# Ein Bild, das Text, Screenshot, Rechteck, Diagramm enthält. Automatisch generierte BeschreibungPulsweitenmodulation (PWM)

Programmieren lernen mit dem nanoBoard – AB 18

## Pulsweitenmodulation (PWM) Informationsblatt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| u |  |  |  |  |
|  | taus | tein | t |  |
|  |  | **T** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PWM-Signale werden in der Praxis häufig genutzt um: | | | | |
| - | die Helligkeit von Glühlampen, | | | |
| - | die Drehzahl von Gleichstrommotoren, | | | |
| - | die Wärmeleistung von Heizungen zu steuern und | | | |
| - | um Informationen zu übertragen (z. B. Temperatursensor, Servomotoren). | | | |
|  |  | | | |
|  | T: |  | Periodendauer |  |
|  | f: |  | Frequenz |
|  | taus: |  | Ausschaltzeit |
|  | tein: |  | Einschaltzeit |
|  | g: |  | Tastgrad |

Der Arduino nano kann sechs unabhängige PWM-Signale generieren. Die PWM-Ausgänge sind im Pinout durch eine Art Wechselspannungssymbol gekennzeichnet. Die PWM-Ausgänge sind die Arduino-Pins 3, 5, 6, 9, 10 und 11.

Ein Bild, das Text, Diagramm, Screenshot, Display enthält.

Automatisch generierte BeschreibungDie Arduino-Pins 10 und 11 sind die Anschlüsse PB2, und PB3.

In nebenstehendem Schaltplanauszug ist erkennbar, dass an diesen Pins die Zweifarb‑LED unseres nano-Boards angeschlossen ist.

Die Befehlszeile für die Nutzung der PWM ist:

**analogWrite(Pinnummer, Wert);**

Die Pinnummer ist die Arduino-Pinnummer.  
Der Wert ist ein integer-Wert von 0 bis 255.

## Ein Bild, das Text, Screenshot, Rechteck, Diagramm enthält. Automatisch generierte BeschreibungPulsweitenmodulation (PWM) Aufgaben

Programmieren lernen mit dem nanoBoard – AB 18

**Aufgabe PWM\_1:**

Das von Ihnen erstellte Programm soll eine Farbe der Zweifarb-LED so steuern,  
dass in einer Endlosschleife die folgenden Phasen ausgeführt werden:

**Phase1:** (Aufdimmen) Lampenhelligkeit von Null zunehmend bis zu maximaler Helligkeit (Dauer ca. 5s)

**Phase2:** (Abdimmen) Lampenhelligkeit vom Maximum abnehmend bis Null (Dauer ca. 5s)

**Phase1:** usw.

Entwickeln Sie den Quellcode mit Kommentaren und testen Sie die Funktion.

[Erstellen Sie PAPs oder Struktogramme für das Programm.]

**Aufgabe PWM\_2:**

Der Programmablauf soll über Taster in der folgenden Weise gesteuert werden:

Taster S3: vollständig Aufdimmen (Phase 1)

Taster S0: vollständig Abdimmen (Phase 2)

Taster S1: Aufdimmen so lange Taster gedrückt

Taster S2: Abdimmen so lange Taster gedrückt

Taster S2 und S3: Vollständiges Auf- und Abdimmen ab der aktuellen Helligkeit

Entwickeln Sie den Quellcode mit Kommentaren und testen Sie die Funktion.

[Erstellen Sie PAPs für das Hauptprogramm und die Prozeduren.]

**!!! PWM mit AD-Wandlung !!!**

**Aufgabe PWM\_3:**

Beim Drücken von Taster S0 kann der Rotwert der Zweifarb-LED durch das Drehen am Potentiometer RP6 eingestellt werden. Beim Drücken von Taster S1 kann der Grünwert der Zweifarb-LED durch das Drehen am Potentiometer RP7 eingestellt werden. Wenn kein Taster gedrückt wird, dann kann keine Farbe verändert werden.

**Aufgabe PWM\_4:**

An allen sechs PWM-Pins sind LEDs angeschlossen. Mit den Tastern S1 und S2 soll die LED auswählbar sein. Mit dem Potentiometer RP7 soll die Helligkeit der ausgewählten LED eingestellt werden können, die dann dauerhaft erhalten bleibt. Diese Werte können auch auf dem OLED-Display ausgegeben werden.