

c't-Bot Sprachmodul SP03 Dokumentation

Harald W. Leschner <hari@h9l.net>

Revidiert für L^AT_EX, März 2008

Zusammenfassung

Dieses Dokument beschreibt die Implementierung des SP03 Sprachmoduls von Devantech Inc. für den c't-Bot Roboter¹. Die Steuerung erfolgt über das I²C - Protokoll und der Code für den Treiber ist in C geschrieben.

Inhaltsverzeichnis

1	Hardware	1
1.1	Devantech SP03 Sprachmodul	1
1.1.1	Technischer Aufbau	1
1.1.2	Elektrische Eigenschaften	2
1.1.3	Anschlüsse des Moduls	2
1.2	Eine Stimme für den c't-Bot!	4
2	Software	5
2.1	Das I ² C Protokoll	5
2.1.1	Die Befehlssequenzen	6
2.1.2	Parameter der Stimmerzeugung	6
3	Montage	7
3.1	Befestigungsplatte	7
3.2	Bilder	8

1 Hardware

Einem Roboter das Sprechen beizubringen wäre vor einigen Jahren noch Science-Fiction gewesen, heute ist es dank der hochintegrierten Chips und Prozessoren im Handumdrehen möglich. Der c't-Bot bietet eine ideale Lern- und Experimentierplattform für die ersten Erfahrungen mit Elektronik, Mikrokontrollertechnik und Softwareprogrammierung; seine Schnittstellen und der Code sind sehr gut dokumentiert und bieten daher optimale Voraussetzungen, um mit eigenen Hardwareerweiterungen rumzuspielen. Zudem ist gerade ein sprechender Roboter interessant, weil diese Fähigkeiten bei der Erlernung der Programmierung neue Einblicke in die KI-Forschung ermöglicht oder Persönlichkeitsfragen der Roboterseele möglich machen. . .

¹Siehe Projektseite unter <http://www.heise.de/ct/projekte/ct-bot/>

1.1 Devantech SP03 Sprachmodul

Aufgrund der geringen Abmessungen und der einfachen Ansteuerung, bietet sich das SP03 als sinnvolle Erweiterung für den Roboter nahezu an. Im folgenden wird die Anbindung an einen **AVR RISC** Prozessor, ATmega644 von Atmel beschrieben. Eine analoge Implementierung des Codes (mit geringfügigen Änderungen) für andere Controller, anderer Roboter, ist problemlos möglich. Mehr dazu im Kapitel Software.

1.1.1 Technischer Aufbau

Das SP03 Sprachmodul kann ASCII Texte direkt in gesprochene Sprache konvertieren. Das erzeugte Sprachsignal wird mittels eines kleinen Lautsprechers akustisch ausgegeben, ist in englischer Sprache und (wie in meinem Fall) mit männlicher oder weiblicher Stimme. Kernbaustein des Moduls ist der **WTS701EM/T** von Winbond, ein hochintegrierter Chip für *Text-to-Speech-Systeme (TTS)* mit einem Textprozessor, diversen Filtern und einem Datenspeicher, alles in einem Gehäuse.

Die Sprachsynthese erfolgt durch Vorverarbeitung des eingehenden Textes in Phoneme, die anschließend in natürlich klingende Wortteile modelliert und abgebildet werden. Dabei versucht der Sprachalgorithmus im passenden Kontext eines Satzes möglichst viele Wortteile bei der Generierung der Satzmelodie zu berücksichtigen. Satzzeichen wie Fragezeichen führen zu einer leichten Anhebung am Ende, Zahlennamen werden aus einzelnen Ziffern verknüpft und E-Mail Adressen oder URLs werden einzeln buchstabiert. Maximal können bei einer einzelnen Übertragung 85 Bytes (3 Steuercodes + 0x00 + 81 Zeichen) verarbeitet werden.

1.1.2 Elektrische Eigenschaften

Praktischerweise verfügt das Modul über 2 Schnittstellen um mit diversen Controllern gesteuert zu werden. Zudem gibt es auch ein Eingabeport worüber auch feste, dauerhaft gespeicherte Sätze direkt programmiert und abgerufen werden können. Die detaillierteren Eigenschaften sind:

- Betriebsspannung: 5 V DC
- Strombedarf: 35 mA Standby, ca 60 – 100 mA bei Sprachausgabe
- Serielle Schnittstelle: RS-232 (38400 baud) oder Controller mit MAX232
- Bussystem: I²C Protokoll
- Eingebauter Lautsprecher: 40 mm
- Integrierter Verstärker: 325 mW
- Paralleler Eingabeport: 5 bits
- Feste speicherbare Sätze: 30 Sätze mit mehr als 1925 Zeichen insgesamt

1.1.3 Anschlüsse des Moduls

Über 2 Stiftleisten werden die I/O-Ports und die Stromversorgung für das Modul angeschlossen. An PL1 befinden sich die serielle Schnittstelle für Programmierung und Steuerung mit dem PC sowie die I²C Ports für die Kommunikation mit einem Mikrokontroller, PL2 ist quasi eine 6 bit parallele Schnittstelle mit Statusbit für die Adressierung der fest eingespeicherten Sätze. Die Stromversorgung erfolgt wahlweise über einen der beiden Buchsen.

Man sollte vor dem Anlegen sämtlicher Spannungen und Ströme immer sicherstellen, daß man die richtigen Pins verbunden hat. Die Anschlußbelegung ist wie folgt:

Pin	PL1	Verwendung
1	+5V	5 V DC - max. 100 mA
2	SDA	I ² C-Bus SDA Leitung
3	SCL	I ² C-Bus SCL Leitung
4	NC	Nicht verbunden
5	GND	Masse
6	SPARE	Undef. Pin
7	GND	Masse
8	RS-232 RX	TX am PC
9	RS-232 TX	RX am PC

Pin	PL2	Verwendung
1	+5V	5 V DC - max. 100 mA
2	STATUS	High=Reden, Low=fertig
3	SEL 4	
4	SEL 3	Binäre I/O-Ports Pins:
5	SEL 2	Adressierung der 30 fest
6	SEL 1	gespeicherten Sätze.
7	SEL 0	
8	GND	Masse

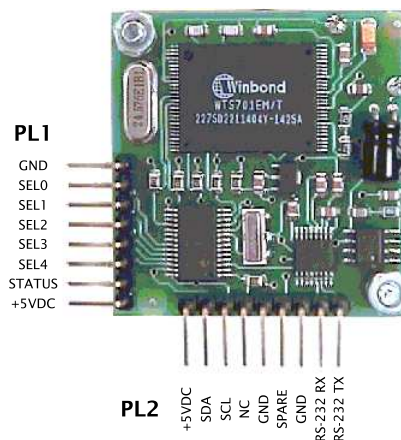


Tabelle 1: Anschlußbelegung SP03

SDA SCL-Bits I²C benötigt eine Datenleitung SDA und eine Clock-Leitung SCL. Beide werden parallel an die entsprechenden Pins des Controller, auch I²C-Master genannt, angeschlossen.

RS-232-Bits Für die serielle Übertragung werden mindestens 3 Leitungen benötigt: TX, RX und GND. Diese Pins werden direkt mit der seriellen Schnittstelle des Computers verbunden. Die Einstellungen lauten: 8 bits, No Parity, 2 stop bits, 38400 baud.

STATUS-Bit Aktuell wird die Leitung STATUS an Pin 2 von PL2 in diesem Code (noch) nicht unterstützt. Eine Abfrage des Zustandes ist hilfreich und nützlich falls sich mehrere Controller das Sprachmodul teilen oder wenn elektromechanische Artikulationen mit der Sprachausgabe synchronisiert werden sollen.

SEL-Bits Hier liegen die 5 bits um binär die 30 gespeicherten Sätze abzurufen.

1.2 Eine Stimme für den c't-Bot!

An dieser Stelle soll auf die offizielle Projektdokumentation² verwiesen werden, die sich mit allen Hardwaredetails des Roboters beschäftigt. Für die Ansteuerung des Sprachmoduls werden nur 2 Pins am ATmega644 benötigt, eine Implementierung der I²C Schnittstelle ist bereits intern im Microprozessor bewerkstelligt.

Um diese Schnittstelle nutzen zu können, werden Pin 7 (SDA) und Pin 8 (SCL) von J8 mit den entsprechenden Pins SDA und SCL des Sprachmoduls verbunden. Die Stromversorgung erfolgt am Besten von J3. Folgendes Bild zeigt den Anschlußplan für die minimalen Anforderungen:

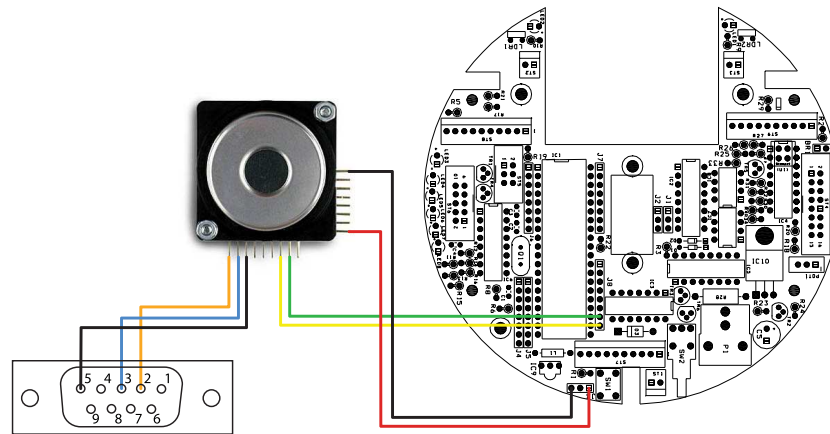


Abbildung 1: Verbindungen SP03 mit c't-Bot und PC

Verkabeln Um die beiden Platinen miteinander zu verbinden, bastelt man sich am besten aus Buchsenstecker entsprechende Kabeladapter. Dazu werden Pfostenbuchsen einzeln und als Pärchen abgetrennt, die Kabelenden um die Anschlußbeinchen gewickelt und verlötet. Schrumpfschlauch vollendet die selbstgefertigten Kabel.

Erster Test Sollten Sie Ihr SP03-Modul noch nie vorher in Betrieb genommen haben, ertönt beim Anlegen der Betriebsspannung ein Willkommensgruß von Devantech. Sie können die vordefinierten Sätze mit dem Programmiertool `sp03.exe`³ beliebig ändern, es wird aber immer Satz Nr. 1 gesprochen, wenn das Modul beim Einschalten unter Spannung gesetzt wird.

Programmierung Um die fest einprogrammierten Sätze auf dem Modul abzuspeichern, verbinden Sie das SP03 mit der seriellen Schnittstelle (RS-232) Ihres Computers wie in Abb. 1 und starten Sie das Programm `sp03.exe`

²<http://www.heise.de/ct/projekte/machmit/ctbot/wiki/FirstSteps>

³<http://www.robot-electronics.co.uk/files/sp03.exe>

2 Software

Die hier beschriebenen Dateien finden Sie in der aktuellsten Version im SVN Repository des Projekts⁴.

```
/include/sp03.h
/mcu/sp03.c
```

2.1 Das I²C Protokoll

Alle Befehle an das SP03 werden über die Ports PC0 und PC1 über die I²C Schnittstelle gesendet. Hierfür benutzt das SP03 folgendes Schema:

Nr.	Befehl
1	START BIT
2	SP03_TWI_ADDRESS
3	SP03_COMMAND_REG
4	SP03_COMMAND_NOP
5	sp03_volume
6	sp03_pitch
7	sp03_speed
8	'H' (0x48)
9	'e' (0x65)
10	'l' (0x6C)
11	'l' (0x6C)
12	'o' (0x6F)
13	SP03_COMMAND_NOP
14	STOP BIT

Tabelle 2: SP03 Befehlssequenz

Der Treiber zerlegt die einzelnen Parameter in Datenpakete, die separat in einem I²C Telegramm versendet werden. Es werden also immer 3 Pakete verschickt, bevor das SP03 loslegen kann.

START I ² C Container					
#	0xC4	I2C Adresse	#	0xC4	I2C Adresse
0	0x00	Kommandoregister	0	0x00	Kommandoregister
1	0x00	NOP - Register	1	0x00	NOP - Register
2	0x01	Volume	2	0x48	H
3	0x03	Pitch	3	0x65	e
4	0x05	Speed	4	0x6C	l
			5	0x6C	l
			6	0x6F	o
			7	0x00	NOP - Register
END I ² C Container					

Tabelle 3: Zerlegte I²C Befehlssequenz

⁴<http://www.heise.de/ct/projekte/machmit/ctbot/browser/devel/ct-Bot>

2.1.1 Die Befehlssequenzen

Gesendet werden die einzelnen Telegramme durch die Funktion `i2c_write()`. Sie kümmert sich um das Timing und die Konfiguration der Pins. Die Daten an das SP03 werden zunächst wie ein `printf()` geparsed. Zu Beginn werden die Stimmparameter mit `sp03_set_voice()` gesetzt, mit dem Aufruf von `sp03_speakf()` erfolgt dann die Übertragung der Textzeichen und anschließend des Sprech-Befehls in einem. Die komplette GUI für die SP03 Ansteuerung ergibt sich somit mit folgenden Befehlen:

```
#include "sp03.h"
...
sp03_set_voice(2,4,3); // Setzt Stimmparameter, gefolgt von
sp03_speakf("I am %d", sensError); // Spricht den Text, oder alternativ nur

sp03_say("I am %d", sensError); // Idem aber mit default Stimme

sp03_speak_phrase(0x02); // Spricht SatzID Nr.2

sp03_get_version(); // Gibt ueber LOG die Versionsnr. aus
```

Der `sp03_say()`; Befehl funktioniert wie der `sp03_speakf()`; nur muss man hier keine vorangestellten Stimmenparameter übertragen, es werden die Default-Stimmenparameter verwendet, die in der Datei `sp03.h` definiert wurden. Leider wird noch kein Status abgefragt und die maximale Textlänge wird nur rudimentär kontrolliert. Man sollte daher vermeiden, den `sp03_say()`; Befehl in Schleifen oder Interruptroutinen einzusetzen, die kürzer als die Sprechdauer des Satzes sind...

2.1.2 Parameter der Stimmerzeugung

Die Parameter in `sp03_set_voice()`; bestimmen die Lautstärke, Geschwindigkeit und Tonlage der erzeugten Stimme. Dabei können folgende Werte gesetzt werden:

Parameter	Min	Max
<code>sp03_volume</code>	0x07 leise	0x00 laut
<code>sp03_pitch</code>	0x00 hoch	0x07 tief
<code>sp03_speed</code>	0x00 langsam	0x03 schnell
<code>sp03_pre_id</code>	0x01 1	0x1E 30

Maximal können 81 Textzeichen in einer Transaktion übertragen werden. Die Anzahl der vordefinierten Sätze liegt bei 30, wobei die Zahlen 0 0x00 und 31 0x1F keinen Text ausgeben.

3 Montage

Als Anhang noch eine kleine Kopfplatte, die unter das Display montiert wird und das SP03 aufnehmen kann. Durch die Öffnung an der Seite kann man das Displaykabel und die Leitungen für das I²C und Stromversorgung führen.

3.1 Befestigungsplatte

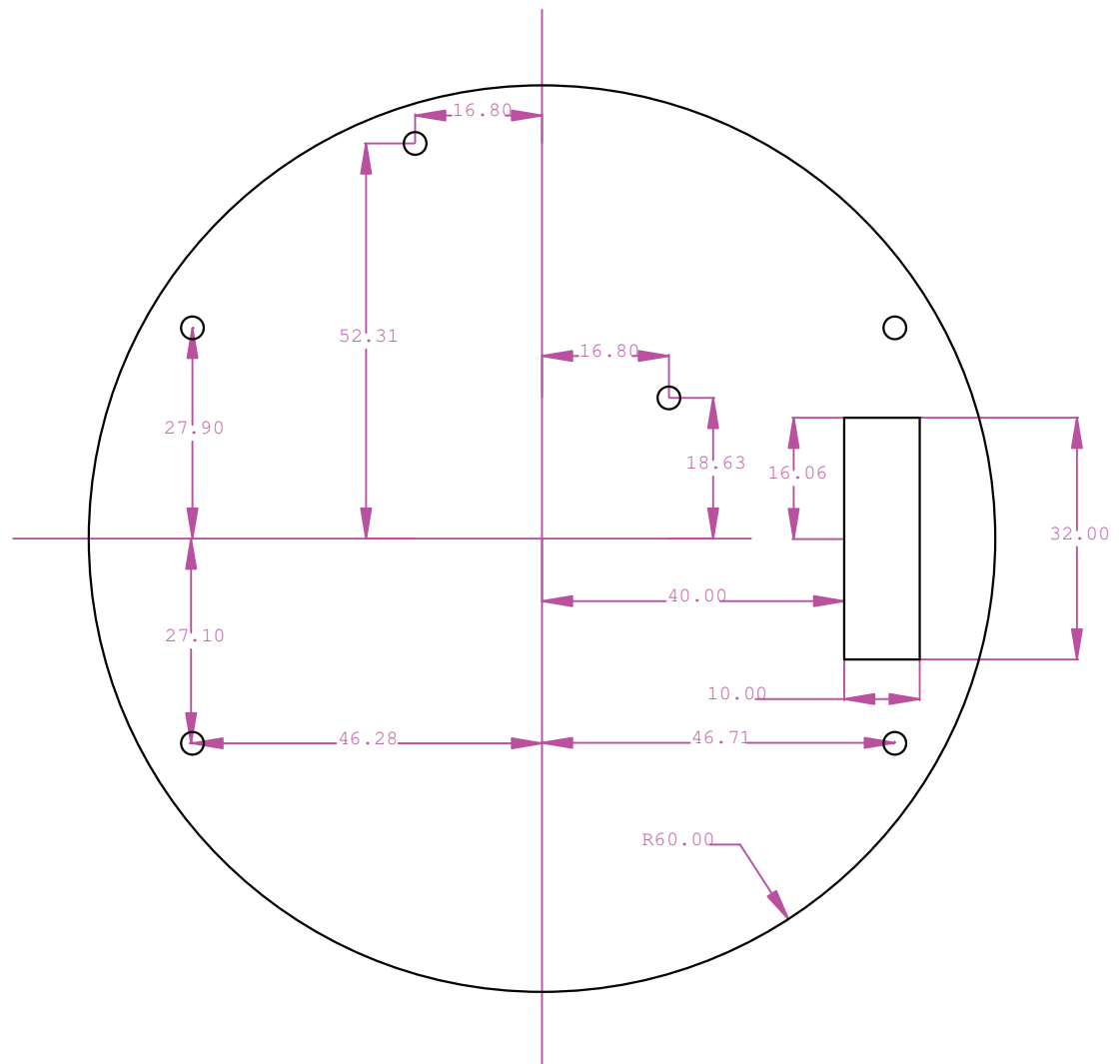


Abbildung 2: Skizze Montageplatte SP03 für c't-Bot

3.2 Bilder

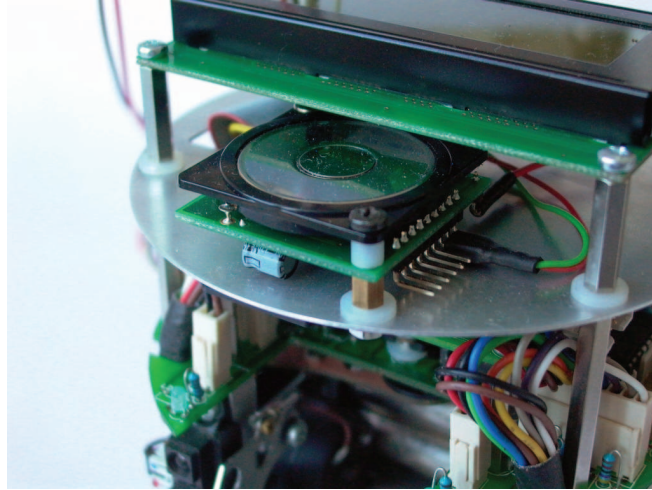


Abbildung 3: Foto SP03 am c't-Bot