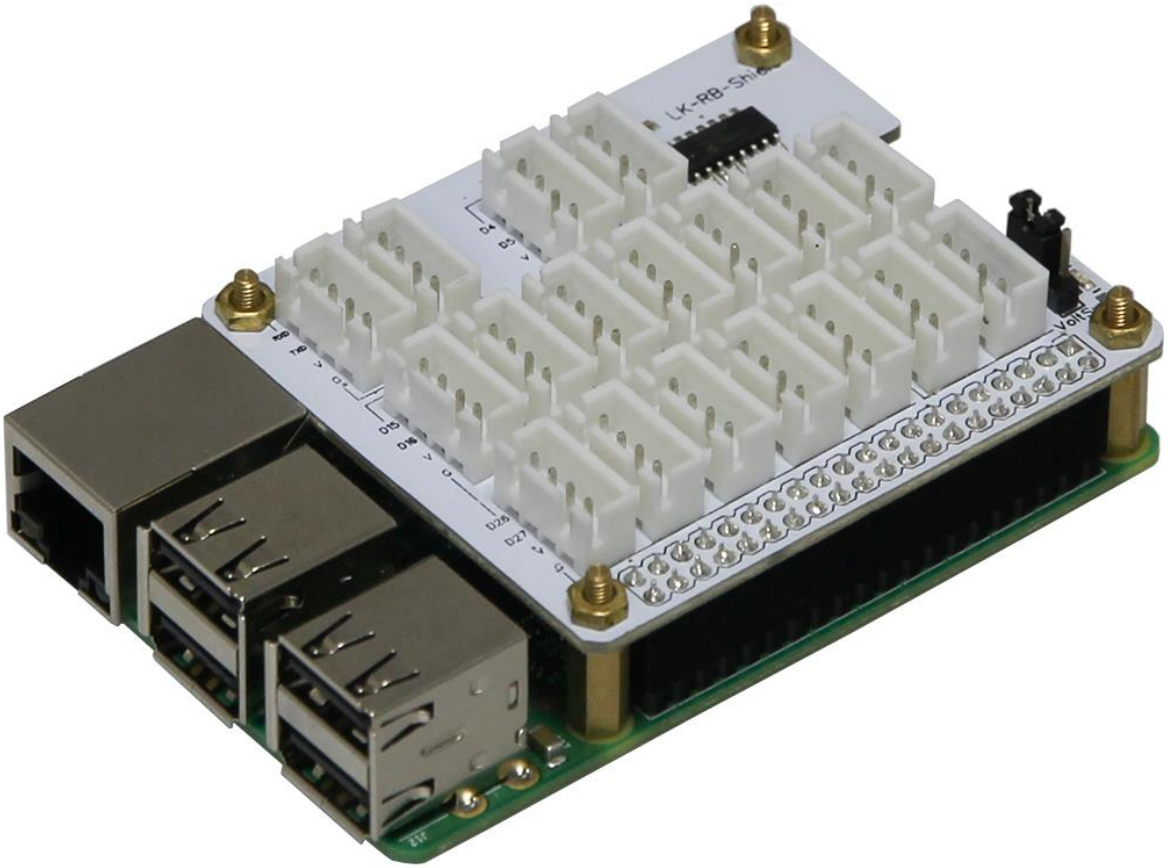
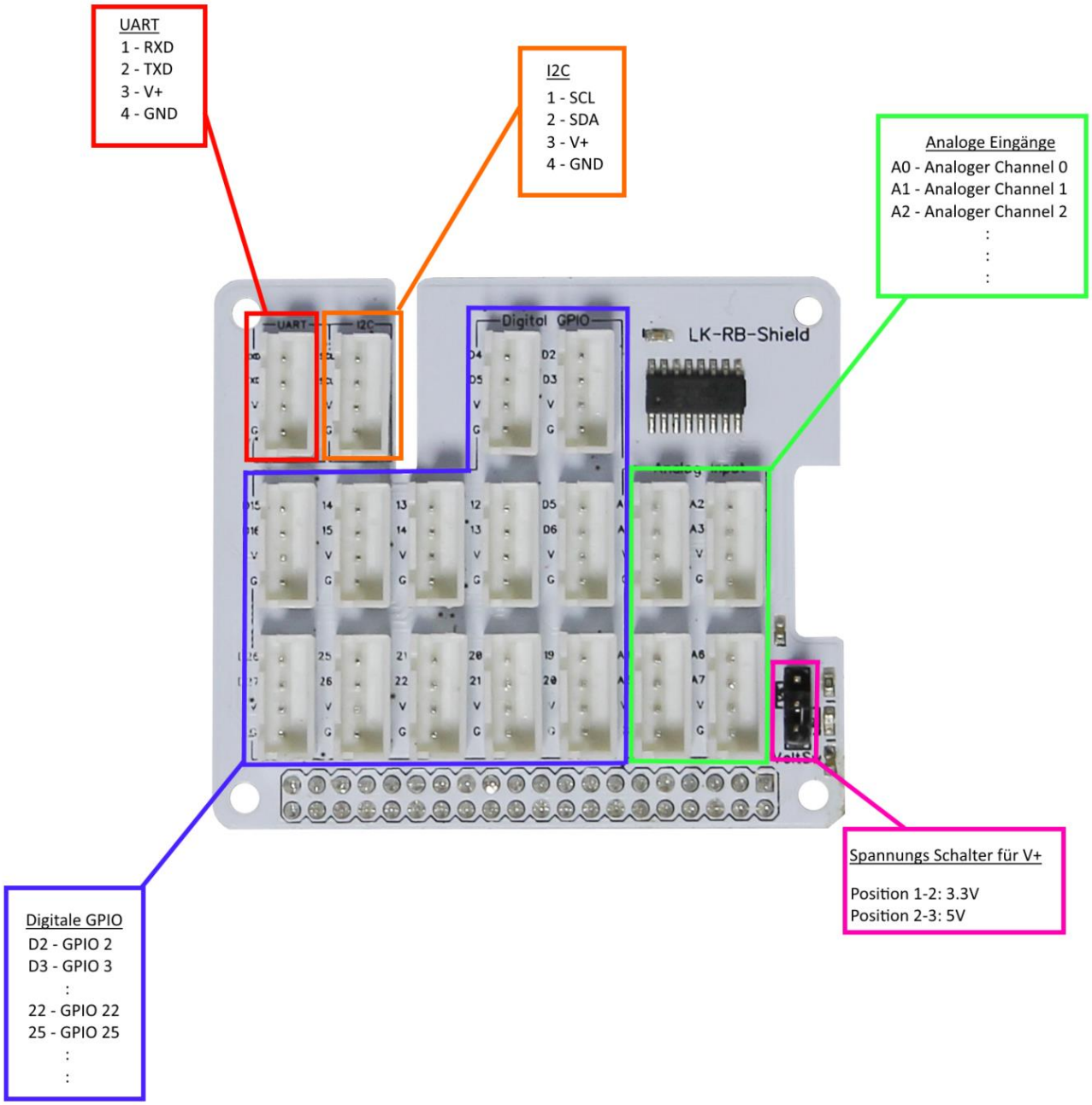
LK-Baseboard für Raspberry Pi B+ / Pi 2

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:





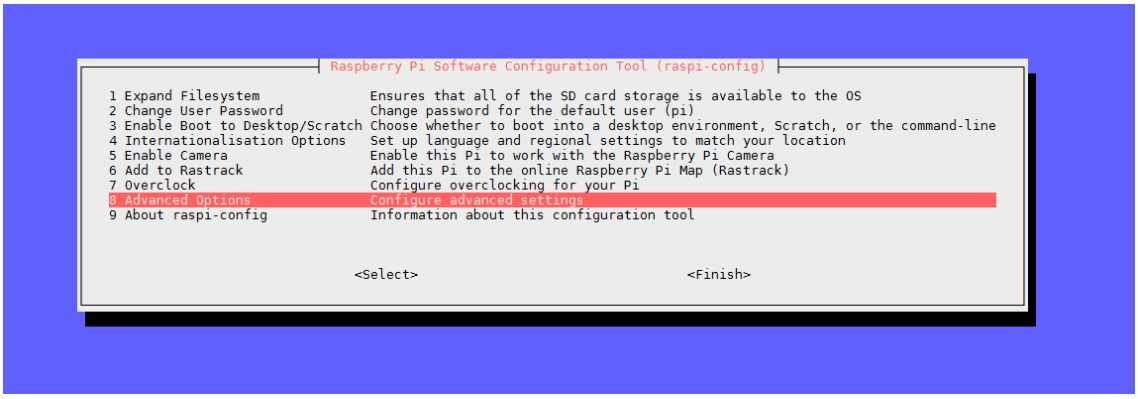
Ansteuern der Analogen Eingänge über den MCP3008

1. Installation der benötigten Module

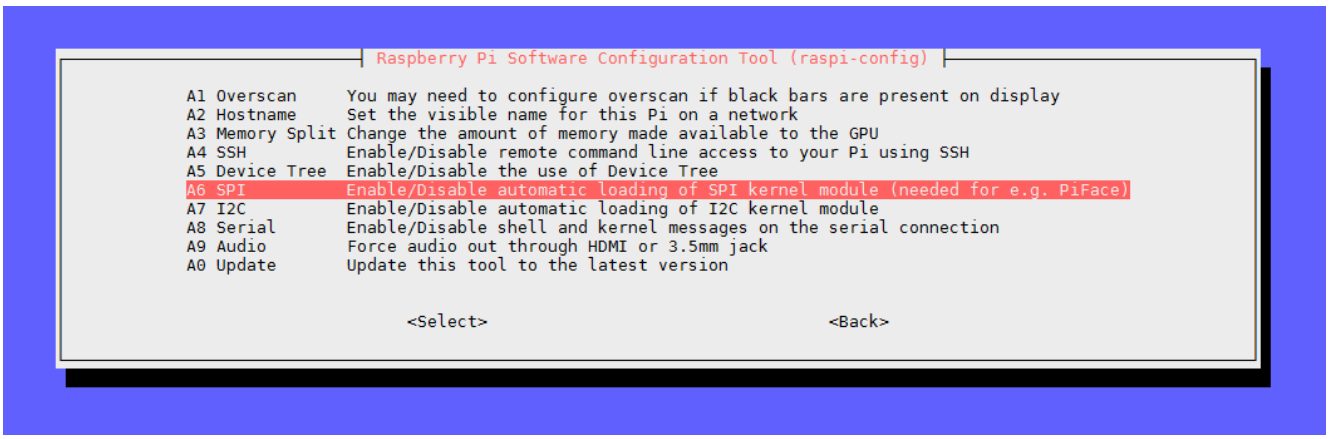
Wir empfehlen hierbei ein aktuelles Raspbian (Debian Wheezy) zu verwenden. Zuerst muss die SPI-Schnittstelle des Raspberry Pi aktiviert werden, damit dieser mit dem ADC des LK-Baseboards kommunizieren kann. Hierzu wird folgender Befehl eingegeben:

```
sudo raspi-config
```

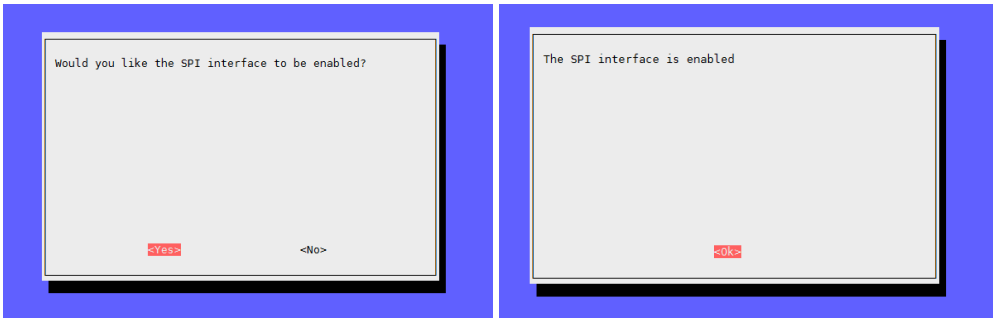
Im darauffolgend auftauchenden Fenster, gehen wir auf den Optionspunkt „Advanced Options“



Danach auf „A6 SPI“

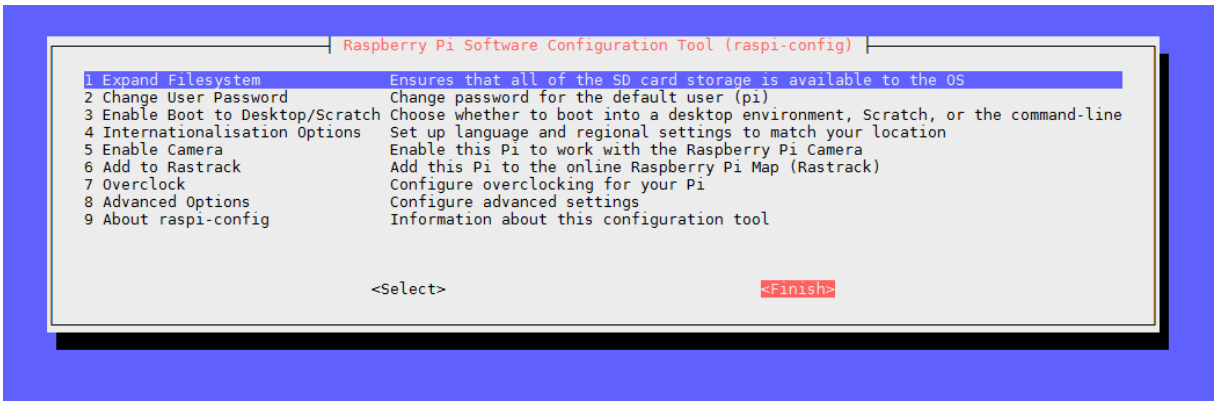


Die nächsten beiden Fenster müssen mit „Yes“ und „OK“ bestätigt werden



Ebenfalls die beiden folgenden

Zum Schluss gehen Sie auf Finish um das „Configuration Tool“ zu beenden…



…und starten den Raspberry Pi mit folgenden Befehl neu:

```
sudo reboot
```

Nach dem Neustart müssen die benötigten Treiber und Module installiert werden. Hierzu geben Sie die folgenden Befehlen in die Konsole des Raspberry Pi ein und bestätigen diese mit [Enter]. Der Raspberry Pi muss dabei mit dem Internet verbunden sein:

```
sudo apt-get update
```

```
sudo apt-get install python-imaging python-imaging-tk python-pip python-dev git
```

```
sudo pip install spidev
```

```
sudo pip install wiringpi
```

Auch hiernach sollte der Raspberry Pi neugestartet werden:

```
sudo reboot
```

Beispiel der Nutzung des ADC Controllers MCP3008 mit Hilfe der Datei testadc.py

```
import spidev
import time
import sys

spi = spidev.SpiDev()
spi.open(0,0)

def readadc(adcnum):
   if adcnum >7 or adcnum <0:
      return-1
   r = spi.xfer2([1,8+adcnum <<4,0])
   adcout = ((r[1] &3) <<8)+r[2]
   return adcout

while True:
   if len(sys.argv) >1:
    for i in range(len(sys.argv)):
     if i == 0:
      print "_____\n"
     else:
      adc_channel = int(sys.argv[i])
      print "Channel " + str(adc_channel)
      value=readadc(adc_channel)
      volts=(value*3.3)/1024
      print("%4d/1023 => %5.3f V" % (value, volts))
      print " "
    print "_____\n"
    time.sleep(1.5)
   else:
    print "_____\n"
    print "Channel 0"
    value=readadc(0)1
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 1"
    value=readadc(1)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 2"
    value=readadc(2)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 3"
    value=readadc(3)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 4"
    value=readadc(4)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 5"
    value=readadc(5)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 6"
    value=readadc(6)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "Channel 7"
    value=readadc(7)
    volts=(value*3.3)/1024
    print("%4d/1023 => %5.3f V" % (value, volts))
    print "_____\n"
    time.sleep(1.5)
```

Die oben aufgezeigte Datei „testadc.py“ zeigt auf, wie aktuelle analoge Werte vom ADC Controller mittels eines Python-Script ausgelesen werden; dabei kann mittels der Funktion readadc() mit der Angabe von ein entsprechenden Kanal (0-7) der Wert ausgelesen werden.

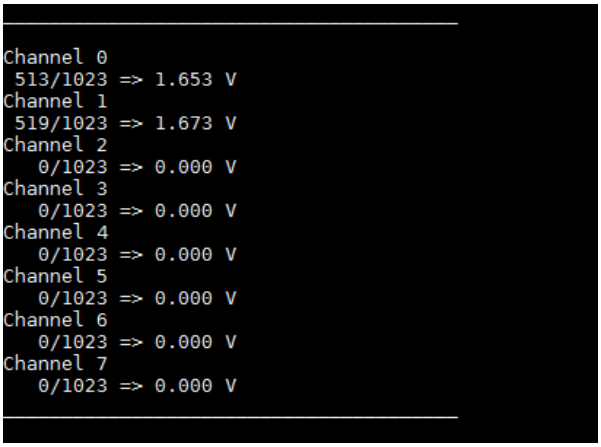
Erstellen Sie eine Datei mit dem Namen „testadc.py“ und kopieren Sie den obenstehenden Inhalt in diese Datei und speichern diese ab (achten Sie hierbei auf die einzelnen Leerzeichen vor jeder Zeile); alternativ können Sie auch die vorbereitete Datei aus diesem Zip-Paket auf den Raspberry Pi in ein entsprechendes Verzeichnis kopieren.

Das Skript kann nun auf zwei Weisen ausgeführt werden:

Ausgabe der Werte aller ADC-Kanäle:

```
sudo python testadc.py
```

Dieser Befehl bewirkt, dass die Werte aller Kanäle angezeigt, jede 1.5s neu ausgelesen und aktualisiert werden.



Ausgabe der Werte einzelner ADC-Kanäle:

```
sudo python testadc.py 3 7
```

Sie können auch nur die Werte einzelner Kanäle abfragen – geben Sie nach dem oberen Befehl, mit einem Leerzeichen Abstand, die benötigten Kanäle, die ausgelesen werden sollen. In diesem Beispiel sind das der Kanal [3] und [7]