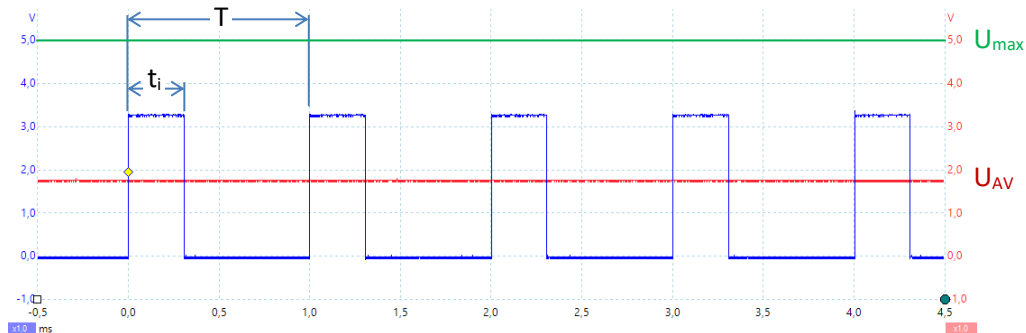


## Pulsweitenmodulation Prinzip

Die Pulsweitenmodulation wird oft eingesetzt, wenn ein Verbraucher in der Leistung stetig gesteuert werden soll. Die PWM ist eine Rechteckspannung mit konstanter Periodendauer  $T$ . Durch variieren der Impulsdauer  $t_i$  wird der Mittelwert der Spannung am Verbraucher eingestellt.



es gilt:

$$U_{AV} = U_{max} \cdot \frac{t_i}{T}$$

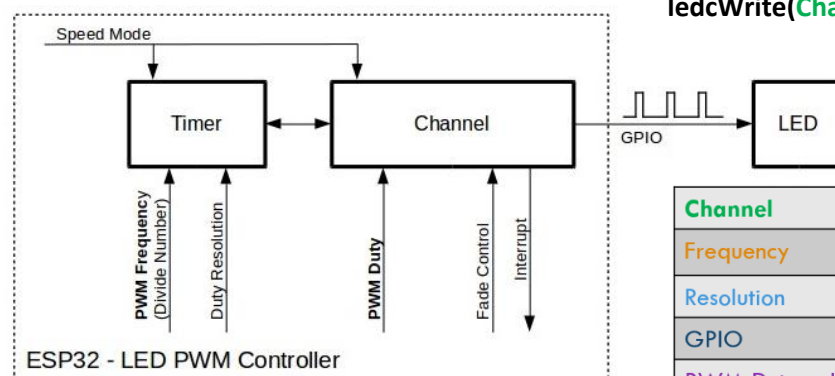
$$U_{AV} = U_{max} \cdot g$$

$g$ : Tastgrad

## PWM Hardware-Modul und Funktionen für ESP32

Der ESP32 besitzt 16 PWM Kanäle, die häufig für die Helligkeitssteuerung von LED eingesetzt werden.

```
ledcSetup(Channel, Frequency, Resolution);
ledcAttachPin(GPIO, Channel);
ledcWrite(Channel, PWM-Dutycycle);
```



Channel	0 ... 15
Frequency	1 kHz, 5 kHz, 8 kHz, 10 kHz
Resolution	1 ... 16 Bit
GPIO	beliebig
PWM-Dutycycle	Abhängig vom Dutycycle

## Arbeitsauftrag

- Erstellen Sie das nebenstehende Programm und überprüfen Sie die Änderung der Helligkeit der LED an GPIO 2.
- Stellen Sie den Jumper "Backlight" auf **pwm/2**. Nun können Sie auch die Helligkeit des LCD dimmen.
- Messen Sie die PWM-Spannung mit dem Picoscope an GPIO 2 und die Analogspannung an GPIO 34 (beide auf der 3,3V Seite). Beobachten Sie die Spannung bei Änderung des Tastgrades.
- Dokumentieren Sie ihre Ergebnisse.

```
const int pwmPin = 2;

#define LED 1

void setup()
{
    // (Kanal, Frequenz, Auflösung)
    ledcSetup(LED, 1000, 12);
    // Pin mit Kanal verbinden
    ledcAttachPin(pwmPin, LED);
}

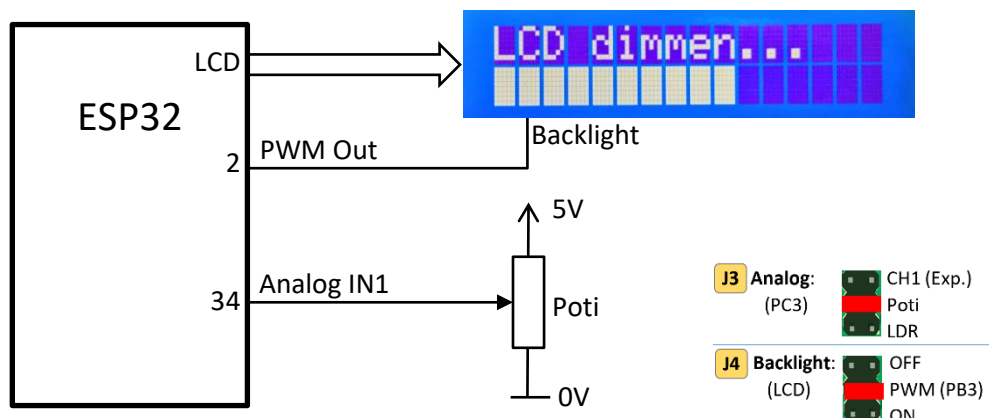
void loop()
{
    // ADC hat 12-Bit Auflösung
    int hell = analogRead(34);
    ledcWrite(LED, hell);

    delay(100);
}
```

## Projekt: Balkenanzeige für Hintergrundbeleuchtung des LC-Displays

Die Helligkeit der Hintergrundbeleuchtung (LED-Backlight) des LC-Displays kann effektiv über eine vom ESP32 bereitgestellte PWM-Einheit realisiert werden. Zur Visualisierung der Displayhelligkeit soll eine Balkenanzeige realisiert werden. Beim Schreiben des Balkens wird das Blocksymbol (█ = 0xFF) von Links an jeder Cursor-Position angezeigt, deren Wert < der aktuellen Helligkeit ist. Alle Cursor-Positionen die > dem aktuellen Helligkeitswert sind, werden mit Leerzeichen (░ = 0x20) befüllt. Die Anzeige der Balkenelemente erfolgt jeweils mit:

```
lcd.write(0xFF); // Block-Symbol █
lcd.write(0x20); // Leerzeichen (Blank) ░
```



## Arbeitsauftrag

- Erstellen Sie das Programm zur Balkendarstellung auf dem LC-Display nach dem nebenstehenden Algorithmus.

Die map()-Funktion wird benötigt, um die 12Bit-Auflösung (0...4095) des AD-Wandlers auf die 16 Spalten der Anzeige runterzumappen. Recherchieren Sie die Benutzung der Funktion in der Arduino-Referenz.

Testen Sie das Programm durch Drehen am Potenziometer.

- Anstelle des Potis kann auch der LDR als Analogeingabe dienen. Für eine adaptive Helligkeitseinstellung, muss die Hintergrundbeleuchtung mit zunehmender Umgebungshelligkeit steigen. Ist es dunkel, reicht eine geringe Displaybeleuchtung. Stecken Sie dazu nur den Jumper J3 (Analog) auf LDR um.

Testen Sie die Steuerung durch Abdunkeln des LDR.

