

Analog I/O

<code>int analog- Read(analogPin)</code>	ließt Spannung auf <i>analogPin</i> als Wert zwischen 0 und 1023
<code>analogRef- erence(mode)</code>	legt Referenzspannung für <i>HIGH</i> fest
DEFAULT, INTERNAL, EXTERNAL	Spannungsreferenzmodi
<code>analog- Write(Wert)</code>	erzeugt PWM-Welle mit einem Wert zwischen 0 und 255

Interrupts

<code>digitalPinToInterrupt(Pin)</code>	wandelt digitalen <i>Input-Pin</i> in Interrupt-Pin um
LOW, CHANGE, RISING, FALLING	Interrupt-Typen
<code>attachInterrupt(Interrupt, Function, Typ)</code>	verbindet <i>Interrupt-Pin</i> mit <i>Funktion</i>
<code>detachInterrupt(Interrupt)</code>	hebt Verbindung zwischen <i>Funktion</i> und <i>Interrupt-Pin</i> auf
<code>interrupts()</code>	aktiviert Interrupts
<code>noInterrupts()</code>	deaktiviert Interrupts

Fortgeschrittene I/O

<code>tone(Pin, Frequenz)</code>	<i>Frequenz</i> in Hz wird auf <i>Pin</i> ausgegeben
<code>tone(Pin, Frequenz, Dauer)</code>	auf <i>Pin</i> wird eine <i>Frequenz</i> für <i>Dauer</i> Millisekunden ausgegeben
<code>noTone(Pin)</code>	Tonausgabe auf <i>Pin</i> wird beendet
<code>pulseIn(Pin, Wert)</code>	misst die Dauer die <i>Pin</i> *Wert annimmt (HIGH/LOW)

Zeit

<code>unsigned int millis()</code>	Millisekunden seit Programmstart. ~50 Tage Overflow
<code>unsigned int micros()</code>	Microsekunden seit Programmstart. ~70 min bis Overflow
<code>delay(ms)</code>	Pause für <i>ms</i> Millisekunden
<code>delayMicroseconds(us)</code>	Pause für <i>us</i> Microsekunden

Mathematik

<code>min(x, y), max(x, y), abs(x)</code>	Minimum, Maximum, absoluter Wert
<code>pow(Basis, Exponent)</code>	Basis ^{Exponent}
<code>sqrt(x)</code>	Wurzel(x)
<code>sin(x), cos(y), tan(z)</code>	Sinus, Cosinus, Tangens

Mathematik (cont)

<code>map(Wert, lim1U, lim1O, lim2U, lim2O*)</code>	skaliert <i>Wert</i> zwischen <i>lim1U</i> und <i>lim1O</i> zu Wert zwischen <i>lim2U</i> und <i>lim2O</i>
<code>constrain(Wert, limU, limO)</code>	beschränkt <i>Wert</i> zwischen <i>unterem</i> und <i>oberem Limit</i>

Zufallszahlen

<code>randomSeed(Wert)</code>	Ausgangswert für Zufalls-generator
<code>random(max)</code>	Zufallszahl zwischen 0 und <i>max</i>
<code>random(min, max)</code>	Zufallszahl zwischen <i>min</i> und <i>max</i>

Serial

<code>Serial.begin(9600);</code>	startet Serial -verbindung mit Baudrate
<code>Serial.println(inhalt);</code>	sendet Zeile mit <i>Inhalt</i>
<code>Serial.print(text);</code>	sendet <i>Text</i>
<code>Serial.write(daten);</code>	sendet <i>Daten</i> als Binärcode
<code>Serial.flush();</code>	wartet bis <i>Daten</i> gesendet sind
<code>Serial.end();</code>	beendet Serial-Verbindung

Servo

<code>#include <Servo.h></code>	fügt Servo--Library hinzu
<code>attach(pin)</code>	fügt <i>Servopin</i> hinzu
<code>write(Winkel)</code>	stellt Servomotor auf <i>Winkel</i> (0-180°)
<code>read()</code>	liest Winkel von Servomotor