Inhaltsverzeichnis

1 Einleitung

Seit 2021 hat sich das Georg-Büchner-Gymnasium dazu entschieden, in der Roboter-AG den neuen *Excelsior* zu verwenden. Dabei handelt es sich um einen Brick (Baustein), ähnlich wie die NXT- und EV3-Bricks von Lego.

Dieser arbeitet mit einem Arduino-Board, hat ein Farbdisplay, 4 Motoranschlüsse, 8 Sensoranschlüsse und 3 Schalter. Der erste Schalter ist zum An- und Ausschalten des Bricks, der zweite zum Abschalten der Motoren, damit man das Display im laufenden Betrieb besser ablesen kann und einem dritten Schalter, dessen Verwendung man selber programmieren kann. Unten befindet sich Platz für 6 Akkus, die manuell geladen bzw. ausgetauscht werden können.

Durch diese bessere Performance löst er viele Probleme, die seine Vorgänger haben. Allerdings gibt es für den *Excelsior* keinen Block-Programmiersprache, d.h. man muss ihn in Textform über die Arduino-Software programmieren.

Zusammengenommen wird der Roboter mit Lego gebaut, außer dass ein schuleigener Brick und schuleigene Farbsensoren verwendet werden und die Programmierung über Arduino erfolgt.

2 Erste Schritte mit Arduino

2.1 Programmaufbau

Ein Arduino-Programm gliedert sich in drei Blöcke.

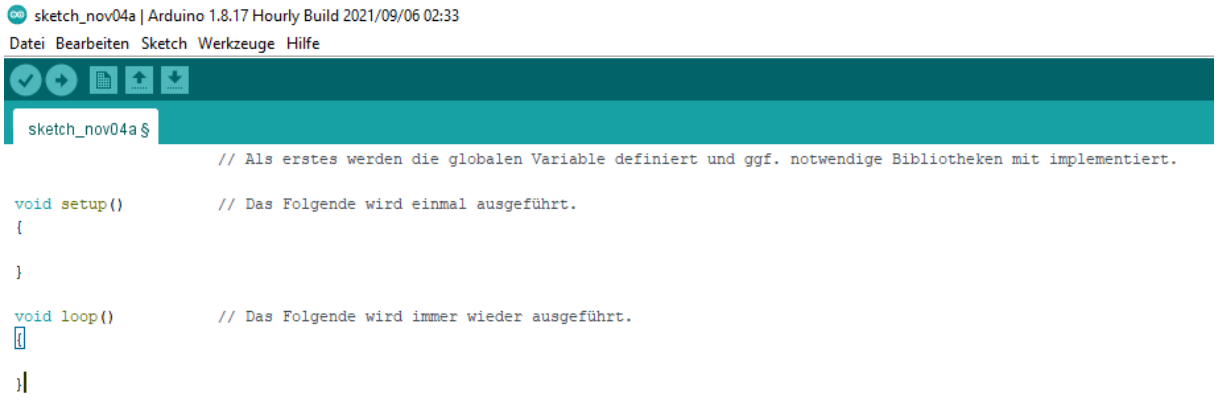

Abbildung 1: Der Aufbau

Am Anfang, noch vor *void setup()*, werden die globalen Variable definiert und ggf. notwendige Bibliotheken mit implementiert. D.h., alle Variablen, die im Laufe des Programms

benutzt werden, werden hier erzeugt und definiert. Man kann keine Variable verwenden, die man vorher nicht erzeugt hat. Des Weiteren werden hier die Bibliotheken genannt, auf die zurückgegriffen wird. Diese Bibliotheken hat jemand anders programmiert und wir verwenden sie. Ein Beispiel dafür ist die Bibliothek für das Display und die Sensoren, welche eine Menge Programmierarbeit erspart.

Im *void setup()* werden die Befehle geschrieben, die nur einmal durchgeführt werden. Z.B. sind das die Initialisierungen der einzelnen Ports oder das seriellen Monitor. Auch eine Startsequenz kann hier stehen.

Das Hauptprogramm befindet sich im *void loop()*. Das ist eine Endlosschleife, die wieder von vorn anfängt, wenn das Ende erreicht ist. Hier befinden sich die Sensorabfragen und Entscheidungen, was in welchem Falle getan werden soll.

2.2 Der Aufbau des Boards

Als einen Ausgang bezeichnet man Informationen, die aus dem Board herausgehen. Z.B. können damit Motoren, Lampen oder das Display gesteuert werden. Bei einem Eingang gehen Informationen in das Board hinein. Das sind jede Art von Sensoren, z.B. Farb- Tast- und Neigungssensoren,

Auf dem Board gibt es digitale und analoge Ports. Die digitalen Ports können als Ein- und als Ausgänge verwendet werden und haben nur die Zustände *HIGH* und *LOW* bzw. 1 und 0. Diese werden mit Nummern von 0 bis 13 (je nach Board kann die letzte Zahl variieren) gekennzeichnet. Im Gegensatz dazu gibt es analoge Ports, die Werte von 0 bis 1023 (2^{10}) ausgeben. Allerdings lassen sich diese nur als Eingang verwenden und nicht als Ausgang. Diese werden mit einem A vor der Nummer gekennzeichnet.

2.3 Ein Eingang (Sensorport)

Wenn ein Sensor verwendet werden soll, muss man im *void setup()* dem Programm sagen, dass der entsprechende Port als Eingang verwendet werden soll. Der Befehl dazu ist *pinMode(Portnummer, INPUT)*. Wenn man z.B. an den Port 13 einen Tastsensor anschließen möchte, dann schreibt man *pinMode(13, INPUT)*. Jetzt steht der Sensor zur Verfügung und kann verwendet werden.

Um weiter auf in zuzugreifen, kann man jetzt mit *digitalRead(Portnummer)*, im obigen Beispiel *digitalRead(13)*, den Wert den Port 13 heraus gibt, im weiteren Verlauf verwenden.

2.4 Ein Ausgang

Wenn ein Motor, eine Lampe o.ä. verwendet werden soll, muss man - ähnlich wie beim Eingang, im *void setup()* dem Programm sagen, dass der entsprechende Port als Ausgang verwendet werden soll. Der Befehl dazu ist *pinMode(Portnummer, OUTPUT)*. Wenn man z.B. an den Port 12 eine Leuchtdiode anschließen möchte, dann schreibt man *pinMode(12, OUTPUT)*.

Jetzt kann die Diode im weiteren Verlauf des Programms mit *digitalWrite(Portnummer, Wert)*, im obigen Beispiel *digitalWrite(12, HIGH)*, angeschaltet und mit *digitalWrite(12, LOW)* ausgeschaltet werden.

2.5 Der serielle Monitor

Um sich Informationen anzeigen zu lassen, kann man das Display des *Excelsiors* verwenden oder den PC-Monitor (letzterer ist deutlich größer). Ähnlich wie bei den Ein- und Ausgängen muss auch hier am Anfang unter *void setup()* der Monitor mit *Serial.begin(9600);* angeschaltet werden. Da alle Informationen über das USB-Kabel laufen und man gleichzeitig viele verschiedenen Geräte verwendet, hat jedes Gerät eine eigene Kennung (wie eine Telefonnummer). Die jetzige ist 9600.

Mit *Serial.print();* kann man in den Klammern Zahlen, z.B. Sensorwerte, oder Wörter (die in Anführungszeichen) schreiben. Möchte man am Ende einen Zeilenumbruch haben, schreibt man *Serial.println();* (*print line* anstatt *print*).

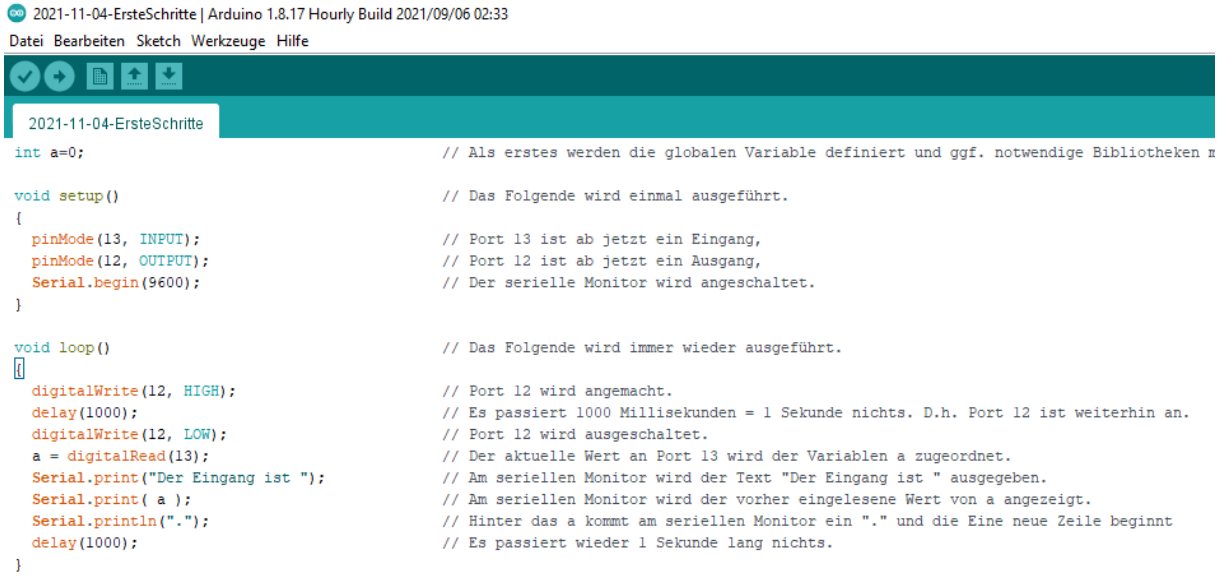

Abbildung 2: Ein einfaches Beispiel