

[illegible]

Analog I/O		Fortgeschrittene I/O		Mathematik (cont)		Servo	
<code>int analog-Read(<i>analogPin</i>)</code>	liest Spannung auf <i>analogPin</i> als Wert zwischen 0 und 1023	<code>tone(<i>Pin</i>, <i>Frequenz</i>)</code>	<i>Frequenz</i> in Hz wird auf <i>Pin</i> ausgegeben	<code>map(<i>Wert</i>, <i>lim1U</i>, <i>lim1O</i>, <i>lim2U</i>, <i>lim2O</i>*)</code>	skaliert <i>Wert</i> zwischen <i>lim1U</i> und <i>lim1O</i> zu Wert zwischen <i>lim2U</i> und <i>lim2O</i>	<code>#include <Servo.h></code>	fügt Servo--Library hinzu
<code>analogReference(mode)</code>	legt Referenzspannung für <i>HIGH</i> fest	<code>tone(<i>Pin</i>, <i>Frequenz</i>, <i>Dauer</i>)</code>	auf <i>Pin</i> wird eine <i>Frequenz</i> für <i>Dauer</i> Millisekunden ausgegeben	<code>constrain(<i>Wert</i>, <i>limU</i>, <i>limO</i>)</code>	beschränkt <i>Wert</i> zwischen <i>unterem</i> und <i>oberem Limit</i>	<code>attach(<i>pin</i>)</code>	fügt <i>Servopin</i> hinzu
DEFAULT, INTERNAL, EXTERNAL	Spannungsreferenzmodi	<code>noTone(<i>Pin</i>)</code>	Tonausgabe auf <i>Pin</i> wird beendet			<code>write(<i>Winkel</i>)</code>	stellt Servomotor auf <i>Winkel</i> (0-180°)
<code>analogWrite(<i>Wert</i>)</code>	erzeugt PWM-Welle mit einem Wert zwischen 0 und 255	<code>pulseIn(<i>Pin</i>, <i>Wert</i>)</code>	misst die Dauer die <i>Pin</i> *Wert annimmt (HIGH/LOW)			<code>read()</code>	liest Winkel von Servomotor
Interrupts		Zeit		Zufallszahlen			
<code>digitalPinToInterrupt(<i>Pin</i>)</code>	wandelt digitalen <i>Input-Pin</i> in Interrupt-Pin um	<code>unsigned int millis()</code>	Millisekunden seit Programmstart. ~50 Tage Overflow	<code>randomSeed(<i>Wert</i>)</code>	Ausgangswert für Zufalls-generator		
LOW, CHANGE, RISING, FALLING	Interrupt-Typen	<code>unsigned int micros()</code>	Microsekunden seit Programmstart. ~70 min bis Overflow	<code>random(<i>max</i>)</code>	Zufallszahl zwischen 0 und <i>max</i>		
<code>attachInterrupt(<i>Interrupt</i>, <i>Function</i>, <i>Typ</i>)</code>	verbindet <i>Interrupt-Pin</i> mit <i>Funktion</i>	<code>delay(<i>ms</i>)</code>	Pause für <i>ms</i> Millisekunden	<code>random(<i>min</i>, <i>max</i>)</code>	Zufallszahl zwischen <i>min</i> und <i>max</i>		
<code>detachInterrupt(<i>interrupt</i>)</code>	hebt Verbindung zwischen <i>Funktion</i> und <i>Interrupt-Pin</i> auf	<code>delayMicroseconds(<i>us</i>)</code>	Pause für <i>us</i> Microsekunden	Serial			
<code>interrupts()</code>	aktiviert Interrupts	Mathematik		<code>Serial.begin(9600);</code>	startet Serial -verbindung mit Baudrate		
<code>noInterrupts()</code>	deaktiviert Interrupts	<code>min(x, y)</code> , <code>max(x, y)</code> , <code>abs(x)</code>	Minimum, Maximum, absoluter Wert	<code>Serial.println(<i>inhalt</i>);</code>	sendet Zeile mit <i>Inhalt</i>		
		<code>pow(<i>Basis</i>, <i>Exponent</i>)</code>	$Basis^{Exponent}$	<code>Serial.print(<i>text</i>);</code>	sendet <i>Text</i>		
		<code>sqrt(x)</code>	Wurzel(x)	<code>Serial.write(<i>daten</i>);</code>	sendet <i>Daten</i> als Binärcode		
		<code>sin(x)</code> , <code>cos(y)</code> , <code>tan(z)</code>	Sinus, Cosinus, Tangens	<code>Serial.flush();</code>	wartet bis <i>Daten</i> gesendet sind		
				<code>Serial.end();</code>	beendet Serial-Verbindung		