry thoroughly before powering it up!**

Connecting it to your system

The bargraph has an IN connector and an OUT connector. You'll connect the IN connector to your Arduino or other microcomputer system. The OUT connector is for daisy-chaining to additional bargraphs if desired.

Bargraph Breakout Kit

2011.12.27

Pin descriptions and connections

label	function	Arduino Uno pin	Arduino Mega pin	Notes
+5V	Power supply	5V	5V	Can draw up to 250mA if all LEDs are on
GND	Ground	GND	GND	
SIN	SPI input (IN connector only)	11 (MOSI)	51 (MOSI)	SPI data to bargraph
SOUT	SPI output (OUT connector only)			SPI data overflow from last bargraph (to SIN of next board if desired)
LAT	Latch input	10 (SS) (default)	53 (SS)	Latch control to bargraph (LOW-HIGH transition makes data visible on LEDs). You can use any Arduino pin for this function.
CLK	Clock input	13 (SCK)	52 (SCK)	SPI clock to bargraph (stores SIN on LOW-HIGH transition).
CLR	Clear input			Provided, but usually not needed. Set LOW to reset all bargraph registers to LOW
30	Bit 30 output (OUT connector only)			The shift registers store 32 bits, but we only use 30 of them;
31	Bit 31 output (OUT connector only)			the unused outputs are available for your use if desired.

The minimum pins needed to operate the bargraph are +5V, GND, SIN, LAT, and CLK. Connect them to your Arduino or other system using the table above.

Note that the CLR input is provided in case you need it, but it is usually not required (it zeroes all of the registers, but you'll generally send a full 32 bits to the board which overwrites the old data). The CLR signal has a pullup resistor attached to it (R1) so it will always read as HIGH (inactive) when disconnected.

How it works

Each of the four 74HC595 chips on the board contains an 8-bit "shift register". The shift register is like a bucket brigade – when the clock input (CLK) transitions from low to high, the first bucket grabs the low or high value present on the serial input (SIN) line, and the rest of the 8 buckets pass the one or zero stored in it down the line to the next one. When you keep clocking data into it, the SIN values "slide" down the chip until you've filled all eight buckets. But wait, there's more!

The last bucket has an output (SOUT) that's very easy to connect to the input of another chip (SIN). So

you can keep linking chips together to make shift registers as long as you like. This board links four 8-bit chips together, providing 32 output pins to drive the 30 LEDs with two outputs left over. And it's easy to link even more boards together if desired. Shift registers are a great way to expand your outputs if you have a limited number of pins.

One more thing. If the LEDs always showed all the data sliding through the shift registers, it would be a very confusing display. But the parts have an additional feature to prevent this: Each bucket in the shift register is separated from the output pin by an additional register, known as a latch. The latch register isolates the internal data from the output pins until you're done shifting in the data you want. Then, when you transition the LAT input from low to high, all the internal data will appear on the output pins simultaneously. Furthermore, the output data is like a snapshot; it will stick around even while you clock new data into the circuit. Only when you again toggle the LAT input will a new copy of the internal shift register data appear at the outputs. Pretty neat, huh?!

See the 74HC595 datasheet for more information:
www.sparkfun.com/datasheets/IC/SN74HC595.pdf

Bargraph Breakout Kit

2011.12.27

Programming

We've put together an Arduino library with some useful routines. It's available from the Bargraph Breakout product page at: www.sparkfun.com/products/10936.

To install the library, unzip it, and copy the SFEbarGraph folder to the "libraries" folder within your Arduino sketch directory (if you don't have a "libraries" folder, you can create one). Restart the Arduino IDE, and the library and example sketches will appear in the menus. Try the example sketches out, and see what they do!

Library functions

Here are descriptions of the library functions you can use in your own sketches. (Also check out the "bargraph_example" sketch.) To use the library, install it as described above, and add the statements:

```
#include <SFEbarGraph.h>
#include <SPI.h>

SFEbarGraph BG;

to the top of your sketch. These load the Bargraph and SPI libraries into your code, and create an Bargraph object (called BG here) for your use. You can then use the below routines in your code.

BG.begin();
```

```
BG.begin(unsigned char numboards);
```

```
BG.begin(unsigned char numboards,  
        unsigned char latchpin);
```

Call BG.begin() once at the beginning of your code to set up the bargraph. Calling it with no parameters will result in default values (only one bargraph board, LATCh pin is 10 on an Uno, 53 on a Mega). You can also call the function with a different number of daisy-chained boards (1-8), and a different LATCh pin. (Note: BG.begin() will return a boolean value indicating whether the library was successfully initialized or not (true == success). You can normally ignore this returned value, but if things aren't working you might check here to see if there was an initialization problem).

```
BG.barGraph(unsigned char bar, unsigned  
char peak);
```

This routine will emulate a standard bargraph. "Bar" should be the number of LEDs to light up starting on the left-hand side (0 = no LEDs). "Peak" will light up an additional single LED if desired to indicate peak values, etc. (0 = no peak). Both values can be 0 to 30 for a single bargraph, or 0 to 240 for up to 8 bargraphs linked together. (This routine will take into account the extra 2 outputs in each bargraph so you don't have to.)

```
BG.clear();
```

```
BG.paint(unsigned char position, boolean  
value);
```

```
BG.send();
```

These routines allow you to control arbitrary LEDs on the bargraph. These routines work on a "canvas" that you erase with BG.clear(). You can then turn individual LEDs on and off with BG.paint(), choosing the position and whether the LED should be on (1 or true or HIGH) or off (0 or false or LOW). Unlike the barGraph function above, the LED positions in this function start at 0 (the least significant LED), and can go as high as 239 for 8 bar graphs (this routine will take into account the extra 2 outputs in each bargraph so you don't have to). Once you're done painting the canvas, you can transfer it to the bargraph with BG.send(). The bargraph will display what you've sent to it until you update it with another BG.send();

```
BG.sendLong(unsigned long number);
```

BG.sendLong() sends an unsigned long number (32 bits) directly to the display. It displays the data immediately so you do not need to call send() afterward. A long variable is 32 bits, or four bytes, so it fits the 32-bit shift registers exactly. The least significant bit will appear in LED 0, and where a bit is 1, that LED will be on. You can use this command to show the binary representation of a variable, demonstrate binary counting, binary arithmetic, bit-shifting, etc. See the example code for some ideas. This command is most effective when using a single display.

Bargraph Breakout Kit

2011.12.27

Writing your own interface

If you're using a non-Arduino system, or don't want to use the supplied library, don't worry, it's very easy to send data to the bargraph. Most processors have either a SPI or SSI module built into them which can be used to send data to the bargraph. When configuring your SPI port, remember that data is clocked into the ICs on the low-to-high transitions, and that the maximum clock speed is 25MHz.

You can also write a routine to manually send data to the bargraph. (See the "bargraph_nolibrary_example" sketch for example code.) You'll need three output lines, for CLK, SIN, and LAT. For each bit you want to clock in, make the CLK line low, set the SIN line to the state you want the next bit to be, make the CLK line high (this actually loads the SIN bit into the board), and repeat. Once you have clocked in all 32 bits, make the LAT line low then high to make the data appear on the LEDs.

Remember that the bits you send are shifted from one position to the next, so the last bit you send will end up in LED 0.

Daisy-chaining bargraphs

These boards are designed to be easily stacked to create bargraphs of any length. The Arduino library supports up to 8 boards (be sure to declare how many you have using the begin command). If you want to connect more than 8 you certainly can, but you'll need to write your own code or modify the library. (To keep memory requirements down, we use 8-bit values in the library, which can only support up to 255 LEDs.)

To connect multiple boards together, simply connect the OUT connector of one bargraph straight across to the IN connector of the next. In most cases you'll only need to connect the first five pins (+5V, GND, SIN/SOUT, LAT, and CLK). Note that SOUT on one board goes straight to SIN on the next.

If you write your own software, remember that you'll need to send 32 bits for each of the attached boards (for two boards, send 64 bits, etc.) Also remember that bits 30 and 31 in each board are not connected to LEDs, but are still necessary to make everything line up correctly, so don't forget to include them in your calculations.

Also, remember that more bargraphs use more power. Each board will use up to 250mA when all the LEDs are turned on, so be sure that your 5V power supply can handle it. An Arduino can safely handle up to two bargraph boards when using a wall-wart or USB cable. If you want an Arduino to run more boards, power the displays from a separate 5V supply, and connect a common ground between the Arduino and the bargraphs.

Multiple bargraphs

You can connect multiple independent bargraphs to a single Arduino by declaring multiple bargraph objects:

```
SFEbarGraph BG1;
SFEbarGraph BG2;
```

Then setting them up to use separate latch pins:

```
BG1.begin(1,10); // one board, pin 10
BG2.begin(1,9); // one board, pin 9
```

To use the bargraphs, just call them by name; the library will handle all the details:

```
BG1.barGraph(x,0);
BG2.barGraph(y,0);
```

Tips:

Although this board is designed for 5V, it will work at 3.3V. However, the yellow and green LED bargraphs may be dim due to their higher forward-voltage requirements. If you wish to run the board at lower voltages, consider using only red LEDs, which have a lower forward voltage than the other colors.

Questions?

We're happy to help with any issues you have, and we love to hear about your projects! Contact us at techsupport@sparkfun.com, or seek other users advice at <http://forum.sparkfun.com>.

Have fun!

- Your friends at SparkFun.