

# UC100 Anwendungen

## A. Verwendung des Motherboards MB100:

### 1. Einfache Testprogramme:

- 1.1. Ausgabe der Chiptemperatur über die serielle Schnittstelle
- 1.2. Ausgabe der Spannungswerte an den Messeingängen über die serielle Schnittstelle
- 1.3. Umwandlung einer gemessenen Spannung in eine Tonfrequenz ( V/f-Wandler )
- 1.4. Ausgabe eines Textes in A1A Morsetelegraphie
- 1.5. Ausgabe eines Textes in F2E Morsetelegraphie
- 1.6. Erzeugen einer DTMF Tonfolge
- 1.7. Ausgabe eines Textes in PSK31
- 1.8. Ausgabe eines Textes und der Chiptemperatur in AX-25 1k2 AFSK
- 1.9. Ausgabe eines Textes und der Chiptemperatur in AX-25 9k6 FSK
- 1.10. Ausgabe der Chiptemperatur über eine LCD Anzeige
- 1.11. Start des Userprogramms über externen Taster (hier: DTMF Aussendung)
- 1.12. Überwachen von Spannungswerten auf Über- oder Unterschreitung vorgegebener Limits

### 2. Verschiedene Messaufgaben

- 2.1. Messung der Temperatur über externen Sensor, Ausgabe in AX-25 1k2 AFSK
- 2.2. Messung von Spannungen höher als Vref, Ausgabe über eine LCD Anzeige
- 2.3. Überwachen von 2 Spannungen und 2 Temperaturen, DTMF Alarm bei Abweichung
- 2.4. Abrufen von Messwerten an verschiedenen Orten

### 3. Amateurfunk Anwendungen

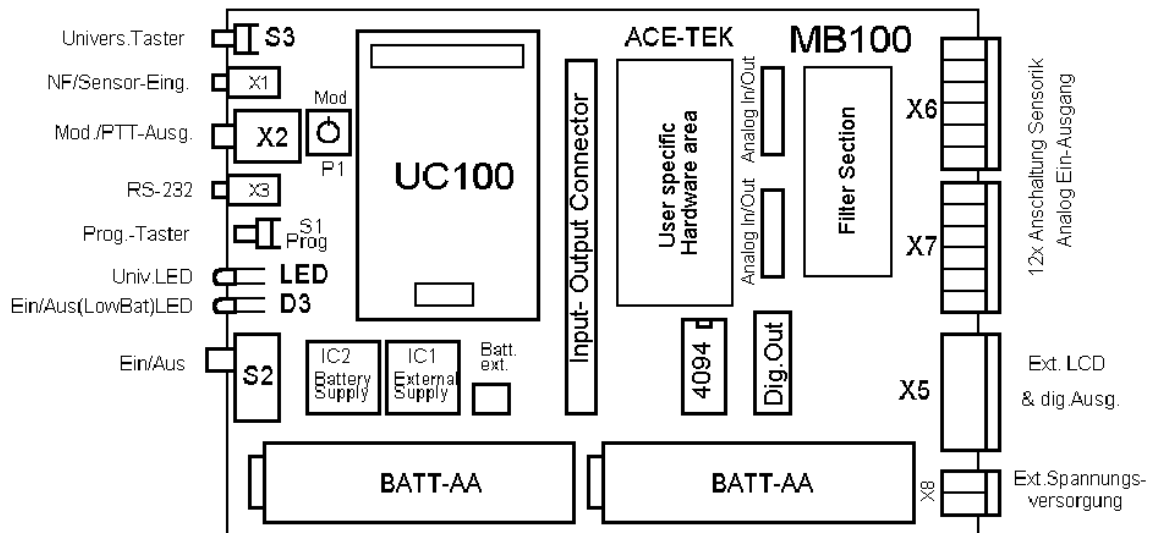
- 3.1. CW CQ-Geber mit variabler Gebegeschwindigkeit
- 3.2. APRS Geber mit universeller NMEA Datenauswertung
- 3.3. Antennenrichtungsanzeige über LCD und akustisch über Morsecode

Die hier beschriebenen Testprogramme können entweder im RAM Mode Zeile für Zeile eingegeben und durch den Befehl run gestartet werden, oder bei den etwas umfangreicheren Versionen zuerst in einen Texteditor gespeichert, und danach entweder im RAM oder BURN Mode in den UC100 geladen werden.

Sie dienen als Anregung für die vielfältigen Einsatzzwecke dieses Controllers, zum Experimentieren, oder auch als Basis für eigene Anwendungen. Anregungen für weitere Aufgaben sind jederzeit willkommen.

## A. Verwendung des Motherboards MB100:

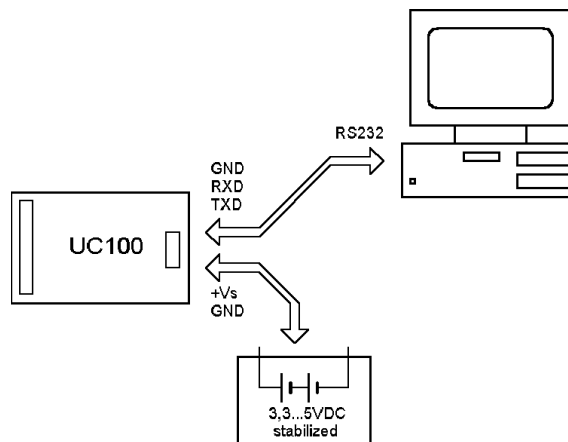
Um eine bedienerfreundliche Anschaltung des UC100 zu ermöglichen, wurde das Motherboard MB100 entworfen. Es ermöglicht den einfachen Zugang zu allen verfügbaren Schnittstellen und eine universelle Spannungsversorgung. Der UC100 wird einfach auf das Motherboard aufgesteckt, prinzipiell brauchen auch nur die Komponenten am MB100 bestückt zu werden, welche für den geplanten Anwendungsfall benötigt werden.



Das MB100 umfasst:

- Spannungsversorgung durch 2xMignonzellen (NiCd und NiMH) und aufsteck- oder lötbaren Schaltwandler (IC2)
- Spannungsversorgung durch Batt.-Pack mit höherer Spannung und linearer Stabilisierung (IC2 direkt bestückt)
- Spannungsversorgung durch externe Quelle mit 3,3 ... 5VDC
- Spannungsversorgung durch externe Quelle mit > 5,5VDC und linearer Stabilisierung (IC1 direkt bestückt)
- Spannungsversorgung durch externe Quelle mit > 5,5VDC und aufsteck- oder lötbaren Schaltwandler (IC1)
- Befestigungsmöglichkeit über 4 Bohrungen
- Dimensionen angepasst an Kunststoff Standardgehäuse LH55 von Pac-Tec (2 Bauhöhen verfügbar)
- Ein/Aus-Schiebeschalter für Batterie- und externe Versorgung
- RS-232 Anschluss über 2,5mm Stereoklinkenbuchse
- Senderansteuerung über 3,5mm Stereoklinkenbuchse (Mod., PTT und GND)
- PTT Signal entweder direkt durch UC100 FET oder über Widerstand an Mod.-Leitung (div. Handgeräte)
- Modulationssignalpegel über Poti und Vorwiderstand einstellbar
- NF Eingang über 2,5mm Klinkenbuchse (z.B. für Tonauswertung), mit Spannungsteiler und b.B. DC-Trennung
- Anschaltmöglichkeit eines Sensors über 2,5mm Klinkenbuchse (auch aktiv mit Spannungsversorgung)
- Programmierertaster (leicht rückversetzt hinter eventuelle Frontplatte)
- Universelle LED parallel zur UC100 LED
- Weitere LED als Ein/Aus-Kontrolle oder Low-Batt.-Anzeige (auch SDN gesteuert möglich)
- Zusätzlicher Taster zur freien Verfügung (z.B. zum manuellen Starten des Userprogramms)
- Bestückung des ext. Schieberegisters (4094) möglich, für LCD Ansteuerung oder Schaltsignalausgabe
- Frei verfügbares Lochrasterfeld zur Bestückung von anwenderspezifischer Hardware
- Stiftleisten mit allen verfügbaren Signalen und Spannungen zum Aufstecken eigener Interfaceplatinen
- HF Filterbereich für alle Sensorsignale und Messspannungen
- Anschlussmöglichkeit für Sensorik und Versorgung über steck- oder lötbare Schraubklemmen
- 16-polige Stiftleiste mit direktem LCD Pinning und Kontrasteinstellung durch Spannungsteiler

## 1.1. Ausgabe der Chiptemperatur über die serielle Schnittstelle



Bei dieser einfachsten Anwendung des UC100 wird die aktuelle Chiptemperatur des Microcontrollers im Sekundentakt zum PC übertragen. Die Programmierung hierfür ist äußerst einfach, da mit der bereits in der Firmware berechneten Variablen „temp“ gearbeitet werden kann. Eine sonst im Userprogramm durchzuführende Umrechnung aus dem gemessenen adc-Wert (0...4095) in einen korrekten Temperaturwert ist nicht notwendig.

### User-Programm 1:

( Anzeigeformat : **XX,X** )

```
pr %d41:temp %n  
sls 1  
loop  
run
```

Ausgabe des Temperaturwertes über die serielle Schnittstelle, danach neue Zeile  
1 Sekunde Wartezeit im Sleep-Mode  
zurück zum Programmstart  
startet das Programm

### User-Programm 2:

( Anzeigeformat : **XX,X Grad C** )

```
pr %d41:temp Grad C%n  
sls 1  
loop  
run
```

Ausgabe des Temperaturwertes über die serielle Schnittstelle, danach neue Zeile  
1 Sekunde Wartezeit im Sleep-Mode  
zurück zum Programmstart  
startet das Programm

### User-Programm 3:

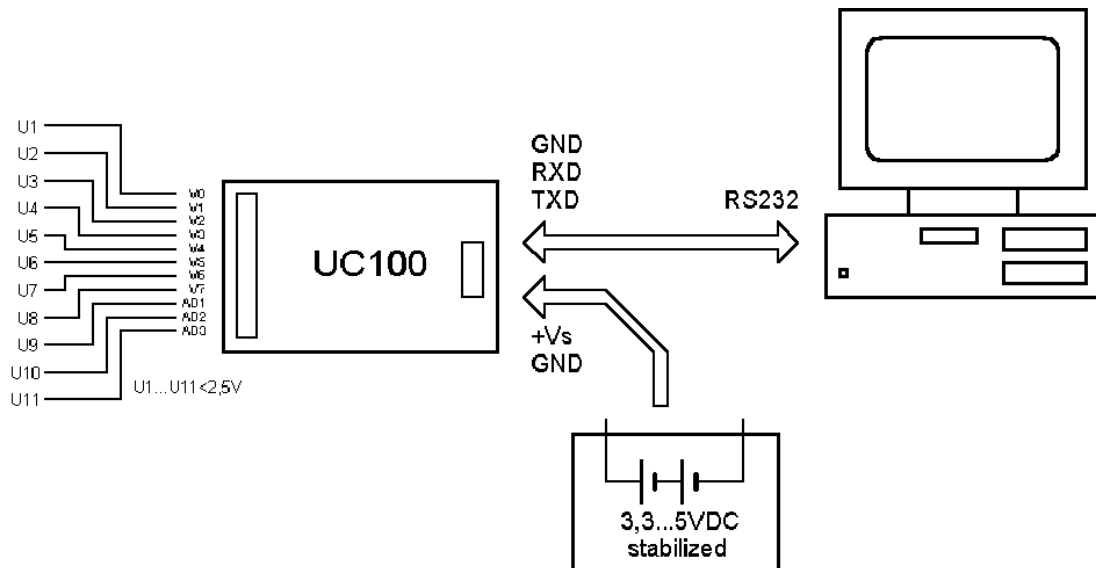
( Anzeigeformat: **Temp XX,X Grad C** )

```
pr Temp %d41:temp Grad C%n  
sls 1  
loop  
run
```

Ausgabe des Temperaturwertes über die serielle Schnittstelle, danach neue Zeile  
1 Sekunde Wartezeit im Sleep-Mode  
zurück zum Programmstart  
startet das Programm

## 1.2. Abgabe der Spannungswerte an den Messeingängen über die serielle Schnittstelle

Dieses Programm schreibt die Momentanwerte an den Ports 0...7 (ADC0 über internen Multiplexer), Port 8 (Chiptemperatur) und Ports 17...18 (ADC1..3 Direkteingänge) über die serielle Schnittstelle auf den Bildschirm.  
Bei Eingabe des Programms Zeile für Zeile braucht der Befehl **ram&** nicht eingegeben zu werden (nur für Download).



Userprogramm	Erklärung
<b>ram&amp;</b>	für Download aus Datei in das RAM ohne Echo
<b>sls 2</b>	2 Sekunden Wartezeit im Sleep-Mode
<b>pr %n %n</b>	2 mal neue Zeile
<b>port=0</b>	mit Port 0 beginnen
<b>1:</b>	Label 1
<b>pr %d3:port :%d4:adc</b>	Portnummer schreiben (0...8), danach „:“ und Messwert am aktuellen Port
<b>port=port+1</b>	Portnummer um 1 erhöhen
<b>if port=9</b>	wenn Portnummer 9 erreicht wird
<b>goto 2</b>	Sprung zu Label 2
<b>goto 1</b>	Sprung zu Label 1
<b>2:</b>	Label 2
<b>pr %n</b>	neue Zeile
<b>port=17</b>	Port 17 anwählen
<b>3:</b>	Label 3
<b>pr %d3:port :%d4:adc</b>	Portnummer schreiben (17...19), danach „:“ und Messwert am aktuellen Port
<b>port=port+1</b>	Portnummer um 1 erhöhen
<b>if port=20</b>	Wenn Portnummer 20 erreicht wird
<b>loop</b>	zurück zum Programmstart ( zu Befehl sls 2 )
<b>goto 3</b>	Sprung zu Label 3
<b>run</b>	startet das Programm

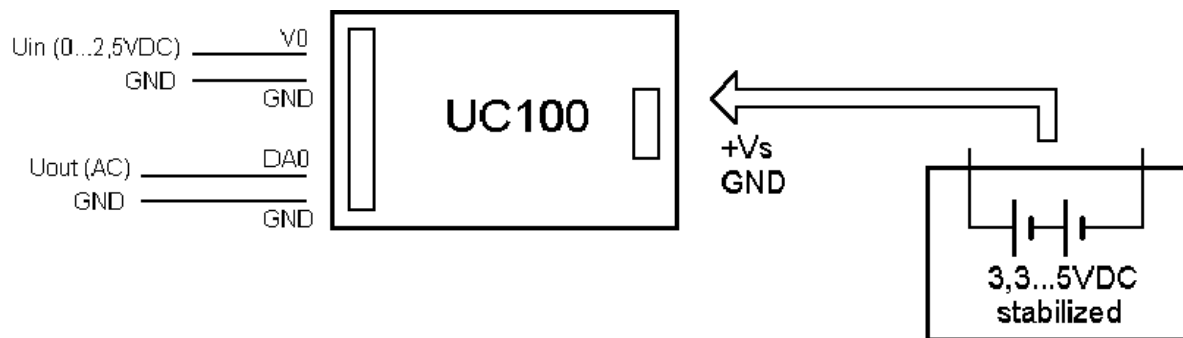
### 1.3. Umwandlung einer gemessenen Spannung in eine Tonfrequenz ( V/f-Wandler )

Dieses Programm wandelt die Gleichspannung an Port 0 (0..2,5V) in eine entsprechende Wechselfspannung an DA0 um. Null Volt Eingangsspannung ergeben auch Null Hz am Ausgang, 2,5V am Port 0 erzeugen eine NF an DA0, die durch bestimmte Parameter vorgegeben ist.

Durch mathematische Bearbeitung der Variablen **adc** kann diese NF beliebig verändert oder mit einem bestimmten Offsetwert versehen werden. In diesem Beispiel wird der ADC-Wert mit der Zahl 2 multipliziert und dadurch auch bei kleineren Messwerten eine brauchbare Ausgangsfrequenz erzielt. Um bei Null Volt Messspannung bereits eine Grundwechselfspannung zu erhalten, muss zu **adc** eine Zahl addiert werden.

Als Messeingang wird hier Ausgangswert Port 0 verwendet, kann jederzeit mit Befehl **port** am Programmanfang verändert werden.

Es wurde, um ungestörte Tests zu ermöglichen, eine permanente Programmschleife verwendet, der Ausstieg aus dem Programm erfolgt durch Drücken der INT-Taste (S1). **Wichtig:** Befehl **tone** nicht in die Schleife mit einbeziehen.



#### Userprogramm

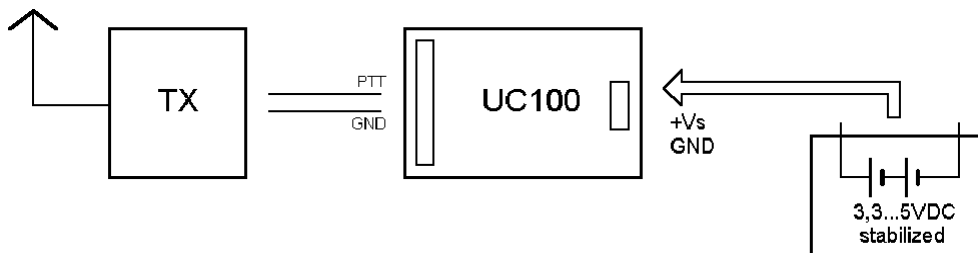
```
tone
1:
f0=adc*2
goto 1
run
```

#### Erklärung

```
Befehl zur Tonwandlung
Label 1
Doppelter adc-Wert wird der Variablen f0 (Tonausgabe) zugewiesen
Sprung zu Label 1
startet das Programm
```

#### 1.4. Ausgabe eines Textes in A1A Morsetelegraphie

Um einen Sender in A1A zu tasten, braucht nur die Leitung PTT des UC100 dazu verwendet werden, entweder ein Relais zu schalten (universelle Lösung, allerdings mit begrenzter Tastgeschwindigkeit), oder, falls der Tasteingang des verwendeten Senders gegen GND geschaltet werden kann (max. 24VDC und 100mA), direkt durch den PTT Schalt-FET des UC100. Im Beispielprogramm wird das Wort „Test“ ausgegeben. Es wird dabei die Ausgangseinstellung für die Gebegeschwindigkeit verwendet (ca. 60 BpM). Änderungen sind über die Befehle **init** und **spd** möglich. Die Ausgabe wird automatisch alle 2 Sekunden wiederholt.



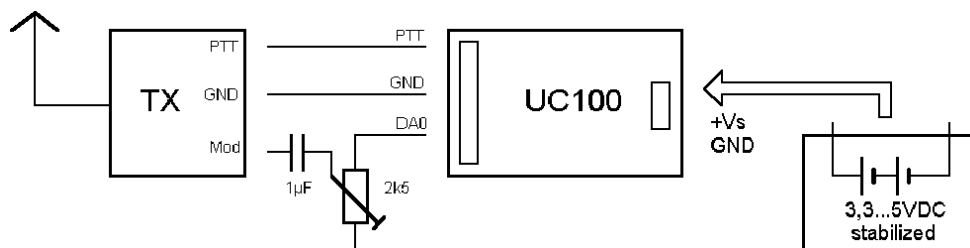
#### Userprogramm

#### Erklärung

<p><b>cwa1 test</b>  <b>sls 2</b>  <b>loop</b>  run</p>	<p>Befehl zur Ausgabe von „test“ in Morsetelegraphie  2 Sekunden Wartezeit im Sleep-Mode  zurück zum Programmstart  startet das Programm</p>
---	--

#### 1.5. Ausgabe eines Textes in F2E Morsetelegraphie

Bei dieser Art der Ausgabe wird zu Beginn der Aussendung der Sender eingeschaltet (PTT), danach erfolgt die CW Sendung durch Tonmodulation. Tonhöhe und Gebegeschwindigkeit haben die Ausgangswerte, Veränderungen sind möglich über die Befehle **init**, **spd** und **f0** (siehe Programmierbefehle). Durch den sehr schnellen Start der Sendung nach dem PTT Signal ist zu empfehlen, als erstes Zeichen ein oder mehrere Leerzeichen zu programmieren.



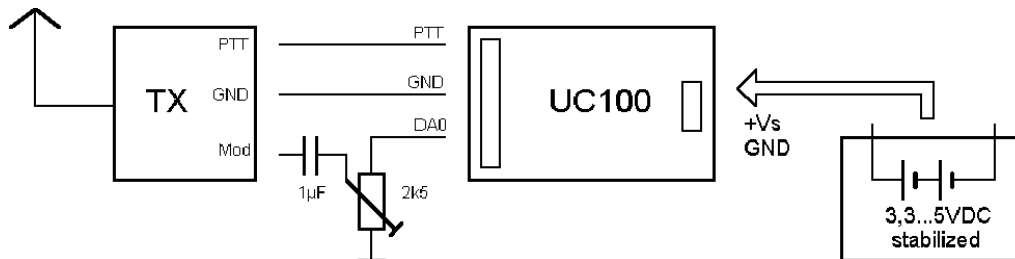
#### Userprogramm

#### Erklärung

<p><b>cwf2 test de aa1bb</b>  <b>sls 5</b>  <b>loop</b>  run</p>	<p>Befehl zur Ausgabe von „test de aa1bb“ in tonmodulierter Morsetelegraphie  5 Sekunden Wartezeit im Sleep-Mode  zurück zum Programmstart  startet das Programm</p>
--	--

## 1.6. Erzeugen einer DTMF Tonfolge

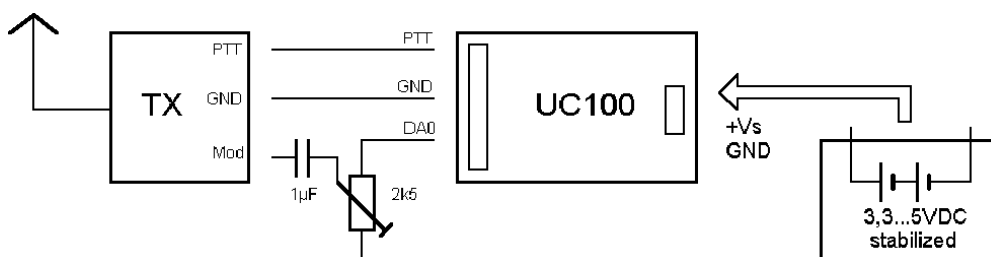
Der UC100 generiert DTMF Töne automatisch und gibt diese über den DAC als sinusähnliches Signal aus. Es braucht lediglich die gewünschte Zahlenfolge nach dem Betriebsartenbefehl eingegeben zu werden. Während der Aussendung wird das PTT Signal eingeschaltet, um einen eventuell angeschlossenen Sender zu tasten. Bei diesem Userprogramm wird eine bestimmte DTMF Tonfolge 2 mal mit einer Pause von 2 Sekunden ausgesendet (Programmstart durch Eingabe von **run** ).



Userprogramm	Erklärung
<b>dtmf 1234567890</b>	Befehl zur Ausgabe von „1234567890“ in DTMF Tönen
<b>sls 2</b>	2 Sekunden Wartezeit im Sleep-Mode
<b>dtmf 1234567890</b>	Befehl zur Ausgabe von „1234567890“ in DTMF Tönen
<b>run</b>	startet das Programm

## 1.7. Ausgabe eines Textes in PSK31

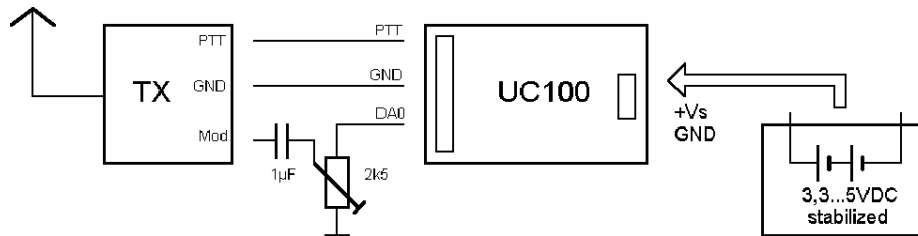
Um eine Aussendung in PSK31 durchzuführen, braucht nur der gewünschte Text oder die Variable nach dem Betriebsartbefehl eingegeben zu werden. Während der Sendung bleibt das PTT Signal geschaltet. Bei diesem Beispiel werden 2 Texte nacheinander ausgesendet, mit 5 Sekunden Pause dazwischen (Programmstart durch Eingabe von **run** ).



Userprogramm	Erklärung
<b>psk test de aa1bb</b>	Befehl zur Ausgabe von „test de aa1bb“ in PSK31
<b>sls 5</b>	5 Sekunden Wartezeit im Sleep-Mode
<b>psk loc is fn11aa</b>	Befehl zur Ausgabe von „loc is fn11aa“ in PSK31
<b>run</b>	startet das Programm

## 1.8. Ausgabe eines Textes und der Chiptemperatur in AX-25 1k2 AFSK

Dieses Programm sendet die aktuelle Chiptemperatur als Kombination aus dem Fixtext *Temperatur* gefolgt von dem aktuellen Wert im Format xx,x (d41). Nach einer Pause von 2 Sekunden im Sleep-Mode beginnt der Zyklus wieder von vorne.



### Userprogramm

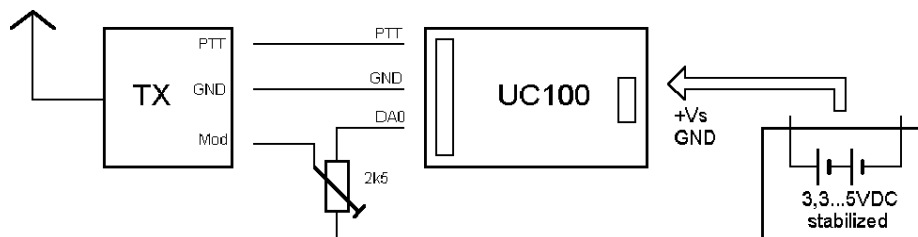
```
h12 :EMPF SEND:Temperatur %d41:temp
sls 2
loop
run
```

### Erklärung

Ausgabebefehl, RX/TX – Adresse, Fixtext und Temp-Variable  
2 Sekunden Wartezeit im Sleep-Mode  
Rücksprung zum Programmbeginn  
startet das Programm

## 1.9. Ausgabe eines Textes in AX-25 9k6 FSK

AX-25 Betrieb mit 9k6 unterscheidet sich grundsätzlich nur durch den geänderten Betriebsartbefehl **h96**. Allerdings muss bei der Ansteuerung des Senders auf die Signaleinkopplung geachtet werden. Die Signalfanken des Modulationssignals sollten so wenig wie möglich beeinträchtigt werden. Der Modulationspegel bewegt sich zwischen 0 und 2,5V und kann durch ein Potentiometer abgeschwächt oder aber auch direkt verwendet werden. User-Testprogramm wie bei 1k2 (Punkt 1.8.)



### Userprogramm

```
h96 :EMPF SEND:Temperatur %d41:temp
sls 2
loop
run
```

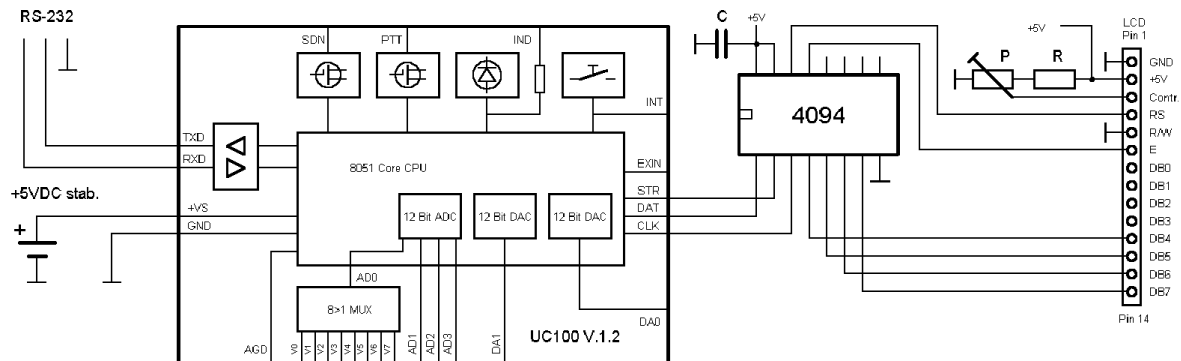
### Erklärung

Ausgabebefehl, RX/TX – Adresse, Fixtext und Temp-Variable  
2 Sekunden Wartezeit im Sleep-Mode  
Rücksprung zum Programmbeginn  
startet das Programm



## 1.10. Ausgabe der Chiptemperatur über eine LCD Anzeige

Durch Einsatz eines externen Schieberegisters ist es möglich, Informationen auch auf Standard LCDs auszugeben (siehe Handbuch „Anschaltung eines LC Displays“). Das folgende Userprogramm schreibt die aktuelle Chiptemperatur alle Sekunden auf ein 2 zeiliges Display mit mindestens je 16 Zeichen (bis 2x40 möglich).



UC100 Ansteuerung eines LC Displays

### Userprogramm

```

lcd %0 %1
1:
slms 50
lcd %128 Chiptemperatur
slms 50
lcd %194 %d31:temp Grad C
sls 1
goto 1
run
    
```

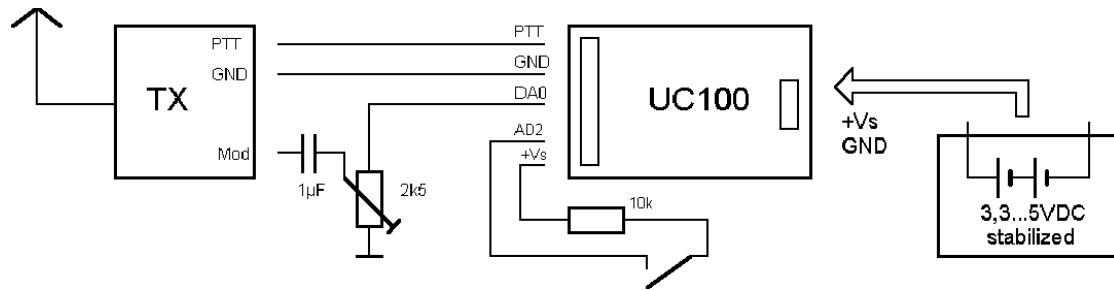
### Erklärung

Steuerbefehl „Löschen der Anzeige“  
 Label 1  
 50ms Wartezeit im Sleep-Mode  
 schreibt das Wort „Chiptemperatur“ an den Beginn der ersten Zeile  
 50ms Wartezeit im Sleep-Mode  
 schreibt die Chiptemperatur an die 3. Stelle der 2. Zeile, gefolgt von „Grad C“  
 1s Wartezeit im Sleep-Mode  
 Sprung zu Label 1  
 startet das Programm

## 1.11 Start des Userprogramms über externen Taster (hier: DTMF Aussendung)

Sollte die Notwendigkeit bestehen, das Userprogramm über einen externen Taster zu starten, so ist das durch den Programmbefehl **wait** sehr einfach möglich. Dazu beschaltet man den UC100 wie unten dargestellt. Vorteilhaft ist die Verwendung eines der 3 direkt anzusprechenden Ports (AD1...AD3 = Port17...19), da an diesen Eingängen bereits Ableitwiderstände von je 1 MOhm vorgesehen sind. Prinzipiell ist jeder der 11 Analogports dafür geeignet (siehe Programmierbefehl **wait** im Handbuch).

Bei erkennen des vorgesehenen Steuersignals am jeweiligen Port startet ein Durchlauf des Userprogramms, danach wird wieder auf das Steuersignal gewartet.



### Userprogramm

```
wait 18
dtmf 001123456789
sls 1
loop
run
```

### Erklärung

Warten auf positive Flanke an Port 18 (AD2)  
 Aussenden der DTMF Tonfolge 001123456789  
 1s Wartezeit im Sleep-Mode  
 Rücksprung zum Programmbeginn  
 startet das Programm

## 1.12. Überwachen von Spannungswerten auf Über- oder Unterschreitung vorgegebener Limits

Die Programmiersprache des UC100 erlaubt es, in sehr einfacher Weise Messwerte auf Einhaltung bestimmter Grenzwerte zu überprüfen. Bei Abweichungen können verschiedenste Aktionen durchgeführt werden um die Über- oder Unterschreitung zu signalisieren. Es können dabei die ADC Werte oder aber auch bereits bearbeitete Variable überwacht werden. Im Folgenden sind verschiedene Möglichkeiten dargestellt, solche Überwachungsaufgaben zu realisieren.

### 1.12.1 Kontrolle der Chiptemperatur

Der Controller soll die Chiptemperatur im Sekundentakt auf Überschreitung überwachen und bei Temperaturen über 35 Grad C einen Beep ausgeben (an DA0).

Userprogramm	Erklärung
<b>sls 1</b>	1s Wartezeit im Sleep-Mode
<b>if temp&gt;350</b>	Falls die Chiptemperatur 35,0 Grad C übersteigt
<b>goto 1</b>	Sprung zu Label 1
<b>loop</b>	Rücksprung zum Programmbeginn
<b>1:</b>	Label 1
<b>beep 1000</b>	Beep mit 1kHz ausführen
<b>loop</b>	Rücksprung zum Programmbeginn
<b>run</b>	startet das Programm

### 1.12.2 Spannungsüberwachung innerhalb eines Bereiches

Eine Spannung von 1,25V an Port 0 (V0) soll auf + und - 50% (+-0,625V) überwacht werden. Entsprechende Warnungen sollen bei Bedarf an die serielle Schnittstelle gegeben werden. Im Normalfall wird die Messspannung alle 3 Sekunden ausgegeben. Keine Umrechnung der gemessenen Werte (Messwerte zwischen 0 und 4095 für 0 bis 2,5VDC).

Userprogramm	Erklärung
<b>sls 3</b>	3s Wartezeit im Sleep-Mode
<b>pr Vm=%d4:adc %n</b>	Sende „Vm=" und danach den ADC Wert an die serielle Schnittstelle, neue Zeile
<b>if adc&lt;1024</b>	Falls die Spannung 0,625V unterschreitet
<b>goto 1</b>	Sprung zu Label 1
<b>if adc&gt;3072</b>	Falls die Spannung 1,875V überschreitet
<b>goto 2</b>	Sprung zu Label 2
<b>loop</b>	Rücksprung zum Programmbeginn
<b>1:</b>	Label 1
<b>pr Unterspannung! %n</b>	Sende das Wort „Unterspannung!“ an die serielle Schnittstelle, neue Zeile
<b>loop</b>	Rücksprung zum Programmbeginn
<b>2:</b>	Label 2
<b>pr Ueberspannung! %n</b>	Sende das Wort „Ueberspannung!“ an die serielle Schnittstelle, neue Zeile
<b>loop</b>	Rücksprung zum Programmbeginn

## 2.1. Messung der Temperatur über externen Sensor, Ausgabe in AX-25 1k2 AFSK

Prinzipiell kann zur Temperaturmessung jeder Sensor mit linearer Kennlinie verwendet werden. Daraus gewonnene Messspannungen im Bereich zwischen 0 und 2,5V lassen sich durch die vielfältigen mathematischen Möglichkeiten des UC100 sehr einfach in die entsprechenden Temperaturwerte umrechnen.

In diesem Beispiel wird als Temperatursensor ein LM335 von National Semiconductors eingesetzt. Mit einem Betriebsstrom von ca. 1mA liefert dieser Sensor eine Ausgangsspannung von 2,98V bei 25°C. Da diese Spannung bereits über der oberen Messgrenze des UC100 liegt, muss sie in geeigneter Weise geteilt werden. Um die höchste Genauigkeit zu erreichen, sollte dazu eine aktive Variante unter Einsatz eines Dual Rail to Rail OPV verwendet werden. Eine stabilisierte Betriebsspannung von 5VDC vereinfacht die Messschaltung, da es dabei nicht unbedingt notwendig ist, zur Versorgung des Sensors eine Konstantstromquelle einzusetzen. Bei schwankender Betriebsspannung oder längeren Zuleitungen sollte eine Stromquelle von ca. 1mA verwendet werden (anstatt R1).

Der Sensorbaustein LM335 ändert seine Ausgangsspannung mit 10mV / °C. Durch ein Teilungsverhältnis von 0,6 (z.B. 22k/33k) wird nicht nur die Messspannung in korrekte Bereiche geteilt, sondern zusätzlich eine Auflösung von 10Bit / °C erreicht. Dadurch entfällt eine zusätzliche Umrechnung in den Temperaturwert, welche wertvollen Speicherplatz kosten würde.

Im folgenden Userprogramm werden im AX-25 Protokoll der dezimale Analogwert des Temperatursensors und zusätzlich der umgerechnete Wert in °C ausgegeben. Als Vergleichswert wird auch die aktuelle Chiptemperatur gesendet.

Die Messwerte aktualisieren sich alle 10 Sekunden.

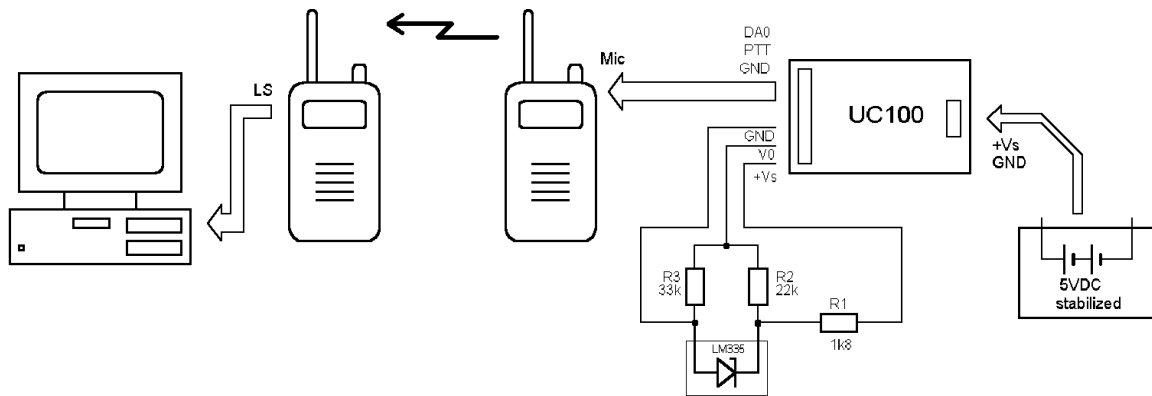


Bild 1. Temperaturmessung mit niedriger Genauigkeit

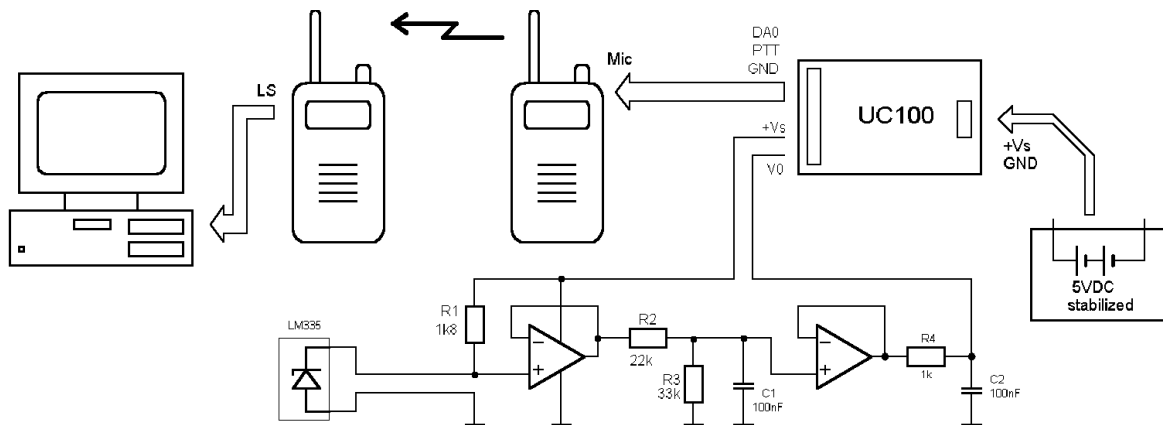


Bild 2. Temperaturmessung mit erhöhter Genauigkeit

### Userprogramm

```

a=adc-2929
a=a+250
h12 :RX TX:%d4:adc %d51:a C
sls 1
h12 :RX TX:Chiptemp=%d41:temp C
sls 10
loop
run
    
```

### Erklärung

vom ADC Wert den 25° Sensornennwert abziehen, Ergebnis=0°C Wert  
zum Ergebnis die Zahl 25,0 addieren  
adc Wert und umgerechneten Temperaturwert senden  
1s Wartezeit im Sleep-Mode  
Chiptemperatur senden  
10s Wartezeit im Sleep-Mode  
Rücksprung zum Programmbeginn  
startet das Programm

## 2.2. Messung von Spannungen höher als Vref, Ausgabe über eine LCD Anzeige

Grundsätzlich können Spannungen in der Höhe bis 2,5VDC direkt an die Analogeingänge des UC100 gelegt und gemessen werden (ergibt Messwerte von 0...4095). Höhere Messspannungen erfordern einen Spannungsteiler und dessen Berücksichtigung bei der Auswertung im Userprogramm.

Die einfachste Methode ist ein ohmscher Spannungsteiler direkt am Analogeingang, dabei sollte jedoch, um ausreichende Genauigkeiten zu erzielen, der Gesamtwiderstand Größenordnungen von 100k nicht übersteigen (Bild 1.). Dieser Wert stellt für manche Spannungsquelle bereits eine nennenswerte Belastung dar. Dann sollte eine Aktive Lösung verwendet werden (Bild 2.). Durch Verwendung eines geeigneten Operationsverstärkers können höhere Messgenauigkeiten und Eingangswiderstände realisiert werden.

Das unten angeführte Userprogramm schreibt die Messspannung alle Sekunden auf ein LC Display (Teilverhältnis 10:1). Durch Verwendung von Widerstandsteilern, welche exakt geradzahlige Bit/V Verhältnisse ergeben, lässt sich die Genauigkeit der Messung deutlich erhöhen.

Dimensionierung der Spannungsteiler:

Verhältnis R1:R2 = 2:1 max. Vm = ca. 5V  
 Verhältnis R1:R2 = 6,8:1 max. Vm = ca. 16V  
 Verhältnis R1:R2 = 10:1 max. Vm = ca. 22V  
 Verhältnis R1:R2 = 20:1 max. Vm = ca. 42V

Umrechnungsfaktor:  $a=a/55$   
 Umrechnungsfaktor:  $a=a/21$   
 Umrechnungsfaktor:  $a=a/149$   
 Umrechnungsfaktor:  $a=a/78$

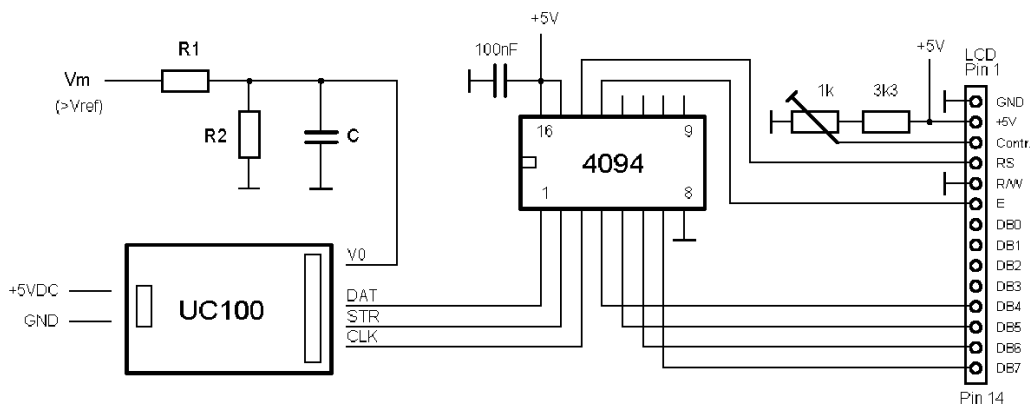


Bild 1. Passiver Eingangsteiler

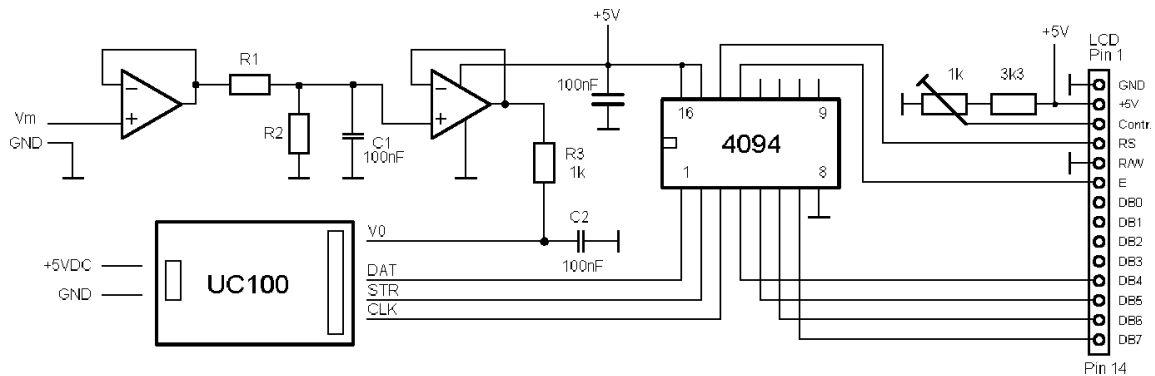


Bild 2. Eingangsteiler mit OPV

### Userprogramm

```

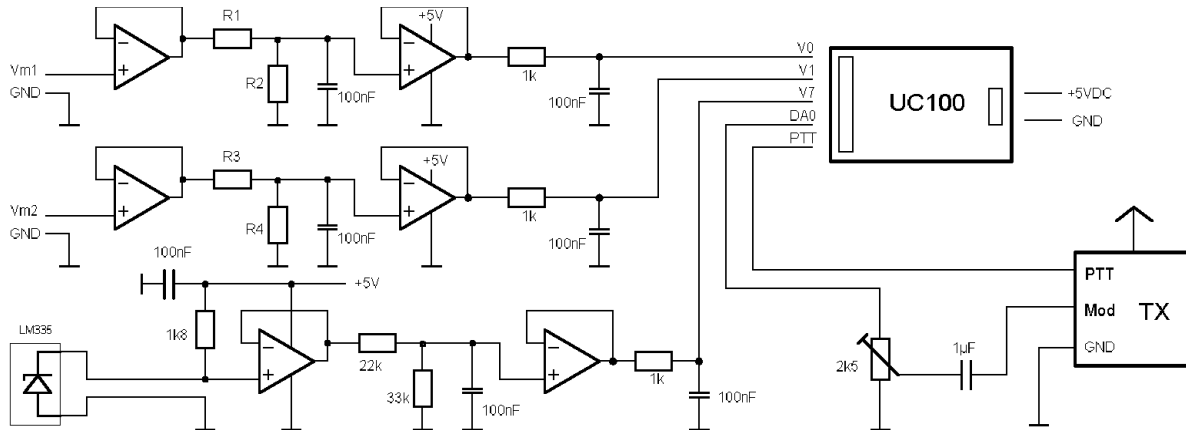
lcd %0 %1
1:
a=adc*10
a=a/149
lcd %128 Messwert
slms 50
lcd %192 Vm=%d41:a V
sls 1
goto 1
run
    
```

### Erklärung

LCD Steuerbefehl „Anzeige löschen“  
 Label 1  
 Der Variablen a wird der aktuelle Messwert zugewiesen und mit 10 multipliziert  
 Teilungsfaktor für 10:1 wie in Liste oben aufgeführt  
 Schreibt das Wort „Messwert“ an den Beginn der 1. Zeile des LCD  
 50ms Wartezeit für zuverlässige Befehlsübernahme beim LCD  
 Schreibt „Vm=“, danach den Messwert, danach „V“ an den Beginn der 2. Zeile  
 1s Wartezeit im Sleep-Mode  
 Sprung zu Label 1  
 startet das Programm

### 2.3. Überwachen von 2 Spannungen und 2 Temperaturen, DTMF Alarm bei Abweichung

Bei dieser Anwendung werden 2 Gleichspannungen (15V+-3V, 12V+-2V) auf Über- und Unterspannung, sowie eine externe Temperatur und die Chiptemperatur auf Überschreitung eines Grenzwertes überwacht (>40 bzw. >50°C). Bei Abweichungen soll ein Alarmsignal als DTMF Code ausgesendet werden. Über AX-25 werden die Messwerte alle 60 Sekunden übertragen, auch die Alarmmeldungen werden in dieser Betriebsart gesendet, allerdings mit kürzeren Intervallen als die Standardinformationen (10s).



#### Userprogramm

```

burn&
a=adc*10
a=a/149
port=1
b=adc*10
b=b/21
port=7
c=adc-2929
c=c+250
port=0
h12 :RX TX:V1=%d41:a V
sls 1
h12 :RX TX:V2=%d41:b V
sls 1
h12 :RX TX:T1=%d41:c C
sls 1
h12 :RX TX :T2=%d41:temp C
sls 1
if a>180
goto 2
if a<120
goto 4
if b>140
goto 2
if b<100
goto 4
if c>400
goto 6
if temp>500
goto 6
sls 60
loop
2:
dtmf 12345
h12 :RX TX:Ueberspannung!
sls 10
loop
4:
dtmf 54321
h12 :RX TX:Unterspannung!
sls 10
loop
6:
dtmf 12121
h12 :RX TX:Uebertemperatur!
sls 10
loop
end
    
```

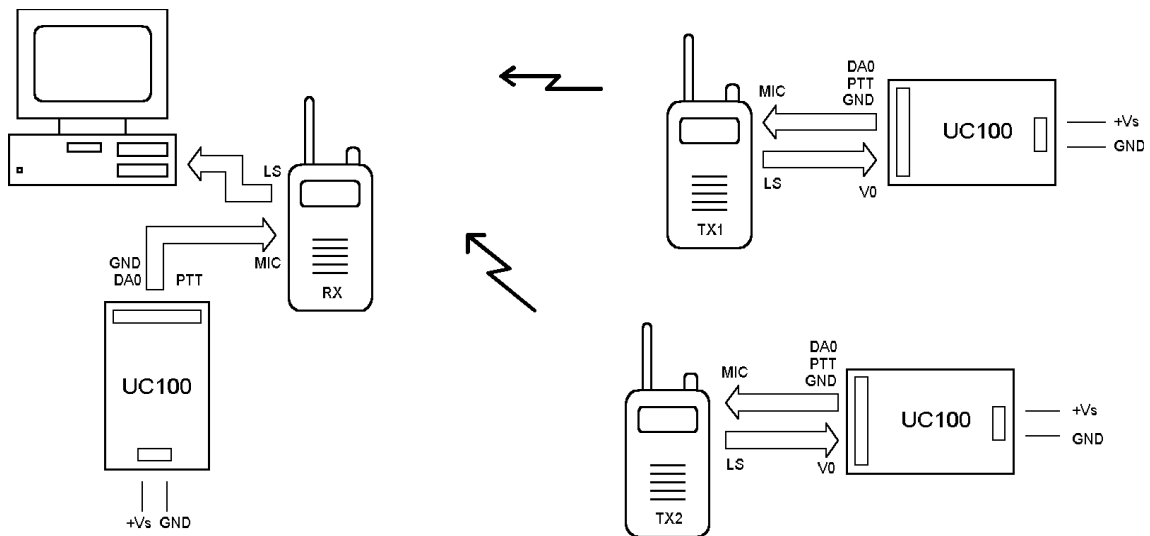
#### Erklärung

Für Download des Userprogramms in das EEPROM des Controllers  
 Der Variablen a wird der aktuelle Messwert zugewiesen (Startport 0) und mit 10 multipliziert  
 Teilungsfaktor für 10:1 zum Messen von Spannungen bis ca. 22V  
 Wechseln zu Port 1 (V1)  
 Der Variablen b wird der aktuelle Messwert zugewiesen (Port 1) und mit 10 multipliziert  
 Teilungsfaktor für 6,8:1 zum Messen von Spannungen bis ca. 16V  
 Wechseln zu Port 7 (V7)  
 vom ADC Wert den 25° Sensornennwert abziehen, Ergebnis=0°C Wert  
 zum Ergebnis die Zahl 25,0 addieren  
 Rückschalten zu Port 0 (V0)  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „V1=“, danach Spannung a und „V“  
 1s Wartezeit im Sleep-Mode  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „V2=“, danach Spannung b und „V“  
 1s Wartezeit im Sleep-Mode  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „T1=“, danach Temperatur c und „C“  
 1s Wartezeit im Sleep-Mode  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „T2=“, danach Chiptemperatur und „C“  
 1s Wartezeit im Sleep-Mode  
 Falls a größer als 18,0V ist  
 Sprung zu Label 2  
 Falls a kleiner als 12,0V ist  
 Sprung zu Label 4  
 Falls b größer als 14,0V ist  
 Sprung zu Label 2  
 Falls b kleiner als 10,0V ist  
 Sprung zu Label 4  
 Falls c größer 40,0°C ist  
 Sprung zu Label 6  
 Falls die Chiptemperatur größer 50,0°C ist  
 Sprung zu Label 6  
 60s Wartezeit im Sleep-Mode (Messwiederholzeit)  
 Rücksprung zum Programmbeginn  
 Label 2  
 Aussenden der DTMF Tonfolge 12345  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „Ueberspannung!“  
 10s Wartezeit im Sleep-Mode (Alarmwiederholzeit)  
 Rücksprung zum Programmbeginn  
 Label 4  
 Aussenden der DTMF Tonfolge 54321  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „Unterspannung!“  
 10s Wartezeit im Sleep-Mode (Alarmwiederholzeit)  
 Rücksprung zum Programmbeginn  
 Label 6  
 Aussenden der DTMF Tonfolge 12121  
 Betriebsartbefehl AX-25/AFSK, senden von Fixtext „Uebertemperatur!“  
 10s Wartezeit im Sleep-Mode (Alarmwiederholzeit)  
 Rücksprung zum Programmbeginn  
 Signalisiert das Ende des Downloads, Start durch Eingabe von **run** oder EIN/AUS

## 2.4. Abrufen von Messwerten an verschiedenen Orten

Oft tritt der Fall ein, Messwerte von mehreren räumlich sehr weit entfernten Messstellen einzuholen. Zur Realisierung dieser Aufgabe eignet sich der Befehl **wait <frequ>**. Dabei wartet der Controller mit der Fortsetzung des Programmablaufs, bis am aktuellen Port ein Tonburst mit der gewünschten Frequenz auftritt. So ist es möglich, die Datenübertragung einer Messstelle gezielt durch Aussendung einer bestimmten Tonfrequenz zu starten. Jeder Messstelle kann eine unterschiedliche Auswertefrequenz zugewiesen werden, dabei kann die Sendersteuerung in der Zentrale ebenfalls von einem UC100 übernommen werden ( Befehl **beep** ).

Im folgenden Beispiel wird die aktuelle Chiptemperatur von 2 unterschiedlichen Messstellen nacheinander alle 10 Sekunden über AX-25 AFSK abgefragt.



### Sender Userprogramm

```
beep 1000  
sls 2  
beep 1500  
sls 10  
loop
```

### Erklärung

Gibt einen kurzen Tonburst mit 1000Hz aus  
2s Wartezeit im Sleep-Mode  
Gibt einen kurzen Tonburst mit 1500Hz aus  
10s Wartezeit im Sleep-Mode  
Rücksprung zum Programmbeginn

### Messstelle 1 Userprogramm

```
wait 1000  
slms 500  
h12 :RX TX1:Temp=%d41:temp C  
loop
```

### Erklärung

Wartet auf einen 1000Hz Ton an Port 0  
500ms Wartezeit im Sleep-Mode  
Betriebsartbefehl AX-25/AFSK, senden von Fixtext „Temp=“, danach Chiptemperatur und „C“  
Rücksprung zum Programmbeginn

### Messstelle 2 Userprogramm

```
wait 1500  
slms 500  
h12 :RX TX2:Temp=%d41:temp C  
loop
```

### Erklärung

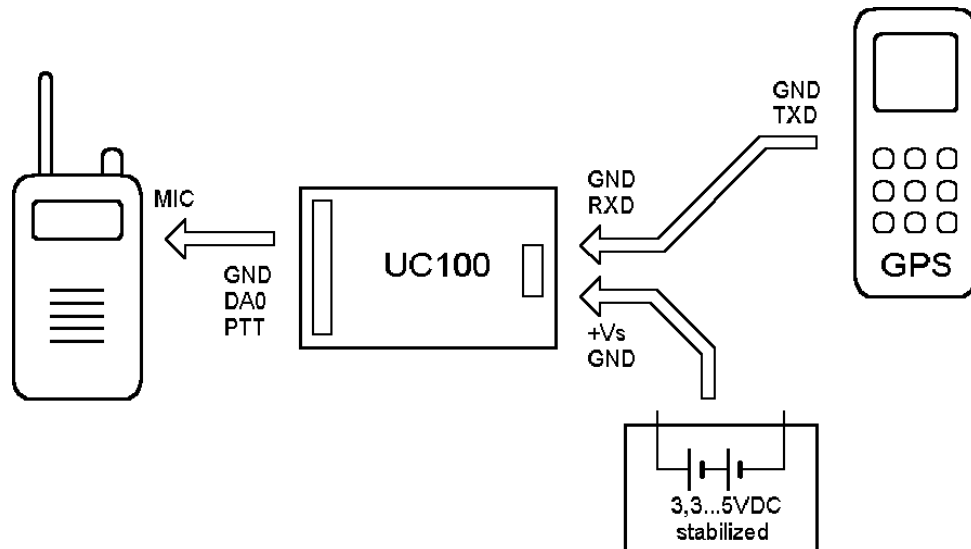
Wartet auf einen 1500Hz Ton an Port 0  
500ms Wartezeit im Sleep-Mode  
Betriebsartbefehl AX-25/AFSK, senden von Fixtext „Temp=“, danach Chiptemperatur und „C“  
Rücksprung zum Programmbeginn





### 3.2. APRS Geber mit universeller NMEA Datenauswertung

Der UC100 ist in der Lage, durch ein geeignetes Userprogramm als APRS Controller zu arbeiten. Das folgende Programm liest NMEA Daten über die serielle Schnittstelle ein, wandelt sie in das APRS Format um und sendet sie in AX-25 1k2 aus. Dieses Userprogramm ist sehr umfangreich, da es auch trotz kleiner Formatunterschiede zwischen verschiedenen Gerätetypen die korrekten Standortinformationen aus dem Datenstring isoliert und verarbeitet. Erweiterungen des Programms sind durch den bis auf nur wenige Byte ausgenutzten Speicherplatz nicht mehr möglich. Für bestimmte GPS Empfängermodelle maßgeschneiderte Userprogramme erlauben dagegen zusätzliche Features.



#### Userprogramm ( Teil 1/3 )

#### Erklärung

```

burn&
##$GPRMC,024221,A,4815.090,N,01302.272,E,000.0,173.0,251003,001.0,E*76
##$GPRMC,142246,A,4804.4864,N,01259.3716,E,5.5,61.1,281103,1.6,E,A*2E
##$GPRMC,142302,A,4804.5062,N,01259.4023,E,14.3,8.8,281103,1.6,E,A*26
##$GPRMC,142304,A,4804.5162,N,01259.4049,E,19.7,10.3,281103,1.6,E,A*16
rsbd 4800
10:
#auf $ im stream warten
if rs<>'$'
goto 10
rdln 1
#if x[2]<>'R'
#goto 10
#if x[3]<>'M'
#goto 10
if x[4]<>'C'
goto 10
if x[13]<>'A'
goto 10
#zeile hat richtigen kopf
#breitengrad:
xl=0
x[0]=x[15]+32
x[1]=x[16]+32
x[2]=x[17]
x[3]=x[18]
x[4]=x[20]
x[5]=x[21]
i=22

```

Mögliches NMEA Format  
 Mögliches NMEA Format  
 Mögliches NMEA Format  
 Mögliches NMEA Format  
 Schnittstellen Baudrate

```
70:
if x[i]='N'
goto 71
if x[i]='S'
goto 72
i=i+1
if i>30
goto 10
goto 70
71:
x[3]=x[3]+32
72:
#laengengrad grad+28 a:
i=i+2
b=x[i]*10
i=i+1
b=b+x[i]
a=b*10
i=i+1
a=a+x[i]
a=a-5300
#laengengrad min+28 b:
i=i+1
b=x[i]*10
i=i+1
b=b+x[i]
b=b-500
#codiere min
if b<38
b=b+60
#laengengrad min 2 kommastellen+28 c:
i=i+2
c=x[i]*10
i=i+1
c=c+x[i]
c=c-500
#west/ost
95:
if x[i]='W'
goto 90
if x[i]='E'
goto 91
i=i+1
if i>45
goto 10
goto 95
90:
x[5]=x[5]+32
91:
#speed d:
i=i+2
d=0
100:
d=d*10
d=d+x[i]
d=d-48
i=i+1
if i>50
goto 10
if x[i]<>'.'
goto 100
#winkel f:
i=i+3
f=0
110:
f=f*10
f=f+x[i]
f=f-48
```

```

i=i+1
if i>60
goto 10
if x[i]<>'.'
goto 110
#codiere speed+winkel in 3 byte d e f save speed in g:
e=d#10
e=e*10
g=f/100
e=e+g
e=e+32
g=d
d=d/10
d=d+28
if d<48
d=d+80
f=f#100
f=f+28
#
if a>=38
goto 30
#laengengrad 0..9
a=a+90
goto 35
30:
if a<128
#laengengrad 10..99
goto 36
if a>=138
goto 32
#laengengrad 100..109
a=a+80
32:
#laengengrad 110..179
a=a-100
35:
x[4]=x[4]+32
36:
#breitengrad im zielcall
if g>1
#wir fahren
goto 40
#wir stehen
h=h+1
if h<6
goto 50
#wir stehen schon n perioden
40:
h=0
h12 :%$-7 AA1BB-9:`%a %b %c %d %e %f >/
50:
sls 30
loop
end

```

Ansprechschwelle für Bewegungserkennung

Aussendungen bei Stillstand (hier alle 6x30s)

Eingabe "Wide" und eigenes Rufzeichen

Intervall zwischen den Aussendungen (hier 30s)

### 3.3. Antennenrichtungsanzeige über LCD und akustisch über Morsecode

Durch die sehr genaue 12 Bit A/D Wandlung im UC100 ist es möglich, die analoge Anzeige bestehender Rotorsteuergeräte durch ein zusätzliches LC Display zu verbessern (1 Grad Auflösung). Voraussetzung ist lediglich eine Gleichspannung, die in linearem Zusammenhang mit der Antennenrichtung steht. Außerdem kann der Wert akustisch im Morsecode ausgegeben werden.

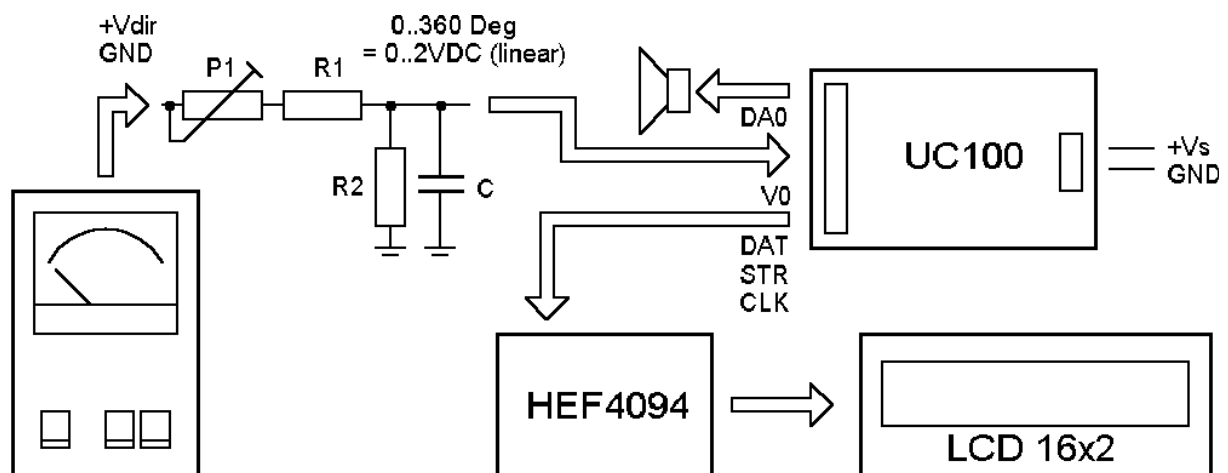
Bei Verwendung von Rotoren mit mechanischem Anschlag im Süden muss der vorhandene 180 Grad Offsetwert im Programm berücksichtigt werden.

Sollte der Maximalwert der Messspannung 2V übersteigen, ist es notwendig, einen Spannungsteiler vor den Analogeingang des UC100 zu schalten. Zum Ausgleich kurzzeitiger Spannungsschwankungen und zur Unterdrückung hochfrequenter Störungen sollte unbedingt der Kondensator C eingesetzt werden. Werte zwischen 10nF und 1µF sollten hier gute Ergebnisse liefern, abhängig von der Dimensionierung des Spannungsteilers. Der Spannungsteiler darf jedoch die Messspannung nicht belasten, in diesem Fall muss die Spannungsteilung aktiv unter Verwendung eines Rail to Rail Operationsverstärkers mit FET Eingang durchgeführt werden (siehe Punkt 2.2. Messung von Spannungen höher als Vref).

Als günstig erweist sich auch ein Widerstandstrimmer P1 in Serie zu R1 mit einem Wert von ca.  $0.1 \times R1$ . Er kann dazu verwendet werden, den Teilungsfaktor so zu beeinflussen, dass der Wert „Bit pro Grad“ eine ganze Zahl ergibt und dadurch die Anzeigegenauigkeit das Maximum erreicht. Eventuelle Offsetspannungen bei Null bzw. 180 Grad werden im Userprogramm rechnerisch eliminiert.

Der UC100 erlaubt die Ansteuerung von Standard LCDs. Dafür werden die Steuersignale des seriellen Datenbusses über ein Schieberegister in parallele Ansteuerimpulse gewandelt. Dazu dient ein CMOS Schieberegister 4094, das die notwendigen Steuer- und Datensignale ( 4 Bit Modus ) an das LC Display liefert.

Zur akustischen Ausgabe dient ein Lautsprecher an DA0 ( gegen GND ), durch den relativ hohen Innenwiderstand des DAC Ausgangs (>100Ohm) empfiehlt es sich, zur Ansteuerung einen Impedanzwandler in Form eines Transistors oder OPV einzusetzen.



Das folgende Userprogramm ist für horizontale Rotoren mit Nordanschlag ausgelegt (Anzeige bereits für zusätzliche Elevationsrotoren ausgelegt). Es erkennt eine Änderung der Messspannung und damit das Drehen der Antenne und reagiert mit Tonpulsen. Sobald sich die Spannung wieder stabilisiert, wird der neue Wert gemessen, umgerechnet, und als Winkelinformation an das Display gesendet. Wenn gewünscht, erfolgt die Ausgabe danach zusätzlich akustisch im Morsecode. Auch ohne Antennendrehung wiederholt sich diese Ausgabe automatisch in gewissen Abständen. Um zu vermeiden, dass bei stärkeren Schwankungen der Messspannung ohne Antennendrehung die automatische Erkennung anspricht, ist im Userprogramm ein Bereich definiert, innerhalb dessen sich die Messspannung bewegen darf, bis ein Messzyklus gestartet wird.

2 individuelle Spannungswerte müssen vorab durch einfache Messung am ADC ermittelt werden:

1. Die Offsetspannung bei 0 Grad (Startposition=Minimalspannung), stellt den Wert xxx im Userprogramm dar

2. Die Spannung bei 360 Grad (Endposition=Maximalspannung)

Die Differenz der beiden ADC-Werte multipliziert mit 10 und geteilt durch 360 ergibt den Wert yy (Bit/Grad)

Mit dem Offsetwert und den Bit/Grad kann im Userprogramm der aktuelle Winkel berechnet und angezeigt werden.

Die Programmzeile e=50 bestimmt die automatische Wiederholzeit der Ausgabe 50 entspricht  $50 \times 200\text{ms} = 10\text{s}$

Die Ansprechtoleranz beträgt hier  $\pm 5$  Bit ( $f=a+5$ ,  $f=a-5$ ), kann individuell angepasst werden.

## Userprogramm

```
burn&
#Auswertung fuer Rotoren mit Nordanschlag
e=0
1:
#Nulloffset bei Norden
c=adc-xxx
#Umrechnung Potispannung in Grad
c=c*10
d=c/yy
if d<0
goto 2
if d>359
goto 3
goto 4
2:
d=d+360
goto 4
3:
d=d-360
4:
#opt.und akust.Ausgabe
lcd %128 Azimut%d40:d Grad
lcd %192 Elevat 000 Grad
pr %d40:d
cwf2 %d3:d
5:
#Erkennung"Antenne dreht sich"
a=adc
#Ansprechtoleranz fuer Erkennung
f=a+5
g=a-5
slms 200
b=adc
e=e+1
6:
if b>f
goto 7
if b<g
goto 7
#Schleife fuer Autowiederholung
if e=1
goto 1
if e=50
loop
goto 5
7:
#Alarmschleife"Antenne dreht sich"
a=adc
f=a+5
g=a-5
slms 800
init beep
spd=20
beep 2000
e=1
b=adc
goto 6
end
```

## Erklärung

Für Download des Userprogramms aus Datei  
Kommentarzeile  
Initialisierungswert 0 für Variable e  
Label 1  
Kommentarzeile  
Variable c=ADC-Wert minus Offsets Spannungswert bei Norden  
Kommentarzeile  
Variable c mit 10 multiplizieren (höhere Genauigkeit)  
Variable d ist Variable c geteilt durch yy (ermittelter Bit/Grad Wert)  
Falls Variable d kleiner Null ist  
Sprung zu Label 2  
Falls Variable d größer 359 ist  
Sprung zu Label 3  
Sprung zu Label 4  
Label 2  
Zu Variable d 360 addieren  
Sprung zu Label 4  
Label 3  
Von Variable d 360 subtrahieren  
Label 4  
Kommentarzeile  
Schreibt „Azimut“, Variable d und „Grad“ in die 1. Zeile der LCD  
Schreibt „Elevat“, „000“ und „Grad“ in die 2. Zeile der LCD  
Sendet den Wert von Variable d zur RS-232 Schnittstelle  
Gibt den Wert von Variable d in Morsecode an DA0 aus  
Label 5  
Kommentarzeile  
ADC Messwert in Variable a einlesen  
Kommentarzeile  
Variable f ist Wert von Variable a plus 5  
Variable g ist Wert von Variable a minus 5  
200ms Wartezeit im Sleep-Mode  
ADC Messwert in Variable b einlesen  
Zur Variablen e die Zahl 1 addieren  
Label 6  
Falls Variable b größer als Variable f ist  
Sprung zu Label 7  
Falls Variable b kleiner als Variable g ist  
Sprung zu Label 7  
Kommentarzeile  
Falls Variable e gleich 1 ist  
Sprung zu Label 1  
Falls Variable e gleich 50 ist  
Rücksprung zum Programmbeginn  
Sprung zu Label 5  
Label 7  
Kommentarzeile  
ADC Messwert in Variable a einlesen  
Variable f ist Wert von Variable a plus 5  
Variable g ist Wert von Variable a minus 5  
800ms Wartezeit im Sleep-Mode  
Änderungen der Beep-Ausgangswerte ermöglichen  
Beep-Dauer auf 200ms setzen  
Beep mit 2kHz ausgeben  
Variable e gleich 1 setzen  
ADC Messwert in Variable b einlesen  
Sprung zu Label 6  
Ende des Downloads, Start durch Eingabe von **run** oder EIN/AUS