

Kurzreferenz für das Arbeiten mit einem Arduino

Eine vollständige Referenz findet man unter
<https://www.arduino.cc/en/Reference/HomePage>

Grundrechenarten:

Symbol	Erläuterung
+	Addition
-	Subtraktion
*	Multiplikation
/	Division (9/2 ist 4: ganzzahlige Division)
%	Modulo (Rest bei der ganzzahligen Division)

Vergleichsoperatoren:

Symbol	Erläuterung
< <=	kleiner kleiner oder gleich
> >=	größer größer oder gleich
==	ist gleich
!=	Ist ungleich

Logische Operatoren:

Symbol	Erläuterung
&&	UND
!!	ODER
!	NICHT

Wichtige Datentypen:

Datentyp	Erläuterung
int	ganze Zahl, 16 Bit
byte	ganze Zahl, 8 Bit
long	ganze Zahl, 32 Bit
unsigned long	ganze Zahl, nur positiv, 32 Bit
float	Gleitkommazahl
String	Zeichenkette
char	Buchstabe
boolean	Wahrheitswert

Kommentare

Symbol	Erläuterung
//	einzeiliger Kommentar
/* */	Mehrzeiliger Kommentar

Variablendeklaration:

Syntax	Erläuterung
<code>int ganzeZahl;</code>	„ganzeZahl“ ist eine integer-Variable
<code>boolean istWahr;</code>	„istWahr“ ist ein Wahrheitswert
<code>int ledRot = 10;</code>	Deklaration der Variablen ledRot und sofortige Wertzuweisung von 10
<code>istWahr = true;</code>	Wertzuweisung von true zu der im Vorfeld deklarierten Variablen istWahr

Wichtige Funktionen:

Funktionsaufruf (Beispiel)	Erläuterung
<code>pinMode(13, OUTPUT);</code>	Pin 13 wird als Ausgang definiert
<code>pinMode(13, INPUT);</code>	Pin 13 wird als Eingang definiert
<code>digitalRead(13);</code>	Liest den digitalen Wert an Pin 13
<code>analogRead(A0);</code>	Liest den analogen Wert an Pin A0
<code>digitalWrite(13, LOW);</code>	Setzt Pin 13 auf LOW (0 Volt)
<code>digitalWrite(13, HIGH);</code>	Setzt Pin 13 auf HIGH (5 Volt)
<code>analogWrite(ledRot, 105);</code>	Setzt den PWM-PIN ledRot auf 105 (der Variablen ledRot muss im Vorfeld eine integer-Zahl zugewiesen sein)
<code>delay(1200);</code>	Warte 1200 Millisekunden
<code>delayMicroseconds(3000);</code>	Warte 3000 Mikrosekunden
<code>map(wert, quelle_start, quelle_ende, ziel_start, ziel_ende);</code>	Ein wert eines bestimmten Bereichs (hier quelle) wird in einen Wert eines anderen Bereichs (hier ziel) umgewandelt, so dass diese einander entsprechen
<code>pulseIn(13, HIGH)</code>	Setzt Pin 13 auf HIGH und misst die Zeit in Mikrosekunden, bis der Pin wieder auf LOW steht. Wird z. B. für Ultraschallsensoren benötigt. Hierbei ist dann der zugehörige Pin der Echopin, zuvor muss jedoch über den Trigger-Pin eine Ultraschallwelle losgeschickt worden sein.

Ausgabe über den seriellen Monitor:

Funktionsaufruf	Erläuterung
<code>Serial.begin(9600);</code>	In der Setup-Methode wird dem Programm mitgeteilt, dass eine serielle Ausgabe mit der Baudrate 9600 stattfindet
<code>Serial.print(„Text“)</code>	Ausgabe von „Text“ ohne Zeilenschaltung
<code>Serial.println(„Text“);</code>	Ausgabe von „Text“ mit Zeilenschaltung

Servos:

Syntax	Erläuterung
<code>#include <Servo.h></code>	Einbinden der zugehörigen Bibliothek
<code>Servo myServo</code>	Erstellen eines Objekts „myServo“ der Klasse „Servo“
<code>myServo.attach(pinNummer)</code>	Dem Programm wird mitgeteilt, an welchen Pin der Servo angeschlossen ist (findet im Setup statt)
<code>myServo.write(gradWinkel)</code>	Der Servo namens „myServo“ wird an die Position mit dem zugehörigen Winkel (gradWinkel) gebracht

Definition eigener Funktionen:

Syntax	Erläuterung
<pre>void meineFunktion1() { // Anweisungen }</pre>	<p>void bedeutet, dass die Funktion keine Rückgabewerte besitzt.</p> <p>Da nichts zwischen den Klammern nach dem Funktionsnamen steht, hat diese Funktion keine Parameter.</p>
<pre>int meineFunktion2(int a, int b) { return a + b }</pre>	<p>Diese Funktion hat einen int-Wert als Rückgabewert. Parameter in dieser Funktion sind die beiden integer-Zahlen a und b</p> <p>Im Rumpf der Funktion wird die Addition von a und b gebildet zurückgegeben.</p>

Wichtige Kontrollstrukturen:

Syntax	Erläuterung
<pre>if (Bedingung) { // Anweisungen }</pre>	<p>Einseitige Auswahl</p> <p>falls „Bedingung“ erfüllt, werden Anweisungen getätigt</p>
<pre>if (Bedingung1) { // Anweisungen1 } else if (Bedingung2) { // Anweisungen2 } else { //Anweisungen3 }</pre>	<p>Mehrseitige Auswahl</p>
<pre>for (Anfangswert; Bedingung; Schrittweite) { //Anweisungen }</pre>	<p>Wiederholungsanweisung mit for (allgemeine Syntax)</p>
<pre>for (int i=0; i<100; i=i+1) { //Anweisungen }</pre>	<p>In diesem Beispiel werden die Anweisungen genau 100 mal wiederholt</p>
<pre>while (Bedingung) { //Anweisungen }</pre>	<p>Wiederholung mit Anfangsbedingung</p>

Liste:

Syntax - Beispiel	Erläuterung
<code>int ganzeZahlen[5];</code>	Eine Liste von ganzen Zahlen der Länge 5;
<code>ganzeZahlen[0]</code>	Erstes Element der Liste
<code>ganzeZahlen[99]</code>	Letztes Element der Liste

