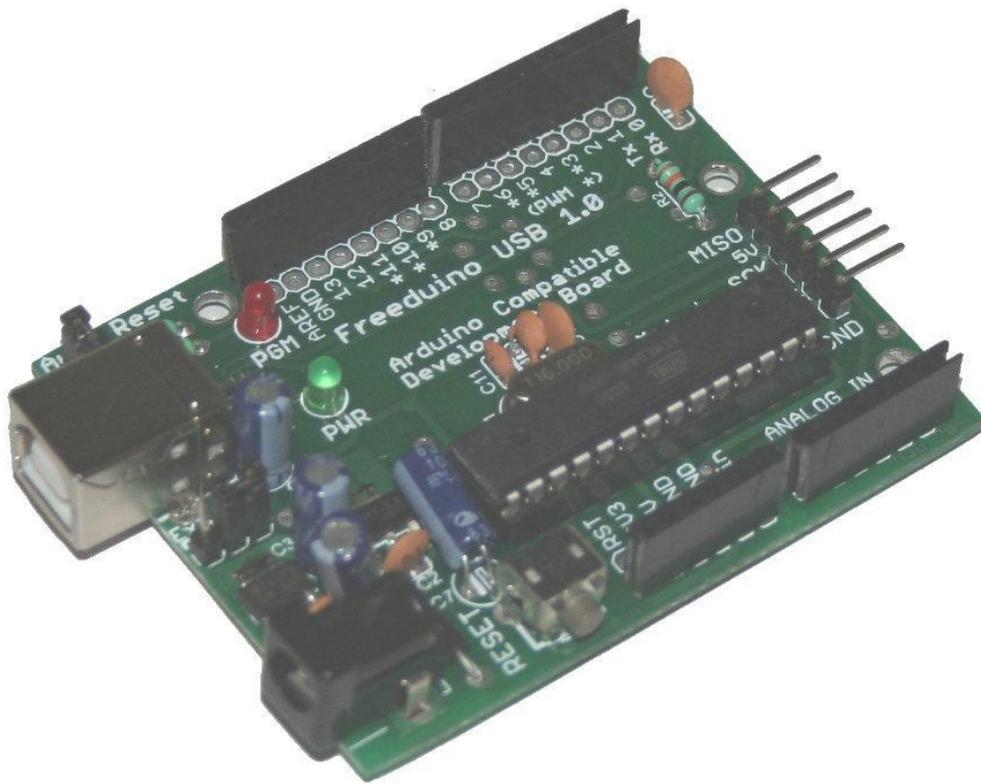


# Freeduino USB 1.0

Arduino Compatible Development Board

## Starter Guide



### 1. Overview

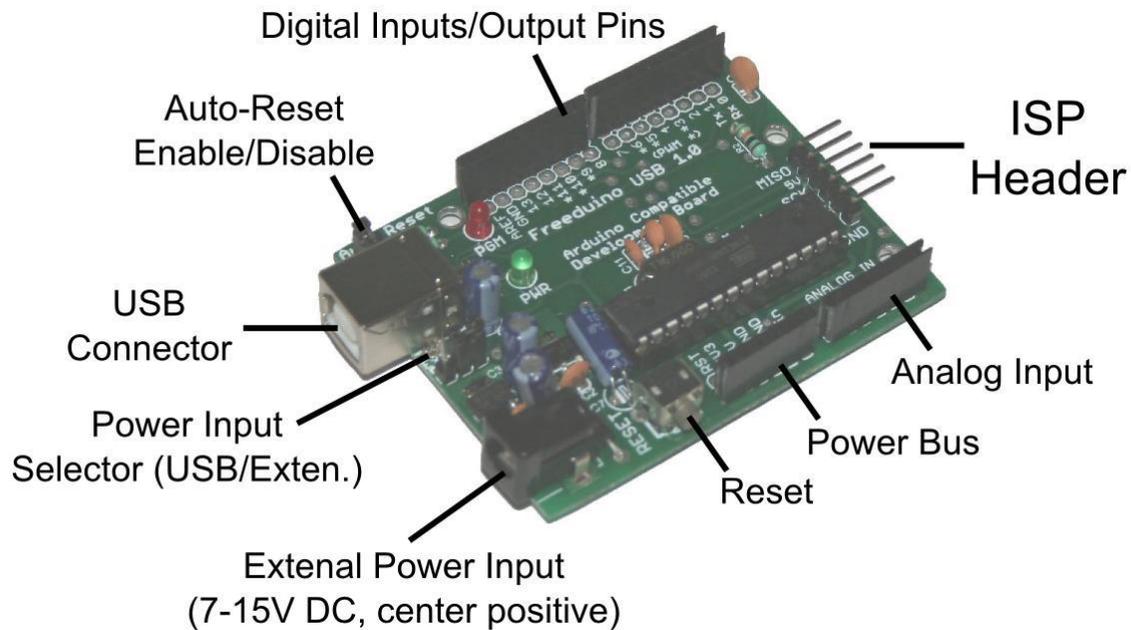
Arduino is an open source embedded development platform consisting of a simple development board based on Atmel's AVR microcontroller and an easy to use development environment for writing, compiling and uploading codes to the board. Freeduino USB is a Arduino compatible board, which is compatible with all Arduino development tools, softwares, codes etc. Freeduino USB comes with only the bare essentials present on an Arduino board that is required for getting started. It contains everything that is required for programming a microcontroller and omits all extra features to keep it simple and cost effective.

Arduino is an opensource development platform, which means all the designs files are available for free. Users may use these freely available design files, which include the board schematics, PCB Files, codes, etc and use them in their projects making suitable modifications as their application may require. A large Arduino user base ensures that you will get an excellent technical assistance and help, which will greatly reduce your development time and increase your learning experience. You get all this apart from our excellent after-sales support and service.

Another great feature of the Arduino is its modularity approach. There are a wide range of addon boards, called Shields, which plug onto an Arduino and help you achieve a specific purpose and greatly simplifying the whole development process. For example, an Arduino motor shield can plug onto an Arduino and help you control motors directly through the main board. There are a wide range of such shields including GSM/GPS shields, Robotics shields, Ethernet Shields, Bluetooth shields and even prototyping shields. Freeduino USB is completely compatible with all Arduino compatible shields.

Freeduino USB can be programmed directly through a USB connection to a PC through the Arduino IDE, which is a simple and easy to use program development environment.

## **2. Features**



The board is built on a high quality FR-4(1.6 mm) board with a green solder mask and a clear and legible white legend

- Compatible with all Arduino development tools, hardware, etc. and can be used as a direct cost effective replacement for the official Arduino
- Completely opensource, hence easy to understand, work with and modify as per individual requirements
- Huge online community for help and assistance
- Can be powered through USB or through an external power input(7 - 15 V DC)
- Power Status LED(Green) along with a general purpose LED connected to Pin 13
- Onboard quartz crystal 16 Mhz oscillator circuit
- A right angled reset switch which can be used even when a shield is plugged onto the board
- All IO pins are brought out to female header pins, for easy plugging of wires and prototyping

- Every IO pin has an empty parallel next to each which can be used for easy expansion by soldering in required connectors or wires from external circuits
- Excellent after-sales support and service

### **3. Using the board**

This section explains the steps to be followed to use the board –  
**Chosing a microcontroller – Atmega8 or Atmega328**

Initially Arduino and compatible boards came with the Atmega8 microcontroller from Atmel. Most Arduino and compatible boards come with the Atmega328 these days. Both the microcontrollers have the same pinout and can be used interchangeably. The Atmega328 offers more memory space for your programs, and more timer options apart from a few other minor advantages. Please go through the datasheet of the two microcontroller for more information. The Atmega8 is a lot cheaper than the Atmega328. Freeduino is available with both the Atmega8 and the Atmega328. Go for the Atmega8 only if cost is a concern.

#### **Power Supply**

Freeduino can be powered in two ways – through USB or through an external input power supply. When powered off USB, the maximum current that can be drawn safely without damaging your USB port is 50 mA, hence this option is only recommended for applications which draw low currents (less than 50mA).

The board can also be powered of an external power supply applied to the power input connector, which is a standard 2.1 mm DC Jack. The board can operate within a wide range of input voltage ranging from 7V to 15V DC. Care should be taken to ensure that the center pin of the connector is connected to the positive potential. The board has an onboard reverse polarity protection circuit, which will prevent damage to the board when power is applied with the wrong polarity. When powered correctly (through USB or through an external power supply) the green Power LED on the board will glow indicating a correct power supply.

There is an onboard 5V, which is required for the operation of the microcontroller. The regulator will output a constant 5V within the supported range of input voltage. There is also an additional 3V3 regulator. These regulators can provide a maximum of 500mA of current. This regulated 5V and 3V3 output is provided at all connectors marked 5V and

3V3 on the board. All connectors marked GND on the board are connected to the negative potential. The connectors marked VI are connected to the positive end of the power supply.

The required power input type can be selected by using a jumper (marked as Power Input Selector in the above figure) at the power selector pin on the board. Putting the jumper in one position will power the board of USB and putting it in the other position will power the board of the external power supply at the power input connector.

Note:

- 1) If you power the board of a AC adapter, ensure a proper grounding connection else you may damage the board.
- 2) Drawing more than 50mA from the USB port through Freeduino USB, may damage your USB port. Most computers come with protection from USB power surges, but it is always safe to use them within the rated power specification.

### **Crystal Oscillator**

The board by default comes with an onboard crystal oscillator circuit consisting of a 16 MHz quartz crystal and the required capacitors.

### **General Purpose LED**

The onboard Orange LED is connected to Pin 13 of the microcontroller. It is connected to Gnd through a suitable resistor as shown the circuit. To control the status of the LED, you will first have to set Pin 13 as output. To switch on the LED you will have to set Pin 13 as high and to switch it off set Pin 13 as low.

### **Reset Button**

The board provides a reset switch which is connected such that it resets the microcontroller every time it is pressed. Upon reset, all registers are set to their default value and the program is executed from the beginning.

The reset switch will also be required while burning programs, called Sketches, to the boards through the Arduino IDE. Freeduino USB provides a right angled reset switch which can be used even when a shield is plugged on top of the board.

### **Auto – Reset**

The reset button is required to be pressed each time you want to upload a Sketch to Freeduino USB. The computer is capable of issuing this reset signal to the board when programming Freeduino USB eliminating the need for the user to reset the board every time a code is to be uploaded.

For this, the onboard Auto-Reset feature will have to be activated by placing a jumper at the autoreset pins. Removing the jumper will disable the autoreset feature.

Note: On some computers, you may face problems with the Auto-Reset and you may be unable to program your Freeduino USB. In which case, just disable the Auto-Reset feature and manually reset Freeduino USB, by pressing the reset switch.

### **ISP Header**

Arduino and Arduino compatible boards require piece of code called the bootloader, to be burnt into the microcontroller for it to communicate with a computer during programming. This bootloader takes up a little space on the microcontroller's total flash storage space. If you would like to use the complete flash storage space available on the microcontroller, you will need an external hardware programmer which can act as a bridge between your computer and Arduino/Freeduino USB and facilitate programming of the microcontroller. The ISP Header brings out the 6 pins required for programming the microcontroller through an external programmer.

## 4. Getting started

This section covers the initial steps required to be followed to step up the Freeduino USB to be used on a computer. The primary requirement to get started is the need for a suitable power supply and a USB cable to connect Freeduino USB to the computer for programming.

You may use one of our AC mains adapters which have output voltage in the range 7 – 15V as the power source for the Freeduino USB. The adapters come with a 2.1 mm DC Barrel Jack as their output pin which mates perfectly with the onboard power connector on Freeduino. For connecting Freeduino USB to a computer, you will need a USB A to B Extension Cord. Before you start programming Freeduino USB, you will need the Arduino IDE. It is basically a simple to use and easy to understand development environment, which you can use to write, compile and upload codes to the Freeduino USB. It is completely free and opensource. You may find it in the Freeduino Product CD. You may also download the latest version directly from Arduino's website – [www.arduino.cc](http://www.arduino.cc) . Copy the Arduino folder onto a suitable folder on your computer.

You are also required to install drivers for Freeduino USB on the computer you wish to use it on. Freeduino USB uses the CP2102 USB to Serial Converter chip from Silicon Laboratories for its USB to Serial Conversion. The drivers are also provided on the Freeduino Product CD. You may also get the latest drivers directly from Silicon Laboratories website. Installing the drivers is a one time process and you have to do it just once – the first time you connect the board to a computer.

The following description covers the steps required to be followed to setup Freeduino USB on a PC with Windows -

Connect Freeduino USB to a computer through a USB cable. Set the jumper such that the board sources its power from the USB port. You should now see the Power LED lit up. After a few seconds, you will see the orange led blink quickly. This is because Freeduino USB is burnt with a sample LED blink program by default.

The computer will detect and signal a new hardware connected to it through message on the screen and will prompt you to install the device drivers for it. Please follow our CP2102 Driver Installation Tutorial to install the drivers. You may find the tutorial under the Resources section of our website. A copy of the installation guide is also provided on the product CD.

Once the drivers are installed you will have to find out the COM port to which Freeduino USB is connected to. To do this right click on the My Computer icon, click on Properties and select the Hardware tab in the dialog box that appears. Under the Hardware tab, click on Device Manager. In the Device Manager Dialog box, open the list – Ports (COM & LPT). Note down the COM port to which appears beside the device name Silicon Labs CP210x USB to UART Bridge. You will need this later when you use the Arduino IDE to tell the software to which port Freeduino USB is connected to. You are now ready to upload codes to Freeduino USB.

## 5. Uploading Codes using the Arduino IDE

Run the Arduino IDE. You will have to first setup the Arduino IDE to match the board you are using. First, you will have to select the board you are using. To do this, select “**Freeduino USB w/ Atmega8**” under the Tools -> Board list. If you are using Freeduino with ATmega168, then select “**Arduino Diecimila, Duemilanove, or Nano w/ ATmega168**”. If you are using Freeduino with ATmega328, then select “**Arduino Duemilanove or Nano w/ ATmega328**” If these names do not appear in the list, then open the file boards.txt, which is found at arduino-002x/hardware/arduino folder, where Arduino-002x is your Arduino installation folder. Just add the code at the end of the file, without modifying any part of the code or the file –

```
#####  
iLabz.name=Freeduino USB w/ ATmega8  
iLabz.upload.protocol=stk500  
iLabz.upload.maximum_size=7168  
iLabz.upload.speed=19200  
iLabz.bootloader.low_fuses=0xDF  
iLabz.bootloader.high_fuses=0xCA  
iLabz.bootloader.path=atmega8  
iLabz.bootloader.file=ATmegaBOOT.hex  
iLabz.bootloader.unlock_bits=0x3F  
iLabz.bootloader.lock_bits=0x0F  
iLabz.build.mcu=atmega8  
iLabz.build.f_cpu=16000000L  
iLabz.build.core=arduino  
#####
```

Save the file and restart the Arduino IDE. You may now select your board in the Boards list.

Once you have selected your board you will have to select the COM port to which your board is connected to. To do this navigate to Tools -> Serial Port and select the COM port to which your board is connected to. It is the same port which shows up the Device Manager, beside the device - Silicon Labs CP210x USB to UART Bridge.

Once you select the port to which the board is connected to you may now start programming the board. There are a lot of sample sketches (programs) available with the IDE. You may under the Files -> Examples list.

### **Uploading programs to the board**

Once the IDE is configured, you may start uploading sketches to the board. You may try the Blink sketch. It flashes the Yellow LED on Freeduino USB on and off continuously. You can open the blink sketch at Files -> Examples -> 1.Basics -> Blink

Once you have the blink sketch opened, you may want to just go through the sketch. To upload the sketch to the board, click on the Upload button on the top of the IDE. The IDE will prepare the sketch to be uploaded to the board and once the code is ready to be uploaded to the board, it will display the size of the sketch and the total size available on the microcontroller's flash memory. You must immediately reset the board, by pressing the reset button on Freeduino USB. If you are using the auto-reset, the computer will issue the reset signal and you need not reset the board. Once you reset the board, the sketch will get uploaded to the board and the IDE will show a success message. You can now see the Orange LED blink after around 6 – 8 seconds.

## **6. Further reading and getting more information**

Freeduino USB, like the Arduino, is a highly capable and configurable platform. You may find more about Arduino at its official site – [www.arduino.cc](http://www.arduino.cc) The site contains tutorials, sample codes, examples and a ton of other resources for learning and even showcasing your creations. You may also contact us for any doubts you may have about Freeduino USB.

## **6. Troubleshooting and Getting Help**

We are committed to ensure that our customers' projects, designs and research go as efficiently and as smoothly as possible. And for this we promise and provide an excellent after-sales support service. Due to the

nature of the products we sell, we are sure to receive a lot of request for support. In anticipation of this, we have created a separate section, "Resources", where we have many tutorials covering how to use our products. Your first step to seek support from us would be to surf through the "Resources" section for answers to your query. We also have our very own forum, which you can use to post your queries and get replies directly from us and our other customers who have used our products and probably faced the same doubts when they first got started. You may access our forum at [If our Resources and Forum sections don't address your query, mail us directly at \[in\]\(#\) for further support.](#) When you contact us for support, please keep the following things in mind -

- you will have to provide us with the invoice information(order no., date of purchase, etc.)
- provide us with all the required details(operating system being used, compiler being used, etc)
- do not ask us questions or doubts about products we do not sell
- do not contact us 2 days before a project deadline for any sort of immediate or urgent support
- do not mail us a rephrased version of your project
- contact us only if you know what you are doing, do not expect any sort of help from our side otherwise