V1.0- Initial commit

V1.1- Updated Parts/Links

***Assembly Guide Overview***

*The EduCase is a fully functional classroom inside a portable case. It contains multiple computers (Raspberry Pi & Android) loaded with content, tutorials, tools and information which educators can use in any area of the world.*

**

*The project is fully self-contained and self-powered and also includes provisions for communication with the outside world using either cellular data or a dedicated portable satellite dish and receiver linked to the "Outernet" satellite data provider. It also houses its own LCD display and utilizes an LED projector enabling the entire classroom to see the content.*

*The EduCase can be deployed to refugee camps, rural areas, developing nations or anywhere else around the world and used to provide content, communications, courses and material we take for granted in the western world.*

*The EduCase can also be deployed into disaster/relief zones and used for sharing news, communication or things as simple as an impromptu movie theater.*

# INTRODUCTION

## Basics

**This document is a work in progress and will be updated OFTEN. Thanks for your understanding!**

The EduCase will be able to deploy technical training, medical knowledge, communication, news, and content from all fields into any area of the world regardless of infrastructure.

## References

Parts Video:

<https://youtu.be/J0L335KiNEU>

isual and Audiong Display either the Lighthouse wifi hotspot or to the cellular network via the Android phoneng the volume avaContact

Support provided via the forum at <http://mkme.org> as well as the GIT repository.

Support for the libraries used to support the code is provided by the original authors only. Restrictions for use can be found on their respective support pages (linked within the Arduino Source Code and Raspberry Pi Files)

Forum Thread:

<http://mkme.org/forum/viewtopic.php?f=32&t=702>

# All files and source code:

All files and updates via GitHub:

<https://github.com/MKme/EduCase-Portable-Classroom>

## Project References

Initial design completed spring 2016. Testing and updates ongoing via Hackaday website:

<https://hackaday.io/project/11010-educase-portable-classroom>

## Assembly

1. Gather the required components for assembly into your case. Pelican style cases are preferred due to the depth and resistance to water intrusion. You can find many options here: <http://www.pelicancases.com/>
2. Assemble the batteries into the case as per the electrical schematic. You can use any time of battery you wish but it is preferable to use lead acid batteries which lend themselves better to these discharge cycles better than LiPo etc.

Ensure you fuse the battery output as per the schematic using protection of your choice. A PTC is preferable but and automotive spade-style fuse can also be used and are easily available from many sources

1. At this time it is best to decide if you will incorporate the Arduino controller into the case. It will be easier to wire the relay power control and temperature sensor now before you finalize the component locations inside the case. If you choose to use the control you can order the printed circuit board using the files I have provided on GitHub. Wire the components as per the layout in the code and schematics and flash the Arduino with the code provided. Download the Aurora firmware from Github and flash using the Arduino IDE from <https://www.arduino.cc/en/Main/Software>
2. If you implement the controller you may wish to mount your Nokia 5110 LCD permanently in the case. Within the files on Github there is a folder for CAD files and documents. Within it you will find a stylish bezel for the LCD. You can 3D print this yourself or use and online source such as Shapeways.
3. Customize the code as you wish to include time, temperature and voltage alarms and control. If you make changes please consider sharing them back with the world open-source.
4. Lay out the components inside your case as per your hardware. You can use the foam inserts and carve them out to snugly fit your monitor, Raspberry Pi and still leave room for storage of components and equipment such as webcams, remotes etc

Route the USB wiring, video and audio cables to suit your installation. I suggest leaving extra HDMI cable easily accessible so the projector and monitor can be used outside the case. This makes presenting to a classroom environment much easier as you can place the case away from your work/presentation area.

1. Add a powered USB hub with the number of ports needed for your particular project. I suggest at least 10. The power for the hub is supplied via the DC/DC converter as per the schematic.

These units can be found as “buck converters” from any electronics source. Some suggested versions are given in the parts list in this project.

Mount the hub so it is convenient to access at least a few ports during normal use for plugging in items for charging, webcams etc.

1. Install the Raspbian operating system on your Raspberry Pi. I have made a video detailing this here: <https://www.youtube.com/watch?v=29ejr-KFxvY>
2. You will want to be installing many files and learning content on the Pi so installing a remote VNC connection will make this much easier. Install remote desktop connection as per the video I made here:

<https://www.youtube.com/watch?v=P5lPmXTgwu0>

1. Copy all the files and content over to your Raspberry Pi. I also place some of the media files on the Android phone. It hooked up to the projector does a much better job at media file playback such as video/movies. I cannot provide the files I am using due to copyright etc so please don’t ask for them. Sorry.
2. If you are also constructing the portable satellite dish you will need to swap out the LNB that comes with the dish to a Linear LNB. They can be sourced easily on eBay or Amazon. Install the LNB in the same position as the original. The distance from the face of the LNB to the dish itself must remain the exact same to retain the correct focus. This can be done with a simple measuring tape.

## Glossary

**Arduino**- Arduino is an open-source computer hardware and software company, project and user community that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world.

The project is based on a family of microcontroller board designs manufactured primarily by SmartProjects in Italy, and also by several other vendors, using various 8-bit Atmel AVR microcontrollers or 32-bit Atmel ARM processors. These systems provide sets of digital and analog I/O pins that can be interfaced to various expansion boards ("shields") and other circuits. The boards feature serial communications interfaces, including USB on some models, for loading programs from personal computers. For programming the microcontrollers, the Arduino platform provides an integrated development environment (IDE) based on the Processing project, which includes support for C,C++ and Java programming languages.