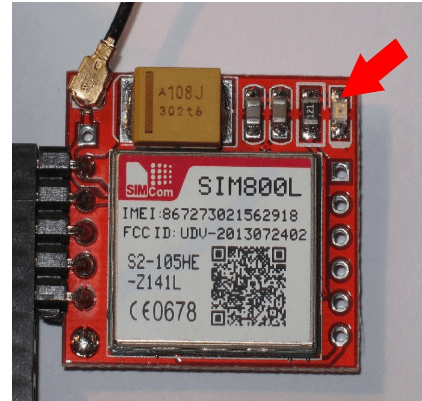

GSM-Modul mit SIM800L

Für schlappe 7 Euro (oder auch weniger) kann man inzwischen in China GSM/GPRS-Module (inkl. Porto) erwerben. Die Preise variieren allerdings sehr stark je nach benutztem IC und den zusätzlichen Komponenten auf dem Modul. Ich habe mich für eine spartanische Version entschieden (vgl. Abb. 1). Im Wesentlichen sind hier nur die wichtigsten I/O-Anschlüsse des SIM800L-Bausteins herausgeführt; daneben gibt es einen Anschluss für eine Antenne sowie eine Kontroll-LED (dazu später mehr). Die Antenne war übrigens bei meinem Modul mit im Preis inbegriffen.



Die Versorgungsspannung des SIM800L darf 4,4 V nicht überschreiten; vorgesehen sind 3,4 V - 4,4 V. Ansonsten kann der Baustein zerstört werden. Die Signalpegel dürfen aber auch 5 V betragen. **Abb.1:** Das SIM800L-Modul

Da der SIM800L im Extremfall bis zu 2A Strom benötigt, ist ein gutes geregeltes Netzgerät bzw. ein entsprechender Akku empfehlenswert. Das Modul aus Abb. 1 hat keinerlei Pegelanpassung; um es über unsere Attiny-Platine zu betreiben, versorgen wir diese nicht über die 9-Volt-Batterie inkl. Spannungsregler, sondern über ein geregeltes Netzgerät. Die entsprechenden Anschlüsse befinden sich neben dem Spannungsregler (vgl. Abb. 2; oben PLUS (rot), unten MINUS).

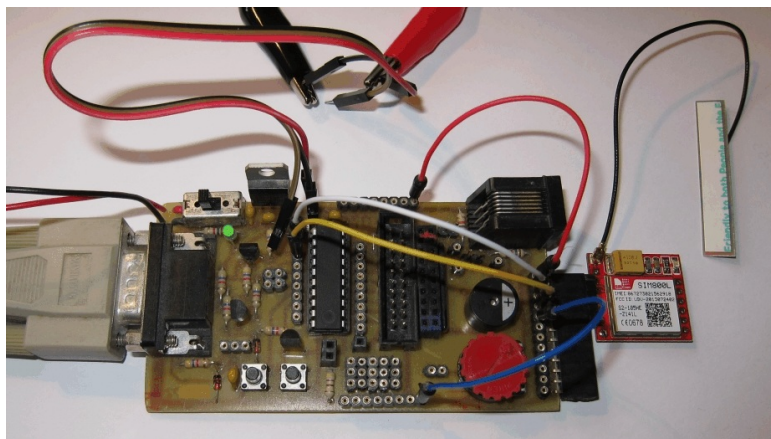


Abb. 2: Anschluss des SIM800L-Moduls an die Attiny-Platine

Bevor es losgeht, müssen wir noch unsere Mikro-Sim-Karte in die entsprechende Halterung auf der Rückseite des SIM800L hineinschieben. Achtung: Die Kontakte der SIM-Karte müssen dabei nach unten weisen und die Aussparung der SIM-Karte muss beim Einschieben auf Südosten liegen (vgl. Abb. 3).

GSM-Modul mit SIM800L

Der SIM800L wird über sogenannte AT-Kommandos, wie sie manchen vielleicht von den Modems noch bekannt sind, gesteuert und kontrolliert. Diese werden nach dem UART-Standard übertragen. Für unsere ersten Versuche benutzen wir die Attiny-Platine nur zur Spannungsversorgung und zur Anpassung der Signale von der RS232-Schnittstelle. Wer mit einem USB-UART-Konverter arbeitet, kann diesen auch direkt an das SIM800L-Modul anschließen; das SIM800L-Modul muss aber wie schon erwähnt über eine leistungsstarke elektrische Quelle gespeist werden. Nicht vergessen: Die GND-Anschlüsse von Konverter und SIM800L-Modul müssen miteinander verbunden sein.

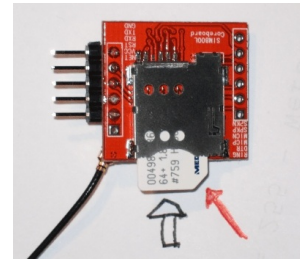


Abb: 3: Einschieben der SIM-Karte

Wenige Sekunden nachdem das Modul mit Strom versorgt worden ist, fängt die LED (s. Pfeil in Abb. 1) an zu blinken (800 ms Blinkrhythmus).

Zum Austausch der Kommandos benutzen wir ein Terminal-Programm, z. B. HTERM oder den Uploader. Wir stellen das Terminal auf 9600 Baud ein und geben das Kommando AT ein. Alle Kommandos müssen mit dem Steuerzeichen CR abgeschlossen werden. Das Modul antwortet immer mit einem Echo und - wenn alles in Ordnung ist - mit einem OK o. Ä..

Auf den folgenden Seiten möchte ich einige Anwendungen des SIM800L vorstellen

GSM-Modul mit SIM800L

Versenden einer SMS

Beginnen wir mit dem kommentierten Protokoll für das Versenden einer kurzen SMS; die Eingaben/Echos sind dabei fett geschrieben.

```
AT
OK
AT
OK
AT+CPIN="xxxx"           //PIN
OK

+CPIN: READY              //Wenn das Modul sich beim Netz eingeloggt hat,
                           dann blinkt die LED im 3000 ms - Rhythmus
Call Ready

SMS Ready
AT+CMGF=1                //SMS-Textmode aktivieren, s. u.
OK
AT+CSMP=17,167,0,0      //verschiedene Parameter setzen, s. u.
OK
AT+CMGS="0177xxxxxxx"  //Telefonnummer
> Hallo... von SIM800L  //Zu sendender Text (ohne "), vor dem CR noch
                           Ctrl-Z (dec 26) übertragen
>
>
+CMGS: 5

OK
```

Das war es: Wenig später zeigt das entsprechende Handy die gesendete Botschaft an, vgl. Abb. 4.

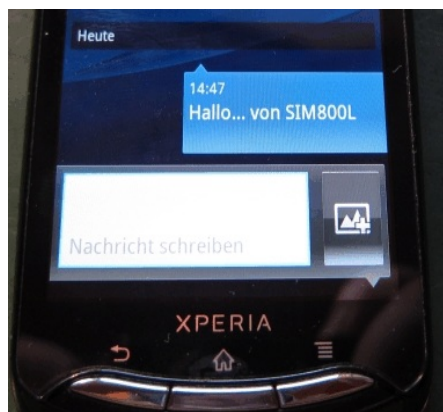


Abb. 4: SMS-Empfang

GSM-Modul mit SIM800L

Einige AT-Commands sind nicht selbsterklärend; deswegen möchte ich sie kurz erläutern (Genauere und weiterführende Informationen findet man in den Datasheets zum SIM800L und GSM03.40):

AT+CMGF=1

Select SMS Message Format

Für eine SMS stehen zwei verschiedenen Formate zur Verfügung: Text-Mode (CMGF=1) und PDU-Mode(CMGF=0). Während der Text-Mode für Texte benutzt wird, können im PDU-Mode auch binäre Daten, z. B. komprimierte Daten, versendet werden.

AT+CSMP=...

Set SMS Text Mode Parameters

Bis zu 4 Parameter vom Type Byte können hier gesetzt werden. Dabei kann die Bedeutung einzelner Bits manchmal auch davon abhängen, wie die anderen Bits gesetzt sind. Hier sollen nur die für uns entscheidenden Bits erläutert werden:

1. Parameter (fo)

fo.0=0 und fo.1=0 bedeutet SMS-DELIVER (Versand von Basis-Station an Handy); fo.0=1 und fo.1=0 bedeutet SMS-SUBMIT (Versand von Handy an Basis-Station).

Fo.3=0 und fo.4=1 aktiviert eine Validitätsperiode, wenn ein vp-Wert vorliegt, s. u.

2. Parameter (vp)

Unser Wert von 167 bedeutet eine Periode von 24 Stunden; für diesen Zeitraum wird die SMS zwischengespeichert, wenn der Adressat nicht erreichbar ist. Danach verfällt die SMS.

GSM-Modul mit SIM800L

ALARM-SMS mit SIM800L versenden

Im ersten Kapitel habe ich dargestellt, wie man mithilfe eines Terminals eine SMS über das SIM800L versenden kann. Nun möchte ich hier zeigen, wie eine SMS von unserem Attiny verschickt werden kann. Dies kann man z. B. ausnutzen, um sich eine Alarm-SMS zukommen zu lassen, wenn z. B. eine Bewegung mit einem PIR-Bewegungsmelder registriert wird. Im Folgenden soll der Alarm der Einfachheit halber durch Betätigen des Tasters Ta0 (an PortD.2) ausgelöst werden. Es dürfte keine Probleme bereiten, den Alarm durch andere Signale auszulösen.

Es ist nicht schwierig, den Attiny die benötigten AT-Commands (s.o.) per print-Befehl an das SIM800L-Modul schicken zu lassen. Hier gilt es lediglich zu beachten, dass das SIM800L-Modul am Ende des Befehls ein CR erwartet; der BASCOM-print-Befehl schließt aber mit den Steuerzeichen CR und LF ab. Mit einem Semikolon können diese beiden Steuerzeichen aber unterdrückt werden; natürlich muss dann das einzelne CR "per Hand" gesendet werden. Der AT-Command sieht dann z. B. so aus:

```
print "AT"; chr(13);
```

Etwas schwieriger fällt die Aufgabe, die vom SIM800L gegebenen Antworten zu analysieren. Diese Analysen sind erforderlich, weil sie zeigen, wann und ob der SIM800L einen Befehl erfolgreich ausgeführt hat. Dazu soll der Attiny die Antwort nach einem signifikanten Suchwort untersuchen. Wir benutzen dafür die Prozedur warte_auf_suchwort (vgl. Programm-Code unten). Diese Prozedur wartet lediglich darauf, dass die Variable "gefunden" den Wert 1 erhält. Die eigentliche Suche findet in der Interrupt-Routine "Onrxd" statt. (Da die Antwort auch CR-Zeichen enthält, kann man nicht einfach mit dem input-Befehl arbeiten!). Um auch die jeweils zuletzt empfangenen Zeichen anzeigen lassen zu können, arbeiten wir mit einem 16 Zeichen langen Ringpuffer.

Vorbereitung der Antwort-Analyse, AT-Command und Analyse sehen z.B. so aus:

```
Suchwort = "OK"  
Call Init warte  
Print "AT" ; Chr (13);  
Call Warte auf suchwort
```

Die restlichen Einzelheiten kann man den Kommentaren im folgenden Programm-Code entnehmen:

```
' Datei für Attiny-Platine von E. Eube, G. Heinrichs und U. Ihlefeldt  
' Wenn Ta0 betätigt wird, schickt der Mikrocontroller über das SIM800L-Modul  
' eine SMS ans Handy. Statt des Tasters kann auch ein anderer Sensor die SMS  
' auslösen.  
,  
' SIM800L und Display anschließen  
' Achtung: Externe Spannungsversorgung mit 4,0 V und min. 2,0 A!!!!!!!!!!!!!!!!!!!!
```

GSM-Modul mit SIM800L

```
,
' Wenn "connect?" im Display erscheint und nach einiger Zeit der Blinkrhythmus
' noch immer knapp 1 s (und nicht 3 s) ist, dann Platine ausschalten und alles
' neu starten.
,
' Die meisten Display-Texte dienen nur zur Ablaufkontrolle und können
' bei Bedarf entfernt werden.
,
-----
$regfile = "attiny2313.dat" 'Attiny2313
$crystal = 4000000 '4 MHz
$baud = 9600

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

Dim Answer As String * 16
Dim Answer_byte(17) As Byte At Answer Overlay 'Ringpuffer für Answer (wegen Speicherknappheit)
Dim Answer_index As Byte
Dim Suchwort As String * 4 'Suchwort max. 4 Zeichen
Dim Suchwort_byte(5) As Byte At Suchwort Overlay
Dim Suchwort_index As Byte
Dim Suchwort_laengel As Byte
Dim Suchende As Word
Dim E As Byte
Dim I As Byte
Dim J As Byte
Dim N As Byte
Dim Gefunden As Byte

Declare Sub Warte_auf_suchwort
Declare Sub Zeige_antwort 'zeigt den Ringpuffer an
Declare Sub Init_warte 'initialisiert die Parameter für Warte_auf_suchwort

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

Ddrb = &B11111111 'Port B als Ausgangsport
Ddrd = &B01110000 'D4, D5, D6 als Ausgang; Rest als Eingang
Portd = &B10001111 'Eingänge auf high legen
Waitms 50 'warte bis Kondensator bei Ta0 geladen

Config Lcd = 16 * 2 'LCD konfigurieren
Config Lcdpin = Pin , Db4 = Portb.0 , Db5 = Portb.1 , Db6 = Portb.2 , Db7 = Portb.3 , E = Portb.4 , Rs = Portb.6
Cursor Off

On Urxc Onrx
Enable Urxc
Enable Interrupts

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

Cls
Lcd "warte..."
Wait 5 'Jetzt sollte die LED am SIM800L blinken (jede 900 ms)
Cls
Lcd "Init..."
Wait 1

'Initialisieren, Synchronisieren
Suchwort = "OK"
Call Init_warte
Print "AT" ; Chr(13); 'nur CR , kein LF
Call Warte_auf_suchwort
Wait 1

Cls
```

GSM-Modul mit SIM800L

```
Lcd "SIM800L"
Lowerline
Lcd "found"
Wait 2

' PIN eingeben und verbinden
Suchwort = "SMS"
Call Init_warte
Print "AT+CPIN=" ; Chr(34) ; "xxxx" ; Chr(34) ; Chr(13) ; 'xxxx = eigene PIN
Call Warte_auf_suchwort

Call Zeige_antwort ' zu Testzwecken, kann zuammen mit dem folgenden Befehl entfernt werden
Do : Loop Until Pind.2 = 0 'warten bis Ta0 gedrückt

Cls
Lcd "connect?" 's. Kommentar am Anfang des Codes
Wait 5 'jetzt sollte die LED jede 3000 ms blinken

' SMS-Mode einstellen
Suchwort = "OK"
Call Init_warte
Print "AT+CMGF=1" ; Chr(13) ;
Call Warte_auf_suchwort
Cls
Lcd "SMS-Mode"
Wait 1

' Parameter setzen
Suchwort = "OK"
Call Init_warte
Print "AT+CSMP=17,167,0,0" ; Chr(13) ;
Call Warte_auf_suchwort
Cls
Lcd "ParamSet"
Wait 1

' Warten auf Alarmknopf; hier kann auch auf andere Ereignisse gewartet werden
Cls
Lcd " Ta0 ->"
Lowerline
Lcd "ALARM..."
Do : Loop Until Pind.2 = 0

' Telefonnummer angeben und ...
Suchwort = ">"
Call Init_warte
Print "AT+CMGS=" ; Chr(34) ; "01774738820" ; Chr(34) ; Chr(13) ;
Call Warte_auf_suchwort

' ... Botschaft senden
Suchwort = "OK"
Call Init_warte
Print "ALARM: Ta0 ist betätigt worden" ; Chr(26) ; Chr(13) ; 'SMS -text mit Ctrl-Z abschließen
Call Warte_auf_suchwort

Cls
Lcd "ALARM"
Lowerline
Lcd "...done"
End

' *****
' ***** Unterprogramme *****

Sub Init_warte
  Gefunden = 0
  Suchwort_index = 1
```

GSM-Modul mit SIM800L

```
Answer_index = 0
For I = 1 To 17
  Answer_byte(i) = 0
Next I
Suchwort_laengel = Len(suchwort)
Suchwort_laengel = Suchwort_laengel + 1
End Sub

Sub Warte_auf_suchwort
  Do
    Loop Until Gefunden = 1 'hier ggf. auch Timeout implementieren
  End Sub

Sub Zeige_antwort ' zu Testzwecken; zeigt den aktuellen Inhalt des Ringpuffers an
  Cls
  For I = 1 To 16
    J = I + 15
    If J > 16 Then J = J - 16
    N = Answer_byte(j)
    Lcd Chr(n)
    If I = 8 Then Lowerline
  Next I
End Sub

' *****
' *****Interruptroutinen*****

Onrx: ' über RxD empfangene Zeichen im Ringpuffer ablegen und mit Suchwort vergleichen
  E = Udr
  If E > 13 Then 'CR und NL nicht berücksichtigen
    Incr Answer_index
    Answer_byte(answer_index) = E
    If Answer_index = 16 Then Answer_index = 0 'An den Anfang des Ringregisters
    If E = Suchwort_byte(suchwort_index) Then
      Incr Suchwort_index
      If Suchwort_index = Suchwort_laengel Then
        Gefunden = 1
      End If
    Else
      Suchwort_index = 1
      If E = Suchwort_byte(suchwort_index) Then
        Incr Suchwort_index
      End If
    End If
  End If
Return

' *****
```

Das Programm kontrolliert nur nach, ob die Antworten des SIM800L jeweils ein Weiterarbeiten zulassen; dies wird auch durch kurze Meldungen auf einem LCD dokumentiert. Wer das Programm entschlacken möchte, kann diese Meldungen auf dem LCD beseitigen. Das Programm erkennt keine Fehlermeldungen; um auch solche zu berücksichtigen, bietet es sich an, bei der Prozedur `warte_auf_suchwort` ein Timeout zu implementieren.

Auch wird - nach der Anmeldung im Handy-Netz - nicht mehr weiter kontrolliert, ob die Verbindung weiterhin besteht.

GSM-Modul mit SIM800L

Mit dem SIM800L ins Internet

Oben habe ich schon dargestellt, wie man mit dem SIM800L-Modul eine SMS verschicken kann. Jetzt zeige ich, wie man mit diesem Modul ins Internet kommt. Natürlich kann auf dem Attiny2313 kein Browser arbeiten; deswegen möchte ich mich hier darauf beschränken, deutlich zu machen, wie man eine Datei über eine HTTP-Anfrage mittels AT-Commands abrufen. Der Einfachheit halber verdeutliche ich dies mithilfe eines Terminals; die Attiny-Platine wird - wie schon in dem Kapitel "Versenden einer SMS" lediglich als Pegelwandler benutzt.

Es folgt nun das kommentierte Protokoll für eine HTTP-Anfrage; die Eingaben/Echos sind dabei fett geschrieben.

```
AT
OK

//PIN eingeben
AT+CPIN="xxxx"
OK

+CPIN: READY

Call Ready

SMS Ready

//im GPRS-Netz eingebucht? (1 -> eingebucht, 0 -> nicht eingebucht)
AT+CGATT?
+CGATT: 1
OK

//Wenn nicht, dann einbuchen mit ...
AT+CGATT=1
OK

//Nutzer-Angaben (ALDI-TALK)
AT+SAPBR=3,1, "Contype", "GPRS"
OK

AT+SAPBR=3,1,"APN","internet.eplus.de"
OK

AT+SAPBR=3,1, "USER", "eplus"
OK

AT+SAPBR=3,1,"PWD","gprs"
OK

//GPRS-Kontext öffnen; LED blinkt dann sehr rasch
AT+SAPBR=1,1
OK

//Nutzer-Anfrage
AT+SAPBR=2,1
```

GSM-Modul mit SIM800L

```
+SAPBR: 1,1,"10.238.167.95" //Nutzer 1 hat lokale IP 10.238.167.95
OK

//HTTP-Dienst initialisieren
AT+HTTPIPINIT
OK

//HTTP-Parameter setzen: Nutzer-ID auf 1 setzen
AT+HTTTPARA="CID" ,1
OK

//HTTP-Parameter setzen: URL angeben
AT+HTTTPARA="URL","http://www.google.de/"
OK

//Abfrage starten (0->GET)
AT+HTTTPACTION=0
OK

+HTTTPACTION: 0,200,19048 //0->GET, 200->OK, 19048->Anzahl der Bytes
//Antwort des HTTP-Servers lesen
AT+HTTTPREAD
+HTTTPREAD: 19048
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage"
lang="de"><head><meta content="text/html; charset=UTF-8" : ...

//HTTP-Dienst beenden
AT+HTTPTERM
OK

//GPRS-Kontext schließen; LED blinkt dann wieder im 3-s-Rhythmus
AT+SAPBR=0,1
OK

//oder ...
AT+CGATT=0
OK
```

GSM-Modul mit SIM800L

SMS empfangen

Zugegeben: Die Überschrift ist eigentlich etwas irreführend; denn zum eigentlichen Empfang einer SMS gibt es nicht viel zu sagen: Sobald ein Handy, insbesondere auch unser SIM800L mit dem GSM-Netz verbunden ist, nimmt es automatisch jede SMS an, die an es gerichtet ist. Die Frage ist also eher: Wie kann man eine empfangene SMS über ein Terminal zur Anzeige bringen?

Wie üblich verbinden wir zunächst unser SIM800L-Modul mit dem GSM-Netz, indem wir unsere PIN benutzen; die Eingaben/Echos schreiben wir im Folgenden fett:

AT+CPIN="xxxx"

OK

AT +CMGF=1

OK

Eingehende SMS werden automatisch auf der SIM-Karte oder einem anderen Speicher abgelegt. Die möglichen Speicher kann man mit

AT+CPMS=?

abfragen. Unser SIM800L-Modul speichert die SMS auf der SIM-Karte. Geben wir den Befehl

AT+CPMS="SM"

ein, erhalten wir als Antwort

+CPMS: 0,30,0,30,0,30

Nur die ersten beiden Parameter sind hier relevant. Sie besagen, dass im Speicher der SIM-Karte 0 von maximal 30 Botschaften vorliegen.

Nun senden wir an unser SIM800L-Modul eine SMS; dabei beobachten wir die Anzeige am Terminal. Sobald die Botschaft empfangen worden ist, erscheint am Terminal die Meldung

+CMTI: "SM",1

Dabei zeigen die Parameter an, dass die Botschaft auf der SIM-Karte unter der Nummer 1 abgespeichert wurde. Wir fragen erneut den SIM-Karten-Speicher ab und erhalten diesmal die Antwort:

+CPMS: 1,30,1,30,1,30

GSM-Modul mit SIM800L

Es liegt also eine SMS vor. Die Botschaft mit der Speichernummer 1 können wir uns anzeigen lassen:

AT +CMGR=1

```
+CMGR: "REC READ","+49177xxxxxxx","","15/12/17,16:17:45+04"  
Test sms an sim8001  
OK
```

Der erste Parameter zeigt an, dass die SMS durch das AnzeigenLesen den Zustand "REC READ" hat; der zweite Parameter gibt die Telefonnummer des Handys an, von dem die SMS verschickt wurde; der dritte und der vierte Parameter geben das Datum und die Uhrzeit der SMS an. Als letzter Parameter wird der Text der Botschaft angegeben.

Gibt man bei dem Befehl `AT+CMGR=nr` für eine Speichernummer `nr` an, in der keine Botschaft abgespeichert worden ist, ist die Antwort lediglich OK.

Man kann auch alle vorhandenen Botschaften auflisten lassen; der Befehl dazu lautet:

AT+CMGL="ALL"

Nun werden alle SMS der Reihe nach inklusive ihrer Speicherplatznummer zur Anzeige gebracht; im Gegensatz zum `CGMR`-Kommando wird hierdurch aber nicht der Lese-Zustand (REC UNREAD / REC READ) geändert.

Irgendwann ist der Speicher der SIM-Karte voll. Spätestens dann ist es an der Zeit, einzelne Botschaften zu löschen. Mit dem Befehl

AT+CMGD=3

wird zum Beispiel die SMS mit der Nummer 3 gelöscht.

GSM-Modul mit SIM800L

Telefonieren mit dem SIM800L

Bislang haben wir den SIM800L-Baustein nur benutzt, um digitale Daten auszutauschen. Kann man aber auch mit diesem Modul wie mit einem Handy telefonieren?

Ja, man kann! Tatsächlich findet man auf der Rückseite links oben Abkürzungen, die auf diese Möglichkeit hinweisen: MICP, MICN, SPKP und SPKN. Diese Bezeichnungen machen deutlich, dass an den darunter befindlichen Kontakten ein Mikrofon (MIC) und ein kleiner Lautsprecher (SPK = Speaker) angeschlossen werden können (Abb. 5).

Bei EBAY habe ich mir kleine 8-Ohm-Lautsprecher (5 Stück für ca. 2 Euro) und Elektret-Mikrofone (10 Stück für ca. 1,60 Euro) besorgt.

Die Lautsprecher habe ich an die Pins SPKP (Speaker Positive) und SPKN (Speaker Negative) angeschlossen. Die Bezeichnungen Positive und Negative weisen auf einen Differenzverstärker hin; für das Anschließen des Speakers spielt die Polung keine Rolle. Der Hersteller des SIM800L rät zwar zu einer zusätzlichen Beschaltung mit Kondensatoren. Ein Test zeigte aber, dass diese nicht unbedingt erforderlich ist.

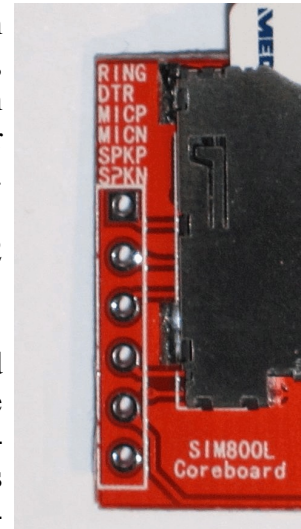


Abbildung 5

Beim Anschließen der Elektretmikrofone kann man leider nicht auf eine solche Beschaltung verzichten. Die Empfehlung des Herstellers ist folgende Schaltung:

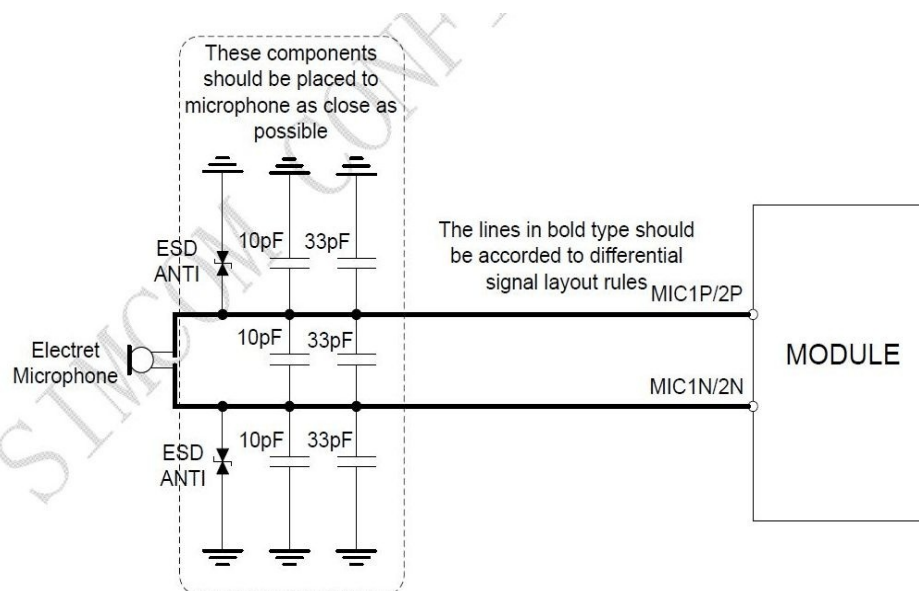


Abbildung 6: Quelle: sim800h_hardware_design_v1.00.pdf des Herstellers

GSM-Modul mit SIM800L

Der Einfachheit halber habe ich die parallel geschalteten Kondensatoren jeweils durch einen einzigen Kondensator von 15 pF ersetzt. Die Abschirmung des Kabels habe ich über einen solchen Kondensator mit der Masse des SIM800L verbunden. Auf den ESD-Schutz habe ich verzichtet. Auf diese Weise konnte eine akzeptable Tonqualität erzielt werden.

Welche Kommandos sind nun für ein Gespräch erforderlich? Dies möchte für den Fall, dass das SIM800L-Modul angerufen wird, erläutern.

Wie üblich muss das Modul im Netz mit Hilfe der PIN eingeloggt werden:

```
AT+CPIN="xxxx"
```

Wenn das Modul im Netz eingeloggt ist, dann blinkt die LED im 3000 ms - Rhythmus, und das Modul meldet:

```
OK
```

```
+CPIN: READY
```

```
Call Ready
```

```
SMS Ready
```

Damit ist das Modul in Bereitschaft versetzt. Jetzt rufen wir es an; dieser Anruf wird auf dem Terminal durch die beiden Meldungen

```
RING
```

```
+CLIP: "0177xxxxxxxxxx",129,"",0,"",0
```

angezeigt.

Diese Meldungen werden fortwährend wiederholt - ähnlich dem Klingeln eines angerufenen Telefons; sie hören auf, wenn man am Terminal den Befehl

```
ATA
```

eingibt (A = answer). Dadurch wird das Gespräch entgegengenommen; auf dem Terminal erhalten wir ein OK.

Will man das Gespräch beenden, gibt man den Befehl

```
ATH
```

ein (H = hang up = Hörer auflegen). Das Terminal meldet dann

GSM-Modul mit SIM800L

NO CARRIER

Auf genau dieselbe Weise erfolgt übrigens auch die Kommunikation zwischen (Hayes-kompatiblen) Modems.