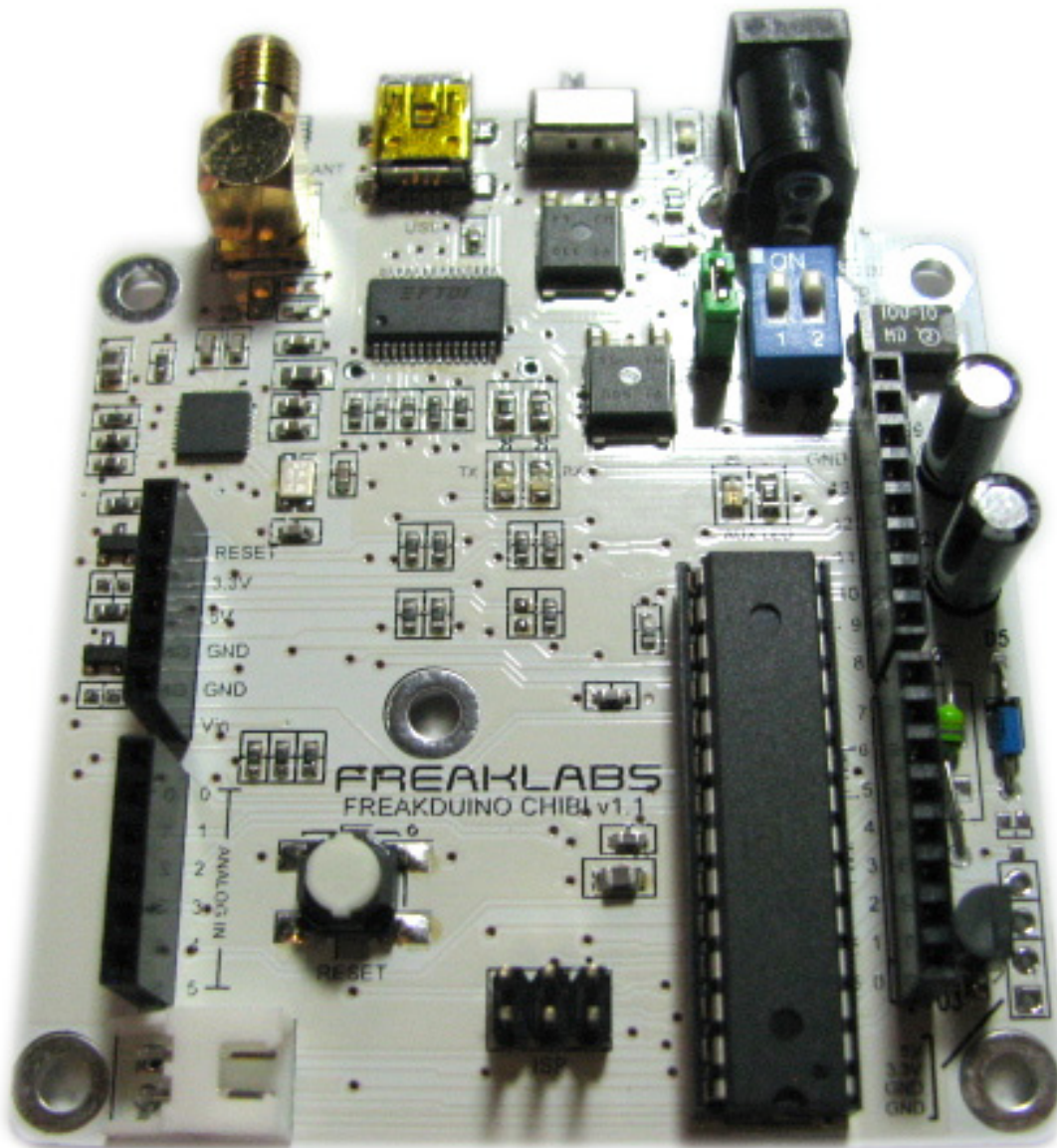# Freaklabs

FREAKDUINO-CHIBI v1.1
Wireless Arduino-Compatible Prototyping Platform
Datasheet

Document Revision History

| Date	Description
2010-11-01	v1.1A Document creation
2010-12-26	v1.1B Added power switch matrix in Connectors/Jumpers/Switches section Added troubleshooting section

Specifications

The FreakLabs FREAKDUINO-CHIBI board is designed for rapid prototyping, evaluation, and deployment of custom wireless devices at low cost. It combines the ease-of-use of the Arduino IDE and toolchain, compatibility with a rich assortment of peripherals in the Arduino shield form factor, and an integrated IEEE 802.15.4 wireless radio for inexpensive prototyping and testing of a wireless device.

The base board has all the functionality of an Arduino-based system with wireless communication and is an inexpensive way to start playing with wireless Arduino designs.
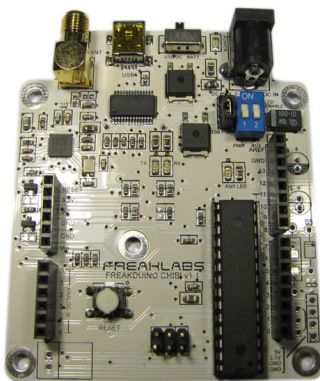
It also has optional features such as battery regulation circuitry, a low-cost bottom-mounted battery case, or a ruggedized enclosure with integrated battery case.
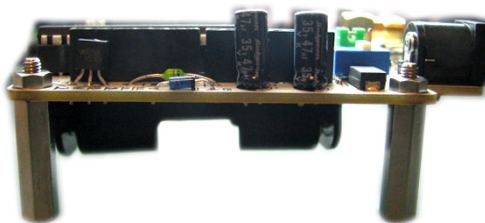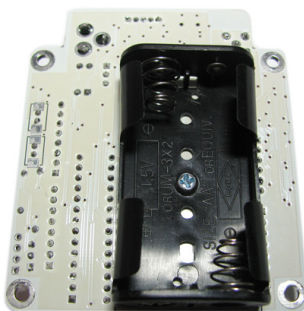
This board is designed to introduce people to wireless sensor networking inexpensively and without having to deal with complex toolchains, protocol stacks, and software. It's fully compatible with the Arduino IDE and toolchain which offers a single click compile and download and a rich ecosystem of open source software and tutorials available on the internet. The electrical specifications and connector form factor are also compatible with the original Arduino hardware so that the board can interface with the large assortment of Arduino peripheral shields. The availability

QUICK SPECS

MCU: ATMega328P

Memory: 32 kB Flash/2 kB RAM

Communications: 802.15.4 wireless, USB

Expansion: Arduino-compatible shield connector

Power: Ext 5VDC, USB, Battery (optional)

Options: Ruggedized enclosure, battery regulation circuit, standalone battery case

of third party peripheral shields and open source software allows this board to be used for many diverse applications.

Enhancements were also added to this board to increase functionality. The addition of an integrated wireless radio based on the 802.15.4 protocol (same radio protocol as the XBee) allows for wireless control of devices or wireless sensor data collection. Battery circuitry was added so that it could function as a true wireless node without any external power cables. The board is also fitted to a ruggedized enclosure so that the design can be transported safely or deployed in remote settings without worrying about damaging the circuit.
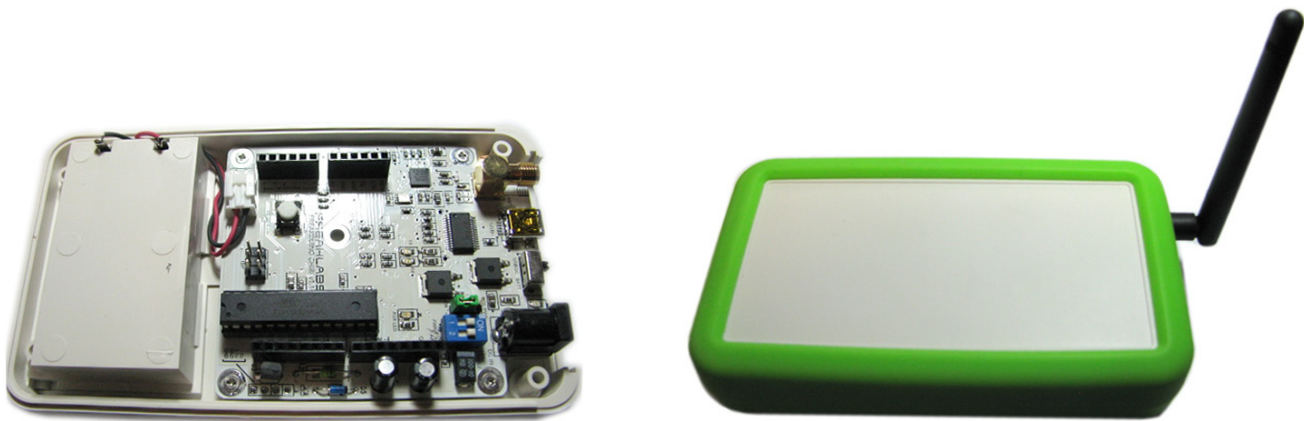
Radio

The main addition to this board is the integrated wireless radio. The radio is based on the 802.15.4 wireless protocol and is the same protocol used by XBee modules and Zigbee devices. The radio operates at 2.4 GHz and comes with an RP-SMA antenna connector which is a standard antenna connector commonly found on Wi-Fi routers. An external antenna was chosen over other options such as a chip antenna or printed antenna because of the improved range and variety of available antennas.

The radio driver software and protocol stack are fully open source and available as an Arduino library. The protocol stack is simplified to three main library functions to make wireless communications as simple as possible. Those functions are: init, send, and receive. This makes it easy to use the radio as a simple extension of a serial port or to set up a peer-to-peer star network where each device can talk with any other device within listening range.

There are also many benefits to using 802.15.4 for communications. At 2.4 GHz, the antennas used are the same as those used for Wi-Fi. Hence there are a large variety of antennas in different sizes, shapes, and power. Omnidirectional antennas such as the standard whip antennas on Wi-Fi routers give moderate range and allow transmission from all directions. Directional antennas can also be used for greatly improved range if the direction of communications is fixed.

Some other benefits of using 802.15.4 is robustness such as automatic acknowledge and retries

and functionality such as auto-discard of frames that don't match the particular network address or node address.

Also, since it's possible to talk directly to the radio, it also allows for more flexibility for advanced users such as a direct readout of the signal strength (RSSI), changing transmission thresholds (CCA), number of retries, or whatever else is available in the radio's registers.

Power

There are three options to provide power to the board. The most common is via the USB, however the board can also be powered via an external adapter connected to the DC jack. This is especially useful when more power is needed than can be provided by the USB. Finally, when no external power is available, the board can also be battery powered.

The USB connector provides up to 500 mA of current at 5V and can directly power the board for most applications. It's convenient when the device is connected to a PC since the USB to serial converter also allows for communications with the PC.

For designs with higher current requirements, such as driving motors or high power LEDs, an external DC adapter can be used. The adapter should be a minimum of 6V and a maximum of 10V.

> **Note:** The power input is determined by the position of the jumper below the DC input jack. The markings on the jumper indicate whether the USB or DC input is being used.

The board also has two separate connectors for batteries. One of the connectors allows a battery case to be mounted on the bottom side of the board. This is convenient since the battery case is directly mounted to the board and allows for battery operation in tight spaces.

The other connector is located near the bottom of the board and is a 2-wire connector that features mating and polarization. It's technically called a JST XH series connector and was chosen to allow easy attachment of external battery cables without worrying about reversing the positive and negative connections. It also makes connecting and disconnecting the battery case of the enclosure to the board much easier.

There is also a battery regulation circuit on the board. The reason this is needed is because the battery voltage varies based on the battery type and amount of charge left. The voltage regulation allows any type of battery to be connected to the board as long as the battery voltage is below 6V and will generate a stable 5V output. The battery regulation circuit has a maximum current output of 200 mA. This is fine for most devices, however devices like motors and high power LEDs (1W or greater) may exceed the maximum current output.

There is a power switch on the board that selects between external line power (via USB or DC jack depending on the jumper) or battery power. The switch can also be used as an OFF switch if either line power or battery power is not present.

There is also a 2-pin DIP switch on the board that enables or disables the power LED and the auxiliary LED. If the board is externally powered, then it's fine to enable both LEDs, but when the board is battery powered, the LEDs will cause the battery to drain much faster. The DIP switch allows the user to easily disable the LEDs during battery operation.

Connectors, Jumpers and Switches

The board contains a number of connectors, jumpers, and switches:

1.   **Power Input Jumper.** The power input jumper is used to select between using the external DC jack or the USB to power the board.

2.   **Main Power Switch.** The main power switch toggles between using line power or battery power. When no battery is connected, this would also serve as the OFF position. The opposite is true when no line power is attached.

3.   **LED Enable Switch.** The LED enable DIP switch is used to enable or disable the main power LED and auxiliary LED. LEDs may consume unecessary power during battery operation so the DIP switch allows the user to enable/disable them when needed.

4.   **External Antenna Connector.** The external antenna connector is an RP-SMA connector that attaches to a compatible antenna.

5.   **Battery Connector.** The battery connector is optionally installed and comes with the battery boost kit.
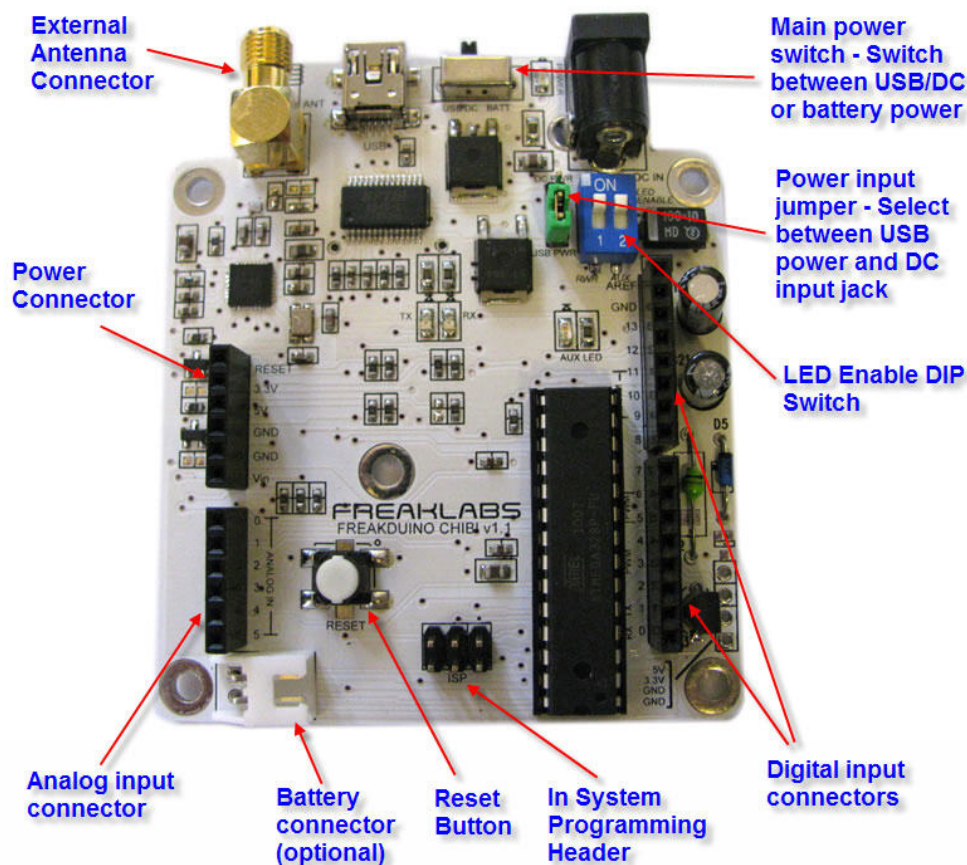
Due to having three ways to power the board, the power switch and power jumper settings need to be configured for your desired input. The switch and jumper is labeled to make things easy, but here's a quick reference chart on how to configure the power input. The positions in the table correspond to the labels on the board.

	Main Power Switch	Power Input Jumper
USB Power	USB/DC	USB PWR
External DC Power	USB/DC	DC PWR
Battery Power	BATT	Ignored

Uploading Code

Compiling and uploading user code into the Freakduino board is very simple:

1. **Download the Arduino IDE.** The first thing you need to do is get the Arduino IDE and install it. It can be <u>found at the Arduino site</u>.

2. **Open the IDE and add code.** We're going to take a shortcut and open up one of the default examples that comes with the Arduino software. Go to the "File/Examples/Basics" menu and select "Blink".
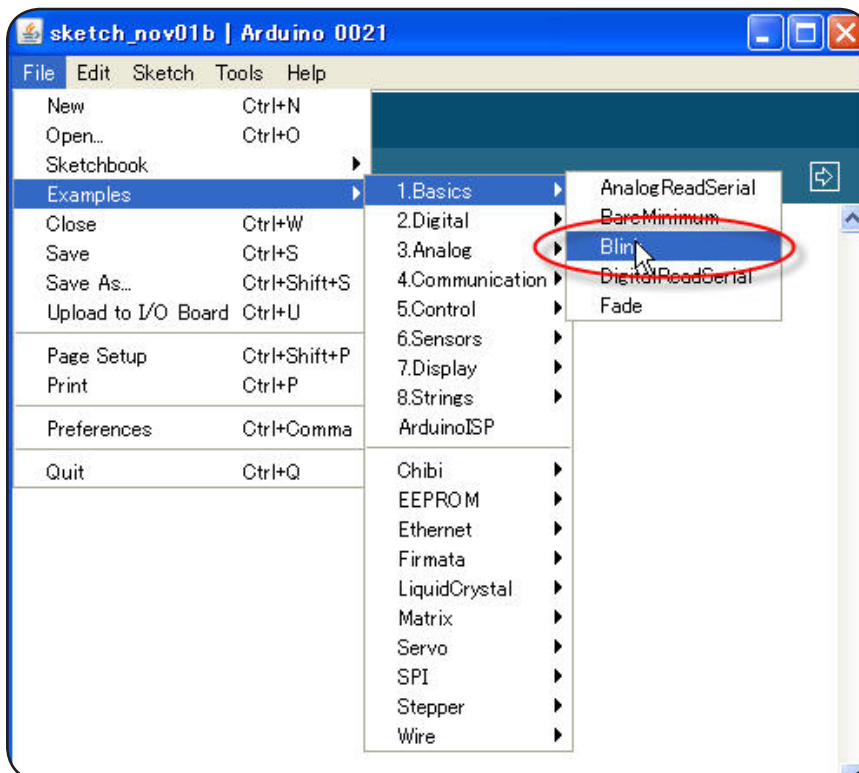
3. **Select the board.** Go to the "Tools/Board" menu and select "Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATMega328". Don't worry that it specifies 3.3 volts. We actually

run at 5V but the main thing is to get the clock frequency and chip correct. In the case of the Freakduino, it uses an ATMega328 running at 8MHz.
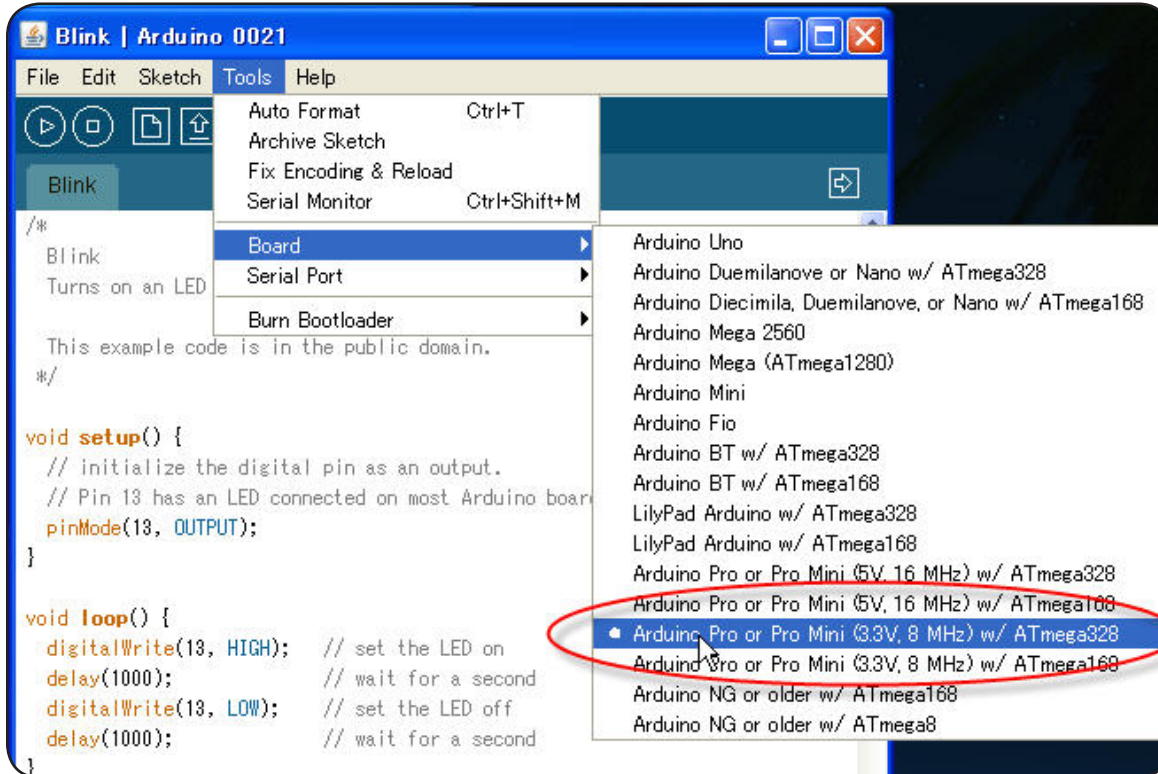
4.   **Select the serial port.** Plug the USB connector into the Freakduino board and select the serial port that it's connected to.
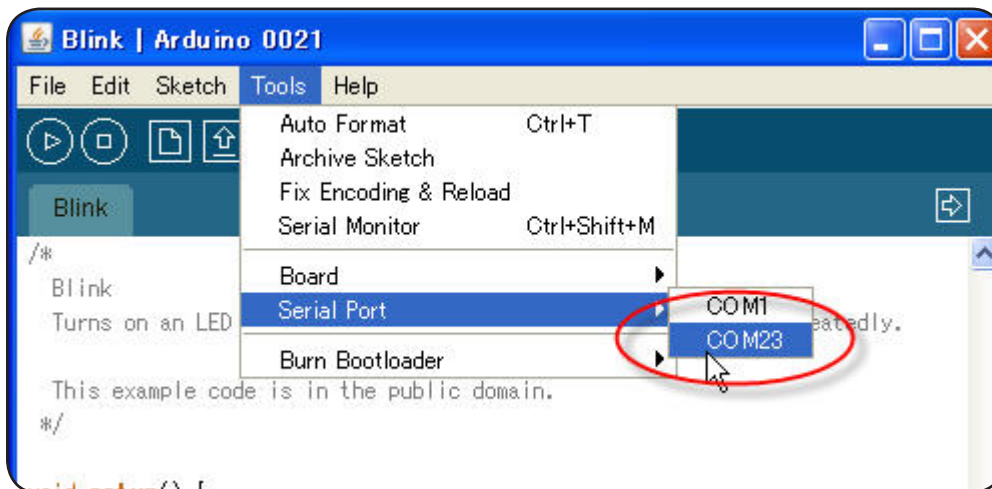
5.   **Click on the "Upload" icon.** The code should compile and start uploading. You'll also see the serial port's TX and RX LEDs blinking on the Freakduino board.
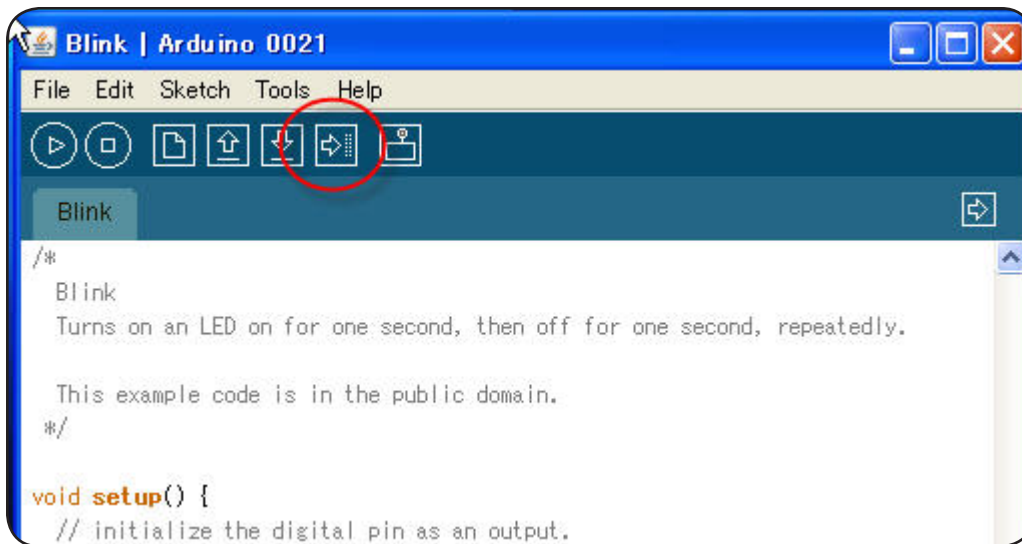
Congratulations! You just uploaded your code to the Freakduino board. You should now be the proud owner of a blinking LED :)

Compatibility

The FREAKDUINO-CHIBI board is essentially an Arduino-based system and a shield integrated together. This means that the peripheral will require using some dedicated pins for the wireless functionality.

Two pins on the analog input connector are not available for use except for standard digital I/O. These are pins Analog 2 and Analog 3. Analog 2 controls the sleep mode and Analog 3 controls the chip select for communications between the microcontroller and radio IC. If the wireless functionality is not used, these pins are available as standard digital I/O but not as analog inputs. If possible, its best to avoid using these pins.

If the wireless functionality is being used, the SPI bus is also required to communicate with the radio. That means that digital pins 10-12 (PB3 to PB5 or MOSI, MISO, and SCLK) will be dedicated SPI pins. Please be especially careful when using the AUX LED on the board. It is located on the SCLK pin required by the SPI. If the radio will be used, then the AUX LED should not be used in the sketch.

If there is any question about compatibility with a particular shield, please post to the FreakLabs forums or email *support@freaklabsstore.com*.

License

The FREAKDUINO-CHIBI hardware design is licensed under the Creative Commons Attribution-ShareAlike license v3.0. Attribution is an option and not a requirement. If you do attribute any derivatives of this design to FreakLabs, I'll think you're really cool, though :)

Disclaimer

The FREAKDUINO-CHIBI board is NOT FCC approved. It is designed to comply with FCC Part 15 rules. However this board is not in a finished product form and is only intended for experimental and research/development purposes. If you wish to use this board in an actual product, you will need to attain certification with the appropriate local regulatory body for the complete system. Additionally, please use the wireless equipment in a responsible manner with regard for others and your surroundings.

Troubleshooting

There are some common problems that people run into when using the Freakduino board. The following is a list of issues people have had and the fixes for them:

1.   **Serial port does not work at speed specified in Arduino sketch.** This is most likely due to having the wrong "Board" setting in the Arduino IDE. A telltale sign is when you can get the serial output to work properly at twice the baudrate. Normal Arduinos use a 16 MHz clock but the Freakduino uses the internal oscillator on the AVR which runs at 8 MHz. Because of that, the "Tools/Board" setting in the Arduino IDE should be:

"Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ATMega328"

2.   **Radio is not working or displays "RADIO NOT INITIALIZED PROPERLY" on serial terminal.** The first thing to try when you see that message is to load the example sketch "chibi_ex4_cmdline" with nothing connected to the Freakduino. If the error message disappears, then its likely that some pins needed for radio communications (like the SPI pins) are also being used in the sketch. One of the main culprits of this issue is when a sketch uses the AUX LED on the board. The LED is standard on Arduinos but is unfortunately connected to the SCLK pin required by SPI. If the radio is going to be used, the AUX LED should not be used. Please see the "Compatibility" section of this document.

If the error message still shows with the example sketch, then there is likely a hardware problem. Please contact me through the FreakLabs website, email me at chris(at)freaklabs(dot)org, or post on the forums.

Schematics

Schematics can be found on the following page:

LOGIC CONVERSION FOR 3.3V/5V INTERFACE

Reset has a pullup to VCC5 so the
pulldown on the voltage divider
should be weaker

SLP_TR has a pullup on the radio side.
When going to sleep mode, just set this
to be an input. That way, there's less
static power drain.

ISP CONNECTOR

LED CONTROL - THIS MAKES
IT POSSIBLE TO TURN OFF
THE LEDS. ESPECIALLY
IMPORTANT WHEN RUNNING ON
BATTERIES.

ATMEGA328-DIP

PUSHBUTTON-DUAL/SMD2
S2

U4

USB-Mini-AB

J1

SHLD 9
SHLD 8
SHLD 7
SHLD 6
+5V
D+
D-
ID
GND

VBUS

DM
DP

C1
0.1 uF

FT232-SSOP28

U1

OSCI 27
OSCO 28
3V3OUT 17
USBDM 16
USBDP 15
RESET_N 19

GND 7
GND 18
GND 21
AGND 25

VCC 20
VCCIO 4

TEST 26
NC 8
NC 24
CBUS4 12
CBUS3 14
CBUS2 13
CBUS1 22
CBUS0 23
DCD_N 10
DSR_N 9
DTR_N 2
RI_N 6
RTS_N 3
CTS_N 11
TXD 1
RXD 5

R6
10k

FTDI RST N

VBUS

VBUS

VBUS

C15
0.1 uF

TXLED N
RXLED N

DTR_N

MCU RX
MCU TX

MCU_RX
MCU_TX

R7
330

D2
GREEN

VBUS

R8
330

D3
GREEN

VBUS

Size A
Date: Wednesday, September 22, 2010

Project Title: FREAKDUINO CHIBI
Page Title: USB

Sheet   5   of   5

Rev
v1.1

FREAKLABS

FREAKDUINO-CHIBI v1.1B   Page 13

POWERJACK/THRU-3

J2

VCC_IN

C8
100uF/10V

VCC_IN

C9
DNI

NJM2391-5.0
NJM2845-5.0
LP-2950, DK#LP2950CDT-5.0-ND

VR2
VIN VOUT
GND

BATT1
JST-XH2P

BATT2
BH321-1P

VBAT

C26
DNI

C21
47 uF

47 uH
L2

47 uH L3
DNI

LX

HT7750A

U3
LX VOUT
GND

D6
DNI

1S4
D5

C19
47 uF

BOOST OUT

C25
DNI

VBUS

JP7
JP3/THRU_VERT

VCC5 IN

SWITCH_SLIDE/THRU-3

S1

VCC5

C3
22uF/6.3V

TP1

NJM2391-3.3
NJM2845-3.3
LP-2950, DK#LP2950CDT-3.3-ND
KF33BDT-TR, DK#497-1148-1-ND

VR1
VIN VOUT
GND

TP2 VCC3

C10
22uF/6.3V

Size
A

Date:
Wednesday, September 22, 2010

Project Title: FREAKDUINO CHIBI
Page Title: POWER

Sheet   3   of   5
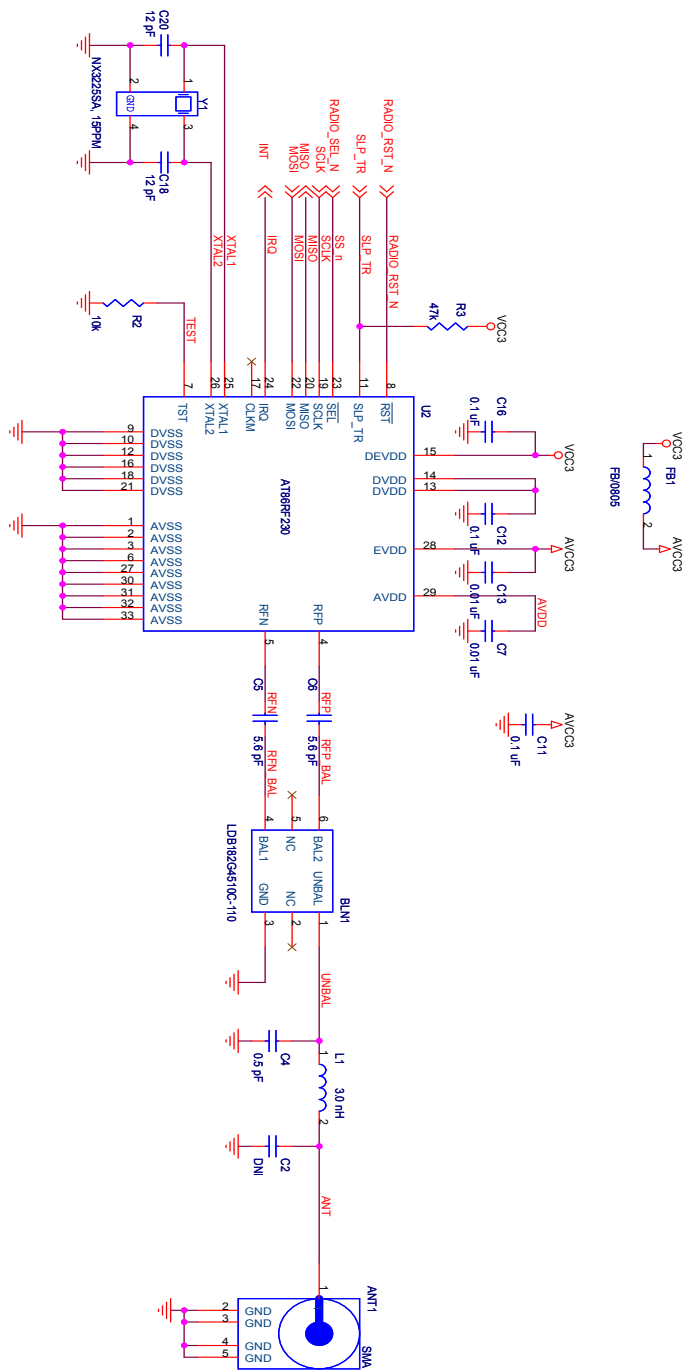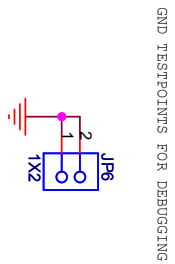
Rev
v1.1

FREAKLABS

FREAKDUINO-CHIBI v1.1B   Page 14

ANALOG AND DIGITAL GROUND WILL BE SEPARATED
WITH THE ONLY CONNECTION POINT BEING
UNDERNEATH THE RADIO.

FREAKLABS

|	
Project Title:	FREAKDUINO CHIBI
Page Title:	RADIO

Size			
B			
Date:	Wednesday, September 22, 2010	Sheet	4 of 5

Rev v1.1

GND TESTPOINTS FOR DEBUGGING

JP6
1X2
1
2

H4
MOUNT_HOLE/MOUNT_HOLE
MOUNT_HOLE/MOUNT_HOLE
1

H2
MOUNT_HOLE/MOUNT_HOLE
1

H5
MOUNT_HOLE/MOUNT_HOLE
1

H1
MOUNT_HOLE/MOUNT_HOLE
1

FREAKLABS

Size	Project Title:	FREAKDUINO CHIBI				Rev
A	Page Title:	RESET/MISC				v1.1

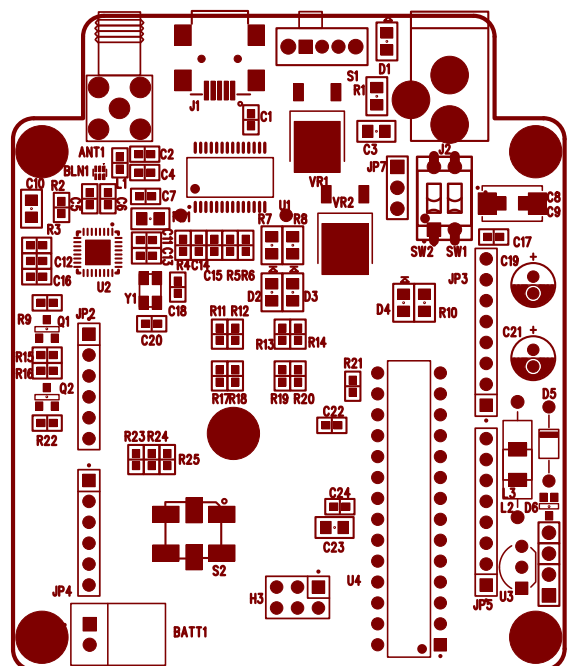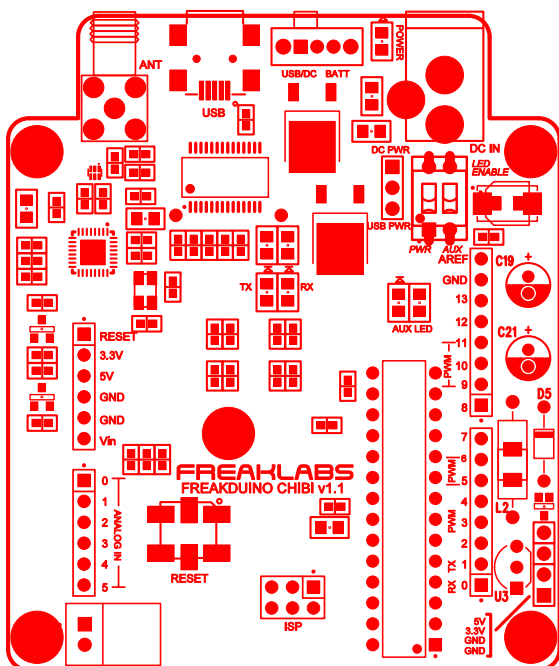Date: Sunday, September 05, 2010

Sheet   2   of   5

2

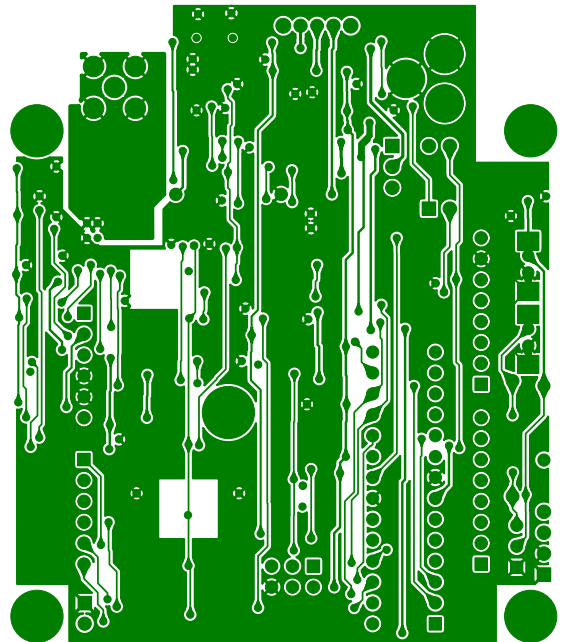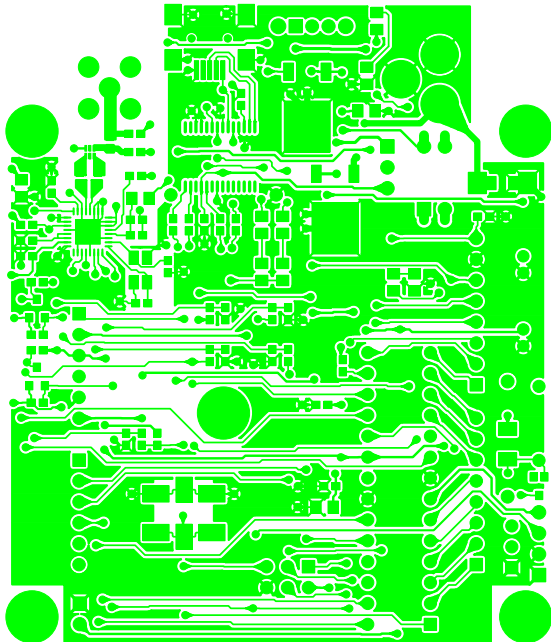# PCB Layout

PCB layout file order:

1. *Mechanical Dimensions*
2. *Top Layer*
3. *Bottom Layer*
4. *Assembly Drawing - Silkscreen View*
5. *Assembly drawing - Reference Designator View*
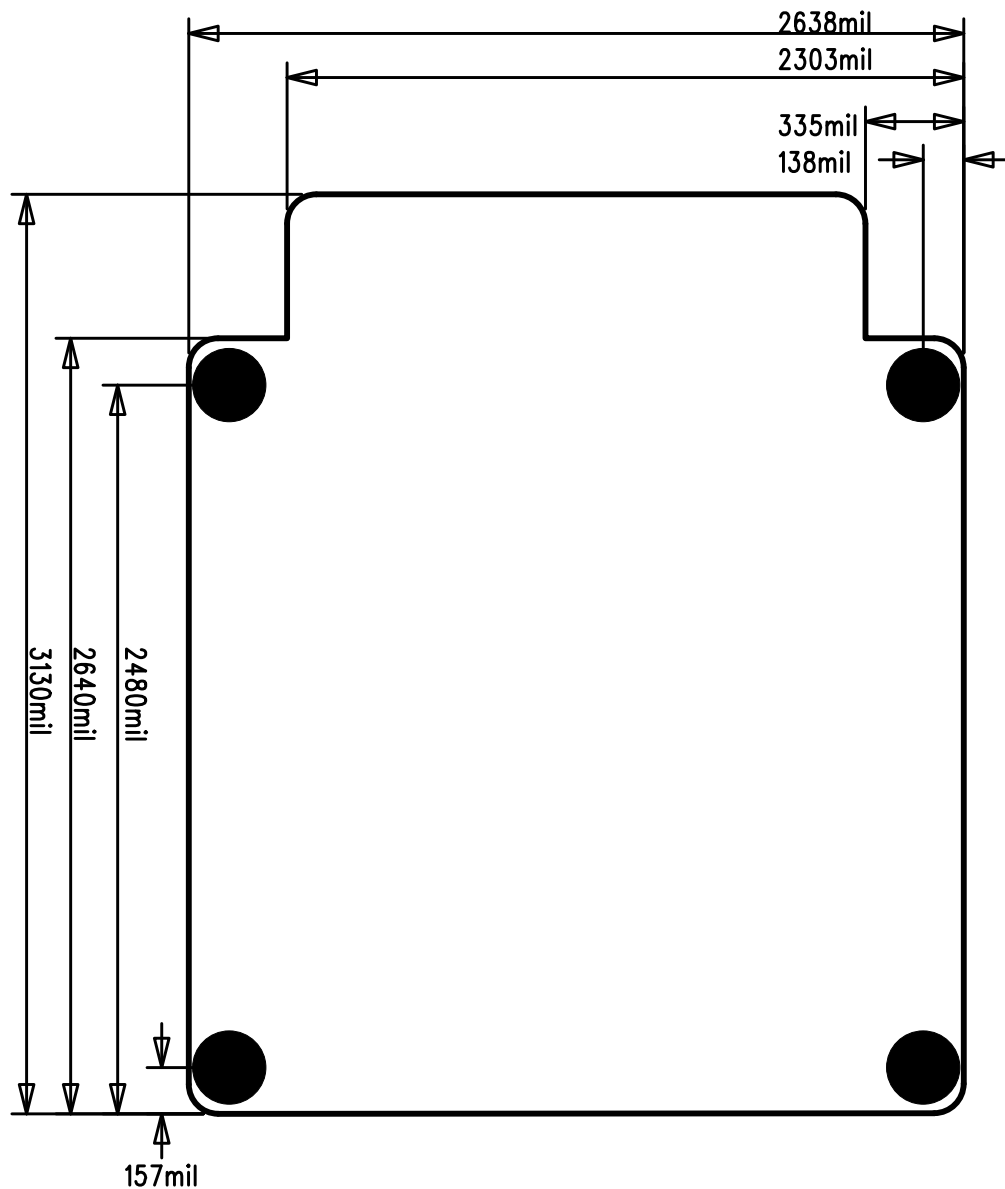
Bill of Materials

Quantity	Reference	Manufacturer	Part Number	Description
1	ANT1	ChangHong	SMA-02-113-TGG	RP-SMA Connector
1	BATT1	JST	JST-XH2P	2-pin mated, polarized connector, male
1	BATT2	COMF	BH321-1P	2-AA battery case
1	BLN1	Murata	LDB 182G4510C-110	100 ohm diff/50 ohm single ended, 2.4 GHz, balun
9	C1,C11,C12,C14, C15,C16, C17,C22, C24	Various		0.1uF/50V, 0603
3	C3,C10,C23	Various		22uF/6.3V, 0805, MLCC
1	C4	Taiyo Yuden	UVK-105CH0R5BW-F	0.5 pF/50V, 0402, high freq
2	C5,C6	Taiyo Yuden	UMK105C-G5R6DW-F	5.6 pF/50V, 0402, high freq
2	C7,C13	Various		0.01 uF/50V, 0603
1	C8	Various		100 uF/10V, Case-D, Tantalum
2	C18,C20	Various		12 pF/50V, 0603
2	C19,C21	Various		100 uF/10V, Case-D, Tantalum
1	D1	Various		Blue LED, 0805
2	D2,D3	Various		Green LED, 0805
1	D4	Various		Red LED, 0805
1	D5	Various		Schottky diode, 200 mA
1	FB1	Various		Ferrite Bead, 0805
1	H3	Various		3x2 Straight male header, 0.100"
2	JP3,JP5	Various		1x8 Straight female header, 0.100"
1	JP4, JP2	Various		1x6 Straight female header, 0.100"
1	JP7	Various		1x3 Straight male header, 0.100"

Quantity	Reference	Manufacturer	Part Number	Description
1	J1	4UCON	09558	USB Mini-AB connector
1	J2	4UCON	05537	DC Power Jack, 2.0 mm center conductor
1	L1	Abracon	ATFC-0402-3N0-BT	3.0 nH, 0402, high freq
2	L2,L3	Various		47 uH inductor, 200 mA
2	Q1,Q2	Diodes, Inc	2N7002	N-channel enhancement FET
4	R1,R7,R8,R10	Various		330 ohms, 0805
14	R2, R6, R9, R11, R13, R16, R17, R20, R21, R24, R25	Various		10 kohms, 0603
3	R12, R14, R18	Various		15 kohms, 0805
2	R3,R23	Various		47 kohms, 0603
2	R4,R5	Various		1 kohms, 0603
1	SW1	Various		2-input DIP switch
1	S1	See-Plus	SK-12D01-VG2	Right angle slide switch
1	S2	Various		SMD SPST Tactile Switch/Pushbutton
1	U1	FTDI	FT232RL	USB to Serial Converter
1	U2	Atmel	AT86RF230	2.4 GHz, 802.15.4 radio
1	U3	Holtek	HT7750A	5V Boost Converter
1	U4	Atmel	ATMEGA328P	AVR Microcontroller
1	VR1	NJRC	NJM2391-3.3	3.3V Regulator
1	VR2	NJRC	NJM2391- 5.0	5.0V Regulator
1	Y1	NDK	NX3225SA	16 MHz crystal, 15 ppm