

# **JOY-IT**

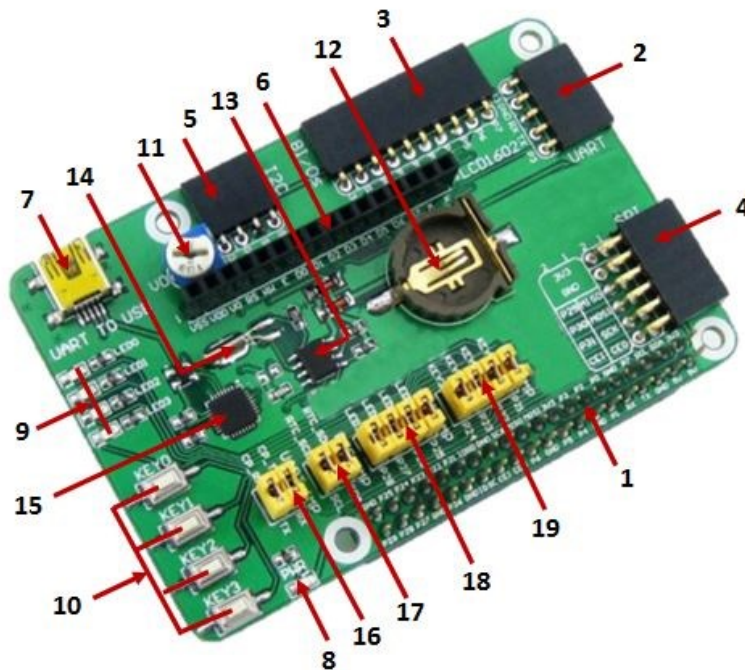
## **EXP500**



## Contents

1. Anschlüsse
2. Einführung
3. Basis Operationen
  - 3.1 System Image-Datei Installation
  - 3.2 Serielle Debug Umgebung installieren
4. Installieren der Bibliotheken benötigt
  - 4.1 LED Demo
  - 4.2 Key Demo
  - 4.3 LCD1602 (Separat erwerbbar) Demo
  - 4.4 PCF8563 RTC Demo
  - 4.5 UART Demo
5. Installation der Bibliotheken für RPi
  - 5.1 Installieren der WiringPi Bibliothek
  - 5.2 Installieren der C Bibliothek bcm2835
  - 5.3 Installieren der Python Bibliothek
  - 5.4 Konfigurieren der Schnittstellen
6. Support

# 1. Anschlüsse



1. Pin-Header zu direkten Aufstecken auf den Raspberry B+, 2B oder 3B
2. UART Schnittstelle: verbinden Sie ganz einfach UART Module wie z.B. RS232, RS485, USB TO UART (TTL-Kabel)
3. 8 x I/O Schnittstelle: verbinden Sie ganz einfach über I/O gesteuerte Module
4. SPI Schnittstelle für SPI angesteuerte Module
5. I2C Schnittstelle für I2C gesteuerte Module
6. LCD Schnittstelle für HD44780 Industrie-Standard-LCD-Module
7. USB TO UART, Konverter mit CP2102 Chipsatz (Mini USB-Anschluss)
8. Power LED
9. 4x programmierbare LED
10. 4x programmierbare Knöpfe
11. Potentiometer: Kontrastkontrolle für LCD Displays (siehe Punkt 6)
12. RTC Batteriehalter
13. PCF8563: onboard RTC chip
14. 32.768K Quarz: RTC Quarz
15. CP2102: onboard USB TO UART Chip, für debugging
16. CP2102: Jumper für: an/aus
17. RTC: Jumper für: an/aus
18. User LEDs Jumper für: an/aus (siehe Punkt 9)
19. User Keys Jumper für: an/aus (siehe Punkt 10)

## 2. Einführung

Raspberry Pi Erweiterung/Evaluation Board, unterstützt verschiedene Schnittstellen

---

## 3. Basis Operationen

### 3.1 System Image-Datei Installation

Laden Sie sich die neuste vorbereitete Image-Datei von unserer Website herunter: [Download](#)

Formatieren Sie eine micro-SD Karte mit der SDFormatter.exe

**Notiz: Die Speichergröße der micro-SD Karte sollte mehr als 4GB betragen. In dieser Operation wird ein micro SD Kartenleser benötigt, welcher separat erworben werden muss.**

Starte die Datei Win32DiskImager.exe und wählen Sie die kopierte System Image-Datei auf Ihrem PC, klicken Sie danach auf die Schaltfläche „Write“ um die System Image-Datei auf der Karte zu programmieren.

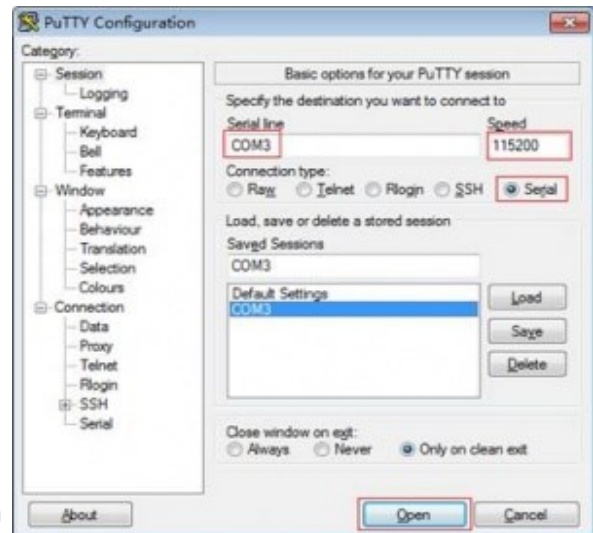
## 3.2 Serielle Debug Umgebung installieren

Verbinden Sie Ihren PC mit der UART zu USB Schnittstelle auf dem ,EXP500 via mini USB Kabel

- **Notiz: Der USB Part der UART zu USB Schnittstelle die auf dem EXP500 vorhanden ist, ist eine mini USB Schnittstelle, während die Schnittstelle auf dem Raspberry Pi eine micro-USB Schnittstelle ist. Bitte Beachten Sie dies.**

Installieren Sie den „cp2102\_driver“ Treiber

Starten Sie das Programm PuTTY.exe und konfigurieren Sie die folgenden Parameter:



- Serial line: wird benutzt um den entsprechenden Seriellen Port auszuwählen. Der momentan benutzte Serielle Port kann per Geräte Manager überprüft werden.
- Schnelligkeit: wird benutzt um die Baudrate zu setzen : 115200
- Verbindungstyp: Diese Option sollte auf Seriell gesetzt werden. Danach klicken Sie auf die Schaltfläche „Open“

Nachdem das System gestartet ist sollten Sie folgende Informationen eingeben: Benutzername: pi und Passwort: raspberry . Danach können Sie das Serielle Terminal öffnen um mit dem Raspberry Pi zu Kommunizieren.

**Notiz: In diesem Dokument wird das Programm PuTTY immer benutzt um den Raspberry Pi per Serieller Port Kommunikation zu steuern, falls nicht anders spezifiziert. Wenn ein Serieller Port für das Raspberry Pi Terminal Debuggen benutzt wird , kann es nicht mehr als ein normaler Serieller Port benutzt werden , da es seitdem von der System Debug Funktion belegt wird.**

## 4. Installieren der Bibliotheken benötigt

Vor dem benutzen des EXP500 Programms sollten Sie die bcm2835, wiringPi und python Bibliotheken auf dem RPi installieren um weitere APIs hinzuzufügen und danach sollten Sie die Einstellungen konfigurieren um den Kern Treiber des I2C, SPI und UART nach dem Installieren der Bibliotheken zu starten.

Sie können den API Quell Code im Verzeichnis /home/pi/EXP500 finden wenn Sie die fertige System Image-Datei benutzen.

**Notiz: Jede Demo sollte unter dem entsprechenden Verzeichnis laufen, z.B. um die bcm2835 Demo der LED zu benutzen, geben Sie folgendes Kommando ein:**

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ sudo ./led
```

Das bedeutet Sie sollten zuerst den Pfad eingeben und dann das Kommando ausführen.

```
sudo ./led
```

 (./led ist eine Kompilierte und ausführbare Datei).

Das Sudo bedeutet das der Befehl als super Benutzer ausgeführt wird.

Das Sudo wird nicht benötigt wenn Sie als Stammverzeichnis Benutzer angemeldet sind.

```
root@raspberrypi:/home/pi/EXP500/LED/bcm2835# ./led
```

Wenn nicht anders spezifiziert, sind die folgenden Demos vor Kompiliert für ausführbare Dateien und können direkt gestartet werden. Falls Sie die C Programme von bcm2835, wiringPi, sysfs, usw. Kompilieren wollen, betreten Sie bitte das Demo Verzeichnis und führen folgendes Kommando make aus um das Programm zu kompilieren und make clean um die Ausführbaren Dateien zu entfernen. Zum Beispiel:

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ make clean  
rm led  
pi@raspberrypi ~/EXP500/LED/bcm2835 $ make  
gcc -Wall led.c -o led -lbcm2835
```

Das Python Programm kann direkt ohne Kompilierung gestartet werden, z.B.

```
pi@raspberrypi ~/EXP500/LED/python $ sudo python led.py
```

## 4.1 LED Demo

### bcm2835 Programm:

1. Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/LED/bcm2835 $ sudo ./led
```

2. Die 4 LEDs werden alle nacheinander leuchten.
3. Drücken Sie die Tasten Strg+c um die Demo zu beenden.

### wiringPi Programm:

1. Öffnen Sie die Linux Konsole führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/LED/wiringPi $ sudo ./led
```

2. Die 4 LEDs werden alle nacheinander leuchten.
3. Drücken Sie die Tasten Strg+c um die Demo zu beenden.

### Python Programm:

1. Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/LED/python $ sudo python led.py
```

2. Die 4 LEDs werden alle nacheinander leuchten.
3. Drücken Sie die Tasten Strg+c um die Demo zu beenden.

### shell Programm:

1. Schalten Sie LED0 mit diesem Kommando ein:

```
pi@raspberrypi ~/EXP500/LED/shell $ sudo ./LED.sh 26 1
```

2. Schalten Sie LED0 mit diesem Kommando aus:

```
pi@raspberrypi ~/EXP500/LED/shell $ sudo ./LED.sh 26 0
```



## 4.2 Key Demo

### bcm2835 Programm:

1. Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. Das Terminal zeigt an wenn eine Taste gedrückt wird.
3. Drücken Sie die Tasten Strg+c um das Programm zu beenden.

### wiringPi Programm:

1. Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. Das Terminal zeigt an wenn eine Taste gedrückt wird.
3. Drücken Sie die Tasten Strg+c um das Programm zu beenden.

### Python Programm:

1. Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/KEY/bcm2835 $ sudo ./key
```

2. Das Terminal zeigt an wenn eine Taste gedrückt wird.
3. Drücken Sie die Tasten Strg+c um das Programm zu beenden.

### 4.3 LCD1602 (Separat erhältlich) Demo

Stecken Sie das LCD1602 in die LCD1602 Schnittstelle auf dem EXP500 ein.  
Öffnen Sie die Linux Konsole und führen Sie folgendes Kommando aus:

```
pi@raspberrypi ~/EXP500/LCD1602 $ sudo ./lcd1602
```

LCD1602 wird die jeweiligen Informationen anzeigen. Sollte auf dem LCD nichts angezeigt werden müssen Sie zuerst den Kontrast einstellen. Die rote Box im Bild zeigt die Position des Potentiometers an womit Sie die Einstellung vornehmen können.



Stecken Sie das LCD1602 in die LCD1602 Schnittstelle auf dem EXP500 ein.

## 4.4 PCF8563 RTC Demo

Setzen Sie die Jumper des EXP500:

- Verbinde RTC\_SDA mit SDA
- Verbinde RTC\_SCL mit SCL

Installieren Sie die i2c-tools. Falls Sie die i2c-tools noch nicht installiert haben , öffnen Sie bitte die Konsole und geben Sie folgendes ein: `root@ raspberrypi:/# apt-get install i2c-tools`

Danach geben Sie folgendes Kommando ein:

```
root@ raspberrypi:/# i2cdetect -y 1
```

Ihnen wird jetzt die Geräte Adresse des PCF8563 angezeigt welches mit dem Raspberry Pi Modul verbunden ist. Hier ist die Geräte Adresse des PCF8563 51, was zeigt das es vom Raspberry Pi erkannt wurde.

```
root@raspberrypi:/home/pi# i2cdetect -y 1
   0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- UU -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- 51 -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
```

PCF8563 wurde vom Raspberry Pi erkannt.

Geben Sie nun das entsprechende Kommando in die Konsole ein:

### bcm2835 Programm:

```
pi@raspberrypi ~/EXP500/PCF8563/bcm2835 $ sudo ./pcf8563
```

### wiringPi Programm:

```
pi@raspberrypi ~/EXP500/PCF8563/wiringPi $ sudo ./pcf8563
```

### python Programm:

```
pi@raspberrypi ~/EXP500/PCF8563/python $ sudo ./pcf8563.py
```

Die Zeit des PCD8563 wird in der Konsole angezeigt.

## 4.5 UART Demo

**Notiz: Der Serielle Port des Raspberry Pi wird normalerweise für das Debuggen über die Konsole benutzt, damit Sie aber UART benutzen können muss diese Funktion ausgeschaltet werden.**

Verbinden Sie den eingebauten UART zu USB Port per mini USB Kabel mit einem PC.

Öffnen Sie die Konsole und führen Sie folgendes Kommando aus:

**wiringPi Programm:**

```
pi@raspberrypi ~/EXP500/UART/wiringPi $ sudo ./UART
```

**python Programm:**

```
pi@raspberrypi ~/EXP500/UART/python $ sudo ./uart.py
```

Starten Sie das Serielle Monitor Programm. Suchen Sie den entsprechenden COM Port und setzen Sie die Baudrate auf 115200. Das Feld was von dem Programm gesendet wird , wird zurückgeschickt und von dem Programm angezeigt.

## 5. Installation der Bibliotheken für RPi

Um unsere API Beispiele zu benutzen , werden Bibliotheken benötigt die manuell installiert werden sollten.

[bcm2835 Bibliothek](#)

[wiringPi Bibliothek](#)

### 5.1 Installieren der WiringPi Bibliothek

[Hier klicken](#) um die WiringPi Bibliotheken herunter zu laden, andernfalls können Sie die neuste Version der Bibliotheken auch von der WiringPi Webseite laden:

<https://projects.drogon.net/raspberry-pi/wiringpi/download-and-install/>

Kopieren Sie das Installationspaket in ihr Eigenes System per USB Stick, öffnen Sie dann den WiringPi Ordner und führen Sie folgenden Befehl aus um das Paket zu installieren :

```
chmod 777 build
./build
```

Führen Sie folgendes Kommando aus um die Installation zu überprüfen:

```
gpio -v
```

### 5.2 Installieren der C Bibliothek bcm2835

Klicken Sie auf den Link zum herunterladen Bibliothek. [bcm2835-1.50 Bibliothek](#).

Sie können die neuste Version auch von der bcm2835 Webseite erhalten: <http://www.airspayce.com/mikem/bcm2835/>

Kopieren Sie das Installationspaket in ihr eigenes System, öffnen Sie den bcm2835 Bibliotheken Ordner und geben Sie folgendes ein zum installieren:

```
./configure
make
sudo make check
sudo make install
```

## 5.3 Installieren der Python Bibliothek

Python Bibliotheken für Raspbian ( enthält RPi, GPIO und spidev Installationspakete.

Siehe: <https://pypi.python.org/pypi/RPi.GPIO> , <https://pypi.python.org/pypi/spidev> ) und bekommen Sie es per apt-get Kommandos.

Bitte beachten Sie das Ihr Raspberry Pi mit dem Netzwerk verbunden sein sollte wenn Sie das Kommando apt-get benutzen um eine Bibliothek zu installieren. Vor der Installation können Sie folgendes Kommando benutzen um Ihre Software Liste zu aktualisieren:

```
sudo apt-get update
```

1. Geben Sie folgendes Kommando ein um das python-dev Paket zu installieren.

```
sudo apt-get install python-dev
```

2. Um das Paket für Rpi, GPIO zu installieren (GPIO Schnittstellen Funktionen), kopieren Sie das Rpi, GPIO Installationspaket auf Ihr RPi Board und entpacken es. Betreten Sie die entpackte Datei per Konsole und geben Sie folgendes Kommando ein um die Bibliothek zu installieren.:

```
sudo python setup.py install
```

3. Geben Sie folgendes Kommando ein um die smbus Bibliothek zu installieren (I2C Schnittstellen Funktionen)

```
sudo apt-get install python-smbus
```

4. Run Geben Sie folgendes Kommando ein um die Serielle Bibliothek, welche UART Schnittstellen Funktionen enthält zu installieren:

```
sudo apt-get install python-serial
```

5. Um die spidev Bibliothek für die SPI Funktionen zu installieren, kopieren Sie das spidev Paket auf Ihr RPi Board und entpacken es. Betreten Sie die entpackte Datei per Konsole und geben Sie folgendes Kommando ein um die Bibliothek zu installieren:

```
sudo python setup.py install
```

## 5.4 Konfigurieren der Schnittstellen

(Vor dem Ausführen der API Codes die wir anbieten sollten Sie die Kern Treiber der Schnittstellen starten. In der System Image-Datei sind beide, I2C und SPI standartmäßig auf "Enable" eingestellt, aber der Serielle Port ist immer noch im "Debug per Konsolen Funktion" Modus.)

1. **Erlauben der I2C Funktion. Geben Sie folgendes Kommando ein um ihr Raspberry Pi Board zu konfigurieren:**

```
sudo raspi-config
```

Wählen Sie Advanced Options -> I2C -> yes, um den I2C Kern Treiber zu starten.

Danach müssen Sie noch die Konfigurationsdatei verändern. Um die Konfigurationsdatei zu Öffnen geben Sie bitte folgendes Kommando ein:

```
sudo nano /etc/modules
```

Fügen Sie folgende zwei Zeilen am Ende der Konfigurationsdatei hinzu:

```
i2c-bcm2708  
i2c-dev
```

Drücken Sie die Tasten Strg+x zum verlassen und geben Sie Y ein um die Einstellungen zu Speichern. Danach starten Sie das Modul um die Einstellung in kraft zu setzen.

**2. Erlauben der Seriellen Funktionen.** Der Serielle Port des RPi ist standardmäßig auf den Debuggen per Konsolen Modus eingestellt. Wenn Sie die Seriellen Port Dienste als normalen IO benutzen wollen, müssen Sie die Einstellungen des RPi verändern. Wenn die Debug per Konsolen Funktion abgeschaltet ist, können Sie nicht auf das RPi Board per Seriellen Port zugreifen. Wenn Sie das RPi steuern wollen müssen Sie die Debug per Konsolen Funktion des Seriellen Ports wieder einschalten.

```
sudo raspi-config
```

Wählen Sie Advanced Options -> Serial. Durch das auswählen der Option „No“ kann die Debug per Konsole Funktion abgeschaltet werden. Danach kann der Serielle Port für die Serielle Kommunikation benutzt werden. Mit dem Wählen der Option „Yes“ kann die Debug per Konsolen Funktion wieder aktivieren.

Sie sollten nun das Modul neustarten damit die Einstellung in kraft treten.

**Notiz: Der Serielle Port des Raspberry Pi 3 Model B ist nicht verfügbar da Pin 14 und 15 mit dem eigenen Bluetooth verbunden sind.**

Starten Sie die spi Funktion und geben Sie folgendes Kommando ein:

```
sudo raspi-config
```

Wählen Sie Advanced Options -> I2C -> „Yes“, um den Kern Treiber des I2C zu starten.



## **6. Support**

Wir sind auch nach dem Kauf für Sie da sollten noch Fragen offen bleiben oder Probleme auftauchen stehen wir Ihnen auch per E-Mail, Telefon und Ticket-Supportsystem zur Seite.

E-Mail: [service@joy-it.net](mailto:service@joy-it.net)

Ticket-System: <http://support.joy-it.net>

Telefon: +49 (0)2845 98469 – 66 (11- 18 Uhr)

Für weitere Informationen besuchen Sie unsere Website:

[www.joy-it.net](http://www.joy-it.net)