

```
def starte_timer():  
    zeit_timer.init(period=1000, mode=Timer.PERIODIC, callback=zeige_datum_und_uhrzeit)
```

Die Funktion `starte_timer()` initialisiert einen Timer, der eine bestimmte Aufgabe (in diesem Fall das Anzeigen von Datum und Uhrzeit) in regelmäßigen Intervallen ausführt. Lass uns die einzelnen Bestandteile der Funktion detailliert erklären:

1. Timer

In MicroPython wird ein **Timer** verwendet, um wiederkehrende oder verzögerte Aktionen auszuführen. Ein Timer kann mit einer bestimmten Zeitspanne (dem **Periodenwert**) konfiguriert werden, und er ruft dann eine sogenannte **Callback-Funktion** auf, wenn der Timer abläuft.

In dieser Funktion wird der Timer mit `zeit_timer.init(...)` konfiguriert.

2. Parameter der `init()`-Methode

Die Methode `init()` wird verwendet, um den Timer zu starten oder zu konfigurieren. Hier sind die verwendeten Parameter und deren Bedeutung:

a. `period=1000`

- Der Timer wird mit einer Periode von **1000 Millisekunden (1 Sekunde)** initialisiert.
- Das bedeutet, dass die Aktion (Callback) **alle 1000 Millisekunden** ausgeführt wird.

b. `mode=Timer.PERIODIC`

- Der **Modus** des Timers ist auf `PERIODIC` eingestellt.
- Das bedeutet, dass der Timer wiederholt wird. Nach Ablauf eines Intervalls (1 Sekunde) startet er automatisch erneut.
- Alternative Modi wären z. B. `Timer.ONE_SHOT`, bei dem der Timer nur einmal abläuft und dann gestoppt wird.

c. `callback=zeige_datum_und_uhrzeit`

- Die **Callback-Funktion** ist eine Funktion, die bei jedem Ablauf des Timers ausgeführt wird.
- Hier ist die Callback-Funktion `zeige_datum_und_uhrzeit`. Diese Funktion wird jedes Mal aufgerufen, wenn der Timer abläuft (alle 1000 Millisekunden).
- Die Callback-Funktion muss keine Argumente annehmen, aber der Timer übergibt automatisch eine Referenz auf sich selbst, wenn erforderlich.

3. Zweck der Funktion

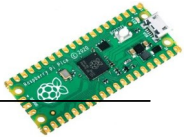
Die Funktion `starte_timer()` setzt den Timer so auf, dass er:

- Jede Sekunde (alle 1000 Millisekunden) den aktuellen Zeitpunkt verarbeitet.
- Die Funktion `zeige_datum_und_uhrzeit` ausführt, um z. B. das Datum und die Uhrzeit auf einem LCD anzuzeigen.

4. Beispielablauf

Hier ist ein möglicher Ablauf, wenn `starte_timer()` aufgerufen wird:

1. Der Timer wird initialisiert.
2. Nach 1000 Millisekunden ruft der Timer die Funktion `zeige_datum_und_uhrzeit` auf.
3. Der Timer wartet erneut 1000 Millisekunden und ruft die Funktion wieder auf.
4. Dies wiederholt sich, solange der Timer aktiv ist.



Zusammenhang mit `zeige_datum_und_uhrzeit`

Die Funktion `zeige_datum_und_uhrzeit` wird vom Timer aufgerufen. Sie könnte wie folgt aussehen:

```
def zeige_datum_und_uhrzeit(timer):  
    jetzt = utime.localtime()  
    datum = f"{jetzt[2]:02d}.{jetzt[1]:02d}.{jetzt[0]}"  
    uhrzeit = f"{jetzt[3]:02d}:{jetzt[4]:02d}:{jetzt[5]:02d}"  
    lcd_schreiben(f"Datum: {datum}", f"Zeit: {uhrzeit}")  
    print(f"Datum: {datum}, Zeit: {uhrzeit}")
```

- Diese Funktion zeigt das Datum und die Uhrzeit auf einem LCD-Display an.
- Sie nutzt `utime.localtime()`, um die aktuelle Zeit abzurufen, formatiert Datum und Uhrzeit als Strings und schreibt sie auf das LCD.

Zusammenfassung

Die Funktion `starte_timer()` startet einen Timer, der alle 1 Sekunde die Funktion `zeige_datum_und_uhrzeit` aufruft, um das Datum und die Uhrzeit regelmäßig zu aktualisieren. So wird sichergestellt, dass die Anzeige immer exakt und aktuell bleibt.