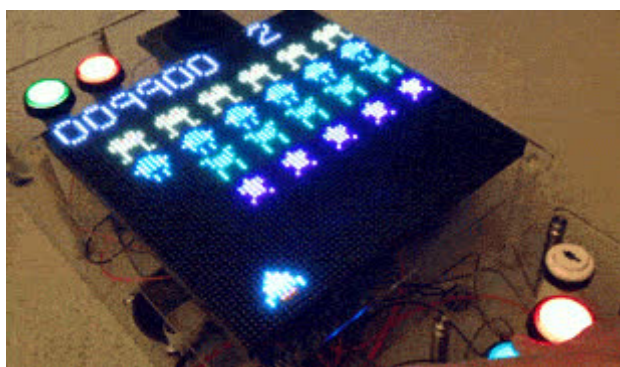




LedGames - a BeagleBone Black 64x64 LED Game

Created by Keith Henrickson



<https://learn.adafruit.com/ledgames-beaglebone-black-64x64-led-game>

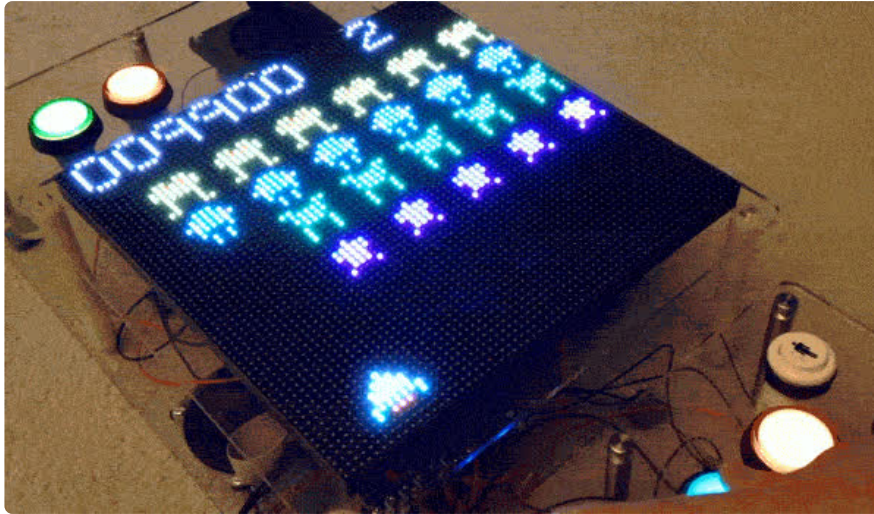
Last updated on 2024-03-08 02:05:34 PM EST

Table of Contents

Overview	5
<hr/>	
<ul style="list-style-type: none">• What is it good for?• How does it work?	
Parts List	7
<hr/>	
<ul style="list-style-type: none">• Disclaimer• Adafruit Parts• Arcade Controls• Other Electronics• Acrylic• General Hardware (it is cheaper to order full lots)• Alternative Parts	
Preparing the Cape	9
<hr/>	
<ul style="list-style-type: none">• Before you build this section• Assembling the Cape• Constructing the LED connector• Wiring the controls• Wiring the Audio	
Installing Software	14
<hr/>	
<ul style="list-style-type: none">• Installing Debian Linux• Upgrade the BeagleBone Black from wheezy to jessie• Disabling HDMI and friends• Enabling USB audio• Installing LEDGames	
Make the Power Bus	17
<hr/>	
<ul style="list-style-type: none">• Method 1• Method 2	
Smoke Testing	20
<hr/>	
<ul style="list-style-type: none">• Powering the test• LED Data Cable• Attach the cape• Attach USB audio• Apply power, and test!• Autostarting...	
Cutting Acrylic	23
<hr/>	
<ul style="list-style-type: none">• Cut the LED Mount• Cut the Base Plate	
Building the Display	25
<hr/>	
<ul style="list-style-type: none">• Wiring the display• Alignment	
Building the Control Panels	28
<hr/>	
<ul style="list-style-type: none">• Wiring• Testing	

Making Sound	30
Placing Electronics	32
• Audio Electronics	
• Power	
Placing Hardware	33
• The bitter end...	
• The very last step	
New Games	35
• Audio	
• Controls	
• Menu Entry	
Downloads	38

Overview



The goal of this project is to build a two-player 'cocktail' style game, with a 64x64 LED display, and arcade controls. The game system has stereo sound, and sits on a tabletop to save space and materials. The number of included sample games is small, but will hopefully grow over time.

This was not an easy project for me to make, and will require some design work on your part in order to successfully build it. Some of the parts can come in various flavors, and it can be hard to tell if you will be getting just the right thing. My advice is to go in this order:

- Read all sections carefully. Do not simply order everything on the parts list without understanding why you will use it.
- Order all the parts you think you will need, from a given supplier. Let it come in, inventory what you have, test fit things together.
- If the parts seem to fit, move on to the next supplier. If a part isn't what you expected or needed, maybe something similar can be added to your next order.
- If a part needs a larger space, or a bigger hole, than is accounted for in the acrylic cutting, adjust the files as needed.
- Finally cut the acrylic and hold your breath that everything fits.



This is the pile of leftover, unused, and unworkable parts that I created while designing this project. This suggests that this is not an easy build.

What is it good for?

I'm glad you asked. With a 64x64 resolution, what sort of games can fit in such a limited space? A surprising amount, although the graphics will be very crude. This is something that you can turn into your advantage. Not an artist, but always wanted to create a game? With graphics this primitive, no one is a great artist.

Extremely low-resolution games have shown up in many places. If the game is fun, HD graphics aren't needed to express it. Think of your favorite game, reduce it to it's bare essentials, use colors instead of resolution, and see what you can make go.

How does it work?

There is a set of software for the BeagleBone Black, called LEDscape, which uses one of the secondary CPU units (called a PRU), to drive attached LED matrix displays. By disabling the BeagleBone Black's HDMI port and a couple of other resources, enough pins can be freed up to drive 64 LED matrix displays.

I didn't want that many panels, 4 would do, so I spent some time finding a way to get it to do **LESS** than the maximum. I made a fork with the LEDscape software configured to support one chain of 4 displays, and included some sample games along with some software to read the control panels.

A link to the repo with the original code can be found in the downloads page, along with a link to my own fork, which is preconfigured for this project.

Parts List

This list is a best attempt at identifying suitable parts for this project. Some things might be missing or extraneous!

Disclaimer

When I link directly to a part, I am simply indicating which part I used, to the best of my documentation or recollection. This is not a 'kit', with all parts collected and guaranteed to work together. In most cases, many parts will work, however, some editing of the laser cutting files may be required. In just as many cases, a subtle difference between one part and another may cause the part to not fit. It can be difficult to determine that you have the 'right parts' without ordering things and trying it out.

Adafruit Parts

- BeagleBone Black. I used a Rev B, but a Rev C should work as well.
- 4x 32x32 LED Matrix Panels, with 4mm pitch (preferably a matched set)
- BeagleBone Prototype Cape
- 2x 3" 8Ω speakers
- 2x Power Distribution Bus
- 5V power supply
- Female DC power adapter
- Male DC power adapter
- Break-away male header

Arcade Controls

- 2x [Super Joystick](https://adafru.it/eMT) (<https://adafru.it/eMT>)
- 4x [Medium Pushbutton](https://adafru.it/eMU) (<https://adafru.it/eMU>)
- 4x (1 box of 10 will do) [6-volt lamps](https://adafru.it/eMV) (<https://adafru.it/eMV>)
- [Player 1 button](https://adafru.it/eMW) (<https://adafru.it/eMW>)
- [Player 2 button](https://adafru.it/eMX) (<https://adafru.it/eMX>)

Other Electronics

- 18" 16-position IDC cable (socket to socket, unshielded, pitch 0.100", polarizing key)

- 3x 4" IDC cable (possibly included with your LED matrix)
- [USB Audio Adapter \(https://adafru.it/eMY\)](https://adafru.it/eMY) (C-Media chipset works natively)
- 3.5mm male headphone plug

Acrylic

- 24"x18" cast acrylic, 1/4" thick

General Hardware (it is cheaper to order full lots)

- 12x Acrylic Standoffs, 12mmx80mm
- 12x #6 screws
- 12x #6 washers
- 12x #6 nuts
- 4x M3 screws, 12-mm below head
- 4x M3 standoffs, 18-mm length, female-female
- 20x M4 screws, 16-mm below head
- 8x M4 standoffs, 30-mm length, female-female
- 4x M4 hex-nut, 7.01-mm width
- 4x M6 standoffs, 64-mm length, male-female
- 4x M6 standoffs, 51-mm length, female-female
- 8x 3/8" binding posts with screw
- 1/4" quick-disconnect crimp connectors
- .187" quick-disconnect crimp connectors
- Rubber Bumpers(feet), 1/2" thick
- Stranded Wire, various gauges 18-26, various colors
- Nylon spacers and washers, variety of sizes from very thin to 1/4".

Alternative Parts

- 10-position barrier block
- Jumpers for the barrier block
- 1/4" spade crimp connectors

Preparing the Cape

Before you build this section

You are going to need some form of:

- Beaglebone Proto Cape
- Male 0.1" header
- Small gauge wire
- 16-conductor ribbon cable

Assembling the Cape

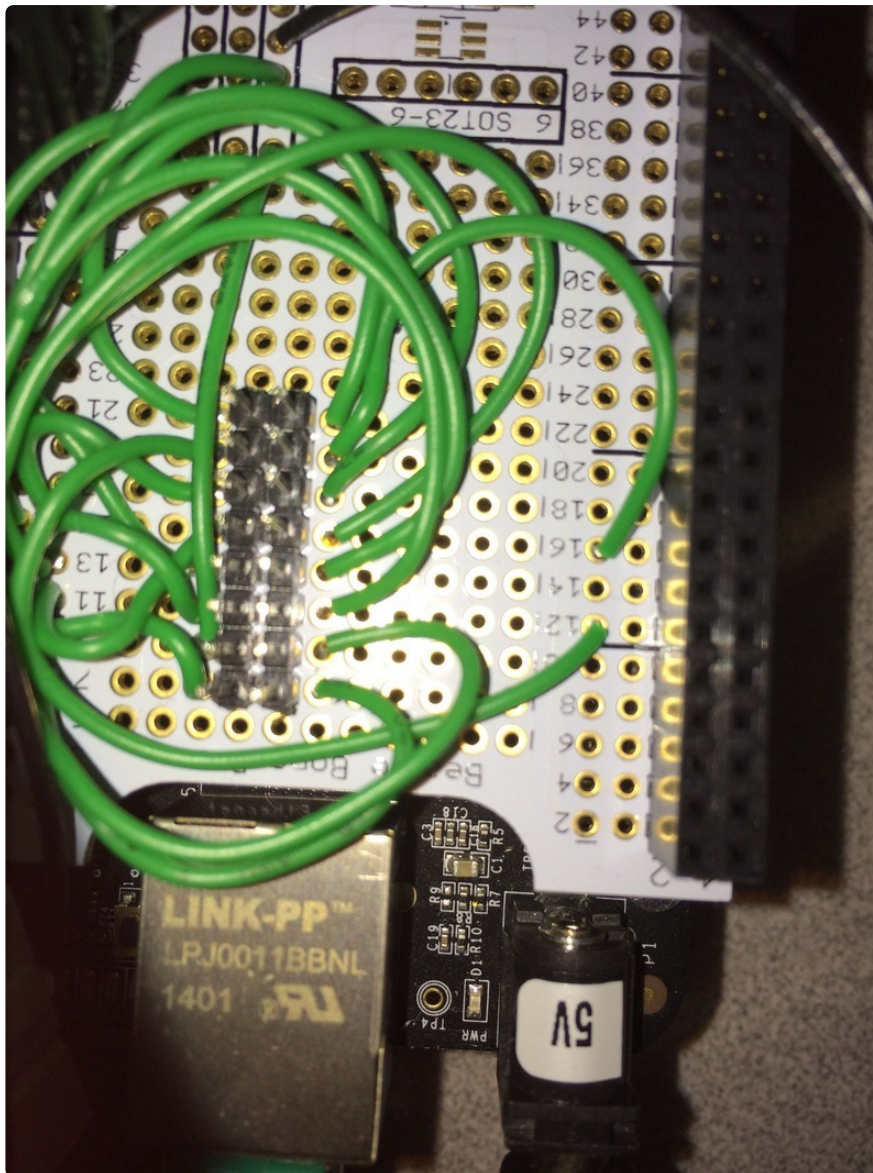
Start by assembling the cape itself. Your cape should at least come with male headers, or possibly male-female headers that allow stacking. I prefer the stacking connectors, as they allow you to shove jumper wires in to test a connection here or there.

There are two connectors, each with two rows of 23 pins each. That means you must make 92 solder joints, while keeping all of the pins perfectly vertical. If you have the male-female stacking connectors, the best way is to insert both connectors into the proto cape, and lie it upside down and work from the bottom. If you have the non-stacking connectors, insert the connectors and solder from the top.

Once you are done, check continuity between the holes where the connectors are inserted, and the pads where the signals are available to you. I had a couple of spots that I missed and skipping this created some troubleshooting headaches for me.

If all the solder joints are good and the signals are arriving on the proto cape, carefully test fit the cape onto the BeagleBone Black. There is a fine line between "enough force to make 96 pins fit in" and "enough force to break one or both boards". Work carefully, and you should be able to verify that all the pins are in alignment. Separate the boards again.

Constructing the LED connector



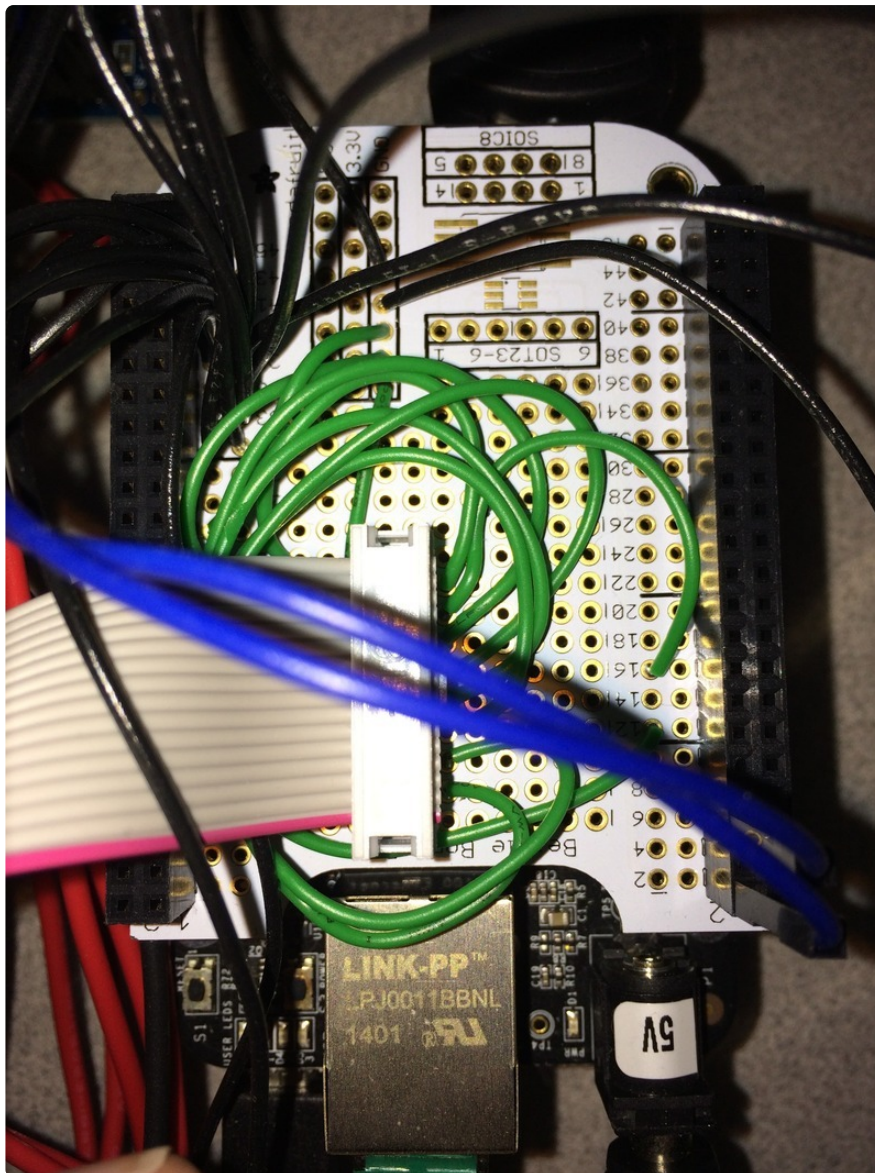
Find a free spot on your proto cape, and install two rows of 0.1" header. Take your narrow-gauge wire, and make the following connections. I used 26 AWG, solid core wire, but I did not have a very easy time making these connections. Any technique that works for working on protoboard will work fine here, as there are no electronics involved, just some electrical connections.

<ul style="list-style-type: none"> • Led Connector • Row 1, Left Pin • Row 1, Right Pin • Row 2, Left Pin • Row 2, Right Pin • Row 3, Left Pin • Row 3, Right Pin • Row 4, Left Pin • Row 4, Right Pin • Row 5, Left Pin • Row 5, Right Pin • Row 6, Left Pin • Row 6, Right Pin • Row 7, Left Pin • Row 7, Right Pin • Row 8, Left Pin • Row 8, Right Pin 	<ul style="list-style-type: none"> • BBB cape • P8-7 • P8-9 • P8-8 • GND • P8-13 • P8-14 • P8-10 • GND • P8-12 • P8-11 • P8-16 • P8-15 • P9-16 • P9-12 • P8-26 • GND 	<ul style="list-style-type: none"> • Signal meaning • Red 1 • Blue 1 • Green 1 • Ground • Red 2 • Blue 2 • Green 2 • Ground • Select A • Select B • Select C • Select D • Clock • Latch • Output Enable • Ground 	<ul style="list-style-type: none"> • BBB signal • Gpio2[2] • Gpio2[5] • Gpio2[3] • Ground • Gpio0[23] • Gpio0[26] • Gpio2[4] • Ground • Gpio1[12] • Gpio1[13] • Gpio1[14] • Gpio1[15] • Gpio1[19] • Gpio1[28] • Gpio1[29]
---	---	---	---

This is an area where you must proceed slowly. Not all LED matrixes are the same. There is a [good article \(https://adafru.it/Cfb\)](https://adafru.it/Cfb) on the number of combinations you may find. I wired my panels for the green/blue swap that article describes, as they seemed to need it.

Wiring the controls

Ideally, you should do similarly to the above, and wire up two male headers, one for each player, and then attach the player controls. After burning my fingertips too many times, and making other errors while assembling the LED connector, I got lazy and soldered wires directly to the cape. The wire ordering is a little odd, as I went from "a little lazy" to "I'm finishing this today at whatever the cost." Everything can be fixed in software.



Do not hook anything to the other end of these wires yet. The control panels come in a later step. You may wish to mark the wires with labels, or pieces of heat shrink tubing, so that you don't have to trace the wires again later. Also, attach two lengths of wire to the GND pads. This will form the 'other end' of the controls.

<ul style="list-style-type: none"> • Player 1 • Up • Down • Left • Right • Action - Primary • Action - Secondary • Start - 1 Player • Start - 2 Players 	<ul style="list-style-type: none"> • Beaglebone Black Pin • P8 - 38 • P8 - 44 • P8 - 41 • P8 - 42 • P8 - 39 • P8 - 40 • P8 - 45 • P8 - 46 	<ul style="list-style-type: none"> • Beaglebone Black Signal • Gpio2[15] • Gpio2[9] • Gpio2[10] • Gpio2[11] • Gpio2[12] • Gpio2[13] • Gpio2[6] • Gpio2[7] 	<ul style="list-style-type: none"> • Software Gpio • 79 • 73 • 74 • 75 • 76 • 77 • 70 • 71
<ul style="list-style-type: none"> • Player 2 • Up • Down • Left • Right • Action - Primary • Action - Secondary 	<ul style="list-style-type: none"> • BeagleBone Black Pin • P8 - 35 • P8 - 31 • P8 - 33 • P8 - 34 • P8 - 32 • P8 - 36 	<ul style="list-style-type: none"> • BeagleBone Black Signal • Gpio0[8] • Gpio0[10] • Gpio0[9] • Gpio2[17] • Gpio0[11] • Gpio2[16] 	<ul style="list-style-type: none"> • Software Gpio • 8 • 10 • 9 • 81 • 11 • 80

These pin assignments reuse pins that have other uses on the BeagleBone Black. The HDMI display must be disabled. Also, the controls will affect the state of the SYSBOOT pins, possibly causing bootup issues.

Wiring the Audio

Luckily, this part is easy. Simply attach a wire to the +5V pads, and a wire to the GND pads. Make sure that the other end is insulated so that you don't short anything out later.

Installing Software

Installing Debian Linux

First, install an 4GB MicroSD card in the MicroSD card slot. This is required on the Rev B and Rev C. The pins used by the eMMC module are also used by the LED port, and both devices cannot share those pins. For information about getting Debian Linux onto a MicroSD card for the BBB, please look at the wiki: [BeagleBoard Wiki \(https://adafru.it/eN0\)](https://adafru.it/eN0)

Either get the latest stable image, and upgrade it in the next step, or get the latest jessie image, and skip the next step.

Upgrade the BeagleBone Black from wheezy to jessie

First, get your BBB powered up and attached to the network. There are many, many guides on how to bring up a BBB for the first time, so I will not reiterate it here. Do not have the cape that you made in the previous step attached at this point.

Once the BBB can access the Internet, the installed Linux must be upgraded from wheezy to jessie, as the sample games use C++11, which is 'experimental' in wheezy. To do this, start by changing your `/etc/apt/sources.list` to the following:

```
deb http://ftp.us.debian.org/debian/ jessie main contrib non-free
```

```
#deb-src http://ftp.us.debian.org/debian/ jessie main contrib non-free
```

```
deb http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
```

```
#deb-src http://ftp.us.debian.org/debian/ jessie-updates main contrib non-free
```

```
deb http://security.debian.org/ jessie/updates main contrib non-free
```

```
#deb-src http://security.debian.org/ jessie/updates main contrib non-free
```

```
#deb http://ftp.debian.org/debian jessie-backports main contrib non-free
```

```
##deb-src http://ftp.debian.org/debian jessie-backports main contrib non-free
```

```
deb [arch=armhf] http://debian.beagleboard.org/packages jessie-bbb main
```

```
#deb-src [arch=armhf] http://debian.beagleboard.org/packages jessie-bbb main
```

Then do:

```
apt-get update
```

Once that completes, do:

```
apt-get dist-upgrade
```

I'd like to say that this process is straightforward, but I have to report that it took hours, and had to resolve certain issues manually. My guess is that any good guide on upgrading from wheezy to jessie can help you here.

Disabling HDMI and friends

There are several on-board devices that will not be used for the purposes of this project, and the pins need to be freed up. To do this, edit the file at `/boot/u-boot/uEnv.txt` and change the following section to uncomment the `cape_disable` line:

```
##BeagleBone Black:
```

```
##Disable HDMI/eMMC
```

```
cape_disable=capemgr.disable_partno=BB-BONELT-HDMI,BB-BONELT-HDMIN,BB-BONE-EMMC-2G
```

Enabling USB audio

The default configuration for ALSA causes it to prevent the USB audio adapter, which is installed later, from being device 0. That rule needs to be changed so that the USB adapter is the default. Change or create the file `/etc/modprobe.d/alsa-base.conf` to:

```
# autoloader aliases

install sound-slot-0 /sbin/modprobe snd-card-0

install sound-slot-1 /sbin/modprobe snd-card-1

install sound-slot-2 /sbin/modprobe snd-card-2

install sound-slot-3 /sbin/modprobe snd-card-3

install sound-slot-4 /sbin/modprobe snd-card-4

install sound-slot-5 /sbin/modprobe snd-card-5

install sound-slot-6 /sbin/modprobe snd-card-6

install sound-slot-7 /sbin/modprobe snd-card-7

# Cause optional modules to be loaded above generic modules

install snd /sbin/modprobe --ignore-install snd && { /sbin/modprobe --quiet
snd-ioctl32 ; /sbin/modprobe --quiet snd-seq ; ; }

install snd-rawmidi /sbin/modprobe --ignore-install snd-rawmidi && { /sbin/
modprobe --quiet snd-seq-midi ; ; }

install snd-emu10k1 /sbin/modprobe --ignore-install snd-emu10k1 && { /sbin/
modprobe --quiet snd-emu10k1-synth ; ; }

# Keep snd-pcsp from beeing loaded as first soundcard

options snd-pcsp index=-2

# Keep snd-usb-audio from beeing loaded as first soundcard
```



```
options snd-usb-audio index=0

# Prevent abnormal drivers from grabbing index 0

options bt87x index=-2

options cx88_alsa index=-2

options snd-atiixp-modem index=-2

options snd-intel8x0m index=-2

options snd-via82xx-modem index=-2
```

Installing LEDGames

The boilerplate stuff is out of the way for the moment, so let's try having some fun. The repository with all the software is at <https://github.com/KeithHenrickson/LEDscape> (<https://adafru.it/eN1>) and you can clone it with:

```
git clone https://github.com/KeithHenrickson/LEDscape.git
```

This is my fork of the original osresearch/LEDscape code, which has been changed to work a bit better with 32x32 panels, and is pre-configured for the matrix layout this project will be using.

Change to that directory and issue a 'make' command and ensure that everything builds. There are some warnings, but there should be no errors. If so, shut down Linux and power down for the moment.

This will install, but not activate, two cape definition files. They are activated later, but the build step does put them in place.

Make the Power Bus

Method 1

This is the method I chose. I used a 10-position barrier block, and used some jumper tabs to join together the rows that shared the same voltage. The jumper blocks should be sold along side the barrier block in the store (or ideally online). I do not

have a specific manufacturer to recommend here, as I grabbed whatever was in my local electronics store and no longer have the packaging.

Power Supply

This will be the input from the supply to the game. Make 4 lengths of 18 AWG wire, 2 red, and 2 black. I used about a foot or so. Trim some insulation from each end.

Now insert the two red wires into the + terminal of the Female DC Power adapter, screwing it down and checking that they are securely fastened. Insert the two black wires into the - terminal of the power adapter, screw it down and ensure that they are secure.

Put both red wires into a spade connector rated for 16 AWG and crimp it down. Put both blue wires into a different spade connector rated for 16 AWG and crimp it down.

Screw the positive spade connector in to the barrier block at one of the end terminals. Screw the negative spade connector into the barrier block 4 positions down. Install jumper blocks between terminals 1 and 2, 2 and 3, and 3 and 4. Install jumper blocks between terminals 5 and 6, 6 and 7, and 7 and 8.

Check to see that there is no short between the + and - rings of the DC Power adapter. If you find any shorts, check the wires, crimps, and jumper blocks.

Test Power Up

Plug the power supply into the female connector. Plug the power supply into the wall. With your voltmeter, check the voltage on each tab. You should find +5V between all terminals in rows 1, 2, 3, and 4 and all terminals in rows 5, 6, 7, and 8, and no connection on rows 9 and 10. If this all works, you have 8 terminals that supply +5V and 8 terminals with GND.



A picture is a thousand words. Here you can see the barrier block, the jumper blocks (the metal tabs that stick up and over, as well as the crimp connectors used to attach the wires to the barrier block.

I have gone ahead and attached the wires from the LED matrix panels. These wires were provided to me in the kits I received from Adafruit. Red wire to any +5V terminal, black wire to any GND terminal. Each wire assembly is capable of powering two panels.

One Last Step

There is still no power available to the BBB. Take a Male DC Power adapter and hook a red 18 AWG wire to the + terminal, and a black 18 AWG wire to the - terminal. Around 6 to 8 inches in length is enough. Attach spade connectors and hook them to available terminals on the barrier block as well. Plug this into the BBB, and it should fire up.

There will be no picture on the LED matrixes yet, but the power wiring is done. Power down and take a break.

Method 2

I have not tried this, so I can only give general guidelines. You may wish to use the Power Distribution Bus as sold by Adafruit. In this case, you should not need any spade connectors, simply screwing the wire into place.

Make all parts and connections as noted above, but with the following exceptions:

- You will need two Power Distribution Bus blocks, one for +5V and one for GND.
- Wires will need to be somewhat longer as the Distribution Bus blocks are installed some distance apart in the final design.
- Install no spade connectors, and cut off any spade connectors provided on the wiring to the LED matrix panels

Smoke Testing

At this point, there should be enough on the screen to test what has been done so far. Overall the test process is:

- Wire up power to everything
- Build a temporary data cable
- Power up the panels and the BBB
- Manually execute a game

Do not plug the USB cable into the BeagleBone Black if there is any possibility that the LED matrix panels do not power.

Powering the test

The BeagleBone Black can draw all its power off the USB cable, which is a nice feature for small projects. However, the LED matrix panels need more current than this can supply, and always having the USB plugged in may not be convenient. So the power will be supplied via the power jack.

A problem can arise if the panels are unpowered and the USB cable is plugged in. The BeagleBone Black will come up and start sending something to the panels, which can cause damage to the electronics of the panels. I have done it accidentally, and the panels are still fine, but I recommend avoiding this at all costs.

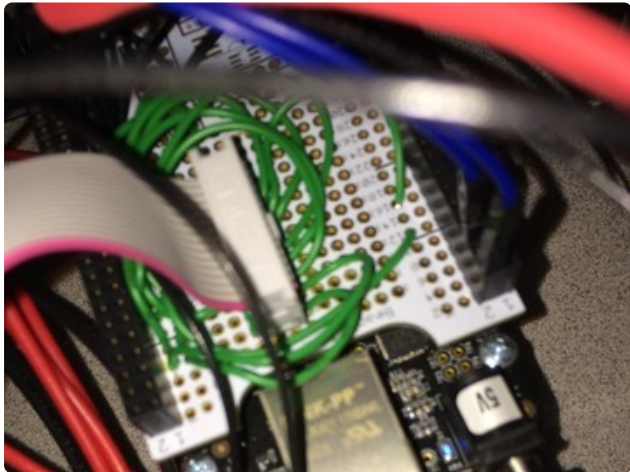
Start by making sure all four panels have the wiring attached, and that the wiring is attached to the power bus from the last step.

Next plug the Male DC Power adapter into the BeagleBone Black.

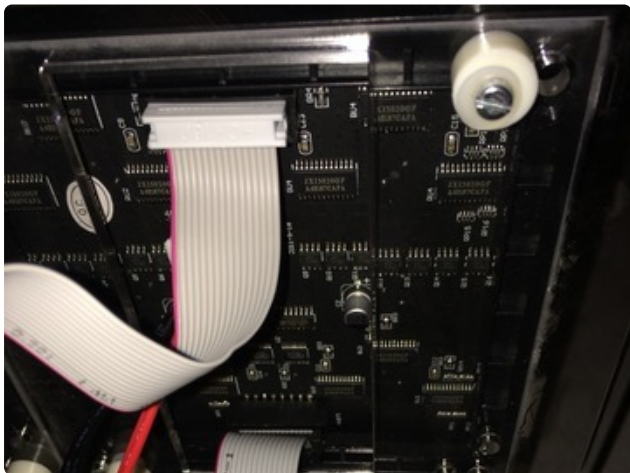
Finally plug the power supply into the Female DC Power adapter. Don't plug it into the wall yet.

LED Data Cable

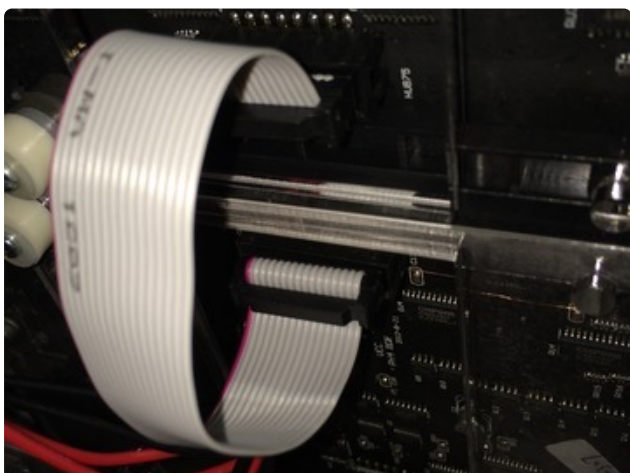
Before you begin this step, take the time to read [this article \(https://adafru.it/Cfb\)](https://adafru.it/Cfb) about wiring LED matrix panels. Hopefully this is just plugging in some cables, but if things look 'wrong', that article can help you correct any issues.



Take a nice, long 16-conductor ribbon cable and plug it in to the LED Connector on the proto cape you built earlier. This should be attached with the red wire at the 'top' edge, that is towards the notch in the PCB.



And this goes to the 'input' side of one of the panels. Any panel will do at the moment, but this will eventually be the bottom right panel.



For the other three panels, just hook the input of one panel to the output of the previous one. This will make a daisy chain going from the BBB, to the lower right panel, to the lower left, to the upper right, to the upper left. Since this is a test, don't worry about positioning them, just lay the chain out on your work table.

Attach the cape

Place the BBB on a firm surface, and make sure that there is nothing under it. Align the pins on the cape with the holes on the BBB, and press down. It will take a lot of force to get all the pins to settle in place.

Attach USB audio

The games depend on the USB audio card being installed, so attach it now. An earphone will be required to hear anything, but even without that, the games will be happy.

Apply power, and test!

Finally you get to plug the power supply into the wall. The lights on the BBB should start to flash as it boots up. Connect to the BBB with SSH, and change to the directory where you cloned the git repo. This is probably ~/LEDscape, so go there now.

To start one of the games, type:

```
bin/run-ledscape
```

After a short delay (about 10 seconds while the BBB injects the cape definition files), the LED matrices should light up, and you should see pieces of the "Bricks" game.

Autostarting...

Of course, a game should come up immediately upon booting, so that an SSH session isn't always required. To set this, do:

```
cp bin/ledscape.service /etc/systemd/system/
```

```
systemctl enable ledscape.service
```

```
systemctl status ledscape.service
```

Cutting Acrylic

Get a copy of the [LEDscape](https://adafru.it/eN1) (<https://adafru.it/eN1>) repo, with the following command:

```
git clone https://github.com/KeithHenrickson/LEDscape.git
```

In the directory 'acrylic', there are four files. One file cuts the mounting panel for the LED matrix panels, and is provided in version for CorelDraw X5 and X7. The other file cuts 'everything else', meaning the base plate and the control panels.

Before cutting acrylic, make sure you have obtained all hardware that will mount to that acrylic. This is doubly important if you will be sending the CDR files out to be cut elsewhere, as some parts come in many styles and it can be difficult to be sure that you will receive exactly what is shown in this tutorial.

Cut the LED Mount

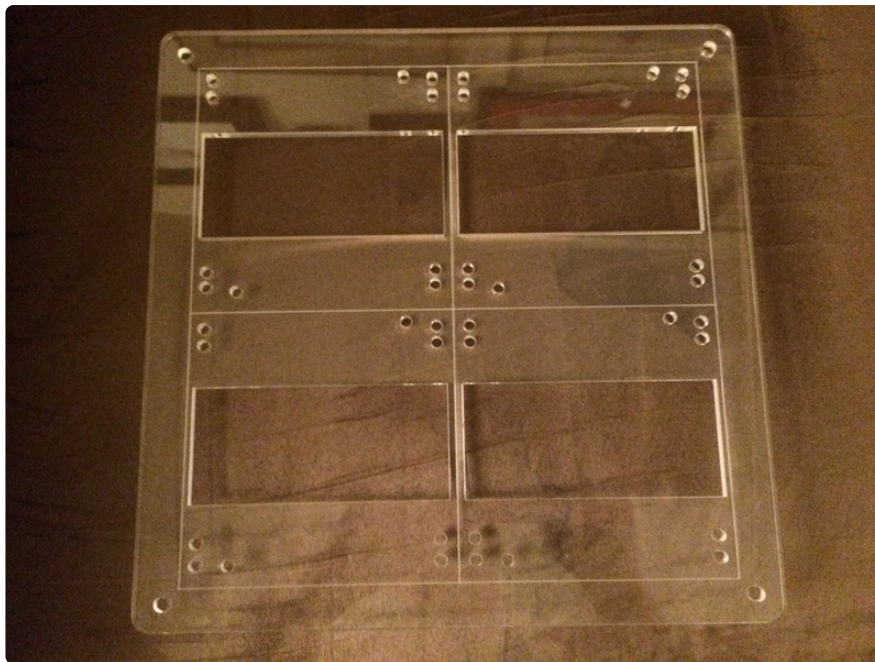
The first step is to cut the LED Matrix mount. This uses a 12x12 piece of 1/4" acrylic (or any piece with that much available room). Load the file LedMount-*.cdr, and you will find two layers:

- Guidelines
- Cutlines

The cutlines layer indicates the path that the laser should cut. The guidelines layer can be sent as lines to be etched, but should not be cut all the way through. This was done to allow the guidelines to be etched as vectors, rather than rastered (yes, there are better ways to do this, I didnt know at the time).

After cutting this, somehow temporarily mark which side is the top. The LED matrix panels will attach to this side.

You should end up with something a little like this:



Cut the Base Plate

This is a little more involved. Three pieces are cut from a single piece of 24"x18" 1/4" thick acrylic. Funnily enough, that is the EXACT size of the laser cutter I had access to. The job will use every last bit of this piece, so make sure that the laser cutter origin is set to the absolute edge of the piece. Even a slight margin, and the laser will skip right off the far edge of the piece.

Load the file `LedBasePlate-*.cdr`, and you will find two layers:

- Don't Cut Me
- Layer 1

Nobody ever said I was original. The "Don't Cut Me" layer is only there as a reminder for where things will go. It also holds all of the 'keep away' areas of each thing that gets installed, so that two controls don't get placed too close together.

Layer 1 contains both vector cutting, as well as raster engraving operations, so make sure that your laser software is set to do both operations. After completing this step, somehow mark which side is the top side, all components will be on this side.

You should end up with something SIMILAR to this, but note how the top edge is lopped off. That's what happens when the origin is not set correctly. This step is tricky. I went through three pieces of acrylic before I got one that was usable.



Building the Display

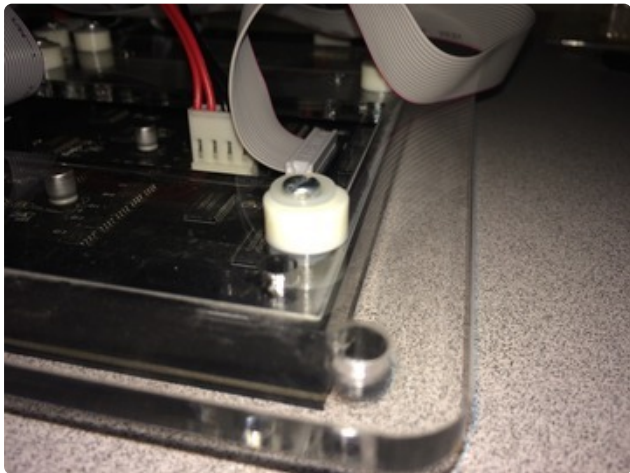
This step is the hardest to describe. You will definitely need:

- M3 screws (I am guessing 10mm would be the right length)
- M3 washers

Because the panels you have may not be a matched set, you may also need:

- The entire contents of a hardware store

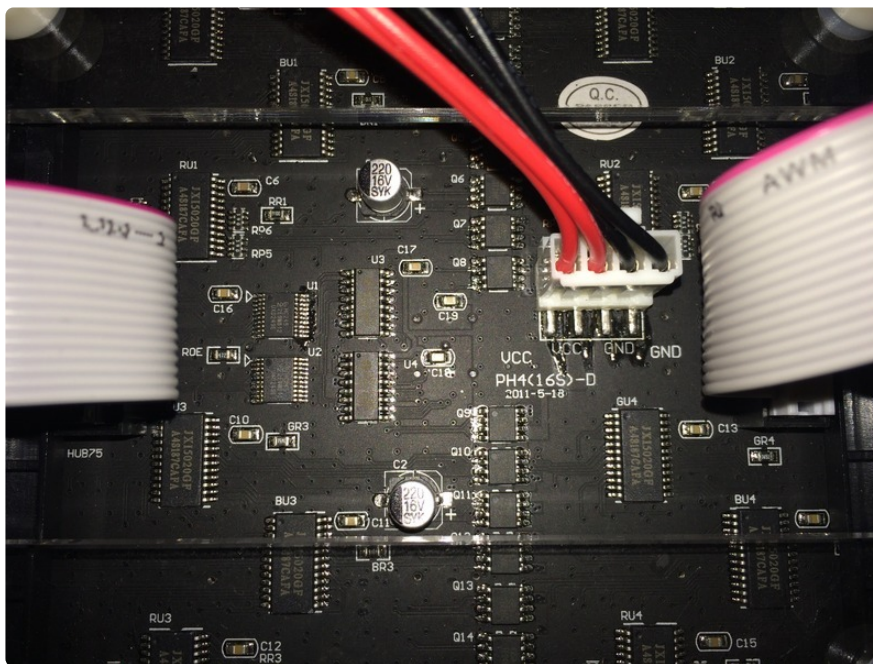
Just a little humor, as I found this step incredibly frustrating.



This picture shows how to screw the LED matrix panels to the acrylic.

My local hardware store only had very short and very long M3 screws, so as you can see in the picture, I used a large nylon spacer to make a gap. If I had ordered a bag of screws online, I would not have needed this spacer.

Start by positioning all of the panels LED side down, with the silk-screened text on the PCB normally readable, like this:

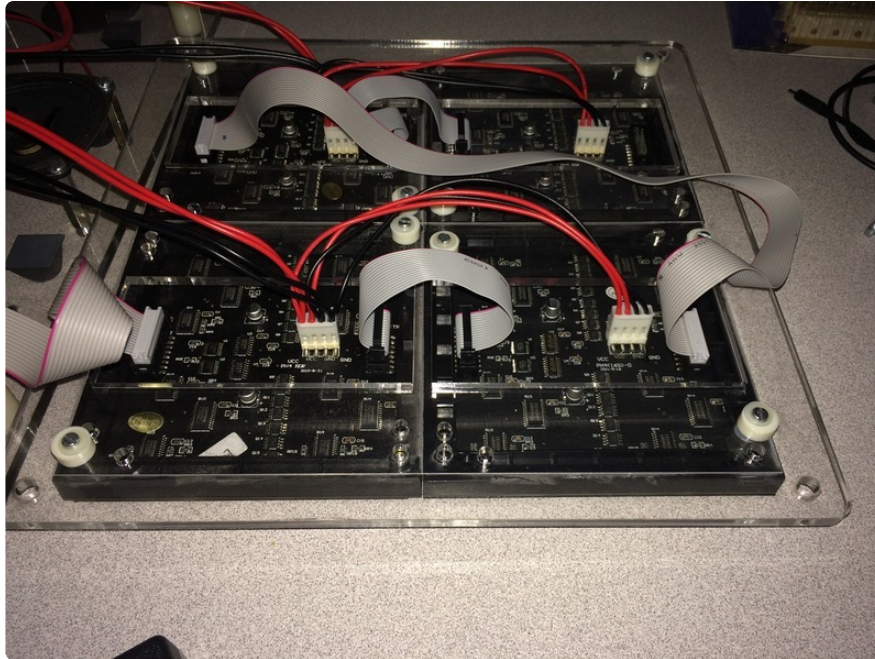


Position the panels in a 2x2 square, as close to each other as possible. Now carefully place the acrylic LED mount from the previous step over the panels. What was the upper surface of the acrylic in the laser cutter, should be touching the back of the LED matrix panels. What was the 'top' edge of the acrylic should remain the 'top' edge. Adjust the panels and the acrylic piece until as many holes as possible line up.

Not all of the holes are used for any given panel. The acrylic piece supports two different kinds of LED matrix displays. Also, some of the holes are used as alignment guides, with small plastic pins on each matrix fitting in to them to help you out.

When it looks like you've got it close, carefully start screwing the M3 screws in. Do not tighten them all the way down yet, there should be some slip in each panel. The holes in the acrylic are intentionally larger than required so that the display may be aligned later.

The final display should look like this:



Wiring the display

First attach two 4" ribbon cables between each horizontal pair, along the center line as shown above. Then take one 18" ribbon cable, and attach it from the lower-right connector to the upper-left connector. All of these connections are keyed, so the cables should only go in one way.

Attach another 18" ribbon cable to the only 'input' jack left, on the lower-left side of the display. This goes to the LED port on the Cape built several steps ago. The cape connector is not keyed, so be careful to plug the red wire in towards the top of the connector.

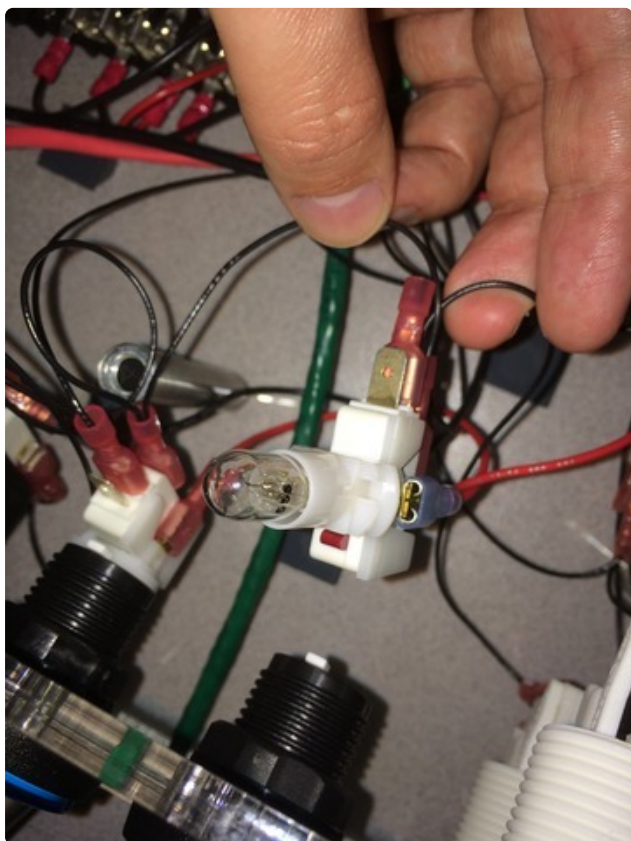
Last step is to hook up the power. Use the same cables from the "Smoke Test", and plug them in as before. The cables I used came with my LED matrix panels, and are shown above.

Alignment

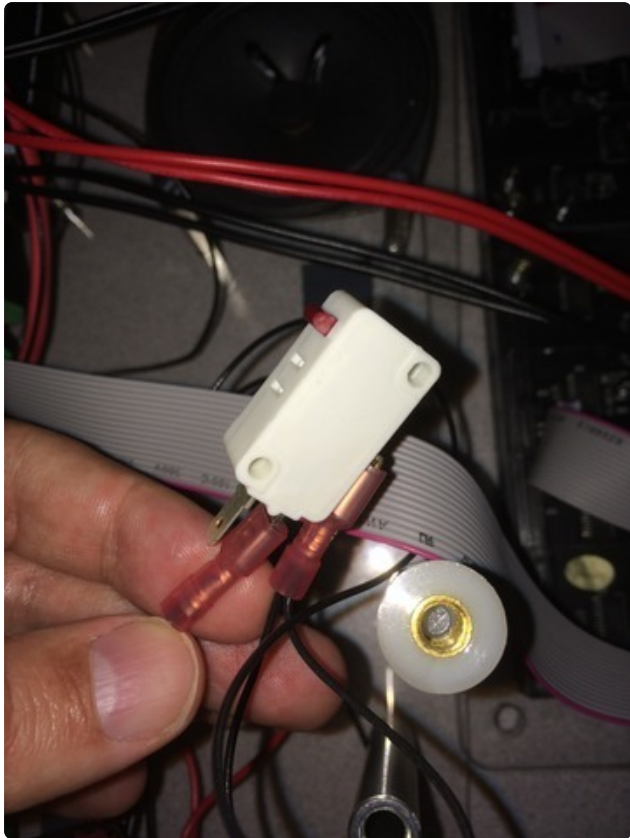
Almost there with a working display, just to align it. Repeating the smoke test should produce a working display at this point. Look to see if all rows and columns are nice and straight. Slowly tighten the screws down, adjusting the panels from time to time. Try and push them together to eliminate any gaps between each panel, and carefully rotate them if they appear out of line.

Building the Control Panels

That last step was difficult, so now it's time for something a bit more fun. Each control panel has a joystick, and two illuminated action buttons. The player 1 panel, has a 1 player and 2 player start button as well.



The illuminated buttons look nice, but ship with 14v bulbs in them. They will look awfully dim (if they illuminate at all) at 5v. They need to be replaced with 6v bulbs, which are pennies a piece. Simply give the base of the button a twist, and remove the entire lower assembly. Pop the bulb out, and replace it with a 6v bulb. Leave this apart for a moment.



The 1 player and 2 player start buttons need the microswitch itself removed from the button before installation.

The pushbuttons each come with a plastic nut that is used to attach them to the acrylic. Unscrew the nut from the button, insert the button through the acrylic, and put the nut on from the bottom. Install the lower assembly in each illuminated button, and reinstall the microswitch on the 1 player and 2 player buttons.

The joysticks have a complicated set of directions to assemble, which I am not going to repost here. Suffice it to say that the main body of the joystick is under the panel, but some pieces are inserted from the top. To secure the joystick in place, I used aluminum "binding posts", and found that 3/8" works the best. The advantage of the binding posts is that they have a smooth surface on the upper side. I have not, however, found a reliable source and my local hardware store seems to be perpetually out of the 3/8" length. Any screw and nut will do here, however.

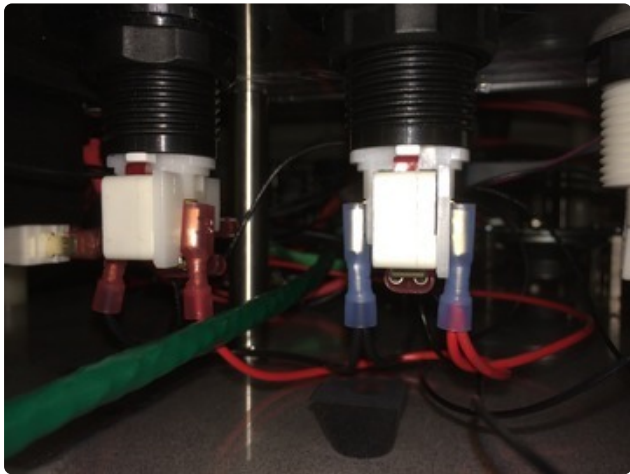
Wiring

This isn't such a bad job, if you kept good notes earlier when building the cape. There is only one tricky thing. The illuminated pushbuttons use **1/4" quick disconnect** connectors. The joysticks and start buttons use **0.187" quick disconnect** connectors. They do have holes, and you can solder to the tabs to save time. I used the connectors.

Each microswitch has a COM terminal. This will be a GND signal, connected to all switches in common. Using quick disconnect crimp terminals, hook a wire to each COM terminal, and then back to one of the GND wires on the cape that were installed during that step.

Now, take each of the signal wires from the cape, and used the appropriate sized quick-disconnect terminal to attach it to the N.O pin of the appropriate microswitch. Watch carefully with the joystick, as the switches are mounted strangely. It is probably easiest to experiment, push the joystick left, and see just which switch gets pressed.

The illumination is easy. To save on using any 'active' compoments, these are just wired straight to the power bus.



Make a short length and a long length of red 14 gauge wire, and a short length and long length of black 14 gauge wire. Using a blue quick disconnect, crimp the short and long red wire together. Do the same with the black wire.

Crimp a red quick disconnect onto the end of the short length of each wire. Crimp a spade connector onto the end of the long length of each wire. Attach the spade connectors to the power bus, and the quick disconnects to the side terminals of the illuminated buttons.

Testing

Again, repeat the smoke test. However, if the control panels are all wired correctly, the Bricks game should now be playable. Take a break and get a glass of lemonade. A lot is working.

Making Sound

I've produced sound by using the MAX98306 amplifier, but any solution will work. The actual sound comes out of a USB sound card, so any solution that takes a 3.5mm headphone jack will work.

Take a look at the assembly instructions for the MAX98306 breakout. +5V and GND will come from the spare wires attached to the Cape in that assembly step. I strongly advise using either jumper wires with female pins installed, or crimping them yourself, as being able to disconnect this makes assembly MUCH simpler. L- and R- will connect to the sleeve on the headphone plug. L+ will connect to the tip, and R+ will connect to the ring. Each headphone plug assembly may be different, so just check the datasheet for the selected plug.



I tagged the left channel wires with a green heat-shrink. For the wires, I used pre-cut and crimped jumper wires, which was expensive, but saved me the effort of crimping the female jumper connectors.

Do not forget to place the included jumper block over one of the sets of amplification select pins. I wasted a good deal of time troubleshooting the audio setup for having missed that.

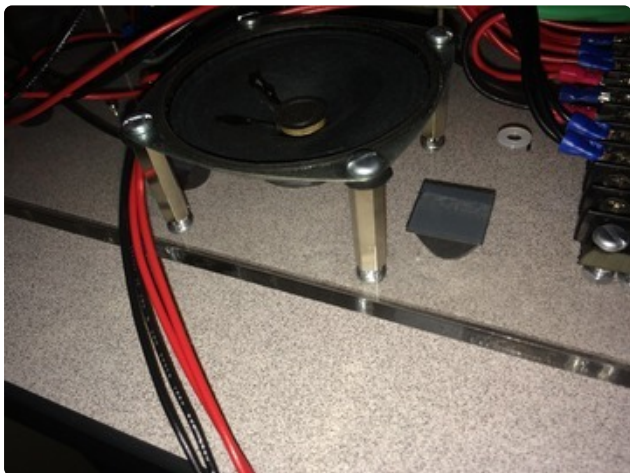
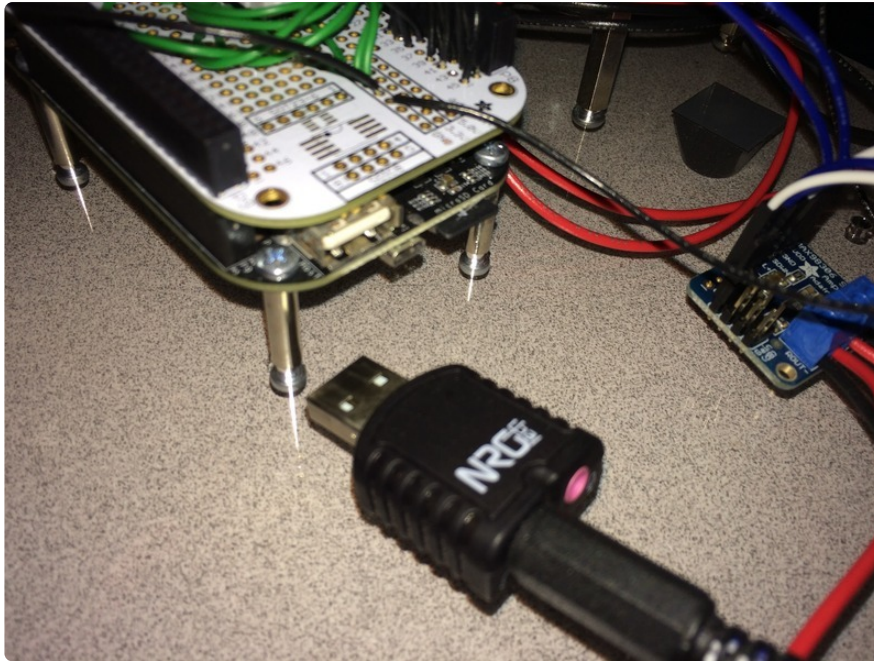
On each speaker, you will see + and - terminals. Either use a narrow gauge speaker wire, or simply solder some red and black 18 gauge wire to the speakers. The kit has screw terminals, as shown in the above picture.

Plug the headphone plug into the output jack of the USB audio card, and plug that into the BBB. Test it again, and there should be a 'bong' as the game starts up. The speakers do seem to emit several seconds of static until the sound card is initialized by the kernel. I have chosen to call this 'normal'. There is a 'shutdown' pin which could be wired to a spare GPIO to inhibit this during startup, but this is an exercise for the reader.

Placing Electronics

The next step is to place the power, BeagleBone Black, and the speakers onto the base plate. Things start to take shape in this step.

First, carefully separate the cape from the BBB. This takes a lot of effort to do, and care must be taken to not damage the boards or the pins as this is done. Use 4 M3 screws to attach 4 M3 standoffs to the bottom of the BBB.



Prepare the speakers for mounting by attaching 4 M4 standoffs with M4 screws to the bottom of each speaker like so. I added rubber washers between the speaker and the standoffs, hoping that this would prevent any strange vibrations. I'm not sure if it does the least bit of good.

Leave these screws somewhat loose at this point, as they will need to slide during installation.

Now position the BBB over the base plate. It goes in the very center of the large piece of acrylic, and is secured with 4 M3 screws. The ethernet jack should point towards the player 1 side. The standoffs are not symmetrical, so if things don't seem to line up, try flipping things around until they do.

Audio Electronics

Secure the speakers to the acrylic with 4 M4 screws each. I am not sure that the holes are positioned correctly in the cutting file. It took more 'fiddling' to make everything line up than I would have expected. By starting out with all of the screws very loose and slowly tightening them down, things seemed to go in place, and the speaker cones are not visibly bent. This is something to check before cutting acrylic.

There are holes in the base plate for the audio amplifier, but I haven't found standoffs that fit the holes in the amplifier board. It can just rest in place, as the wires are stiff enough to prevent it from moving around too much.

Power

The last step for now is to attach the power bus. I have pictured the barrier block that I used below. There are two places on the base plate where a barrier block can be mounted. The outer four holes will be used by a barrier block. Your barrier block may be different.

The inner two holes of the power mounting area are for the white Power Bus blocks. I believe that they fit, but have not used them in my assembly.

To attach either kind of block, use M4 screws, and M4 nuts. The holes are cut to this size in the base plate.

Once the blocks have been mounted, power can be run to the BBB, and the wires from the display panel can be attached.

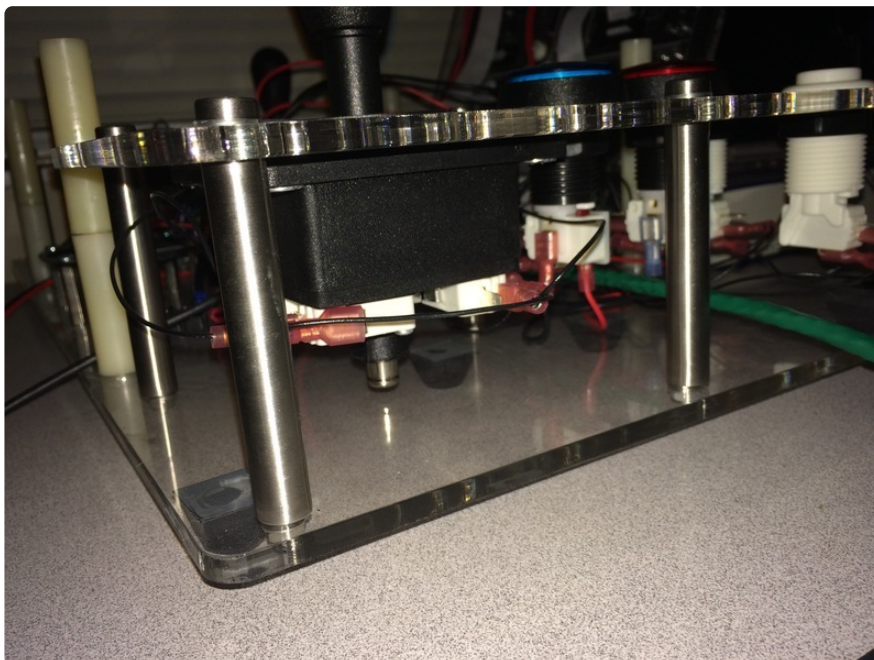
Placing Hardware

Finally the end of this project approaches. First build the supports for the LED display. The nylon standoffs from the parts list need to be screwed together, one male-female standoff is screwed into one female-female standoff, to make a longer female-female standoff. Each one of these is attached to the base plate with an M6 screw, using one of the four large holes that surround the electronics package.



In the picture, the seam where the two standoffs were screwed together is visible. Attach them slightly loosely to the base plate at this point, as things may have to shift when the matrix display is installed.

The control panels are attached with more attractive 80mmx12mm stainless steel acrylic standoffs. These are attached to the base plate with a #6 (not an M6) screw, and a #6 washer and nut. The screw is inserted down the hollow tube of the standoff, and put through the base plate. The nut and washer are attached from below.



I don't have a specific source for these standoffs, as finding stainless steel standoffs this long is often some type of special order. Make sure they are 80mm or longer, as the joysticks just about hit bottom.

The bitter end...

Finally, install the cape onto the BeagleBone Black. Position the control panels over the aluminum standoffs, Player 1 on the side with the ethernet jack. Screw the aluminum caps down to secure the panels, and tighten the nuts on the bottom of the base plate.

Attach the data cable to the cape, red wire towards the top of the connector, and set the display panel on top of the nylon standoffs. Attach it with M6 screws, and tighten the screws down on the bottom as well.

The very last step

Install rubber feet on the bottom of the base plate. Because some of the screws hang down a lot, the feet that are in the parts list are 0.52" high.

Power up, and enjoy!

New Games

Of course, a game system isn't any good without games to play on it. There are a couple of problems with this right now:

- It's new, so I've only written two games for it, and am working on a third
- The game engine... isn't.

Luckily, a complicated engine isn't required to get a game of your own running on here. The display is simply a 64x64 array, that is written with a single call. The engine just added some sprite-like copies for convenience later on.

The sample games should give you plenty of reference code for getting things on the display. At the core, you need to do:

```
int width = 64;  
int height = 64;
```

```

ledscape_config_t * config = &ledscape_matrix_default;

ledscape_t * const leds = ledscape_init(config, 0);

uint32_t * const p = (uint32_t*)calloc(width*height,4);

ledscape_draw(leds_, p);

```

The array 'p', is stored as XRGB, where X is "don't care", and R, G, and B are red, green and blue respectively. Each uint32 represents a single pixel.

Audio

Audio is done simply by using SDL in an audio-only mode. It speaks to the default audio device, which is provided here by a USB audio adapter.

Controls

The control panels simply show up as inputs to the Linux GPIO interface. The table of GPIO inputs is given in the Preparing the Cape section. The software is easier to use. To initialize it, do:

```

player_controls[0] = new controls_t(1);
player_controls[1] = new controls_t(2, true);
player_controls[2] = new controls_t(3);

```

The player number is passed in as the first argument to the controls_t constructor. Player 1 and 2 are obvious. Player 3 is the "1 Player" and "2 player" start buttons, as Primary and Secondary action button.

The 'true' option inverts the joystick. Because the players face each other, if players are playing simultaneously, one player is seeing everything upside down, and it can be helpful to invert the joystick.

Once each frame, the controls must be refreshed. This involves some amount of device file access, so it is only done on command:

```

player_controls[current_player]->refresh_status();

```

To get the state of the controls, just do:

```
player_controls[current_player]->is_pressed(joystick_left)
```

The complete list of controls is:

```
joystick_left  
joystick_right  
joystick_up  
joystick_down  
button_a  
button_b
```

Menu Entry

The last step is to add your new game to the menu. The file `Menus.png` is a 'sprite sheet', containing a series of 64x32 sprites. Each sprite should contain some interesting graphics, as well as the menu text, and left/right arrows. The sample menu contains 3 pictures.



The menu program simply loads the sprite sheet, and the code must be modified to change the menu items. The vector `'game_sprites'` holds all of the menu sprites, so just add any new sprites to that vector.

The menu program will return an exit code containing the number of the sprite selected. This is used by the `'run-menu'` script to actually run the next game.

Downloads

There are a few git repos which you will want a clone of, or at least a copy:

The original LEDscape project, of which this is a derivative:

<https://github.com/osresearch/LEDscape> (<https://adafru.it/eN7>)

The original LEDscape firmware

<https://adafru.it/eUA>

My fork of the LEDscape project, which is pre-set to the 64x64 panel layout that I've used here:

<https://github.com/KeithHenrickson/LEDscape> (<https://adafru.it/eN1>)

**Forked LEDscape, with sample
games**

<https://adafru.it/eN1>