

Libraries

The Arduino environment can be extended through the use of libraries, just like most programming platforms. Libraries provide extra functionality for use in sketches, e.g. working with hardware or manipulating data. To use a library in a sketch, select it from Sketch > Import Library.

A number of libraries come installed with the IDE, but you can also download or create your own. See [these instructions](#) for details on installing libraries. There's also a [tutorial on writing your own libraries](#). See the [API Style Guide](#) for information on making a good Arduino-style API for your library.

Standard Libraries

- [EEPROM](#) - reading and writing to "permanent" storage
- [Ethernet](#) - for connecting to the internet using the Arduino Ethernet Shield
- [Firmata](#) - for communicating with applications on the computer using a standard serial protocol.
- [GSM](#) - for connecting to a GSM/GPRS network with the GSM shield.
- [LiquidCrystal](#) - for controlling liquid crystal displays (LCDs)
- [SD](#) - for reading and writing SD cards
- [Servo](#) - for controlling servo motors
- [SPI](#) - for communicating with devices using the Serial Peripheral Interface (SPI) Bus
- [SoftwareSerial](#) - for serial communication on any digital pins. Version 1.0 and later of Arduino incorporate [Mikal Hart's](#) NewSoftSerial library as SoftwareSerial.
- [Stepper](#) - for controlling stepper motors
- [TFT](#) - for drawing text , images, and shapes on the Arduino TFT screen
- [WiFi](#) - for connecting to the internet using the Arduino WiFi shield
- [Wire](#) - Two Wire Interface (TWI/I2C) for sending and receiving data over a net of devices or sensors.

The Matrix and Sprite libraries are no longer part of the core distribution.

Due Only Libraries

- [Audio](#) - Play audio files from a SD card.
- [Scheduler](#) - Manage multiple non-blocking tasks.

Due and Zero Libraries

- [USBHost](#) - Communicate with USB peripherals like mice and keyboards.

Zero Only Libraries

- [AudioZero](#) - Play audio files from a SD card.
- [RTC](#) - Real Time Clock to schedule events.

Esplora Only Library

- [Esplora](#) - this library enable you to easily access to various sensors and actuators mounted on the Esplora board.

Arduino Robot Library

- [Robot](#) - this library enables easy access to the functions of the Arduino Robot

Arduino Yún Bridge Library

- [Bridge Library](#) - Enables communication between the Linux processor and the Arduino on the Yún.

USB Libraries (Leonardo, Micro, Due, Zero and Esplora)

- [Keyboard](#) - Send keystrokes to an attached computer.
- [Mouse](#) - Control cursor movement on a connected computer.

Contributed Libraries

If you're using one of these libraries, you need to install it first. See [these instructions](#) for details on installation. There's also a [tutorial on writing your own libraries](#).

Communication (networking and protocols):

- [Messenger](#) - for processing text-based messages from the computer
- [NewSoftSerial](#) - an improved version of the SoftwareSerial library
- [OneWire](#) - control devices (from Dallas Semiconductor) that use the One Wire protocol.
- [PS2Keyboard](#) - read characters from a PS2 keyboard.
- [Simple Message System](#) - send messages between Arduino and the computer

- [SSerial2Mobile](#) - send text messages or emails using a cell phone (via AT commands over software serial)
- [Webduino](#) - extensible web server library (for use with the Arduino Ethernet Shield)
- [X10](#) - Sending X10 signals over AC power lines
- [XBee](#) - for communicating with XBees in API mode
- [SerialControl](#) - Remote control other Arduinos over a serial connection

Sensing:

- [Capacitive Sensing](#) - turn two or more pins into capacitive sensors
- [Debounce](#) - for reading noisy digital inputs (e.g. from buttons)

Displays and LEDs:

- [GFX](#) - base class with standard graphics routines (by [Adafruit Industries](#))
- [GLCD](#) - graphics routines for LCD based on the KS0108 or equivalent chipset.
- [Improved LCD library](#) fixes LCD initialization bugs in official Arduino LCD library
- [LedControl](#) - for controlling LED matrices or seven-segment displays with a MAX7221 or MAX7219.
- [LedControl](#) - an alternative to the Matrix library for driving multiple LEDs with Maxim chips.
- [LedDisplay](#) - control of a [HCMS-29xx](#) scrolling LED display.
- [Matrix](#) - Basic LED Matrix display manipulation library
- [PCD8544](#) - for the LCD controller on Nokia 55100-like displays (by [Adafruit Industries](#))
- [Sprite](#) - Basic image sprite manipulation library for use in animations with an LED matrix
- [ST7735](#) - for the LCD controller on a 1.8", 128x160 TFT screen (by [Adafruit Industries](#))

Audio and Waveforms:

- [FFT](#) - frequency analysis of audio or other analog signals
- [Tone](#) - generate audio frequency square waves in the background on any microcontroller pin

Motors and PWM:

- [TLC5940](#) - 16 channel 12 bit PWM controller.

Timing:

- [DateTime](#) - a library for keeping track of the current date and time in software.
- [Metro](#) - help you time actions at regular intervals
- [MsTimer2](#) - uses the timer 2 interrupt to trigger an action every N milliseconds.

Utilities:

- [PString](#) - a lightweight class for printing to buffers
- [Streaming](#) - a method to simplify print statements

[Reference Home](#)

Corrections, suggestions, and new documentation should be posted to the [Forum](#).

The text of the Arduino reference is licensed under a [Creative Commons Attribution-ShareAlike 3.0 License](#). Code samples in the reference are released into the public domain.