

Messen mit nano-Board und Bluetooth-Modul HC-05

In der Einführung zum HC-05 (<http://www.forum.g-heinrichs.de/viewtopic.php?f=16&t=133>) haben wir schon ausführlich dargelegt, wie man die App "Bluetooth Electronics" mit dem Bluetooth-Modul kommunizieren lässt. Insbesondere haben Sie dort schon gelernt, wie man das Koppeln und Verbinden durchführt sowie ein Steuerelement auf ein Panel legt und damit arbeitet.

In dem Beitrag "Nano-Board mit dem Bluetooth-Modul HC-05 steuern" (vgl. auch: <http://www.forum.g-heinrichs.de/viewtopic.php?f=16&t=134>) konnten Sie außerdem u. A. erfahren

- wie man solche Steuerelemente konfiguriert,
- in welcher Form die Daten von der App gesendet werden.

Diese Kenntnisse lassen sich leicht auf das Darstellen von Messwerten erweitern. Grob gesagt geht es hierum: Der Mikrocontroller sammelt mit Hilfe von bereits eingebauten Komponenten (z. B. dem AD-Wandler) oder auch über externe Module (z. B. dem LM75-Temperatur-Modul) Messwerte und sendet diese über das Bluetooth-Modul HC-05 an das Handy. Dort können die Daten von der App "Bluetooth Electronics" entgegen genommen und auf einem Panel angezeigt werden. Dabei kann die Anzeige auf sehr unterschiedliche Weise erfolgen. Hier wollen wir uns drei dieser Möglichkeiten genauer anschauen:

Messwert als Zahl in einer "Textbox" anzeigen ("Text")

Messwert mit einem analogen Zeigerinstrument anzeigen ("Gauge")

Messreihe durch ein Diagramm darstellen ("Graph")

In allen drei Fällen wollen wir Spannungswerte am Eingang PortC.3 messen. Dazu schließen wir ein Potentiometer (ca. 10 k Ω) wie in Abb. 1 an das nano-Board an. Dabei muss der Mittenabgriff an PortC.3 angeschlossen werden.

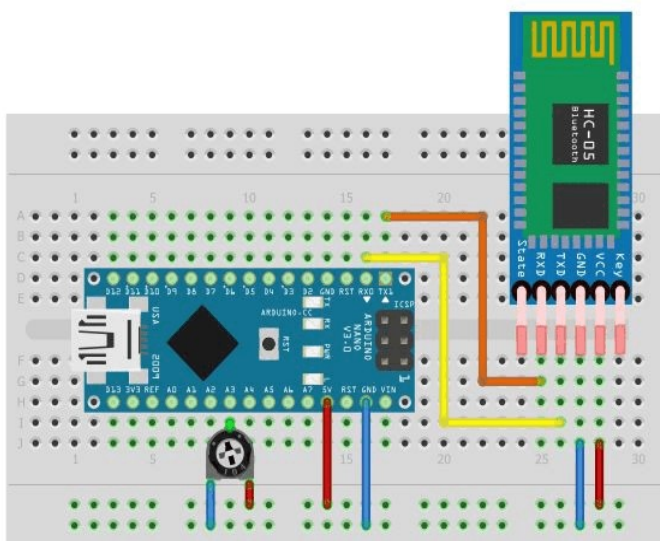


Abbildung 1

1. Messwert in einer Textbox anzeigen ("Text")

Auf einem Panel platzieren wir eine Textbox; diese finden wir in der Liste der Steuerelemente ganz oben. Standardmäßig sind solche Textboxen statisch; dies bedeutet, dass der Inhalt der Textbox beim Editieren festgelegt wird und sich dann in der Ausführungsphase nicht mehr ändert. Im Editierfeld wird darauf durch das Stichwort "static" hingewiesen.

Zur fortwährenden Anzeige der Messwerte muss diese Eigenschaft unserer Textbox abgeändert werden. Dazu betätigen wir die Schaltfläche "Edit". In dem Editierbereich löschen wir zunächst den Standard-Text ("Default Text"). Die Ausrichtung des Textes ("Justification") sowie die Farbe des Textes ("Text Color") lassen wir unverändert. Als Empfangszeichen ("Receive Character") tragen wir das Zeichen T ein. Die Eingaben bestätigen wir dann mit der OK-Schaltfläche. In dem Editierfeld sehen wir, dass der angezeigte Text nicht mehr statisch ist, sondern davon abhängig ist, was mit der Kennung "*T" als Startzeichen empfangen wird.

Wie sieht nun ein passendes Programm aus? Zunächst muss das Programm Spannungswerte mit Hilfe des eingebauten AD-Wandlers bestimmen. Zunächst konfigurieren wir den AD-Wandler:

```
Admux = &B01100011
Adcsra = &B11100010
```

Als Referenzspannung dient hier die Versorgungsspannung $V_{cc} = 4,7\text{ V}$ des nano-Boards. Damit bildet der Mikrocontroller fortwährend 8-Bit-Rohwerte für die gemessene Spannung und legt sie in dem Register ADCH ab. Zudem muss PortC.3 als Eingang konfiguriert werden:

```
Ddrc.3 = 0
```

In einer Schleife lesen wir diese im ADCH-Register bereit gestellten Rohwerte etwa jede 200 ms aus und senden sie über das Bluetooth-Modul an das Handy; zur Kontrolle lassen wir bei jeder einzelnen Übertragung die Test-LED an PortB.5 ein Mal aufblinken:

```
Do
  Rohwert = Adch
  Print "*T" ; Rohwert ; "*" ;
  Portb.5 = 1
  Waitms 100
  Portb.5 = 0
  Waitms 100
Loop
```

Zwei Bemerkungen noch zum `print`-Befehl: Dieser Befehl überträgt seine sämtlichen Parameter als Zeichenketten; insbesondere wird der Wert der Variablen Rohwert zur Übertragung in eine Zeichenkette umgewandelt. Die Übertragung hat hier also dieselbe Form, wie wir sie schon in dem Beitrag zum Steuern mit Bluetooth kennen gelernt haben. Üblicherweise schließt der `print`-Befehl die Ausgabe mit dem Senden der beiden Steuerzeichen für Carriage-Return und Line-Feed ab. Die Semikola bewirken nun, dass diese Steuerzeichen nicht gesendet werden.

Wir geben das Programm im Bascom-Compiler ein, ergänzen es um die Deklaration von der Variablen Rohwert sowie die üblichen Konfigurationen, kompilieren es und laden es auf den Mikrocontroller. **Dabei unbedingt beachten:** Während dieses Uploads sollten die RxD- und TxD-Verbindungen zwischen dem Bluetooth-Modul und dem nano-Board getrennt sein. Außerdem muss man beachten: Wenn vor dem Upload

ein Programm auf dem Mikrocontroller läuft, dass über RxD und TxD Daten austauscht, dann kann dies den Upload (der sich ja auch dieser Ein- und Ausgänge bedient) stören. Es kommt dann zu einer Fehlermeldung beim Upload, ggf. auch zu einem Absturz von Bascom. Um dies zu vermeiden, geht man beim Upload folgendermaßen vor: Wir halten den Reset-Taster beim Nano-Board gedrückt, starten dann im Bascom-Programm den Upload und lassen erst danach den Reset-Taster wieder los. Jetzt erfolgt nach wenigen Sekunden der Upload.

Wenn das Handy und Bluetooth-Modul gekoppelt und verbunden sind und unser Panel gestartet wurde, dann verändern wir die Position des Potentiometers. In der Textbox beobachten wir dann, wie die Werte sich verändern: Je größer die Spannung wird, desto größer wird der angezeigte Rohwert.

Statt des Rohwertes soll nun der wirkliche Spannungswert angezeigt werden. Dazu ändern wir den Anfang unserer Do-Loop-Schleife leicht ab:

```
Spannungswert = Adch * 0.0185                                     Print  
"*T" ; Spannungswert ; "*" ;
```

Durch die Multiplikation mit 0,0185 werden die Rohwerte aus dem Bereich [0; 255] auf den Bereich [0; 4,7] abgebildet. Bei der Deklaration von der Variablen Spannungswert benutzen wir den Typ *single*.

2. Messwert mit einem analogen Zeigerinstrument anzeigen ("Indicators: Gauge")

Das Zeigerinstrument erhält die anzuzeigenden Messwerte auf ähnliche Weise wie eine Textzeile. Als Kennbuchstaben benutzen wir hier "D". Ist der übertragene Wert 57, dann lautet das vollständiges Datenpaket "*D57*". Nur ganze Zahlen sind hier zugelassen. Aus diesem Grunde müssen die Parameter, mit denen das Zeigerinstrument konfiguriert wird, sorgfältig aufeinander abgestimmt werden, wenn es den tatsächlichen Spannungswert (und nicht nur den Rohwert) anzeigen soll. Wie immer sind die Parameter schon vorgelegt. Als Hilfestellung geben wir die erforderlichen Werte mit kurzen (deutschen) Erläuterungen in der folgenden Tabelle an:

Bezeichnung	Bedeutung	Wert
Min Text	angezeigter Wert am linken Anschlag	0
Max Text	angezeigter Wert am rechten Anschlag	5
Min value	Rohwert am linken Anschlag	0
Max value	Rohwert am rechten Anschlag	269
Number of Major Divisions	Anzahl der Hauptintervalle	5
Subdivisions	Unterteilungen für ein Hauptintervall	10
Value	Aktueller Wert	0

Der Wert für den Rohwert am rechten Anschlag wurde so gewählt, dass der Zeiger bei einem Rohwert von 255 auf der Skala möglichst genau auf 4,7 V zeigt (s.o.).

Das Zeigerinstrument sieht dann so aus wie in Abb. 2



Abbildung 2

und das zugehörige Programm so:

```
$regfile = "m328pdef.dat"
$crystal = 16000000
$framesize = 32
$swstack = 32
$hwstack = 64
$baud = 9600

!*****
!***** Deklarationen *****
!*****

Dim Rohwert As Byte

!*****
!***** Initialisierung *****
!*****

Ddrc.3 = 0
Admux = &B01100011
Adcsra = &B11100010

!*****
!***** Hauptprogramm *****
!*****

Do
  Rohwert = Adch
  Print "*D" ; Rohwert ; " ";
  Waitms 200
Loop
```

Programm 1

3. Messreihe durch ein Diagramm darstellen ("Graph")

Auf unser Panel platzieren wir ein Diagramm vom Typ "x-y 6x4". Es soll so aussehen wie in Abb. 3 und einen Graphen aus insgesamt 51 Messpunkten bilden; benachbarte Messpunkte werden dabei automatisch durch Strecken verbunden.

Im Editiermodus geben wir als Erstes "G" für den Kennbuchstaben des Diagramms ein. Weiteren Einstellungen sind in der folgenden Tabelle angegeben:

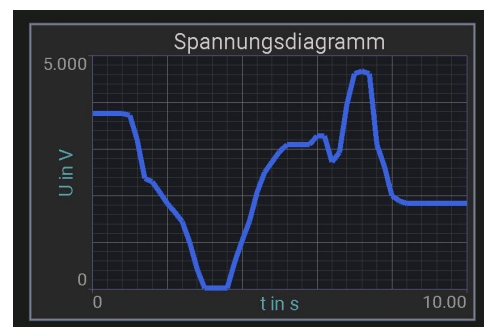


Abbildung 3

Bezeichnung	Bedeutung	Wert
Graph Title	Titel des Graphen	Spannungsdiagramm
x-Axis Title	Beschriftung der x-Achse	t in s
y-Axis Title	Beschriftung der y-Achse	U in V
Data Points	Anzahl der Messpunkte	51
Max X value	Max. Wert auf der x-Achse	10
Min X value	Min. Wert auf der x-Achse	0
Max Y value	Max. Wert auf der y-Achse	5
Min Y value	Min. Wert auf der y-Achse	0

Außerdem werden noch die folgenden Punkte aktiviert:

Draw Grid Lines (Gitterlinien zeichnen)

Draw X-Axis Max and Min Values (Maximal- und Minimalwerte für die x-Achse zeichnen)

Draw Y-Axis Max and Min Values (Maximal- und Minimalwerte für die y-Achse zeichnen)

Trace Lines: Medium

Trace Markers: None

Number of Traces: 1 (Wir wollen nur eine einzige Größe graphisch darstellen.)

Für *Trace 1* (Graph 1) wählen wir ein helles Blau; das erzeugen wir mit Hilfe der Scrollbalken für die Grundfarben.

Damit im Diagramm z. B. der Datenpunkt ($t = 2,7$ s; $U = 3,4$ V) angefügt wird, muss unser Programm die Zeichenkette `"*GX2.7Y3.4"` senden. (Diagramm-Steuerelemente können mit Dezimalzahlen umgehen - im Gegensatz zu Zeigerinstrumenten.) Für einen vollständigen Graphen müssen 51 solcher Zeichenketten gesendet werden. Im Anschluss daran soll das Programm eine Weile anhalten. Danach wird der Graph mit `"*GC"` gelöscht, und es kann ein neuer gezeichnet werden.

In dem Programm 1 ersetzen wir somit die Do-Loop-Schleife durch:

```

Do
  For I = 0 To 50                                '51 Messwerte
    Spannungswert = Adch * 0.01855
    Zeitwert = I / 5
    Print "*GX" ; Zeitwert ; "Y" ; Spannungswert ; "*";
    Waitms 200
  Next I
  Wait 5
  Print "*GC*"                                    'Graph löschen
Loop

```

Vor dem Kompilieren fügen wir noch die erforderlichen Deklarationen ein. Dann lassen wir das Programm laufen. Bei jedem Schleifendurchlauf werden über einen Zeitraum von ca. 10 s Spannungswerte gemessen und in unserem Diagramm eingetragen. Am Ende wartet das Programm 5 s; anschließend wird der Graph gelöscht und alles beginnt von vorne.