CHEATHERO SHEETSHERO

Arduino Cheatsheet

A quick reference guide for Arduino development, covering essential concepts, code snippets, and hardware information for both beginners and advanced users.



Arduino Fundamentals

Basic Structure

Every Arduino program (sketch) has two essential functions:	int
<pre>void setup() {} - Runs once at the beginning of the program. Used for initialization.</pre>	byt e
<pre>void loop() {} - Runs repeatedly after setup() finishes. This is where the main</pre>	
program logic goes.	flo
Example:	t
<pre>void setup() {</pre>	boo ean
<pre>// Initialize serial communication Serial.begin(9600);</pre>	
}	
<pre>void loop() {</pre>	
<pre>// Print "Hello, world!" to the serial</pre>	
monitor	
<pre>Serial.println("Hello, world!");</pre>	
delay(1000); // Wait for 1 second	
3	

Analog I/O & Serial Communication

Analog I/O

analog Read(pi n)	Reads the value from the specified analog pin (0 to 1023).
analog Write(p in, value)	Writes an analog value (PWM signal) to a pin (0 to 255). Only works on PWM enabled pins (marked with ~ on the board).
<pre>analog Referen ce(type)</pre>	Configures the reference voltage used for analog input (DEFAULT , INTERNAL , EXTERNAL).

Data Types

int	Integer (whole number). Typically 2 bytes (-32,768 to 32,767).
byt	Unsigned integer (0 to 255).
lon	Long integer. Typically 4 bytes (-2,147,483,648 to 2,147,483,647).
floa t	Floating-point number (number with decimal point). 4 bytes.
bool ean	Boolean value (true or false).
cha r	Character. 1 byte.

Digital I/O

pinMode(pi	Sets the specified pin as INPUT,
n, mode)	OUTPUT, or INPUT_PULLUP.
digitalWri	Writes HIGH or LOW to a digital
te(pin,	pin (only if the pin is set as
value)	OUTPUT).
digitalRea d(pin)	Reads the value (HIGH or LOW) from a digital pin.

Serial Communication

Serial.b	Initializes serial communication at
egin(baud	the specified baud rate (e.g., 9600,
Rate)	115200).
Serial.p rint(data)	Sends data to the serial port as human-readable ASCII text.
Serial.p	Sends data to the serial port,
rintln(da	followed by a carriage return and
ta)	line feed.
Serial.a	Gets the number of bytes
vailable((characters) available for reading
)	from the serial port.
Serial.r ead()	Reads the first available byte of incoming serial data.

Time Functions

dela y(mil lisec onds)	Pauses the program for the amount of time (in milliseconds) specified as parameter.
<pre>mill is()</pre>	Returns the number of milliseconds since the Arduino board began running the current program. This number will overflow (go back to zero) after approximately 50 days.
<pre>micr os()</pre>	Returns the number of microseconds since the Arduino board began running the current program. This number will overflow (go back to zero) after approximately 70 minutes.

Control Structures & Operators

Control Structures

if	(condition)	{		}	- Executes code
block if the condition is true.					

if (condition) { ... } else { ... } -Executes one code block if the condition is true and another if the condition is false.

for (initialization; condition; increment)
{ ... } - Repeats a block of code a specific
number of times.

while (condition) { ... } - Repeats a block
of code as long as the condition is true.

(do { ... } while (condition); - Executes a block of code once, and then repeats as long as the condition is true.

switch (expression) { case value1: ... break; case value2: ... break; default: ... } - Selects one of several code blocks to execute based on the value of an expression.

Attaches an interrupt to the specified pin. **ISR** is the

interrupt service routine, and

RISING, or FALLING.

specified pin.

mode can be LOW , CHANGE ,

Detaches the interrupt on the

Re-enables interrupts (after they

have been disabled by

noInterrupts()).

Disables interrupts.

Advanced Arduino

Interrupts

attachInte

rrupt(digit

alPinToInte

rrupt(pin),

ISR, mode)

detachInte

rrupt(digit alPinToInte

rrupt(pin)

interrupts

noInterrup

)

()

ts()

Operators

- Assignment operator (assigns a value to a variable).
- Equality operator (checks if two values are equal).
- Inequality operator (checks if two values are not equal).
- > Greater than operator.
- < Less than operator.
- Logical AND operator (returns true if bothconditions are true).
- Logical OR operator (returns true if at leastone condition is true).
- ! Logical NOT operator (reverses the logical state of its operand).

Libraries

Libraries provide extra functionality to your sketches. To include a library: #include <LibraryName.h>

Examples:

- #include <Wire.h> For I2C
 communication
- #include <SPI.h> For SPI communication
- #include <Servo.h> For controlling servo motors
- #include <LiquidCrystal.h> For LCD display

Math Operators

- + Addition.
- Subtraction.
 Multiplication.
- / Division.
- % Modulo (returns the remainder of a division).

EEPROM

EEPROM.w rite(addr ess, value)	Writes a byte to the EEPROM at the specified address.
EEPROM.r ead(addre ss)	Reads a byte from the EEPROM at the specified address.
EEPROM.u pdate(add ress, value)	Writes a byte to the EEPROM at the specified address, only if the new value is different from the old value.