

NWT KA 1 Programmierung

Grundwissen:

- Es gibt mehrere Programmiersprachen -> Für Arduino wird C++ verwendet
- Immer ; nach Ende eines Commands
- Arduino ist ein Microcontroller

Programmierung:

Funktionen:

- Funktionen werden normalerweise zur Ausführung von bestimmten Zwecken „gebaut“
- Funktionen starten immer mit einem **void**
- void setup und void loop sind auch Funktionen

```
void funktion() {  
  }  
}
```

BSP.:

```
int i;

void setup() {

}

void loop() {
  addiereEins();
}

void addiereEins() {
  i++;
}
```

Bei dieser Funktion steigt der Wert von i bei jedem Durchlauf von loop um eins.

Datentypen:

-Es gibt mehrere Datentypen:

- Integer: Im Code mit `int` abgerufen, kann positive und Negative Zahlen speichern. !KEINE KOMAZAHLEN!
- Float: Im Code mit `float` abgerufen, kann positive und negative Komazahlen speichern
- Boolean: Im Code mit `bool` abgerufen, kann wahr(true) oder falsch(false) sein.

-Es gibt noch andere Spezifischere Datentypen wie long, double, uint ... diese sind allerdings nicht relevant!

Header:

-Header sind Dateien welche weitere Funktionen und Datentypen dem Code hinzufügen(grob, natürlich nicht wirklich richtig aber ausreichend für KA)

-Wir benutzen die Header Servo.h und IRremote.h

```
#include <Servo.h>
#include <IRremote.h>

void setup() {
    // put your setup code here, to run once:
}
```

-Um ein Header einzubinden benötigt man #include

Arduino spezifische Commands:

-Servo:

- Servo [name]; -> Definiert Servo
- [name].attach([pin]); -> Sagt Arduino welcher pin für Servo ist
- [name].write([gradzahl]); -> Setzt den Servo auf eine bestimmte gradzahl (0-180)

-Infrarot:

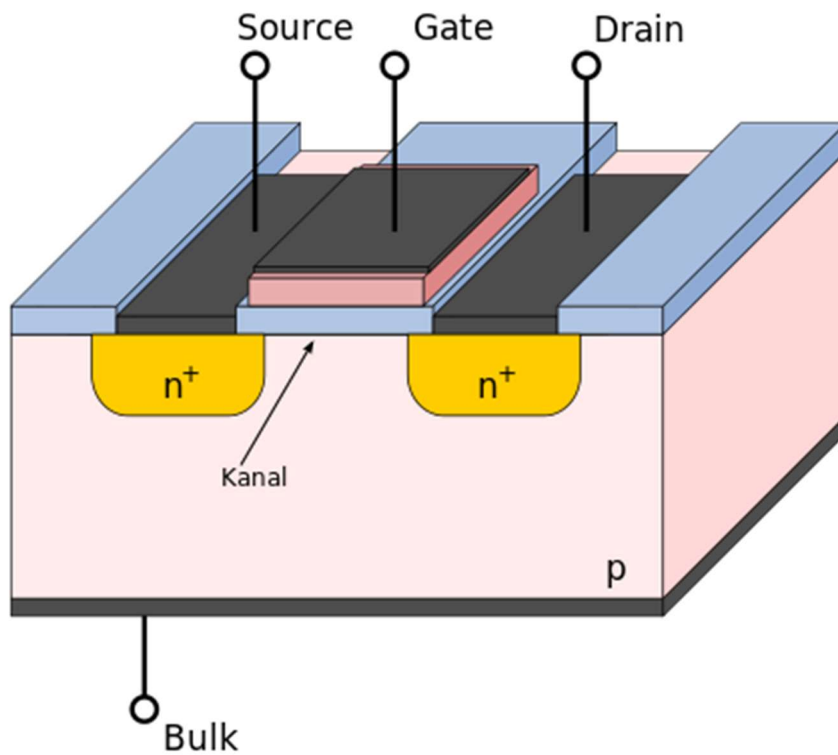
- IRrecv [nameRec]([inputPin]); -> Definiert Infrarotpin
- decode_results [nameResults]; -> Definiert Ergebnisvariable für Infrarot
- if([nameRec].decode(&[nameResults])){} -> Fragt auf Input von Infrarot ab
- [nameRec].resume(); -> Setzt suche nach Signalen fort

-Generell:

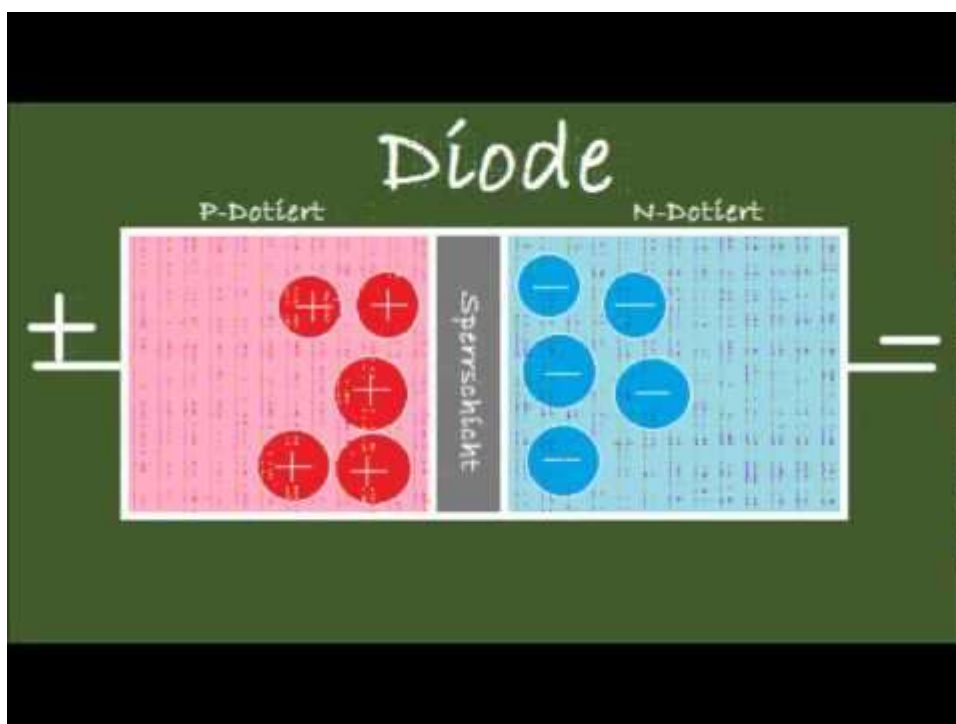
- pinMode([pin], INPUT/OUTPUT); -> Sagt Arduino ob Pin Signale sendet oder empfängt
- digitalWrite([pin], HIGH/LOW); -> Lässt einen Pin Signal HIGH oder LOW Senden
- analogWrite([pin], [value]); -> Lässt einen Pin ein Signal zwischen 0 und 255 senden
- Serial.begin(9600); -> Startet Seriellen Monitor
- Serial.println([Kontext]); -> Lässt etwas auf Seriellem Monitor anzeigen

Hardware:

Transistor:

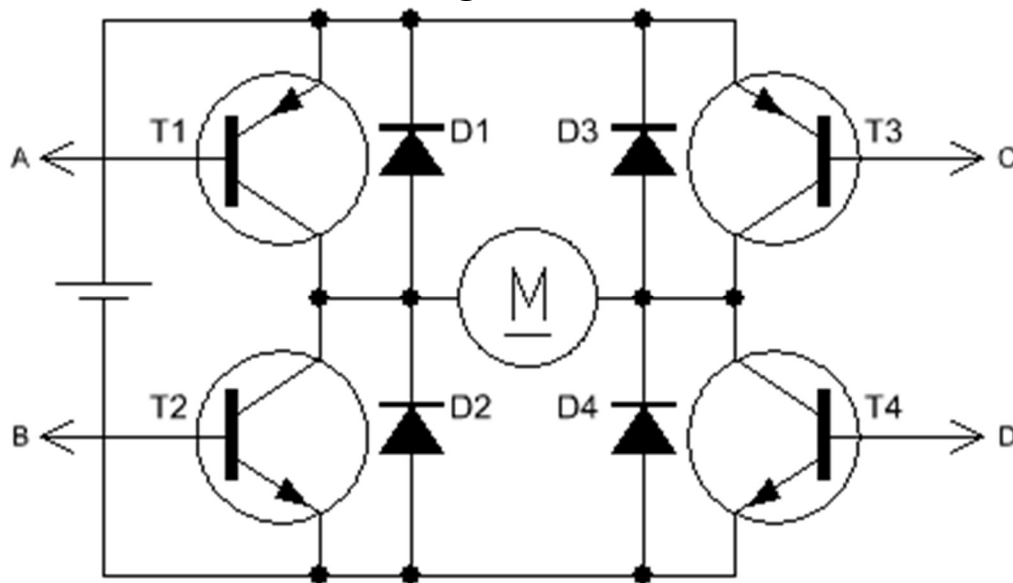


Diode:



Vielleicht Transfer?:

-Wie ist eine H-Brücke aufgebaut?



Für so eine Steuerung werden jeweils 2 PNP und 2 NPN-Transistoren benötigt. Für induktive Lasten, wie Motoren oder Magnete müssen die Transistoren noch mit Freilaufdioden geschützt werden.

Um hier den Motor anzusteuern, muss T1 und T4 oder T3 und T2 angesteuert werden. Dies geschieht über die Anschlüsse A und D oder C und B. Aber ein Problem gibt es hierbei dennoch. Man muss auf jeden Fall vermeiden, dass T1 und T2 oder T3 und T4 gleichzeitig aktiv sind. Dies erzeugt einen Kurzschluss und führt mit sehr hoher Wahrscheinlichkeit zur Zerstörung der Transistoren.

So eine Schaltung nennt sich, wie bereits schon in der Überschrift zu erkennen ist, H-Brücke. Es werden auch hin und wieder die Begriffe Vollbrücke oder seltener auch die Bezeichnung Vierquadrantensteller verwendet.

Letzter Teil aus:

<http://dieelektronikerseite.de/Lectons/H-Bruecke%20-%20Die%20Andersherum-Schaltung.htm> kopiert