Mathematische Berechnungen

Microcontroller und Computer sind zwar ziemlich schnell im Rechnen, aber auch echt dumm und eingeschränkt. Du musst zum Beispiel vorher schon wissen, welche Sorte Zahl zu herausbekommen möchtest. Bei der Auswahl legst Du auch fest, wie viel Speicherplatz Du für diese Zahl im Arduino benutzen möchtest. Das macht die Sache echt kompliziert.
Zahlen können Namen bekommen. Namen müssen mit einem Buchstaben anfangen, üblich ist ein Kleinbuchstabe. Nach dem ersten Buchstaben kann der Name dann auch Zahlen enthalten. Umlaute und Sonderzeichen lässt Du bitte weg, das macht nur Probleme.

Befehl	Beschreibung	Beispiel
`byte name = Wert;`	Erzeugt eine Zahl, die von 0 bis 255 rechnen kann. Braucht 1 Byte Speicher im Arduino.	`byte klein = 45;` `Serial.println(klein);`

`char name = Wert;`
`char name = 'Buchstabe';`
	Erzeugt eine Zahl, die von -128 bis 127 rechnen kann. Wird auch für einzelne Buchstaben verwendet, wenn der Buchstabe in eine Zahl umgerechnet werden soll.
	Braucht 1 Byte Speicher im Arduino.
	Englisch: character = Zeichen
		`char negativ = -23;`
		`char eff = 'f';`
		`Serial.println(negativ);`
		`Serial.println(eff);`

`word name = Wert;`	Erzeugt eine Zahl, die von 0 bis 65535 rechnen kann. Braucht 2 Byte Speicher im Arduino. Englisch: word = Ausdruck / Vokabel / Wort	`word groesser = 54321;`
`int name = Wert;`	Erzeugt eine Zahl, die von -32768 bis 32767 rechnen kann. Braucht 2 Byte Speicher im Arduino. Englisch: integer = ganzzahlig	`int kleiner = -22222;`
`long name = Wert;`	Erzeugt eine Zahl, die von -2147483648 bis 2147483647 rechnen kann. Eine solche Zahl braucht ein L am Ende. Braucht 4 Byte Speicher im Arduino. Englisch: long = lang / weit	`long riesig = 2000000000L;` `long huhu17 = -2000000000L;`

Tipp: *int* ist eine verbreitete Zahlenart. Mach Dir erstmal keine Gedanken über den Speicherverbrauch. Bei kleinen Hobby-Projekten ist es wahrscheinlich, dass der Speicher des Arduino nicht voll wird. Der Arduino Uno hat 2048 Bytes Speicher. Das reicht für viele Zahlen.

Elektronik- und Roboter-Basteln
Spickzettel - Arduino

Befehl	Beschreibung	Beispiel
`+`	Addition, Zusammenzählen, Plus-Rechnen. Du kannst auch andere Zahlen bei ihrem Namen zum Rechnen benutzen. Es gilt die Regel Punkt vor Strich.	`byte a = 4 + 5;` `byte b = a + 7; // ergibt dann 16`
`-`	Subtraktion, Abziehen, Minus-Rechnen. Es gilt die Regel Punkt vor Strich.	`byte hossa = 60;` `word wow = 5000 - hossa;`

`*`	Multiplikation, Malnehmen, Vervielfachen.
	Achte darauf, dass Ergebnis auch noch in die Zahl reinpasst, sonst passieren seltsame Sachen. Computer und Microcontroller können wirklich dumm sein…
	Es gilt die Regel Punkt vor Strich.
		`byte zahl = 70;`
		`word quadrat = zahl * zahl; // = 4900`
		`byte falsch = zahl * zahl;  // = 36?!`

`/`	Division ohne Rest, Teilen ohne Rest.
	Der Rest geht einfach verloren.
	Es gilt die Regel Punkt vor Strich.
		`byte lollis = 27;`
		`byte kinder = 6;`
		`byte jeder = lollis / kinder; // = 4`

`%`	Der Rest vom Teilen.
	Wenn man 27 Lollis an 6 Kinder verteilt, bleiben 3 übrig.
	Es gilt die Regel Punkt vor Strich, wobei % wie / eine Punkt-Rechenart ist.
		`byte lollis = 27;`
		`byte kinder = 6;`
		`byte rest = lollis % kinder; // = 3`

`(` und `)`	Klammern, zum Ändern von Reihenfolgen (Aufhebung von Punkt vor Strich) oder zur besseren Lesbarkeit.	`byte eins = (-1) * (-1);` `byte zwei = (5 - 1) / 2;`

Befehl	Beschreibung	Beispiel

`float name = Wert;`	Erzeugt eine Kommazahl mit etwa 7 Stellen Genauigkeit.
	Das Komma wird als Punkt geschrieben (alles ist Englisch).
	Kommazahlen können sehr verrückt sein*). Verzichte auf Kommazahlen, wann immer Du kannst.
	Braucht 4 Byte Speicher im Arduino.
		`float wieBitte = 333333333.0 / 3.0;`
		`Serial.println(wieBitte); // 111111112`
		`7 genau`

*) Das Problem ist, dass eine Rechnung wie 1 geteilt durch 3 eine unendlich lange Kommazahl gibt, für die man unendlich viel Speicher bräuchte. Da das nicht geht, wurde eine Abwägung zwischen Genauigkeit und Speicherverbrauch gemacht. Das Ergebnis dieser Abwägung ist, dass das Ergebnis ungenau ist. In Einzelfällen sogar sehr ungenau. Irgendwann kann ich es Dir mal erklären – das Prinzip nennt sich IEEE754 und ist ein Buch mit 70 Seiten nur zum Thema Kommazahlen im Computer.