Developer Guide – Computer, Arduino, and Bluetooth Module

Something_Cool

Ezra Kramer

933016089

System Overview

This project is a PC controlled power supply. Ideally used for low-power in-lab applications, this device provides an adjustable DC voltage through two independent channels, capable of 1.5A anywhere from 2V to 14V. Figure 1 below shows the blocks that make up the system, and provides a framework for understanding how the system operates.

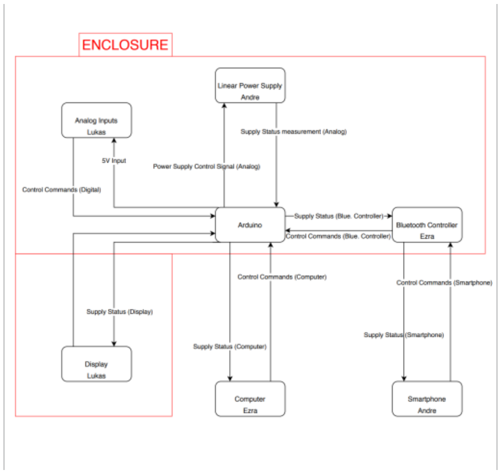

ENCLOSURE
Linear Power Supply
Andre
Analog Inputs
Lukas
Supply Status measurement (Analog)
5V Input
Power Supply Control Signal (Analog)
Control Commands (Digital)
Arduino
Supply Status (Blue. Controller)
Control Commands (Blue. Controller)
Bluetooth Controller
Ezra
Control Commands (Computer)
Control Commands (Smartphone)
Supply Status (Display)
Supply Status (Smartphone)
Supply Status (Computer)
Display
Lukas
Computer
Ezra
Smartphone
Andre

Figure 1: Full system block diagram

The most important block here is the block labeled Linear Power Supply, which using a full bridge rectifier and a control signal from the Arduino, modulates its two channels of output to produce the desired result. The Analog inputs include physical switches and dials that can be used to modulate the output voltage of both channels independently. The display displays the current status of the power supply – this includes the output voltage and current of each channel, as well as the current target output voltage, and any error messages or warnings such as over current or voltage notifications. The smartphone app builds off of the Adafruit Bluefruit libraries that are used in conjunction with the Bluetooth module we have selected, and provides a user interface to both monitor and control the device.

This developer guide will focus on the control systems that operate the power supply and interact with the user. These are the Computer and Bluetooth Controller blocks shown above in Figure 1. The central control unit for this power supply is an Arduino UNO. Through these systems, the user can monitor and control the power supply. The communication with the Arduino that allows this to happen is as follows.

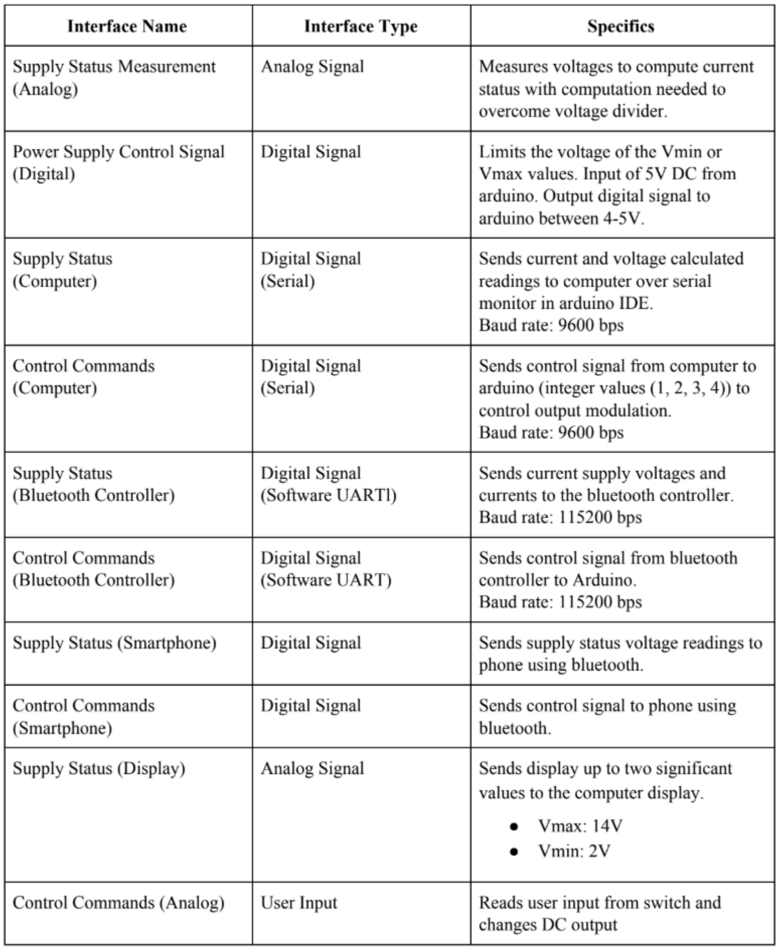
For the Bluetooth Module (Specifically the Adafruit Bluefruit LE UART Friend) This communication happens via software serial communication between the Arduino Uno and the

Bluefruit. On the smartphone end, a Bluetooth connection is established and communication takes place using the UART standard – terminating in an app developed by Adafruit designed to make this communication easy. The computer communicates with the Arduino via USB Serial terminating in the Arduino IDE Serial Monitor.

The data received by both the smartphone and the computer includes all aspects of the current state of the machine that could be relevant, including the output voltage and current for each channel, as well as the active target voltage, and any information regarding overvoltage, overcurrent, and establishing communication.

Electrical Specifications

- Output Voltage Range: 2-14V
- Output Current Range: 0-1.5A
- Operating Temperature Range: -40 to 85 degrees Celsius
- Serial Bodrate: 115200
- Output Refresh Rate: 0.5Hz (max)

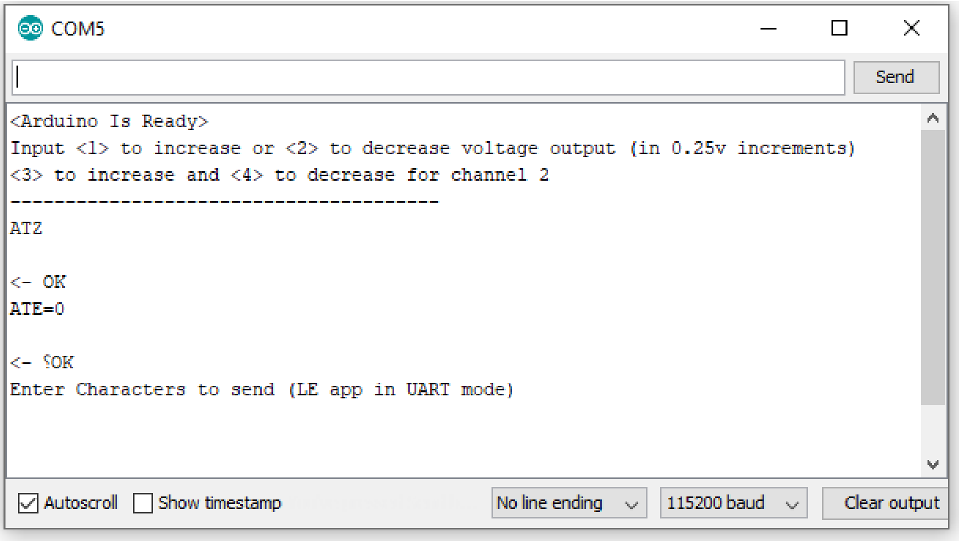


Interface Name	Interface Type	Specifics
Supply Status Measurement (Analog)	Analog Signal	Measures voltages to compute current status with computation needed to overcome voltage divider.
Power Supply Control Signal (Digital)	Digital Signal	Limits the voltage of the Vmin or Vmax values. Input of 5V DC from arduino. Output digital signal to arduino between 4-5V.
Supply Status (Computer)	Digital Signal (Serial)	Sends current and voltage calculated readings to computer over serial monitor in arduino IDE. Baud rate: 9600 bps
Control Commands (Computer)	Digital Signal (Serial)	Sends control signal from computer to arduino (integer values (1, 2, 3, 4)) to control output modulation. Baud rate: 9600 bps
Supply Status (Bluetooth Controller)	Digital Signal (Software UARTl)	Sends current supply voltages and currents to the bluetooth controller. Baud rate: 115200 bps
Control Commands (Bluetooth Controller)	Digital Signal (Software UART)	Sends control signal from bluetooth controller to Arduino. Baud rate: 115200 bps
Supply Status (Smartphone)	Digital Signal	Sends supply status voltage readings to phone using bluetooth.
Control Commands (Smartphone)	Digital Signal	Sends control signal to phone using bluetooth.
Supply Status (Display)	Analog Signal	Sends display up to two significant values to the computer display. ● Vmax: 14V ● Vmin: 2V
Control Commands (Analog)	User Input	Reads user input from switch and changes DC output

Figure 2: Interface Definitions for individual parts of the system

User Guide

Using this device requires a USB serial connection to a computer with the Arduino IDE. If this is the first time using this system, set the “#DEFINE FACTORYRESET_ENABLE” to 1 and hit compile and run. This purges any settings preloaded into the Adafruit Bluefruit, and will ensure that the system functions predictably. Open the serial monitor to see the output of the Arduino as it attempts to connect to the Bluefruit. If this is successful you should see “Enter Characters to send (LE app in UART mode)". Turn on the Bluetooth on your smartphone and connect to the “Bluefruit LE” in the connect tab of the Bluefruit Connect app. Once a connection is established, press on the “UART Communication” tab. This should open a text window and keyboard, and data from the Arduino should begin printing both to the serial monitor and the smartphone screen. If this happens the setup has been successful.



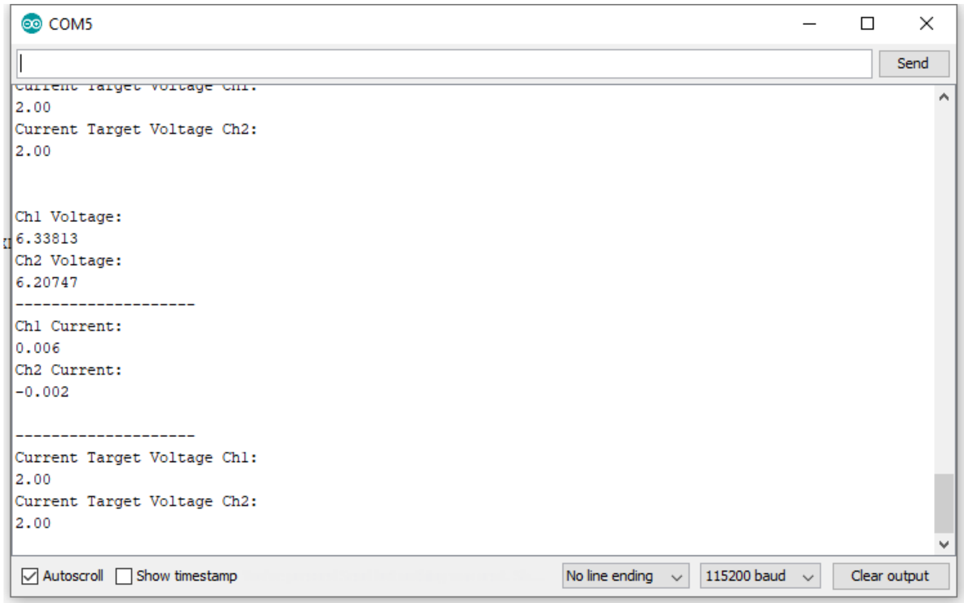
```
<Arduino Is Ready>
Input <1> to increase or <2> to decrease voltage output (in 0.25v increments)
<3> to increase and <4> to decrease for channel 2
---------------------------------------
ATZ

<- OK
ATE=0

<- ?OK
Enter Characters to send (LE app in UART mode)
```

☑ Autoscroll □ Show timestamp No line ending 115200 baud

Figure 3: This is the output you should see in the serial monitor if the system is ready. Once you see this hit connect in the Bluefruit app.



```
Current Target Voltage Ch1:
2.00
Current Target Voltage Ch2:
2.00


Ch1 Voltage:
6.33813
Ch2 Voltage:
6.20747
--------------------
Ch1 Current:
0.006
Ch2 Current:
-0.002

--------------------
Current Target Voltage Ch1:
2.00
Current Target Voltage Ch2:
2.00
```

☑ Autoscroll □ Show timestamp No line ending 115200 baud

Figure 4: This is the output you should see once you hit connect (it is advised to keep Autoscroll on in the serial monitor as the data is refreshed in real time). The output seen in the UART tab of the Bluefruit app should be identical to the output seen on the serial monitor in real time.

Once set up, as long as the Arduino remains powered, the computer is no longer necessary, as all power supply updates as well as control signals can be done with the smartphone. The control signals are as follows:

Input	System Response
<1>	Increments the target voltage for channel 1 by 0.25V
<2>	Decrements the target voltage for channel 1 by 0.25V
<3>	Increments the target voltage for channel 2 by 0.25V
<4>	Decrements the target voltage for channel 2 by 0.25V

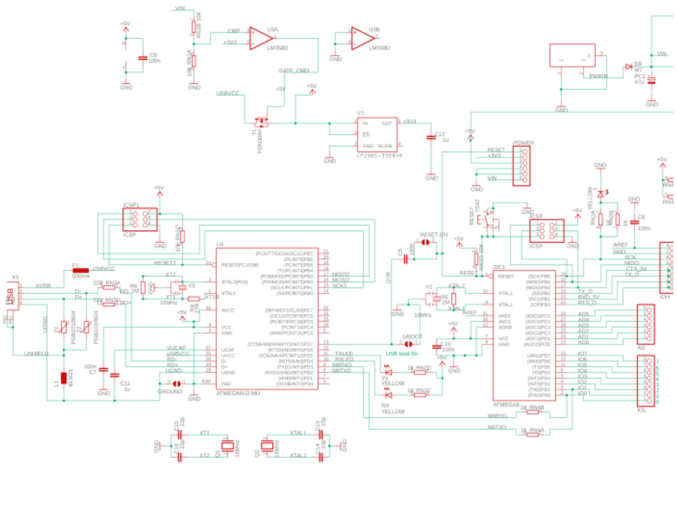
Figure 5: These control signals are all you need to control the power supply

These inputs can be entered into the smartphone text box, or the serial monitor window. The output of the power supply should change to match the target as long as the current draw and supply remain within the acceptable range (1V for draw, 1.5V for output).

Invalid input should not be a problem for either the serial monitor or the Bluefruit connect app, though you will see an error message, the system should continue to function normally. Additionally if the maximum current or voltage is exceeded, the supply will automatically reset the target to the minimum value (2V).

Design Artifacts

The following is a set of design artifacts that details the physical and electrical structure of the unit itself. See Figure 1 above for the block diagram of the overall system, and Figure 2 above for the interface definitions of those blocks. The artifacts shown below detail the Arduino, Computer, and Bluetooth blocks.



5/14/2020 1:38 PM C:\Users\Court\Documents\EAGLE\projects\Ezra\combined.sch (Sheet: 1/1)

Figure 6: System Schematic Part 1

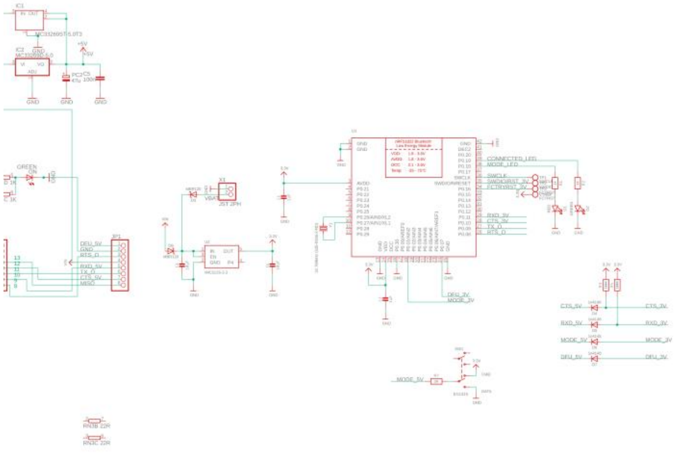


Figure 7: System Schematic Part 2

The above schematics detail the Arduino Uno, Adafruit Bluefruit, and PCB daughterboard used to connect them together. Key to note here is that the Bluefruit and Arduino work well together, requiring no analog parts aside from traces to pins in the PCB daughterboard. This makes the overall system cheap and efficient, using little in the way of custom parts and allowing for fast prototyping and assembly.

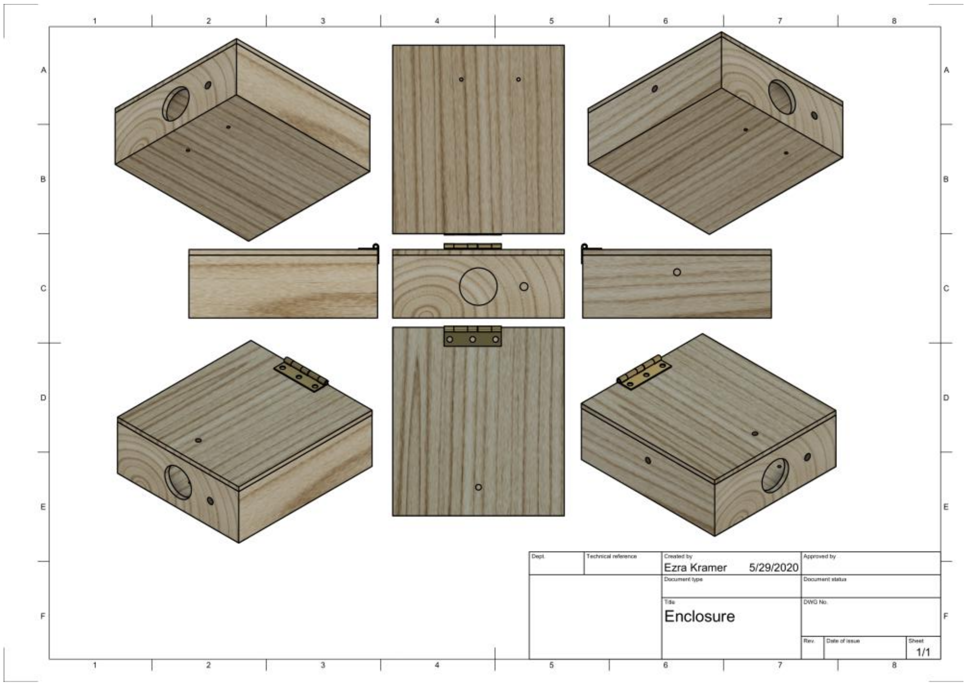


Figure 8: Enclosure CAD shows a view of the enclosure from each side.

This enclosure is robust and allows for routing of all cables used to connect the control structure to the rest of the device. The materials used are cheap, and the wood is thin enough that it can be laser cut and manufactured easily. Additionally, all other mounting hardware including hinges and screws is off the shelf and can be assembled quickly. The nails of the hinge are set up to mesh in such a way that the box remains closed well when closed, and mounting holes secure the system in place within the enclosure. The material choice ensures that additional ports can be made for routing of wires to customize the placement of this module within the power supply.

PCB

A custom PCB was created in order to connect the Arduino to the Bluefruit module. This PCB contains header pins that allow for it to fit onto the Arduino as a daughterboard, and header pins to connect directly to the Bluefruit board. Figure 9 below shows a CAD model of this PCB, and Figure 10 shows the PCB layers ready to be sent for manufacturing.

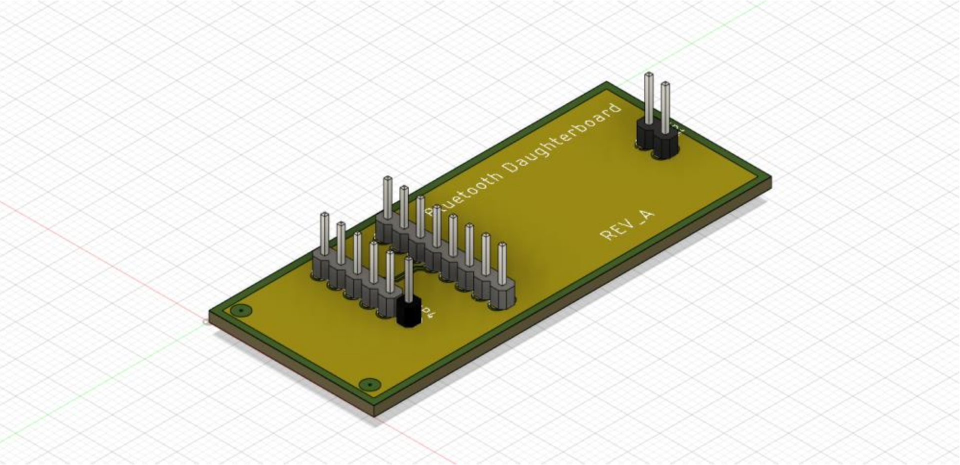
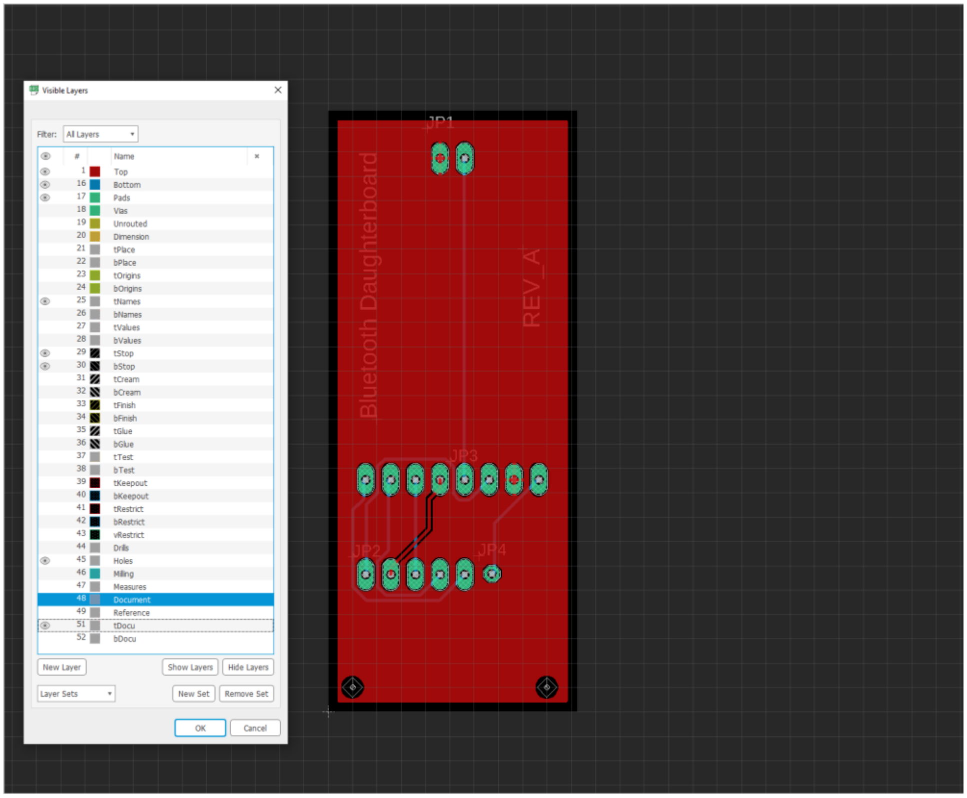


Figure 9: 3D PCB representation

This PCB is light and easy to use. It is set up in such a way that the orientation of the middle 8-pin connector can be soldered to the other side of the PCB, allowing the 5 and 2 pin connecters to enter directly into the Arduino, while the Bluefruit board sits on top. This makes for no wires or solder joints within the system, allowing it to be robust and very stable, as well as easy to modify by simply removing the PCB from the Arduino.


Visible Layers

Filter: All Layers

◉	#		Name
◉	1	🟥	Top
◉	16	🟦	Bottom
◉	17	🟩	Pads
	18	🟩	Vias
	19	🟨	Unrouted
	20	🟨	Dimension
	21	⬜	tPlace
	22	⬜	bPlace
	23	🟩	tOrigins
	24	🟩	bOrigins
◉	25	⬜	tNames
	26	⬜	bNames
	27	⬜	tValues
	28	⬜	bValues
◉	29	▨	tStop
◉	30	▨	bStop
	31	▨	tCream
	32	▨	bCream
	33	▨	tFinish
	34	▨	bFinish
	35	▨	tGlue
	36	▨	bGlue
	37	⬜	tTest
	38	⬜	bTest
	39	■	tKeepout
	40	■	bKeepout
	41	■	tRestrict
	42	■	bRestrict
	43	■	vRestrict
	44	⬜	Drills
◉	45	⬜	Holes
	46	🟩	Milling
	47	⬜	Measures
	48	⬜	Document
	49	⬜	Reference
◉	51	⬜	tDocu
	52	⬜	bDocu

Figure 10: PCB Layers in EAGLE PCB

By using only two layers, this PCB is easy to manufacture and passes the DRC check built into EAGLE PCB. Two holes located in the corners allow for additional mounting locations for use in different orientations to maximize the customizability of the system. The spacing of the pins is such that the PCB fits perfectly into the Arduino, requiring no wires.

Part Information

This unit requires no analog components, connectors, or other circuitry, The Arduino, PCB, and Bluefruit unit is complete with those three components only. Standard header Pins will however need to be soldered directly to the PCB, which can be done manually. The total estimated cost for this system including all the components, shipping, and production of the PCB is approximately $76.