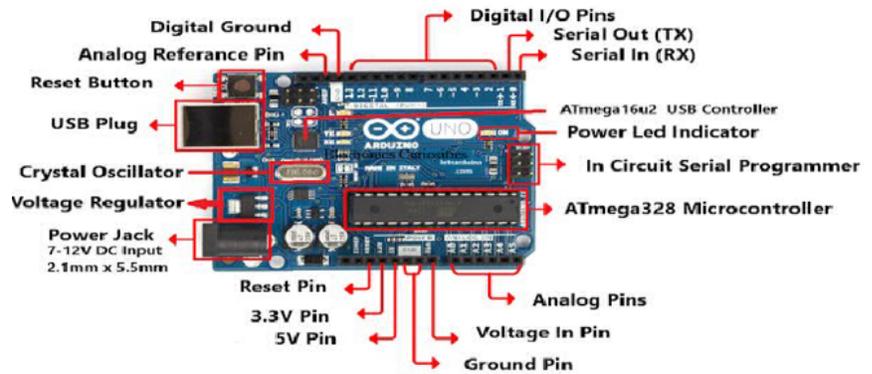


2. Demonstrate Arduino Uno and its pins interfacing with IDE

Remark:

Signature:

The Arduino manages connected components, such as motors or LEDs, by sending output signals to them. Sensor data, received by the Arduino, is considered input. It provides 14 digital input/output pins (pins 0–13) that can be configured as either input or output. A detailed pin reference table is available in Appendix B.

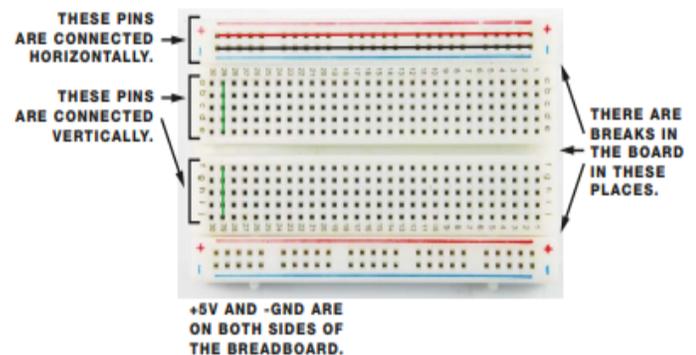


Power

The Arduino Uno is powered through a computer's USB port during program uploads. When not connected to a PC, it can run independently using a 9-volt AC adapter or battery pack with a 2.1 mm jack. Insert the jack into the Arduino's power socket, ensuring the center pin is connected to the positive wire (as depicted in Figure 0-2).

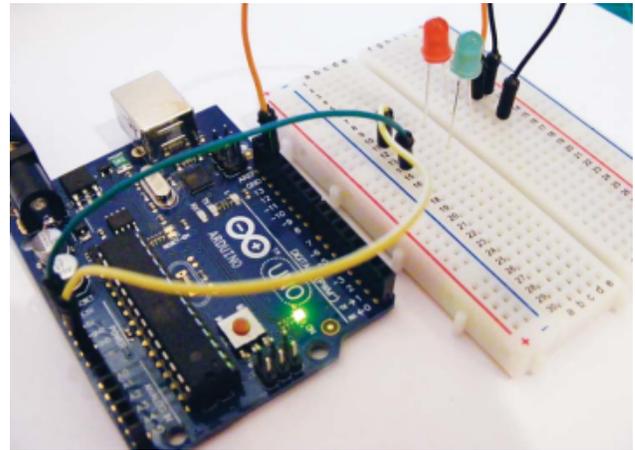
Breadboards

A breadboard, used in the projects here instead of soldering, is a base for electronics prototyping. Originating from wooden boards, early steps involved hammering nails and wrapping wires around them. Today's plastic breadboards, like the one in Figure 0-3, have predrilled holes (tie points) for components or wires. Clips secure them, and conductive strips underneath connect the holes.



Jumper Wires

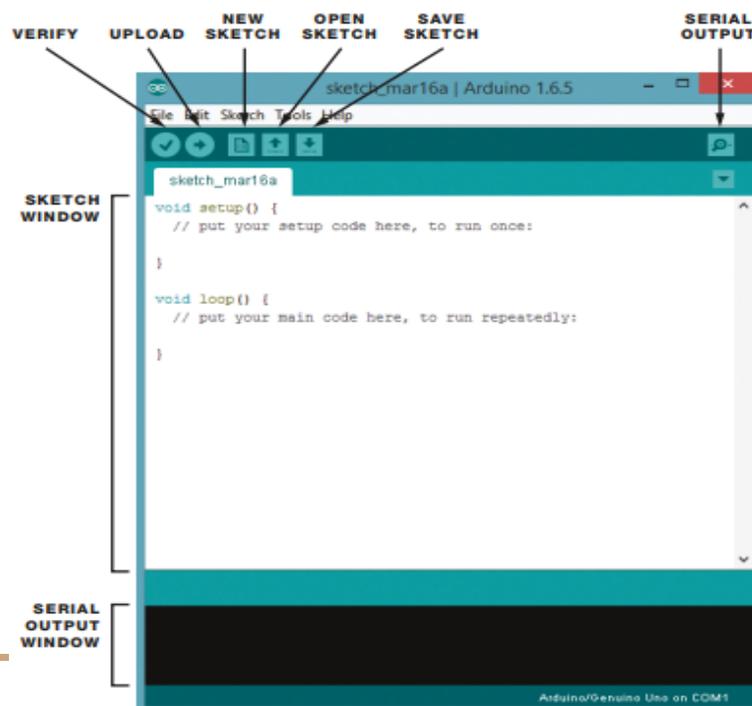
Jumper wires, crucial for breadboard connections, are solid-core wires with molded plastic holders on each end for easy handling. Use solid core wire, as stranded wire lacks the strength needed to push into the hole clips. When inserted into a breadboard hole, the jumper wire is secured beneath the board by a small spring clip, establishing an electrical connection in the row of typically five holes. This setup enables the placement of components in adjacent holes to construct circuits, as shown in Figure 0-5.



Programming the Arduino

To control projects, we use the Arduino Integrated Development Environment (IDE). Downloadable free from <http://www.arduino.cc/>, it runs on Windows, OS X, and Linux. This tool facilitates the creation of computer programs, known as sketches in the Arduino world, which are uploaded to the Arduino using a USB cable. The Arduino then carries out instructions, interacting with the external environment to perform desired tasks.

The IDE Interface



Arduino Sketches

To control projects, we use the Arduino Integrated Development Environment (IDE). Downloadable free from <http://www.arduino.cc/>, it runs on Windows, OS X, and Linux. This tool facilitates the creation of computer programs, known as sketches in the Arduino world, which are uploaded to the Arduino using a USB cable. The Arduino then carries out instructions, interacting with the external environment to perform desired tasks.

Libraries

In the Arduino world, a library is a concise code piece performing a specific function. Instead of repeatedly entering the same code in your sketches, you can use a command to borrow code from the library, saving time and simplifying connections to items like sensors or displays. The Arduino IDE includes built-in libraries, such as LiquidCrystal for LCD displays, and more are available online.

To create the book's projects, import these libraries: RFID, Tone, Pitches, Keypad, Password, Ultrasonic, NewPing, IRRemote, and DHT, accessible at <http://www.nostarch.com/arduinohandbook/>. After downloading, install them by following these steps in Arduino version 1.0.6 and higher:

1. Choose Sketch > Include Library > Add .ZIP Library.
2. Browse to the downloaded ZIP file and select it.

For older Arduino versions, unzip the library file and place the entire folder and its contents into the sketchbook/libraries folder on Linux, My Documents\Arduino\Libraries on Windows, or Documents/Arduino/libraries on OS X. If installed manually, uncompress the ZIP file, creating a folder (e.g., keypad), containing files like keypad.cpp and keypad.h. Move this folder to the libraries folder on your operating system and restart the Arduino application.