

Das BLE-UART-Modul SH-HC-08

1 Überblick

Bei **Bluetooth Low Energie** (BLE) handelt es sich um einen Bluetooth-Standard, welcher auf die Funk-Übertragung kleiner Datenmengen mit möglichst wenig Energieeinsatz ausgerichtet ist. BLE ist nicht kompatibel mit dem ursprünglichen Bluetooth-Standard BR/EDR. Der Baustein SH-HC-08 ist ein BLE-Modul der Klasse 2. Er benutzt einen TI CC2541 Chip. Mit Hilfe von AT-Kommandos kann der Benutzer die Baudrate der seriellen Schnittstelle, den Namen des Moduls, das Passwort und weitere Parameter je nach Bedarf abändern (s. Abschnitt 5). Der SH-HC-08 bietet eine ideale Lösung für Entwickler an, die BLE-Technologie in ihre Entwicklungen integrieren wollen, ohne sich um die Details dieser Technologie kümmern zu müssen. (Wer sich intensiver mit BLE beschäftigen möchte, **Abb. 1** der sei auf <http://www.g-heinrichs.de/wordpress/> verwiesen.)



2 Einige Merkmale

Der SH-HC-08 benutzt das BLE-Protokoll, wie es in der Bluetooth Spezifikation V4.0 beschrieben ist. In freier Umgebung kann dieses Protokoll unter günstigen Bedingungen eine Kommunikation über bis zu 100 m ermöglichen. In den meisten Anwendungen geht es aber eher um Entfernungen von wenigen Metern.

Der SH-HC-08 wird mit einer Spannungsquelle von 3,6 V - 6 V versorgt. Die 3,3V-Signalleitungen sollen nach Angaben des Herstellers auch mit 5V-Pegeln zu betreiben sein ("the interface level 3.3V, can be directly connected the various SCM (51, AVR, PIC, ARM, MSP430, etc.), the 5V MCU also can be connected directly").

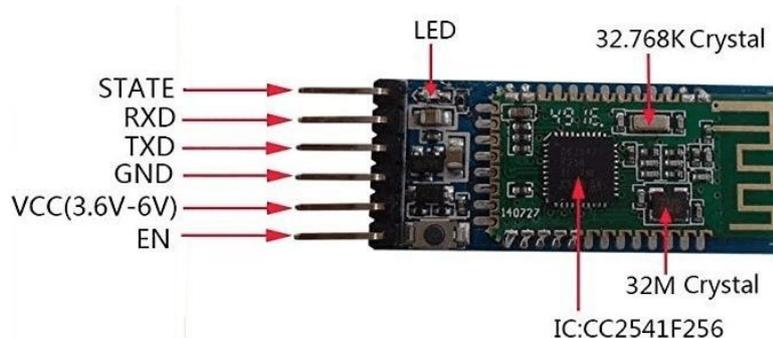


Abb. 2

Für erste Experimente empfehle ich einen USB-UART-Konverter wie z. B. den cp210x. Dieser dient nicht nur zur Spannungsversorgung des SH-HC-08; mit ihm lassen sich auch Daten zwischen dem SH-HC-08 und einem PC austauschen. Das Verbindungsschema ergibt sich aus folgender Tabelle:

cp210x	SH-HC-08
GND	GND
+ 5 V	VCC
RXD	TXD
TXD	RXD

3 Funktionsweise

Der SH-HC-08 arbeitet als **Slave**. Das bedeutet: Nach dem Einschalten sendet er zunächst automatisch sogenannte Advertising-Datenpakete aus. Diese Datenpakete beinhalten eine Reihe von Informationen, insbesondere die BLE-Adresse (manchmal auch als MAC-Adresse bezeichnet) und den Namen des Bausteins; der Name ist standardmäßig "SH-HC-08", kann aber geändert werden (vgl. den Abschnitt 5 über AT-Kommandos). In dieser Advertising-Phase blinkt die LED auf dem Modul.

Mit einem BLE-fähigen Handy können diese Advertising-Pakete empfangen und entschlüsselt werden. Dazu benötigt man eine passende App, z. B. **nRF Connect** oder **BLE Scanner**. Der Hersteller des Moduls bietet selbst auch eine einfache App mit dem Namen **DSD TECH Bluetooth** an. Diese soll nun auch im Folgenden benutzt werden. (Die genannten Apps sind sämtlich kostenlos über den Play Store zu beziehen.)

Wir starten nun die App **DSD TECH**. Die App arbeitet zunächst als **Scanner**: Sie durchforstet alle eingehenden Bluetooth-Advertising-Datenpakete und zeigt deren Namen und BLE-Adresse auf dem Bildschirm an. (s. Abb. 4). Mit der zugehörigen **Connect**-Schaltfläche sendet das Handy ein Connection Request (Anfrage zum Verbindungsaufbau) an unser BLE-Modul.

Sobald die Verbindung aufgebaut worden ist, leuchtet die LED am SH-HC-08 konstant. Von nun an sendet das Modul keine Advertising-Datenpakete mehr aus; es arbeitet jetzt als **Server**. Das Handy beendet den Scan-Vorgang und arbeitet jetzt als **Client**. In dieser Rolle kann es Daten an das Modul senden oder auch anfordern. Entscheidend ist: in diesem Connection-Zustand können nur Daten zwischen diesen beiden Partner ausgetauscht werden. Bei Bedarf kann diese Kommunikation auch mit einem Passwort abgesichert werden; standardmäßig funktioniert das Modul aber ohne Passwort (vgl. den Abschnitt 5 über AT-Kommandos).

4 Erste Experimente

Ziel dieses Abschnittes soll es sein, Daten zwischen dem PC und einem Handy auszutauschen. Dazu benötigen Sie:

- Einen PC mit einem Terminal-Programm, z. B. **DSDTECH-Tools** von der Hersteller-Firma des SH-HC-08. Dieses Programm ist kostenlos von der Seite www.dsdtech-global.com zu beziehen.
- Ein BLE-fähiges Handy mit einer der im letzten Abschnitt genannten Apps, z. B. **DSD TECH Bluetooth**.
- Einen USB-UART-Konverter wie den cp210x. Windows sollte die Treiber-Installation automatisch starten, wenn Sie das Kabel an den PC anschließen. Weitere Informationen finden Sie unter <http://www.g-heinrichs.de/attiny/USBCOMKabel.pdf>.
- Eine Verbindung zwischen dem SH-HC-08 und dem cp210x (vgl. Abschnitt 2). Hierfür bieten sich z. B. Jumper-Kabel an.

Starten Sie nun das Programm DSDTECH-Tools. Hier wählen Sie zunächst die COM-Schnittstelle für die UART aus. Dann wählen Sie für die Baudrate den Wert 9600. Anschließend öffnen Sie die Schnittstelle, indem Sie die Schaltfläche **Open** betätigen. Das Programm sollte dann im unteren Fenster die Meldung **OPEN UART Success** ausgeben (vgl. Abb. 3).

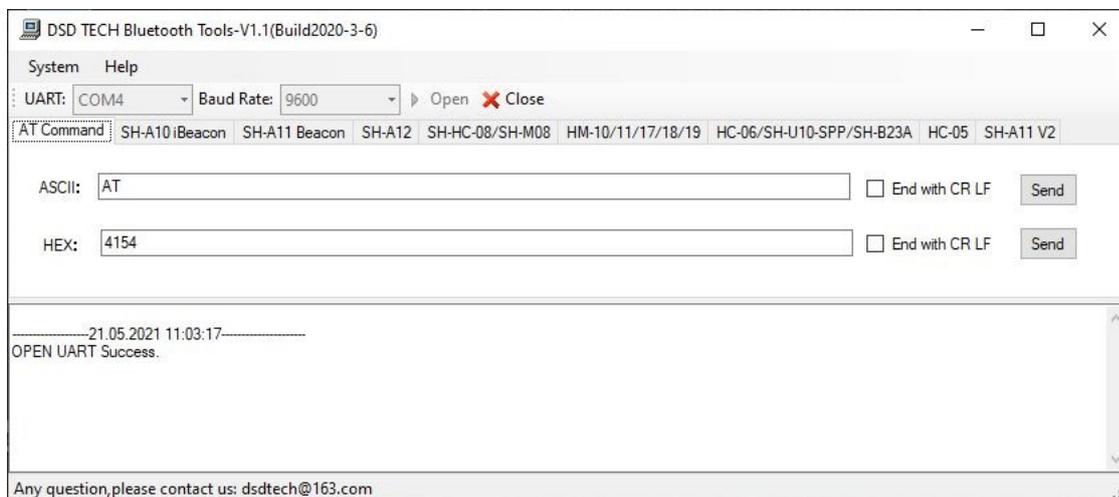


Abb. 3

Bei unserem Handy kontrollieren wir zuerst, ob Bluetooth aktiviert ist. Dann öffnen wir die App **DSD TECH Bluetooth**. Die App führt nun automatisch einen Scan-Vorgang durch. In der Regel wird sie mehrere Geräte finden; diese werden auf dem Display angezeigt (vgl. Abb. 4). Um unser BLE-Modul mit dem Handy zu verbinden, betätigen wir jetzt den Button **CONNECT**. Der Verbindungsaufbau geht sehr rasch; man erkennt ihn daran, dass die LED am SH-HC-08 nun dauerhaft leuchtet.

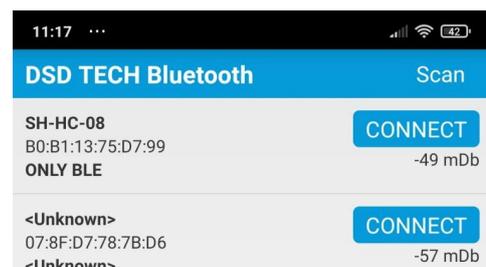


Abb. 4

Auf dem Handy wird jetzt eine einfache Kommunikationsmöglichkeit zur Verfügung gestellt (vgl. Abb. 5). Im unteren Feld geben wir z. B. den Text "Hallo SH-HC-08" ein und betätigen den **SEND**-Button. Gleich darauf finden wir im unteren Fenster unseres Terminal-Programms die Meldung:

```
-----21.05.2021 11:33:46-----
Receive:Hallo SH-HC-08
```

Jetzt versuchen wir auch einmal, vom Terminal-Programm aus eine Botschaft an das Handy zu senden, z. B. "Hallo Handy!". Diesen Text geben wir in das ASCII-Feld ein. (Zeichen, die sich ggf. schon in diesem Feld befinden, löschen wir vorher.) Anschließend betätigen wir die **Send**-Schaltfläche rechts neben dem ASCII-Feld. Nun wird der Text an das Handy gesendet; wir finden ihn in dem oberen Text-Feld (Abb. 5). Wie die Empfangs-Aktivität so wird auch die Sende-Aktivität im Terminal-Programm mit Datum und Uhrzeit dokumentiert:

```
-----21.05.2021 11:41:16-----
Send:HALLO HANDY!
```

Auffällig ist, dass **DSDTECH-Tools** offensichtlich vor dem Senden sämtliche Kleinbuchstaben in Großbuchstaben umwandelt. Dementsprechend kommen auch beim Handy nur Großbuchstaben an. Versuche mit anderen Terminals (welche diese Umwandlung nicht durchführen) zeigen, dass gesendete Kleinbuchstaben auch als Kleinbuchstaben beim Handy ankommen.

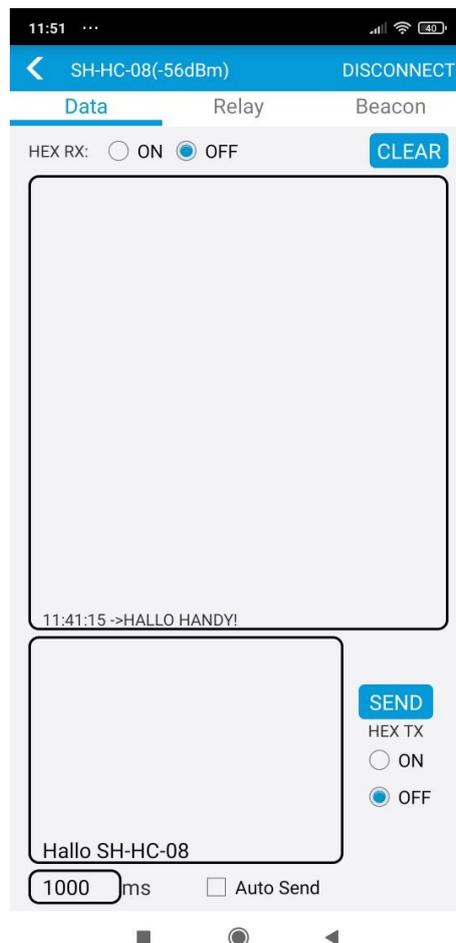


Abb. 5

Sie können mit eigenen Experimenten nun selbstständig weiteren Fragen nachgehen, z. B.:

- Werden auch deutsche Sonderzeichen übertragen?
- Können auch längere Zeichenketten übertragen werden?
- Welche Bewandnis hat es mit den Optionen HEX RX on/off bzw. HEX TX on/off bei der Handy-App?
- Wofür steht die Option Auto Send bei der Handy-App?
- Welche Bedeutung hat die Option CR LF in dem Programm **DSDTECH-Tools**?

5 AT-Kommandos

AT-Kommandos wurden ursprünglich von der Firma Hayes Microcomputer Products entwickelt, um Modems zu konfigurieren; sie stellen einen Industriestandard dar, der gerne auch auf andere

Geräte angewandt wird. Hier dienen sie dazu, einige Parameter bei unserem Modul abzufragen oder auch abzuändern.

AT-Kommandos werden in das ASCII-Feld des Terminal geschrieben und mit der Send-Schaltfläche an das Modul übertragen. **Dazu muss sich das Modul im Advertising-Zustand befinden; d. h. die LED am Modul muss blinken.** Sollte sie dauerhaft leuchten, muss die Verbindung zuerst unterbrochen werden, z. B. mit der **DISCONNECT**-Schaltfläche unserer Handy-App (oben rechts in Abb. 5).

AT-Kommandos beginnen grundsätzlich mit der Abkürzung **AT** für Attention (Achtung). Bei den AT-Kommandos für den SH-HC-08 schließt sich ein Pluszeichen an, gefolgt von einer Parameterbezeichnung. Bevor wir die wichtigsten AT-Kommandos für unser Modul vorstellen, wollen wir an einem Beispiel zeigen, wie man damit umgeht.

Beispiel: Namen des Moduls abfragen und ändern

Zum Abfragen des Modulnamens benutzen wir das Kommando

```
AT+NAME?
```

Nach dem Absenden dieses Befehls sehen wir im Protokollfenster:

```
-----21.05.2021 15:32:54-----  
Send:AT+NAME?  
-----21.05.2021 15:32:54-----  
Receive:OK+Name:SH-HC-08
```

Die Parameterbezeichnung ist hier NAME. Dass hier eine Abfrage erfolgt, wird durch das Fragezeichen deutlich gemacht.

Jetzt wollen wir den Namen abändern, z. B. in MY_BLE. Dazu schreiben wir diesen neuen Namen (ohne Fragezeichen) **direkt** hinter die Parameterbezeichnung:

```
AT+NAMEMY_BLE
```

Im Protokollfenster erscheint nach dem Senden:

```
-----21.05.2021 15:37:09-----  
Send:AT+NAMEMY_BLE  
-----21.05.2021 15:37:09-----  
Receive:OK+Set:MY_BLE
```

Sie sollten nun einmal selbst diese AT-Kommandos eingeben, anschließend die Stromversorgung zum Modul kurz unterbrechen und danach mit dem Befehl `AT+NAME?` überprüfen, ob sich das Modul den neuen Namen dauerhaft gemerkt hat. Wenn Sie jetzt mit der Handy-App einen neuen Scan durchführen, wird Ihr Modul unter diesem neuen Namen angezeigt werden.

Wir merken uns:

- Beim Abfragen eines Parameterwerts folgt auf die Parameterbezeichnung ein Fragezeichen.
- Bei einer Parameteränderung schreiben wir den neuen Wert direkt hinter die Parameterbezeichnung.

Es folgen eine Reihe von AT-Kommandos mit kurzen Erläuterungen. Weitere Informationen findet man im Datenblatt (www.dsdtech-global.com/2017/07/dsd-tech-sh-hc-08-bluetooth-40-ble.html).

Kommando	Erläuterung
AT+BAUD	Baudrate abfragen oder festlegen Parameterwerte: 4 für 9600; 5 für 19200; 6 für 38400; 7 für 57600; 8 für 115200. Standardwert: 4
AT+NAME	s. o.
AT+PASS	PIN-Code abfragen oder eingeben Parameterwerte von 000000 bis 999999. Standard: 000000
AT+TYPE	Authentifizierung abfragen oder festlegen Parameterwerte: 0: Not need PIN Code; 1: Auth not need PIN; 2: Auth with PIN Standard: 0
AT+RENEW	Werkseinstellungen wiederherstellen
AT+RESET	Reset; gleiche Funktion wie Reset-Taster
AT+ADDR	BLE-Adresse abfragen; Änderung nicht möglich!
AT+TXPW	Sendeleistung abfragen oder festlegen 0 für -23 dbm; 1 für -6 dbm; 2 für 0 dbm; 3 für 4 dbm; Standard: 2
AT+VERSION	Versionsbezeichnung abfragen

6 Messwerte vom Nano-Board per BLE an ein Handy senden

An einem Beispiel wollen wir nun zeigen, wie das SH-HC-08-Modul eingesetzt werden kann, um Messwerte per BLE von einem Nano-Board an ein Handy zu senden. Dabei wollen wir der Einfachheit halber keine echten Messwerte benutzen; statt dessen lassen wir jede 3 Sekunden eine Zufallszahl zwischen 50 und 69 erzeugen; diese wird dann über eine Software-UART des Nano an den RXD-Eingang des SH-HC-08-Moduls ausgegeben. Wir können uns z. B. vorstellen, dass es sich bei diesen Werten um Herzschlag-Raten handelt. (Das erinnert an den Heart Rate Service, wie er bei Einführungen in das Thema BLE gerne benutzt wird.)

Das SH-HC-08-Modul wird folgendermaßen an das Nano-Board angeschlossen:

```
#-----#
# Nano-Board --- SH-HC-08 #
# +5V --- VCC #
# GND --- GND #
# A3 (C.3) --- RXD #
#-----#
```

Dabei gehen wir davon aus, dass Port C.3 (Pin A3) als Ausgang der Software-UART benutzt wird (s. u.).



Abb. 6

Sobald Handy und Modul verbunden worden sind, gehen die Werte auf dem Handy ein und werden - versehen mit einem Zeitstempel - auf dem Display angezeigt.

Das Programm für den Atmega328P ist recht einfach:

```
$regfile = "m328pdef.dat"
$crystal = 16000000
$framesize = 32
$swstack = 32
$hwstack = 64

'*****
'***** Deklarationen *****

Dim Hr As Byte

'***** Initialisierung *****

'Software-UART (Ausgang an PortC.3 = Pin A3)
Ddrc.3 = 1
Open "COMC.3:9600,8,n,1" For Output As #1

'*****
'***** Hauptprogramm *****

Do
  Hr = Rnd(20) 'Zufallszahl zwischen 0 und 19
  Hr = Hr + 50
  Print #1 , Hr ;
  wait 3
Loop
```