A

Data Types in C

The following are the data types used in the C programming language:

Type	Definition	Size in memory
`void`	This particular type is used only in function declaration.	
`boolean`	It stores `false` or `true.`	1 byte (8 bits)
`char`	It stores single quoted characters such as `a` as numbers following the ASCII chart. It is a signed type and stores numbers from -128 to 127; it can be unsigned and then stores numbers from 0 to 255.	1 byte
`byte`	It stores numbers as 8-bit unsigned data, that means from 0 to 255.	8 bits
`int`	It stores numbers as 2 bytes signed data, which means from -32768 to 32767. It also can be unsigned and then stores numbers from 0 to 65535.	2 bytes (16 bits)
`word`	It stores numbers as 2 bytes unsigned data exactly as unsigned `int` does.	2 bytes (16 bit)
`long`	It stores numbers as 4 bytes signed data, which means from -2147483648 to 2147483647 and can be unsigned and then stores numbers from 0 to 4294967295.	4 bytes (32 bit)
`float`	It basically stores numbers with a decimal point from -3.4028235E+38 to 3.4028235E+38 as 4 bytes signed data. Be careful to set the required precision; they have no more than 6 to 7 decimal digits and can give strange rounded results sometimes.	4 bytes (32 bit)

Type	Definition	Size in memory
`double`	It generally stores float with a precision two times that of a float. Be careful, in the Arduino IDE and board, double implementation is exactly the same as float, that means with only six to seven decimal digits of precision.	4 bytes (32 bit)
`array`	An ordered structure of consecutive elements of the same type that can each be accessed with an index number.	Number of elements * size of elements' type
`string`	It stores text strings in an array of `char` where the last element is `null`, which is a particular character (ASCII code `0`). Be careful of the small 's' at the beginning of string.	Number of elements * 1 byte
`String`	It is a particular structure of data named a class, which provides a nice way to use and work with strings of text. It comes with a method/function to easily concatenate, split strings, and much more. Be careful to write the capital 'S' at the beginning of String.	Available each time with the `length()` method

B
Operator Precedence in C and C++

Precedence	Operator	Description	Associativity
1 (highest)	`::`	Scope resolution (C++ only)	Left-to-right
2	`++`	Suffix increment	
	`--`	Suffix decrement	
	`()`	Function call	
	`[]`	Array subscripting	
	`.`	Element selection by reference	
	`->`	Element selection through pointers	
	`typeid()`	Runtime type information (C++ only)	
	`const_cast`	Type cast (C++ only)	
	`dynamic_cast`	Type cast (C++ only)	
	`reinterpret_cast`	Type cast (C++ only)	
	`static_cast`	Type cast (C++ only)	

Precedence	Operator	Description	Associativity
3	`++`	Prefix increment	Right-to-left
	`--`	Prefix decrement	
	`+`	Unary plus	
	`-`	Unary minus	
	`!`	Logical NOT	
	`~`	Bitwise NOT	
	`(type)`	Type cast	
	`*`	Indirection (dereference)	
	`&`	Address-of	
	`sizeof`	Size-of	
	`new`, `new[]`	Dynamic memory allocation (C++ only)	
	`delete`, `delete[]`	Dynamic memory deallocation (C++ only)	
4	`.*`	Pointer to member (C++ only)	Left-to-right
	`->*`	Pointer to member (C++ only)	
5	`*`	Multiplication	
	`/`	Division	
	`%`	Modulo (remainder)	
6	`+`	Addition	
	`-`	Subtraction	
7	`<<`	Bitwise left shift	
	`>>`	Bitwise right shift	
8	`<`	Less than	
	`<=`	Less than or equal to	
	`>`	Greater than	
	`>=`	Greater than or equal to	
9	`==`	Equal to	
	`!=`	Not equal to	
10	`&`	Bitwise AND	
11	`^`	Bitwise XOR (exclusive OR)	
12	`|`	Bitwise OR (inclusive OR)	
13	`&&`	Logical AND	
14	`||`	Logical OR	

Precedence	Operator	Description	Associativity
15	`?:`	Ternary conditional	Right-to-left
	`=`	Direct assignment	
	`+=`	Assignment by sum	
	`-=`	Assignment by difference	
	`*=`	Assignment by product	
	`/=`	Assignment by quotient	
	`%=`	Assignment by remainder	
	`<<=`	Assignment by bitwise left shift	
	`>>=`	Assignment by bitwise right shift	
	`&=`	Assignment by bitwise AND	
	`^=`	Assignment by bitwise XOR	
	`|=`	Assignment by bitwise OR	
16	`throw`	Throw operator (exceptions throwing, C++ only)	
17 (lowest)	`,`	Comma	Left-to-right

C
Important Math Functions

The `Math.h` header file contains the trigonometry functions prototype, so does the Arduino core. A few of those functions are as follows:

- `double cos (double x)`: Returns cosine of *x* radians
- `double sin (double x)`: Returns sine of *x* radians
- `double tan (double x)`: Returns tangent of *x* radians
- `double acos (double x)`: Returns A, an angle corresponding to cos (A) = x
- `double asin (double x)`: Returns A, an angle corresponding to sin (A) = x
- `double atan (double x)`: Returns A, an angle corresponding to tan (A) = x
- `double atan2 (double y, double x)`: Returns arctan (y/x)

The Arduino core also implements the following:

- `double pow (double x, double y)`: Returns *x* to power *y*
- `double exp (double x)`: Returns exponential value of *x*
- `double log (double x)`: Returns natural logarithm of *x* with *x* greater than 0
- `double log10 (double x)`: Returns logarithm of *x* to base 10 with *x* greater than 0
- `double square (double x)`: Returns square of *x*
- `double sqrt (double x)`: Returns square root of *x* with *x* greater than or equal to 0
- `double abs (double x)`: Returns absolute value of *x*

Of course, mathematical rules, especially considering range of values, have to be respected. This is why I added some conditions about *x* in the table.

D

Some Useful Taylor Series for Calculation Optimization

This Appendix mentions some useful Taylor series formulas for efficient calculation optimization. The following are a few of the formulas:

$$\sin x \sim x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \ldots\ldots\ldots \text{for all x}$$

$$\cos x \sim 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} \ldots\ldots\ldots \text{for all x}$$

$$\tan x \sim x + \frac{x^3}{3} + \frac{2x^5}{15} \ldots\ldots\ldots \text{for } (x) < \frac{p}{2}$$

$$\sqrt{(x^2 + a)} \sim x + \frac{a}{2x}$$

E

ASCII Table

Decimal	Octal	Hex	Binary	Value	Description
0	0	0	0	NUL	Null char
1	1	1	1	SOH	Start of header
2	2	2	10	STX	Start of text
3	3	3	11	ETX	End of text
4	4	4	100	EOT	End of transmission
5	5	5	101	ENQ	Enquiry
6	6	6	110	ACK	Acknowledgment
7	7	7	111	BEL	Bell
8	10	8	1000	BS	Backspace
9	11	9	1001	HT	Horizontal tab
10	12	00A	1010	LF	Line feed
11	13	00B	1011	VT	Vertical tab
12	14	00C	1100	FF	Form feed
13	15	00D	1101	CR	Carriage return
14	16	00E	1110	SO	Shift out
15	17	00F	1111	SI	Shift in
16	20	10	10000	DLE	Data link escape
17	21	11	10001	DC1	XON / Device control 1
Decimal	Octal	Hex	Binary	Value	
18	22	12	10010	DC2	Device control 2
19	23	13	10011	DC3	XOFF / Device control 3
20	24	14	10100	DC4	Device control 4
21	25	15	10101	NAK	Negative acknowledgement
22	26	16	10110	SYN	Synchronous idle

Decimal	**Octal**	**Hex**	**Binary**	**Value**	**Description**
23	27	17	10111	ETB	End of Transmission Block
24	30	18	11000	CAN	Cancel
25	31	19	11001	EM	End of medium
26	32	01A	11010	SUB	Substitute
27	33	01B	11011	ESC	Escape
28	34	01C	11100	FS	File separator
29	35	01D	11101	GS	Group separator
30	36	01E	11110	RS	Request to send / Record separator
31	37	01F	11111	US	Unit separator
32	40	20	100000	SP	Space
33	41	21	100001	!	Exclamation mark
34	42	22	100010	“	Double quote
35	43	23	100011	#	Number sign
36	44	24	100100	$	Dollar sign
37	45	25	100101	%	Percent
38	46	26	100110	&	Ampersand
39	47	27	100111	‘	Single quote
40	50	28	101000	(	Left/opening parenthesis
41	51	29	101001	)	Right/closing parenthesis
42	52	02A	101010	*	Asterisk
43	53	02B	101011	+	Plus
44	54	02C	101100	,	Comma
45	55	02D	101101	-	Minus or dash
46	56	02E	101110	.	Dot
Decimal	Octal	Hex	Binary	Value	
47	57	02F	101111	/	Forward slash
48	60	30	110000	0	
49	61	31	110001	1	
50	62	32	110010	2	
51	63	33	110011	3	
52	64	34	110100	4	
53	65	35	110101	5	
54	66	36	110110	6	
55	67	37	110111	7	

Decimal	**Octal**	**Hex**	**Binary**	**Value**	**Description**
56	70	38	111000	8	
57	71	39	111001	9	
58	72	03A	111010	:	Colon
59	73	03B	111011	;	Semicolon
60	74	03C	111100	<	Less than
61	75	03D	111101	=	Equal sign
62	76	03E	111110	>	Greater than
63	77	03F	111111	?	Question mark
64	100	40	1000000	@	At symbol
65	101	41	1000001	A	
66	102	42	1000010	B	
67	103	43	1000011	C	
68	104	44	1000100	D	
69	105	45	1000101	E	
70	106	46	1000110	F	
71	107	47	1000111	G	
72	110	48	1001000	H	
73	111	49	1001001	I	
74	112	04A	1001010	J	
75	113	04B	1001011	K	
Decimal	Octal	Hex	Binary	Value	
76	114	04C	1001100	L	
77	115	04D	1001101	M	
78	116	04E	1001110	N	
79	117	04F	1001111	O	
80	120	50	1010000	P	
81	121	51	1010001	Q	
82	122	52	1010010	R	
83	123	53	1010011	S	
84	124	54	1010100	T	
85	125	55	1010101	U	
86	126	56	1010110	V	
87	127	57	1010111	W	
88	130	58	1011000	X	

Decimal	**Octal**	**Hex**	**Binary**	**Value**	**Description**
89	131	59	1011001	Y	
90	132	05A	1011010	Z	
91	133	05B	1011011	[	Left/opening bracket
92	134	05C	1011100	\	Back slash
93	135	05D	1011101	]	Right/closing bracket
94	136	05E	1011110	^	Caret/circumflex
95	137	05F	1011111	_	Underscore
96	140	60	1100000	`	
97	141	61	1100001	a	
98	142	62	1100010	b	
99	143	63	1100011	c	
100	144	64	1100100	d	
101	145	65	1100101	e	
102	146	66	1100110	f	
103	147	67	1100111	g	
104	150	68	1101000	h	
Decimal	**Octal**	**Hex**	**Binary**	**Value**	
105	151	69	1101001	i	
106	152	06A	1101010	j	
107	153	06B	1101011	k	
108	154	06C	1101100	l	
109	155	06D	1101101	m	
110	156	06E	1101110	n	
111	157	06F	1101111	o	
112	160	70	1110000	p	
113	161	71	1110001	q	
114	162	72	1110010	r	
115	163	73	1110011	s	
116	164	74	1110100	t	
117	165	75	1110101	u	
118	166	76	1110110	v	
119	167	77	1110111	w	
120	170	78	1111000	x	
121	171	79	1111001	y	

Decimal	Octal	Hex	Binary	Value	Description
122	172	07A	1111010	z	
123	173	07B	1111011	{	Left/opening brace
124	174	07C	1111100	|	Vertical bar
125	175	07D	1111101	}	Right/closing brace
126	176	07E	1111110	~	Tilde
127	177	07F	1111111	DEL	Delete

F
How to Install a Library

A library often comes as a ZIP file. Uncompress it.

It usually contains a folder with the same name as the library, which contains the following files:

- Header files (`.h`)
- Source files (`.cpp`)
- The `examples` folder
- The `keywords.txt` file for code coloration in the IDE

You have to take this whole folder and to move it to a specific location on your computer.

On Windows systems, this location is `My Documents\Arduino\libraries\`. On OS X and Linux systems, this location is `Documents/Arduino/libraries/`.

You have to restart the IDE and the new library will be available.

G

List of Components' Distributors

Here is a non-exhaustive list of components and circuits distributors in the world. This is a personal selection according to my own experience and you can trust all of them for fast and accurate shipping!

SparkFun Electronics

You can find everything, from small components to Arduino boards, shields, sensors, and much more. They use UPS, and they deliver fast, all over the world.

URL: `https://www.sparkfun.com`

Location: USA

Semageek

Bunches of Arduino boards are always available here among a lot of other nice stuff, such as sensors, LEDs, and potentiometers. This is the official Arduino distributor in France and also a friend of mine.

URL: `http://boutique.semageek.com`

Location: France

Adafruit

The website and shop of Limor Fried, a former MIT engineer.

A huge list of products are available here, including the famous wearable Arduino and other chip shields. They manufacture a lot of circuits themselves for worldwide distribution.

URL: `http://www.adafruit.com`

Location: USA

Farnell

This global and general distributor provides products in large quantities at good prices.

URL: `http://www.farnell.com`

Location: UK

Parallax

They make the Basic Stamp family of microcontrollers and also the Propeller microcontroller. A lot of other stuff is available too.

URL: `http://www.parallax.com`

Location: USA

Mouser

This general distributor provides a very wide range of products from discrete components to more sophisticated circuits.

URL: `http://www.mouser.com`

Location: USA

Schmartboard

Great site for all projects where you might want to use **surface mount technology** (**SMT**) to keep things even smaller. They have a specific inexpensive soldering technique with which they show you how to use for all their surface mount technology components, including ICs.

URL: `www.schmartboard.com`

Location: USA

H
Useful Links to Keep in Mind

C programming for Arduino

This is the official website of the book maintained and powered by me directly.

URL: `http://cprogrammingforarduino.com`

C programming for Arduino Facebook page

This is the official Facebook page of the book. Things will be published here progressively and new circuits and ideas will be posted.

URL: `https://www.facebook.com/C.Programming.for.Arduino`

Official Arduino website

The official Arduino website really is a huge source of information. Some forums are full of advice and answers to a lot of real-life questions. You can find documentation, a references page, and of course the Arduino IDE to download.

URL: `http://www.arduino.cc`

Processing

This is the official Processing website. Forums and references are present here too.

URL: `http://processing.org`

Fritzing

This is the official Fritzing website. Forums and references are present here too.

URL: `http://fritzing.org`

Cycling 74

This is the website for the official Max 6 framework and related stuff. Forums, documentation, and tutorials are present here too.

URL: `http://cycling74.com`

julienbayle.net

This is a website for my more artistic side. There is a blog and you can find a bunch of information and Max 6 patches around there. You'll also find my whole internet's entry points and more.

URL: `http://julienbayle.net`

Design the Media

This is the website that I have designed; it provides courses based on art and technology, from Ableton Live to Max6 and of course Arduino and Processing.

URL: `http://designthemedia.com`