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"

1 Expand Filesystem 2 Change User Password 3 Enable Boot to Desktop/Scratch 4 Internationalisation Options 5 Enable Camera 6 Add to Rastrack 7 Overclock 8 Advanced Options	Ensures that all of the SD ca Change password for the defau Choose whether to boot into a Set up language and regional Enable this P1 to work with th Add this P1 to the online Ras Configure overclocking for yo Configure advanced settings	rd storage is available to the OS It user (p) desktop environment, Scratch, or the command-line settings to match your location he Raspberry Pi Camera pberry Pi Map (Rastrack) ur Pi
9 ADOUT raspi-contig	Information about this config Select>	<finish></finish>

Danach auf "A6 SPI"

A2 Hostname	Set the visible name for this	Pi on a network
A3 Memory Spli	t Change the amount of memory m	ade available to the GPU
A4 SSH	Enable/Disable remote command	line access to your Pi using SSH
A5 Device Tree	Enable/Disable the use of Dev	ice Tree
A6 SPI	Enable/Disable automatic load	ing of SPI kernel module (needed for e.g. PiFace)
A7 120	Enable/Disable automatic load	Ing of 120 Kernel module
A8 Serial	Enable/Disable shell and Kern	at messages on the serial connection
A0 Update	Update this tool to the lates	t version
	<select></select>	<back></back>

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...

1 Expand Filesystem	Change password for th	a SD card storage is available to the OS	d-line
2 Change User Password	Change password for th	e default user (pi)	
3 Enable Boot to Desktop/Scr.	otch Choose whether to boot	into a desktop environment, Scratch, or the comman	
4 Internationalisation Option	is Set up language and re-	gional settings to match your location	
5 Enable Camera	Enable this Pi to work	with the Raspberry Pi Camera	
6 Add to Rastrack	Add this Pi to the onl.	ine Raspberry Pi Map (Rastrack)	
7 Overclock	Configure overclocking	for your Pi	
8 Advanced Options	Configure advanced set	tings	
9 About raspi-config	Information about this	configuration tool	
	<select></select>	<finish></finish>	

... und starten den Raspberry Pi mit folgenden Befehl neu:

		udo 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

ipor e oprace	
nport time	
nport sys	
ni – cridau (riDau()	
bi.open(0,0)	
ef 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	
hile 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_cnannel)	
rint(%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))	
value=readade(1)	
v_{0} 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 (%40/1025 -> %5.51 V % (Value, Volts))	
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(b)	
volts=(Value '5.5)/1024 print("%4d/1023 => %5.2f\/" % (volue, volte))	
print(7040/1025 -> 705.31 V 70 (Value, Volts)) nrint "Channel 7"	
value=readadc(7)	
volts=(value*3.3)/1024	
r = r + (1000 - 0.0) + 202 + 200 +	
print(%40/1023 => %5.31 V % (Value, Volts))	
print(%40/1023 => %5.31 V % (Value, Volts)) print "	\n"

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

Channel 0		
513/1023 => 1.653	V	
Channel 1		
519/1023 => 1.673	V	
Channel 2		
0/1023 => 0.000	V	
Channel 3		
0/1023 => 0.000	V	
Channel 4		
0/1023 => 0.000	V	
Channel 5		
0/1023 => 0.000	V	
Channel 6		
$0/1023 \implies 0.000$	V	
Channel 7		
0/1023 => 0.000	V	
	· · · · · · · · · · · · · · · · · · ·	

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]

