# Entwicklerdokumentation

## 1. Hardware:

Breadboard, Raspberry Pi Pico (W)

7 x LED mit Vorwiderstand 270 OHM

Diverse Vebindungsleitungen

## 2. Software:

Sprache: C

Unterteilt in:

* Dice.c // Hauptprogramm mit Main()
* Hardware.c/h // GPIO init, Ruhezustand

### 2.1 Dice

Im Dice befindet sich das Hauptprogramm hier werden folgende Funktionen realisiert:

Abhängigkeiten: Hardware.h, pico\_stdlib.h, stdudio.h

* Würfel Logik
* Hauptschleife main()
* Zusammenführung aller Features

### 2.2 Hardware.c / Hardware.h

Abhängigkeiten: pico/sleep.h, dice\_harware.h, pico/stdlib.h, hardware/timer.h, hardwar/time.h

#### 2.2.1 dice\_hardware\_init:

Diese Funktion wird vom Hauptprogramm aufgerufen, um die Hardware des Würfels zu initialisieren.

Es werden 7 GPIO Ausgänge und 1 GPIO Eingang konfiguriert. Da der Tastendruck den Eingang mit dem Null-Potential verbindet, wird der Eingang außerdem als Pull up eingerichtet.

Damit die Betätigung des Buttons während jedem Programmstatus registriert wird, erfolgt bei der Initialisierung die Konfiguration des Buttons als Interrupt. Bei Betätigung des Buttons wird dadurch bei einer steigenden Flanke eine später weiter beschriebene callback Funktion ausgelöst.

##### 2.2.2 button\_callback:

Der button Callback wird nach jedem Tastendruck aufgerufen. Innerhalb der Callbackfunktion wird durch den Vergleich des Zeitstempels des Aufrufmoments mit dem Zeitstempels des loslassens ein langer Tastendruck erkannt. Die Erkennung läuft, bis der Taster los gelassen wird. Am Ende der Funktion wird immer der Auslösetimer für den Sleep-mode angehalten.

##### 2.2.3 timer\_start:

Der Würfel soll nach einer bestimmten Zeit in einen Ruhemodus gehen. Um diese Funtionalität zu gewährleisten, kann durch Aufruf dieser Funktion ein Repeating timer gestartet werden. Die repeating\_timer-Funktion im Pico SDK ruft eine Callback-Funktion in regelmäßigen Abständen auf, ohne das Hauptprogramm zu blockieren. Dies wird durch die Timer-Hardware auf dem Pico ermöglicht, die genaue Zeitmessungen durchführt. Nach Ablauf des Timers wird eine timer Callback funktion aufgerufen.

##### 2.2.4 timer\_callback:

Nach Ablauf des repeating\_timers wird diese Funktion aufgerufen, um den Ruhemodus einzuleiten. Nach einer Blink Animation erfolgt der Wechsel auf einen xosc timer. Im Anschluss erfolgt der Wechsel in den Dormant mode, welcher sich durch das drücken des Buttons beenden lässt.

##### 2.2.4 timer\_stop:

Damit der repeating timer nicht immer wider auslöst, wird durch timer\_stop() eine möglichkeit bereit gestellt diesen zu beenden.

**Hardware:** Es werden LED 1 bis 7 benannt, initialisiert und als Konstanten verfügbar gemacht.

Der Button wird nicht direkt im Hauptprogramm abgefragt, sondern während der Initialisierung als callback Interrupt Button konfiguriert. Dadurch wird im Falle das Drückens eine globale Variable in hardware.h gesetzt. Diese kann über die Funktion getButton() im Hauptprogramm abgefragt werden.

**Ruhezustand:** Es werden Funktionen zur Verfügung gestellt, die nach bestimmter Zeit den Ruhezustandsmodus automatisch starten, bzw den Ruhemodus deaktivieren.

Als Ruhemodus wurde hier der Dormant- Mode gewählt. Dieser ist der sparsamste Modus des 2040. Der Dormant-Mode ermöglicht außerdem ein Aufwecken, durch einen Tastendruck.

Der Einstieg in den Ruhemodus, wird durch einen repeating timer callback gestartet. Vor Dem Einstieg in den Ruhemodus, hält sich dieser Timer in seiner Callback Funktion selbst an. Die repeating\_timer-Funktion im Pico SDK ruft eine Callback-Funktion in regelmäßigen Abständen auf, ohne das Hauptprogramm zu blockieren. Dies wird durch die Timer-Hardware auf dem Pico ermöglicht, die genaue Zeitmessungen durchführt.

Der Ruhezustand lässt sich durch einen langen Tastendruck de oder aktivieren. Der normale Modus ist: Ruhezustand = true, nach 5 Sekunden. Erfolgreiches Umschalten wird durch ein Blinken von 1 s signalisiert.