OPEN HARDWARE WORKSHOP

COLLEGE OF ENGINEERING TRIVANDRUM

CONTENTS

OPEN HARDWARE

The term "open source" refers to something that can be modified because its design is publicly accessible. Open hardware, or open source hardware, refers to the design specifications of a physical object which are licensed in such a way that said object can be studied, modified, created, and distributed by anyone. "Open hardware" is a set of design principles and legal practices, not a specific type of object. The term can therefore refer to any number of objects like automobiles, computers, robots or even houses. Like open source software, the "source code" for open hardware schematics, blueprints, logic designs, Computer Aided Design (CAD) drawings or files, etc. is available for modification or enhancement by anyone under permissive licenses. Users with access to the tools that can read and manipulate these source files can update and improve the code that underlies the physical device. They can add features or fix bugs in the software. They can even modify the physical design of the object itself and, if they wish, proceed to share such modifications. Open hardware's source code should be readily accessible, and its components are preferably easy for anyone to obtain. Essentially, open hardware eliminates common roadblocks to the design and manufacture of physical goods; it provides as many people as possible the ability to construct, remix, and share their knowledge of hardware design and function.

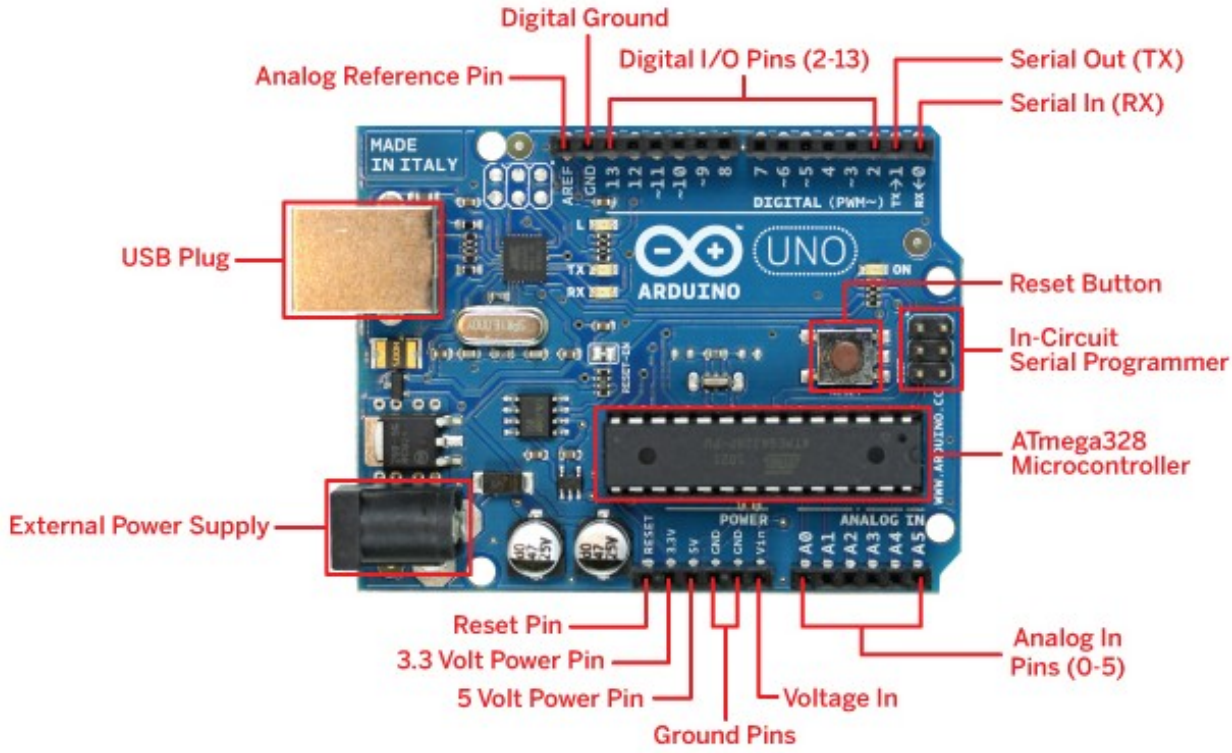
ARDUINO

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP) The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free.

The advantages of arduino are :

- inexpensive
- cross platform
- simple clear programming environment
- open source and extensible software
- open source and extensible hardware

Arduino Components



INSTALLATION

1.Installation for Linux

Go to the folder where the file is located. The file may be a zip file. To extract file, right click in the folder---> open in terminal; type the following commands

tar -xvf “file name”

sudo sh insatll.sh (enter your system's password if asked)

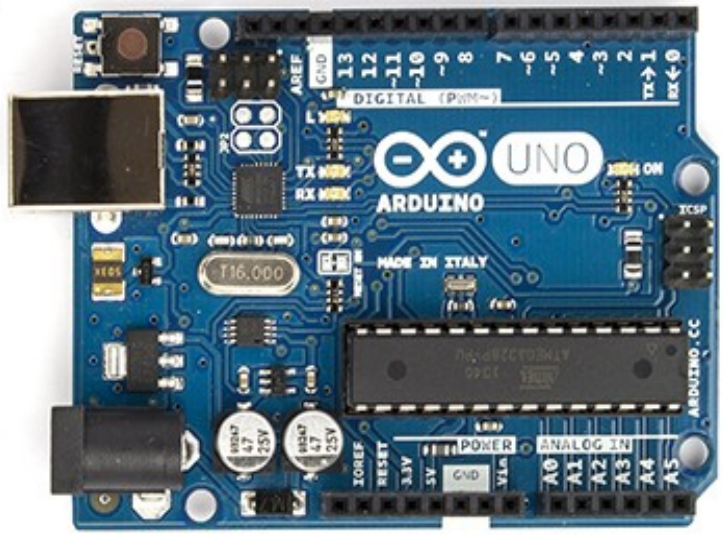
sudo arduino

2.Install the drivers (for Windows)

- Plug in your board and wait for Windows to begin it's driver installation process. After a few moments, the process will fail, despite its best efforts
- Click on the Start Menu, and open up the Control Panel.
- While in the Control Panel, navigate to System and Security. Next, click on System. Once the System window is up, open the Device Manager.
- Look under Ports (COM & LPT). You should see an open port named "Arduino UNO (COMxx)". If there is no COM & LPT section, look under "Other Devices" for "Unknown Device".
- Right click on the "Arduino UNO (COMxx)" port and choose the "Update Driver Software" option.
- Next, choose the "Browse my computer for Driver software" option.
- Finally, navigate to and select the driver file named "arduino.inf", located in the "Drivers" folder of the Arduino Software download (not the "FTDI USB Drivers" sub-directory). If you are using an old version of the IDE (1.0.3 or older), choose the Uno driver file named "Arduino UNO.inf"
- Windows will finish up the driver installation from there.
- You can check that the drivers have been installed by opening the Windows Device Mananger (in the Hardware tab of System control panel). Look for a "USB Serial Port" in the Ports section; that's the Arduino board.

3. Connecting Arduino to Computer

Get an Arduino board and USB cable:



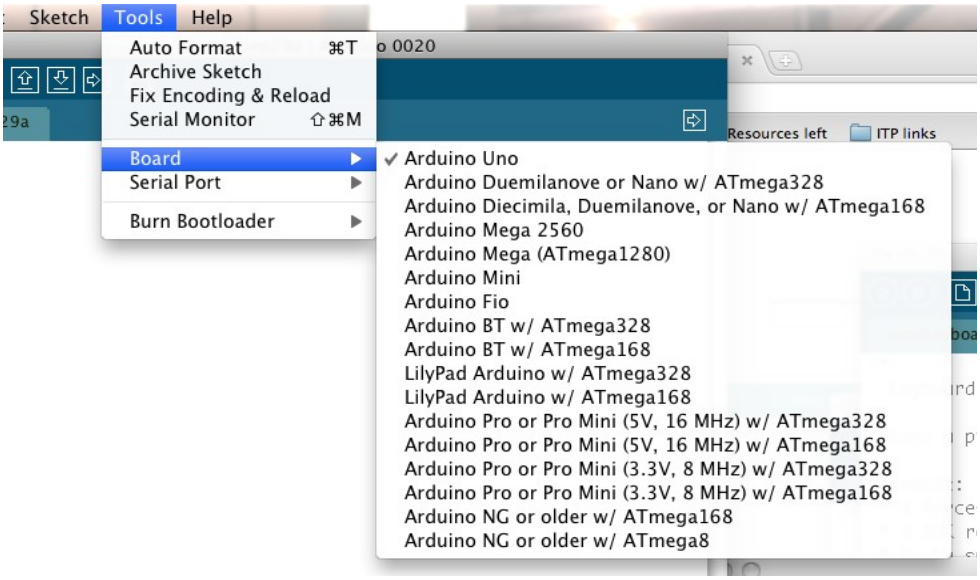
You also need a standard USB cable (A plug to B plug): the kind you would connect to a USB printer, for example. (For the Arduino Nano, you'll need an A to Mini-B cable instead.)

Connect the board:

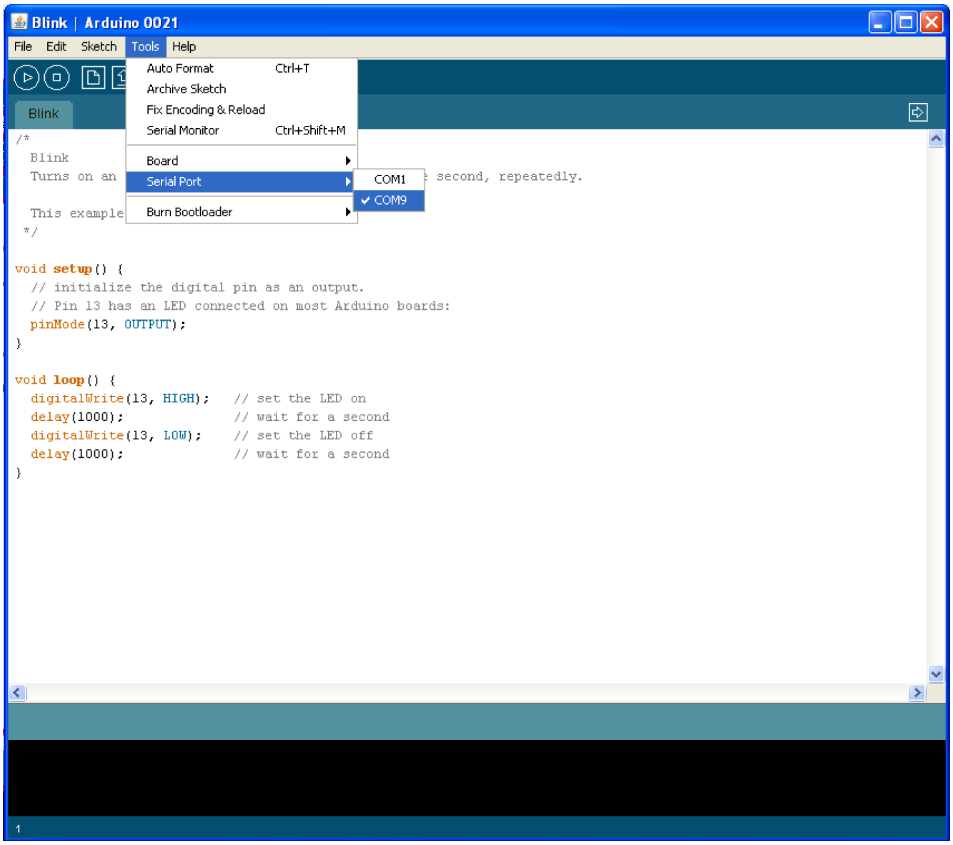
The Arduino Uno automatically draw power from either the USB connection to the computer or an external power supply. Connect the Arduino board to your computer using the USB cable. The green power LED (labelled PWR) should go on.

4.Selecting Device and port in IDE

- You'll need to select the entry in the Tools > Board menu that corresponds to your Arduino.

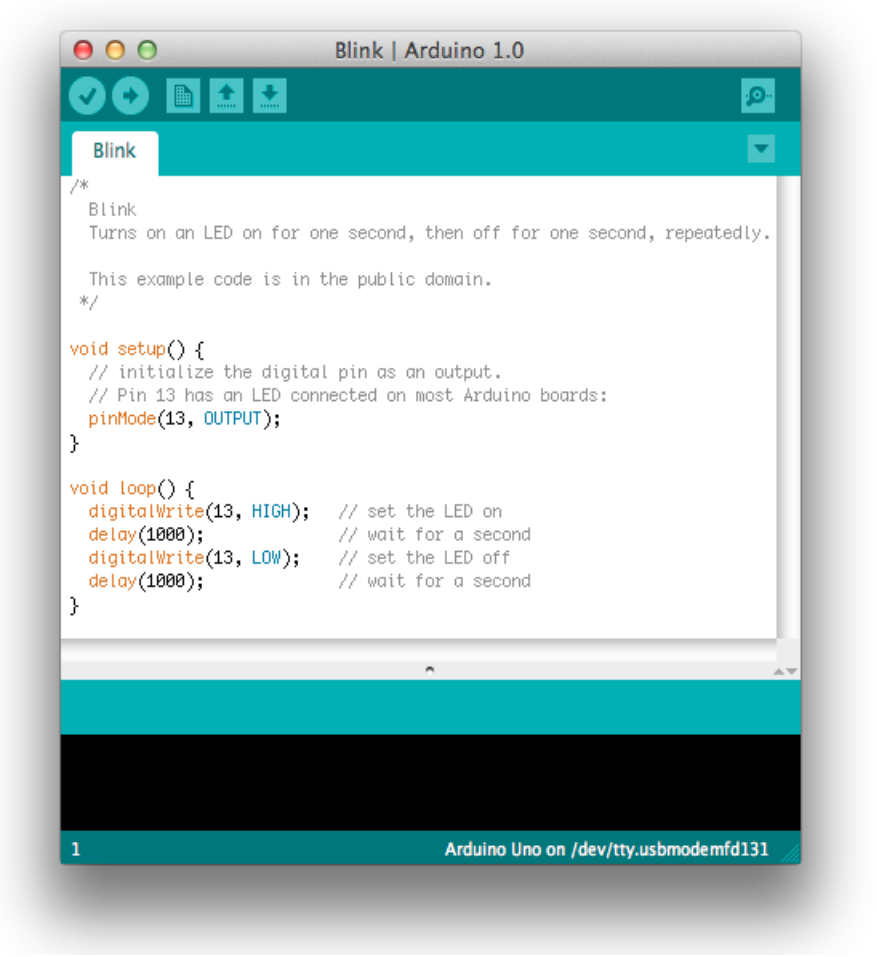


- Select the serial device of the Arduino board from the Tools | Serial Port menu. This is likely to be COM1 or more in Windows and /dev/ttyUSB0 or more in Linux. To find out, you can disconnect your Arduino board and re-open the menu; the entry that disappears should be the Arduino board. Reconnect the board and select that serial port.



5.Uploading sample programs to Arduino

Open the LED blink example sketch: File > Examples > 1.Basics > Blink.



Now, simply click the "Upload" button in the environment. Wait a few seconds - you should see the RX and TX leds on the board flashing. If the upload is successful, the message "Done uploading." will appear in the status bar.

- A few seconds after the upload finishes, you should see the pin 13 (L) LED on the board start to blink (in orange). If it does, congratulations! You've gotten Arduino up-and-running.
- If you have problems, please see the troubleshooting suggestions

  You might also want to look at the examples for using various sensors and actuators.

READING DATA FROM SENSORS

LM35

LM35 is a precision IC temperature sensor with its output proportional to the temperature (in oC). The sensor circuitry is sealed and therefore it is not subjected to oxidation and other processes. With LM35, temperature can be measured more accurately than with a thermistor. It also possess low self heating and does not cause more than 0.1 oC temperature rise in still air. The LM35 is a temperature sensor calibrated to an accuracy of 1 ° C. Its measurement range from -55 ° C to 150 ° C. The output is linear and each degree Celsius equals 10mV.

Features:

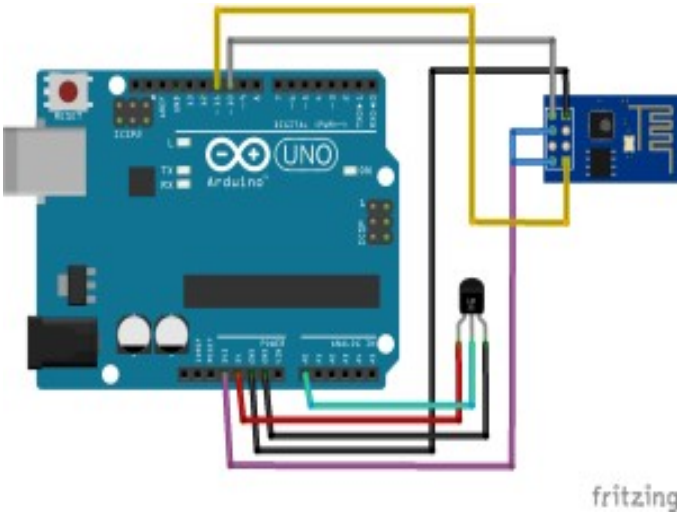
Its main characteristics are:

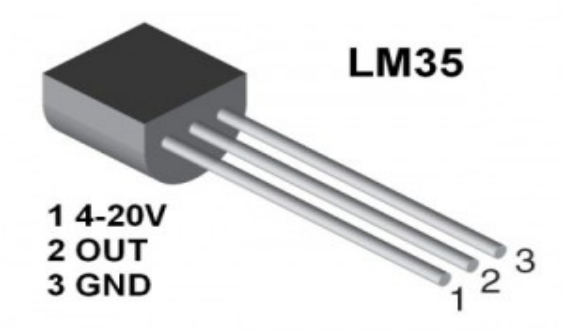
- It is calibrated directly in degrees Celsius.
- The output voltage is proportional to the temperature.
- It has a guaranteed accuracy of 0.5 ° C to 25 ° C.
- Low output impedance.
- Low supply current (60uA).
- Low cost.

Reading through ADC pins:

Interfacing with Arduino:

Connect the LM35 temperature sensor using the pinout shown below. A0 pin should be connected to the out pin of LM35.





Pin No	Function	Name
1	Supply voltage; 5V (+35V to -2V)	Vcc
2	Output voltage (+6V to -1V)	Output
3	Ground (0V)	Ground

ESP8286

ESP8266 offers a complete and self-contained Wi-Fi networking solution, allowing it to either host the application or to offload all Wi-Fi networking functions from another application processor. When ESP8266 hosts the application, and when it is the only application processor in the device, it is able to boot up directly from an external flash. It has integrated cache to improve the performance of the system in such applications, and to minimize the memory requirements. Alternately, serving as a Wi-Fi adapter, wireless internet access can be added to any microcontroller-based design with simple connectivity through UART interface or the CPU AHB bridge interface. ESP8266 on-board processing and storage capabilities allow it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime. With its high degree of on-chip integration, which includes the antenna switch balun, power management converters, it requires minimal external circuitry, and the entire solution, including front-end module, is designed to occupy minimal PCB area.

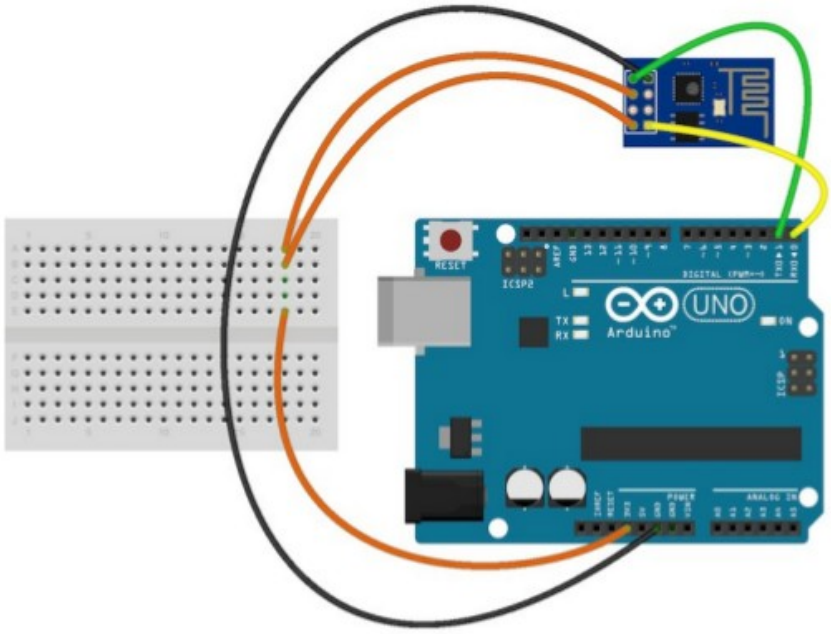
Features

ESP8266 Applications

- Smart power plugs
- Home automation
- Mesh network
- Industrial wireless control
- Baby monitors
- IP Cameras
- Sensor networks
- Wearable electronics
- Wi-Fi location-aware devices
- Security ID tags
- Wi-Fi position system beacons

Interfacing ESP8266 with Arduino:

The module ESP8266 can be connected to the Arduino as shown below.



-
- Plug in the WiFi module.
- Choose the correct serial port from the Tools > Serial Port menu.
- Open the Serial Monitor via the Tools menu or “magnifying glass” icon on the editor window.
- Ensure Carriage return is selected in the line ending pop-up menu at the bottom of the serial monitor.
- For the default firmware version, ensure the communication speed is set to 115200 baud. (For later versions or if it has been modified it will need to be 9600 baud or similar).

ESP8266 power supply and current specs:

- The ESP8266 requires 3.3V power–do not power it with 5 volts!
- The ESP8266 needs to communicate via serial at 3.3V and does not have 5V tolerant inputs, so you need level conversion to communicate with a 5V microcontroller like most Arduinos use.

Mode	Min	Typ	Max	Unit
Transmit 802.11b, CCK 1Mbps, POUT=+19.5dBm		215		mA
Transmit 802.11b, CCK 11Mbps, POUT=+18.5dBm		197		mA
Transmit 802.11g, OFDM 54Mbps, POUT =+16dBm		145		mA
Transmit 802.11n, MCS7, POUT=+14dBm		135		mA
Receive 802.11b, packet length=1024 byte, -80dBm		60		mA
Receive 802.11g, packet length=1024 byte, -70dBm		60		mA
Receive 802.11n, packet length=1024 byte, -65dBm		62		mA
Standby		0.9		mA
Deep sleep		10		uA
Power save mode DTIM 1		1.2		mA
Power save mode DTIM 3		0.86		mA
Total shutdown		0.5		uA

Web Interfacing Concept (IoT)

The Internet of Things (IoT, sometimes Internet of Everything) is the network of physical objects or "things" embedded with electronics, software, sensors, and connectivity to enable objects to exchange data with the manufacturer, operator and/or other connected devices based on the infrastructure of International Telecommunication Union's Global Standards Initiative. The Internet of Things allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration between the physical world and computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Internet of Things, is about how to connect various devices to the network so that they can both send data and receive commands.

Logging temperature through serial port:

https://elementztechblog.wordpress.com/2015/05/13/esp8266-based-temperature-data-logger-using-arduino/