Mein erster Sketch

Sketch? Machen wir jetzt ein Spässle.

Nein, Sketch heißen die Programme im Arduino. Und da es immer so schön heißt „learning by doing“ fangen wir jetzt direkt an.

Also Arduino anschliessen, IDE starten und los geht's.

So sollte es jetzt bei euch aussehen:



Unten rechts in Ecke, seht ihr, welche Hardware ihr angeschlossen haben sollte. Bei mir ist es ein Duemilanove. Sollte bei euch das nicht stimmen, dann geht über das Menü Tools, Board und wählt das richtige Board aus.
Dahinter steht die Schnittstelle. Auch die könnt ihr wählen Menü Tools, Serieller Port.

Mehr braucht man eigentlich nicht zu tun. Jetzt können wir loslegen.

Ein Sketch besteht aus min 2 Programmteilen, einen zum Initialisieren und einen für das eigentliche Programm. Wir könnten jetzt natürlich das Blink Beispiel öffnen, aber ads wäre ja einfach…

Wir machen uns das gleich selbst und fangen jetzt an.

Zunächst einmal sollten wir uns klarmachen, was wir machen wollen und wir sollten das auch mal hinschreiben. Zum Beispiel:

```
Dieses Programm soll eine LED zum Blinken bringen
```

Schon ganz gut. Aber damit will der Arduino noch nicht so recht. Außerdem selbst wenn er das verstehen könnte, würde er fragen

> „Welche LED? Wie schnell?“

2. Versuch:

```
Dieses Programm soll eine LED an dem Ausgang 13 zum Blinken bringen,
Frequenz 1Hz.
```

Super. Aber verstehen kann der Arduino das leider noch immer nicht. Also müssen wir das anders machen. Vor dem Text machen wir ein /* und nach dem Text ein */ und schon haben wir den ersten Kommentar.

```
/*
Dieses Programm soll eine LED an dem Ausgang 13 zum Blinken bringen,
Frequenz 1Hz.
*/
```

Kommentare sind sehr wichtig. Dadurch verstehen wir, was wir uns dabei gedacht haben. Denn der Programmcode ist manchmal recht kryptisch, aber durch Kommentare wird er erst lesbar. Das was wir gemacht haben ist ein sog. Blockkommentar. Schön wäre übrigens auch noch, wenn man wüßte, von wem das Programm ist. Also keine scheu, schreibt euch da rein.

Neben Blockkommentaren gibt es auch noch Zeilenkommentare. Diese fangen mit // an und ab da bis zum Zeilenende wird alles als Kommentar gewertet.

Weiter geht's. Nachdem wir was eingegeben haben, sollten wir das ganze auch speichern, sonst ist das plötzlich mal weg. (z.B. weil Waldi an der Leitung kaute oder so...) Speichern geht über Datei/Speichern. Hier gibt man jetzt einfach den Namen der Projektes an. z.B. Blinken. Ab jetzt kann man schnell mal zwischendurch mit STRG+S speichern.

Wie ich oben bereits sagte, ein Sketch besteht aus 2 Teilen, der Initialisierung und dem eigentlichen Programm.

Die Initialisierung dient dazu, alles für das Programm vorzubereiten. Das nennt sich bei Arduino Setup. Also schreiben wir nun:

```
void setup() {
}
```

void bedeutet Funktion ohne Rückgabewert. Und wer sich so ein bisschen mit Programmiersprachen auskennt wird bemerken, das das ganze sehr nach C aussieht. Arduino basiert auf C hat aber ganz viele „makros“ und fertig definierte Funktionen. Für die anderen. Ist wurscht, wonach das aussieht. Jedenfalls muss da immer so aussehen. Übrigens: für unsere Tippfaulen, die gerne mal Gross/Kleinschreibung vergessen. Hier geht's nicht ohne. Setup ist was anderes als setup. Also immer schön aufpassen.

Die 2. Funktion für unser Programm ist das Programm selber, das nennt sich Loop

```
void loop() {
}
```

Sieht genauso aus wie setup macht aber was anderes. Wenn unser PRogramm hinterher läuft wird beim Starten einmal die Setup Routine durchlaufen. (Nach jedem Reset und Neustart von aussen.) Danach wird die Loop Routine aufgerufen. Wird diese Beendet, ruft der Arduino diese wieder erneut auf. (Loop → Schleife)

Hier mal unser ganzes Programm:

```
/*
```

```
Dieses Programm soll eine LED an dem Ausgang 13 zum Blinken bringen,
Frequenz 1Hz.
*/

void setup() {
}

void loop() {
}
```

So jetzt haben wir unser Grundgerüst. Jetzt geht's ans Programmieren.
Zunächst die Setup Routine. Am Anfang müssen wir dem Comtroller sagen, das die LED ein Ausgang sein soll. Befehl dazu: pinMode(Nummer, Mode) Also schreiben wir:

```
void setup() {
  // LED Pin wird zum Ausgang
  pinMode(13, OUTPUT);
}
```

Fertig und nun zum Blinken. Wir müssen die LED einschalten, warten, auschalten, warten und dann sind wir fertig. Wir erinnern uns, die Loop wird immer wieder neu aufgerufen...Aha, da haben wir also unsere Schleife, brauchen wir nciht selber machen. LED schalten geht mit digitalWrite(pin, Wert) Pin ist unser Anschluss und Wert ist 1 oder 0, oder auch HIGH oder LOW. Man kann beides verwenden. Warten mit delay(msec). Der Wert dabei sind die Millisekunden, die der Controller warten soll, bis er den nächsten Befehl ausführt. Also schreiben wir folgendes:

```
void loop() {
  // LED einschalten,
  digitalWrite(13,HIGH);
  // warten,
  delay(500);
  // auschalten,
  digitalWrite(13,LOW);
  // warten
  delay(500);
}
```

Ob ihr die Kommentare über den Befehl schreibt oder dahinter überlass ich euch. Aber schreibt welche hin. In 1 Jahr wißt ihr sonst nicht mehr, was ihr da gemacht habt. UNd falls mal jemand fremdes drüberschauen soll, wäre es auch recht hilfreich, wenn derjenige wüßte, was ihr eigentlich machen wolltet. Prima wir sind fertig. HALT STOP, irgendwie muss das doch jetzt in den Controller. Stimmt, wir müssen das ganze jetzt kompilieren und uploaden. (Ja so heist das...) Das macht man entweder mit STRG+U oder mit dem PFeil nach rechts , oder übers Menü mit Datei/Upload. Vorher speichern nicht vergessen. Unten in der IDE wird überigens der Verlauf dargestellt. Hier sieht man welche Fehler man gemacht hat, oder ob der Upload funktioniert hat. Ist alles Fehlerfrei gelaufen, sollte jetzt eine LED auf dem Board blinken. (That's Magic, ich hab euch 'ne LED hingezaubert, nein, die Entwickler haben das mit Absicht eingebaut, damit man testen kann, ob die Komunikation mit dem Gerät funktioniert. Ihr glaubt garnicht wieviel ihr jetzt schon geschafft habt, von der Einleitung habt ihr alle Punkte (1-9) erfolgreich überstanden.)

Das fertige Programm:

```
/*
Dieses Programm soll eine LED an dem Ausgang 13 zum Blinken bringen,
Frequenz 1Hz.
*/

void setup() {
  pinMode(13,OUTPUT);
}

void loop() {
  // LED einschalten,
  digitalWrite(13,HIGH);
  // warten,
  delay(500);
  // auschalten,
  digitalWrite(13,LOW);
  // warten
  delay(500);
}
```

Jetzt ist Zeit einen Kaffee zu trinken.

From:
https://wkla.no-ip.biz/ArduinoWiki/ - **Arduino im Modellbau**

Permanent link:
https://wkla.no-ip.biz/ArduinoWiki/doku.php?id=arduino:tutorial:mein-erster-sketch

Last update: **2018/11/04 10:51**