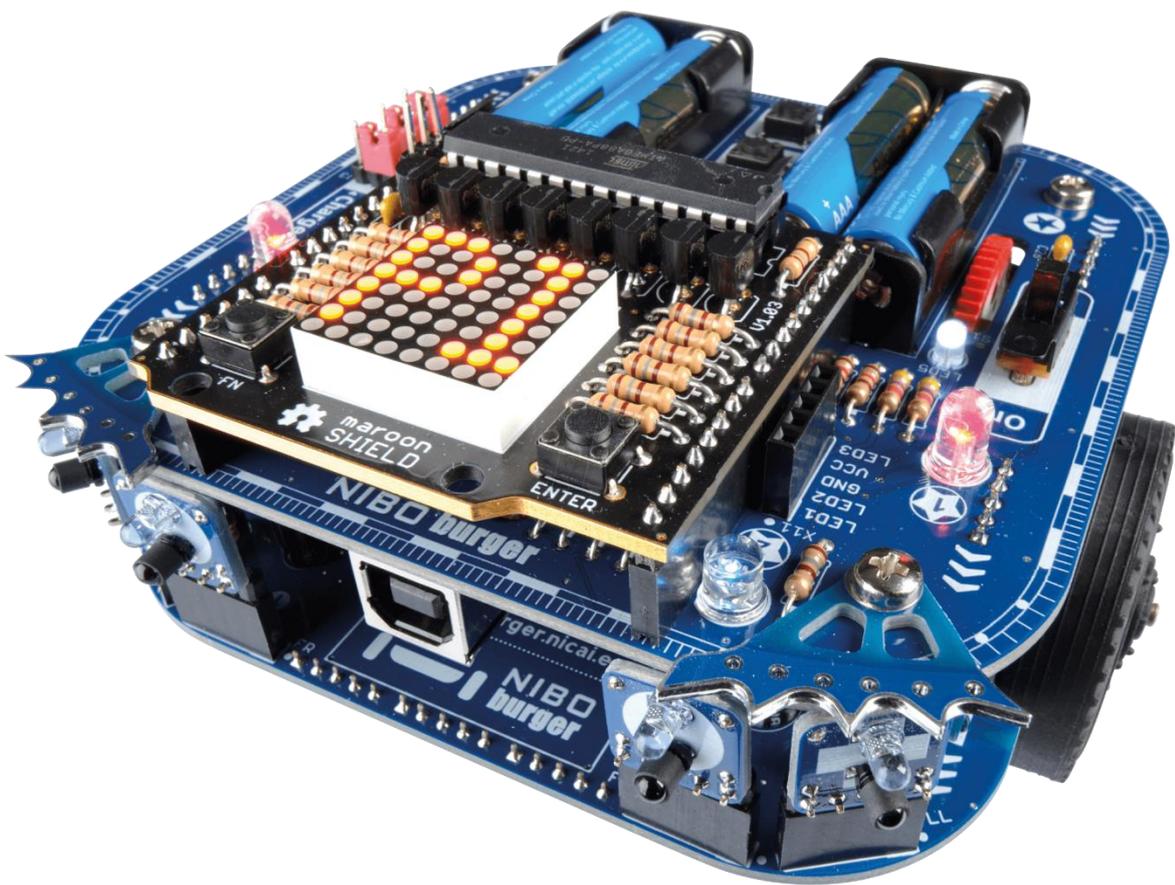


Nibo-Burger Dokumentation



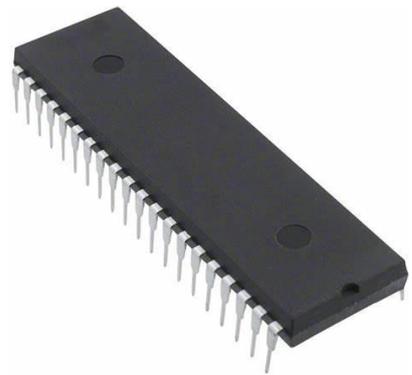
Inhalt

Aufbau:	3
Was ist ein Mikrocontroller?	3
Motorbrücke.....	3
PWM Signal	3
IR Sensor	4
RGB-Sensors	4
Flash und EEPROM Speicherplatz.....	4
Programmieren:	5
Variablen	5
If / Else Schleife	5
For-Loop	6
Switch	6
Schlusswort	7

Aufbau:

Was ist ein Mikrocontroller?

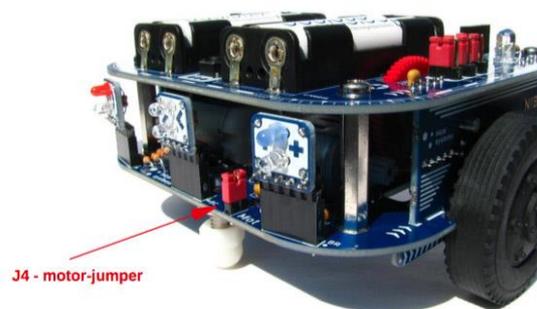
Ein Mikrocontroller ist ein kleiner Computer auf einem einzelnen Halbleiter-Chip. Dazu gehört ein Prozessor, der Programme ausführen kann, Arbeits- und Programmspeicher sowie Schnittstellen, die eine Kommunikation mit der Umgebung ermöglichen.



Motorbrücke

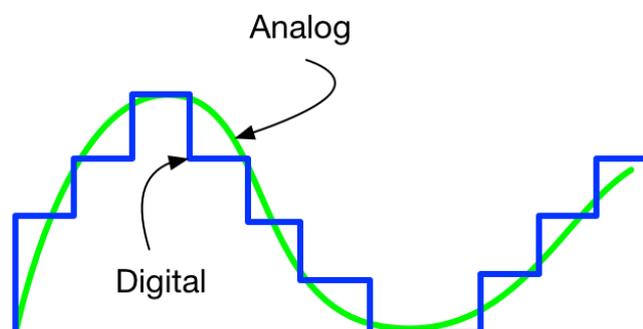
Die Motorbrücke wird für die Stromverstärkung der Mikrocontrollersignale gebraucht. Der Motor bekommt vom Vierquadrantensteller drei mögliche Signalkombinationen vorwärts, rückwärts oder kurzgeschlossen.

Der kurzgeschlossene Betrieb wird zur besseren Energieausnutzung bei der PWM-Ansteuerung gebraucht, da der Strom so nicht gegen die Versorgungsspannung fließen muss.



PWM Signal

Das PWM Signal wird gebraucht um ein Analoges Signal aus einem Digitalen Signal zu machen. Es ermöglicht es, mit der Hilfe der Odometriesensoren, konstant gerade aus zu fahren.



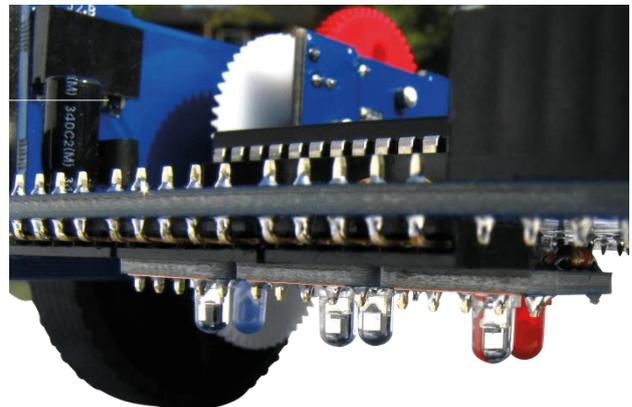
IR Sensor

Der Infrarot Sensor empfängt jene Strahlung deren Wellenlänge sich unterhalb der Wellenlänge des für das menschliche Auge sichtbaren Rotlichts befindet. Dies wird hier gebraucht durch ein Zahnrad mit Löchern (Lichtschranke) und aussen am Roboter zur Erkennung von Gegenständen. Wenn sich das Rad nun bewegt empfängt der Infrarot Sensor das Infrarot Licht durch die einzelnen Löcher. So kann gemessen werden wie viele Umdrehungen auf einer Seite gemacht werden.



RGB-Sensors

Der Namen des RGB-Sensors kommt von den Farben Rot, Grün, Blau. Der Nibo Burger hat diese Drei Sensoren unten angebracht. So kann er erkennen wenn er über eine der Farben fährt.



Flash und EEPROM Speicherplatz

Der Hauptunterschied zwischen EEPROM und Flash ist die Art der Logikgatter, die sie verwenden. Das EEPROM verwendet das schnellere NOR und Flash verwendet den langsameren NAND. Der NOR-Typ ist viel schneller als der NAND-Typ. Ausserdem ist der NOR Typ wesentlich teurer als der NAND-Typ.

Ein weiterer Vorteil von EEPROM gegenüber Flash besteht darin, wie sie auf die gespeicherten Daten zugreifen und diese löschen können. EEPROM kann auf die Daten bytewise zugreifen. Im Vergleich kann Flash dies nur blockweise tun. Um das Ganze zu vereinfachen, werden einzelne Bytes in eine kleinere Anzahl von Blöcken gruppiert.



Programmieren:

Wir haben den Nibo-Burger in der Programmiersprache C programmiert. Das Programm war anfänglich sehr ungewohnt, da ich bisher nur Arduino programmiert habe. Mit der Zeit habe ich mich doch daran gewöhnt. Was ich in dieser programmier Sprache besser fand ist das befehle vorgeschlagen wurden. Im Arduino muss man immer jeden Buchstaben eingeben.

Variablen

Variablen können beliebige werte annehmen. Um dem Programm dies mitzuteilen, kann man zb. int LED schreiben. Das Programm weiss nun mit welcher Datentyp er benutzen muss. Hier wäre es Int. Es gibt noch viele weitere Arten von Variablen wie zb. long, Float, array.

If / Else Schleife

Eine If / Else Schleife fragt ab ob ein gewollter Zustand zutrifft. Wenn dies nicht der Fall ist wird das ausgeführt was in der Else Schleife ist.

```
if (Mitte < 20)
{
motpid_setSpeed(30, 30);
}
else
{
}
```

For-Loop

Die For-Schleife kann gebraucht werden für eine Funktion, die eine gewisse Anzahl von Abläufen hat. Dies ist sehr hilfreich, da man zb. so sagen kann das ein Rad 10 Umdrehungen machen soll.

Das Programm für die For-Schleife sieht so aus:

```
for (LED=1; LED<=4;LED++)
{

}
```

Hier wird die Variable LED zuerst auf 1 geschalten danach wird immer 1 dazu gezählt bis 4.

Switch

Dies kann gebraucht werden um verschiedene Fälle abzuspielen.

Switch (state)

```
{
  case 1:
    motpid_stop(0);
    if (key == 'A')
    {
      state = 2;
    }
    break;
  case 2:
    if (Mitte < 20)
    {
      motpid_setSpeed(30, 30);
    }
    else
    {
      state = 3;
    }
    break;
```

Hier wird nach reihe abgefragt was zutrifft. Case 1 trifft zu wenn der Knopf A gedrückt wird. Case 2 trifft zu wenn die Variable „Mitte“ kleiner ist als 20.

Schlusswort

Das Zusammen bauen des Roboters ist mir gut gelungen. Die meisten Lötstellen sind sauber und haben einen guten Kontakt. Bis auf eine Stelle die für die Odometrie Sensoren gebraucht wurde. Beim erstmaligen einschalten funktionierte alles ausser die Odometrie Sensoren. Das Problem zu finden war jedoch recht schwer. Durch Messungen mit dem Multimeter und dem Schema konnte ich dem Fehler jedoch nach gehen. Dabei stellte sich heraus, dass meine Batterien nicht genügend geladen waren. Nach Behebung des Problems funktionierte nun eine der beiden Seiten. Also ging ich erneut auf Fehlersuche und fand eine unsaubere Löt stelle die keinen Kontakt hatte. Sobald die Stelle sauber gelötet wurde funktionierten nun beide Odometrie Sensoren.

Das Programmieren war sehr verständlich, jedoch gab es immer wieder Probleme mit dem hochladen. Die Programmiersprache C gefiel mir recht gut, da das Programm Vorschläge für Befehle machte (Nicht wie bei Arduino). Die Aufgaben waren zu dem gut gegliedert, dass man nach und nach darauf aufbauen kann.