Arduino: Reference / Reference

Reference  Language (extended)  Libraries  Comparison  Board

Language Reference

See the extended reference for more advanced features of the Arduino languages and the libraries page for interfacing with particular types of hardware.

Arduino programs can be divided into three main parts: structure, values (variables and constants), and functions. The Arduino language is based on C/C++.

Structure

An Arduino program run in two parts:

- void setup()
- void loop()

setup() is preparation, and loop() is execution. In the setup section, always at the top of your program, you would set pinMode, initialize serial communication, etc. The loop section is the code to be executed -- reading inputs, triggering outputs, etc.

- Variable Declaration
- Function Declaration
  - void

Control Structures

- if
- if...else
- for
- switch...case
- while
- do...while
- break
- continue
- return

Further Syntax

- ; (semicolon)
- {} (curly braces)
- // (single line comment)
- /* */ (multi-line comment)

Arithmetic Operators

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Functions

Digital I/O
- pinMode(pin, mode)
- digitalWrite(pin, value)
- int digitalWrite(pin)

Analog I/O
- int analogRead(pin)
- analogWrite(pin, value) - PWM

Advanced I/O
- shiftOut(dataPin, clockPin, bitOrder, value)
- unsigned long pulseIn(pin, value)

Time
- unsigned long millis()
- delay(ms)
- delayMicroseconds(us)

Math
- min(x, y)
- max(x, y)
- abs(x)
- constrain(x, a, b)
- map(value, fromLow, fromHigh, toLow, toHigh)
- pow(base, exponent)
- sqrt(x)

Trigonometry
- sin(rad)
- cos(rad)
- tan(rad)

Random Numbers
- randomSeed(seed)
- long random(max)
- long random(min, max)

Serial Communication

Used for communication between the Arduino board and a computer or other devices. This communication happens via the Arduino board's serial or USB
Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)
- ! (not)

Compound Operators

- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)

Variables

Variables are expressions that you can use in programs to store values, such as a sensor reading from an analog pin.

Constants

Constants are particular values with specific meanings.

- HIGH | LOW
- INPUT | OUTPUT
- true | false

- Integer Constants

Data Types

Variables can have various types, which are described below.

- boolean
- char
- byte
- int
- unsigned int
- long
- unsigned long
- float
- double
- string
- array

Reference

connection and on digital pins 0 (RX) and 1 (TX). Thus, if you use these functions, you cannot also use pins 0 and 1 for digital i/o.

- Serial.begin(speed)
- int Serial.available()
- int Serial.read()
- Serial.flush()
- Serial.print(data)
- Serial.println(data)

Didn't find something? Check the extended reference or the libraries.
Arduino Reference (extended)

The Arduino language is based on C/C++ and supports all standard C constructs and some C++ features. It links against AVR Libc and allows the use of any of its functions; see its user manual for details.

Structure

In Arduino, the standard program entry point (main) is defined in the core and calls into two functions in a sketch. `setup()` is called once, then `loop()` is called repeatedly (until you reset your board).

- void `setup()`
- void `loop()`

Control Structures

- if
- else
- for
- switch case
- while
- do...while
- break
- continue
- return

Further Syntax

- ; (semicolon)
- { } (curly braces)
- // (single line comment)
- /* */ (multi-line comment)
- #define
- #include

Arithmetic Operators

- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Functions

Digital I/O

- `pinMode(pin, mode)`
- `digitalWrite(pin, value)`
- `int digitalRead(pin)`

Analog I/O

- `analogReference(type)`
- `int analogRead(pin)`
- `analogWrite(pin, value) - PWM`

Advanced I/O

- `shiftOut(dataPin, clockPin, bitOrder, value)`
- `unsigned long pulseIn(pin, value)`

Time

- `unsigned long millis()`
- `delay(ms)`
- `delayMicroseconds(us)`

Math

- `min(x, y)`
- `max(x, y)`
- `abs(x)`
- `constrain(x, a, b)`
- `map(value, fromLow, fromHigh, toLow, toHigh)`
- `pow(base, exponent)`
- `sqrt(x)`

Trigonometry

- `sin(rad)`
- `cos(rad)`
- `tan(rad)`

Random Numbers

- `randomSeed(seed)`
- `long random(max)`
- `long random(min, max)`

External Interrupts

- `attachInterrupt(interrupt, function, mode)`
- `detachInterrupt(interrupt)`

Interrupts
Boolean Operators
- && (and)
- || (or)
- ! (not)

Pointer Access Operators
- * dereference operator
- & reference operator

Bitwise Operators
- & (bitwise and)
- | (bitwise or)
- ^ (bitwise xor)
- ~ (bitwise not)
- << (bitshift left)
- >> (bitshift right)
- Port Manipulation

Compound Operators
- ++ (increment)
- -- (decrement)
- += (compound addition)
- -= (compound subtraction)
- *= (compound multiplication)
- /= (compound division)
- &= (compound bitwise and)
- |= (compound bitwise or)

Variables

Constants
- HIGH | LOW
- INPUT | OUTPUT
- true | false
- integer constants
- floating point constants

Data Types
- void keyword
- boolean
- char
- unsigned char
- byte
- int
- unsigned int
- long
- unsigned long
- float
- double

Serial Communication
- Serial.begin(speed)
- int Serial.available()
- int Serial.read()
- Serial.flush()
- Serial.println(data)
• string
• array

Variable Scope & Qualifiers

• static
• volatile
• const
• PROCMEM

Utilities

• cast (cast operator)
• sizeof() (sizeof operator)

Reference

• keywords
• ASCII chart
• Atmega168 pin mapping

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.