

Raspberry Pi 5



AddOn

WiringPi

Version 1.01

Hallo zusammen,

in diesem *RasPi-AddOn* möchte ich ein paar Worte über *WiringPi* verlieren. *WiringPi* ist eine Softwarebibliothek, die speziell für den Raspberry Pi entwickelt wurde, um die GPIO-Pins (General Purpose Input/Output) einfach zu steuern. Sie ermöglicht es Entwicklern, GPIO-Pins über Programmiersprachen wie C oder Python anzusprechen, ohne sich um die komplizierte Low-Level-Hardwareprogrammierung kümmern zu müssen.

Mit *WiringPi* können Nutzer Pins als Eingänge oder Ausgänge definieren, Daten lesen oder schreiben und verschiedene Peripheriegeräte steuern, z.B. LEDs, Motoren oder Sensoren. Die Bibliothek stellt eine einfache API zur Verfügung, die die Arbeit mit Hardware auf dem Raspberry Pi erheblich vereinfacht.

WiringPi wurde ursprünglich von *Gordon Henderson* entwickelt, aber es wird mittlerweile von ihm nicht mehr offiziell gepflegt. Dennoch wird sie weiter von anderer Stelle unterstützt.

Wie kann die Bibliothek installiert werden?

Unter der folgenden Internetadresse ist WiringPi zu finden.

WiringPi
 <https://github.com/WiringPi/WiringPi/releases>

Dort stehen die folgenden Dateien zum Download bereit, die vom Januar 2025 sind.

 wiringpi_3.12_arm64.deb	66 KB	last month
 wiringpi_3.12_armhf.deb	59.6 KB	last month
 Source code (zip)		last month
 Source code (tar.gz)		last month

Abbildung 1 - WiringPi-Installationsdateien

Nach dem Herunterladen auf dem Raspberry Pi liegt die Datei (arm64) im Downloadordner.



Abbildung 2 - Die WiringPi-Installationsdatei

Über das Kontextmenü und der Auswahl des Punktes *Paketinstalltion* wird das Paket installiert.



Abbildung 3 - Installation über Paketinstallation einleiten

Im Anschluss erfolgt die Frage, ob das Paket installiert werden soll, was mit der Bestätigung über die *Installieren*-Schaltfläche erfolgt.

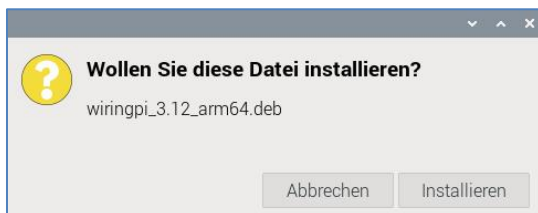


Abbildung 4 - Die Installation des WiringPi-Paketes

Anschließend muss noch das Passwort eingegeben werden.

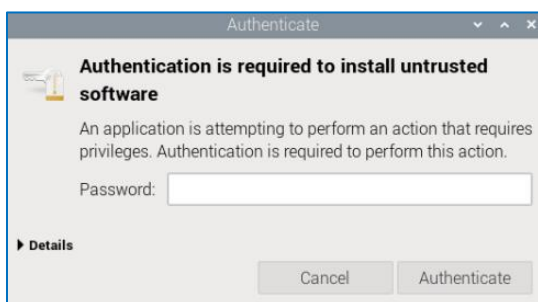


Abbildung 5 - Die Eingabe des Passwortes

Im Anschluss wird das Paket installiert.

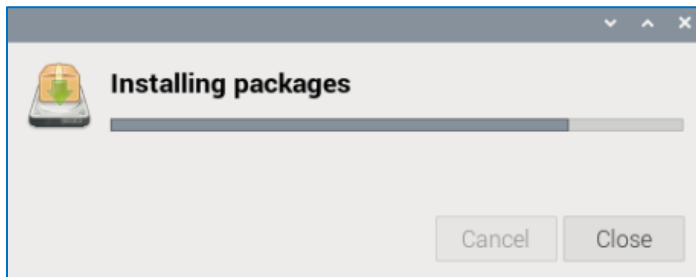


Abbildung 6 - Die Installation hat begonnen

Die Abfrage der Version

Über die folgende Eingabe in einem Terminal-Fenster kann die Version abgefragt werden.

```
erik@raspi: ~  
Datei Bearbeiten Reiter Hilfe  
erik@raspi:~ $ gpio -v  
gpio version: 3.12  
Copyright (c) 2012-2025 Gordon Henderson and contributors  
This is free software with ABSOLUTELY NO WARRANTY.  
For details type: gpio -warranty  
  
Hardware details:  
  Type: Pi 5, Revision: 00, Memory: 8192MB, Maker: Sony  
  
System details:  
  * Device tree present.  
    Model: Raspberry Pi 5 Model B Rev 1.0  
  * Supports full user-level GPIO access via memory.  
  * Supports basic user-level GPIO access via /dev/gpiomem.  
  * Supports basic user-level GPIO access via /dev/gpiochip (slow).  
erik@raspi:~ $
```

Abbildung 7 - Die Abfrage der Version von WiringPi

Ein erster Test - Eine LED ansteuern

Starten wir doch mit einem ersten Versuch und schließen dazu eine LED an den GPIO 14-Pin an wie das nachfolgend zu sehen ist.

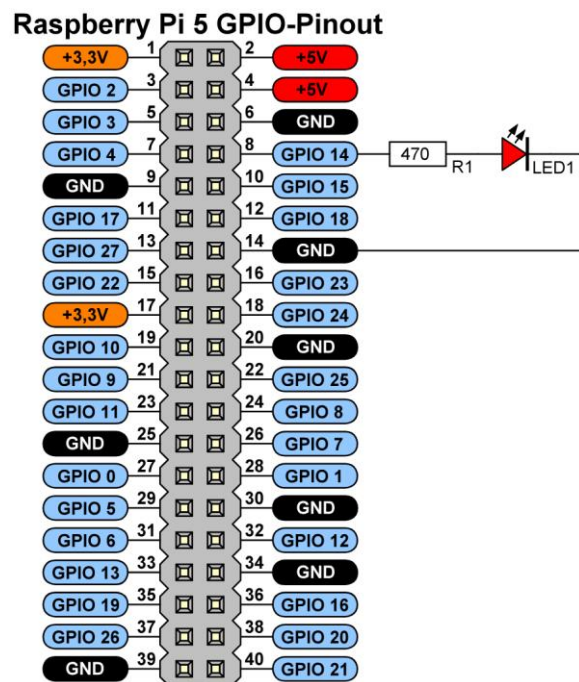


Abbildung 8 - Der Schaltplan zur Ansteuerung einer LED an GPIO 14

Jetzt wird ein Terminal-Fenster geöffnet und die folgenden Kommandos eingegeben.

```
erik@raspi: ~  
Datei Bearbeiten Reiter Hilfe  
erik@raspi:~$ gpio -g mode 14 out  
erik@raspi:~$ gpio -g write 14 1  
erik@raspi:~$
```

Abbildung 9 - Die Ansteuerung der LED an GPIO 14

Der Parameter `-g` in den folgenden Kommandos legt fest, dass die GPIO-Nummern des BCM-Chips verwendet werden. Wenn die Pin-Nummer verwendet werden sollen, dann muss das `-g` durch `-1` ersetzt werden.

Durch das nächste Kommando wird dann der Zustand am GPOP-Pin gesetzt, was durch *write* und den gewünschten Pegel erfolgt. 1 bedeutet einen High-Pegel und 0 einen Low-Pegel.

```
erik@raspi: ~
Datei Bearbeiten Reiter Hilfe
erik@raspi:~ $ gpio -g write 14 0
erik@raspi:~ $
```

Abbildung 10 - Das Ausschalten der LED an GPIO 14

Es ist sogar möglich, die Kontrolle über ein C-Programm zu erreichen, das wie folgt aussieht. Zuvor sollten wir jedoch wissen, dass WiringPi mit anderen Pin-Bezeichnungen arbeitet, die über das folgende Kommando angezeigt werden. Bei unserem GPIO 14 wäre das die Pummer 15, die gleich im C-Programm verwendet werden muss.

```
erik@raspi:~ $ gpio readall
```

Pi 5											
BCM	wPi	Name	Mode	V	Physical	V	Mode	Name	wPi	BCM	
		3.3v			1	2		5v			
2	8	SDA.1	ALT3	1	3	4		5v			
3	9	SCL.1	ALT3	1	5	6		0v			
4	7	GPIO. 7	-	0	7	8	0	TXD	15	14	
		0v			9	10	1	RXD	16	15	
17	0	GPIO. 0	-	0	11	12	0	GPIO. 1	1	18	
27	2	GPIO. 2	-	0	13	14		0v			
22	3	GPIO. 3	-	0	15	16	0	GPIO. 4	4	23	
		3.3v			17	18	0	GPIO. 5	5	24	
10	12	MOSI	ALT0	0	19	20		0v			
9	13	MISO	ALT0	0	21	22	0	GPIO. 6	6	25	
11	14	SCLK	ALT0	0	23	24	1	CE0	10	8	
		0v			25	26	1	CE1	11	7	
0	30	SDA.0	IN	1	27	28	1	SCL.0	31	1	
5	21	GPIO.21	-	0	29	30		0v			
6	22	GPIO.22	-	0	31	32	0	GPIO.26	26	12	
13	23	GPIO.23	-	0	33	34		0v			
19	24	GPIO.24	-	0	35	36	0	GPIO.27	27	16	
26	25	GPIO.25	-	0	37	38	0	GPIO.28	28	20	
		0v			39	40	0	GPIO.29	29	21	

```
erik@raspi:~ $
```

Abbildung 11 - Die Anzeige aller Pins über WiringPi

Doch nun zum C-Programm, das ich in den Editor *Geany* eingegeben habe.

```
1  /*  
2  * blink.c  
3  * Ein einfaches Blink-Programm um die wiringPi-Library zu testen  
4  */  
5  
6  #include <wiringPi.h> // Benoetigte wiringPi-Library  
7  #include <stdio.h>  
8  #include <stdlib.h>  
9  #include <stdint.h>  
10  
11 int main(void){  
12     printf("Raspberry Pi Blink-Programm mit wiringPi");  
13     if(wiringPiSetup() == -1)    // wiringPi initialisieren  
14         exit (1);              // Fehler? -> Programmende  
15     pinMode(15, OUTPUT);        // Pin als Ausgang programmieren  
16     for(;;){  
17         digitalWrite(15, HIGH); // Pin auf HIGH-Pegel setzen  
18         delay(1000);            // 1000ms warten  
19         digitalWrite(15, LOW);  // Pin auf LOW-Pegel setzen  
20         delay(1000);            // 1000ms warten  
21     }  
22     return 0;  
23 }
```

Abbildung 12 - Ein C-Programm zur Ansteuerung des GPIO-Pins

Bevor der Quellcode jedoch kompiliert werden kann, müssen einige Anpassungen in Geany erfolgen. Der Menüpunkt ist *Erstellen > Kommandos zum Erstellen konfigurieren*.

Im sich öffnenden Dialog müssen die markierten Änderungen vorgenommen werden.

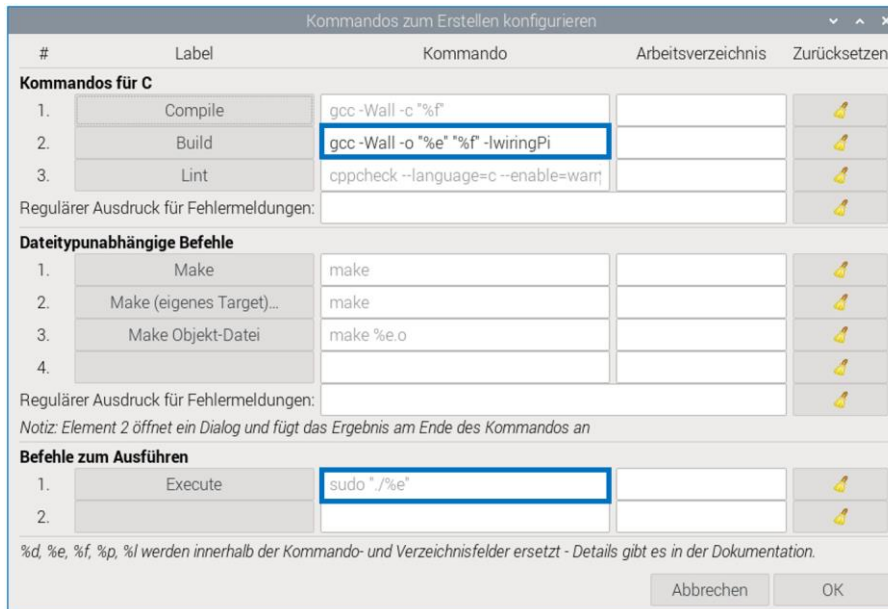


Abbildung 13 - Die Anpassungen im Geany-Editor

Ich habe die betreffenden Zeilen in Blau markiert. In der ersten muss der Zusatz `-lwiringPi` am Ende hinzufügen. Da das Programm zum ausführe Root-Rechte benötigt, ist im unteren blau markierten Bereich der Zusatz `sudo` erforderlich. Danach sollte das Erstellen bzw. das Starten des Programms aus Geany heraus keine Probleme mehr bereiten.

Viel Spaß beim Frickeln

Erik Bartmann

<https://erik-bartmann.de/>