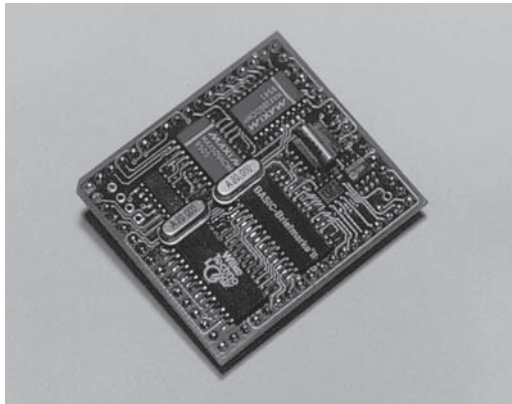


# **BASIC- Briefmarke® II F / G**





**BASIC-  
Briefmarke® II  
F / G**

**Benutzer Handbuch**

**Copyright** © Wilke Technology GmbH  
Krefelder Str. 147  
52070 Aachen / Germany

Version 1.0

Dieses Handbuch, sowie die Hard- und Software, die es beschreibt, ist urheberrechtlich geschützt und darf ohne ausdrückliche schriftliche Genehmigung von Wilke Technology GmbH in keiner Weise vervielfältigt, übersetzt oder in eine andere Darstellungsform gebracht werden.

**Warenzeichen** Diejenigen Bezeichnungen in dieser Publikation von Erzeugnissen und Verfahren, die zugleich Warenzeichen sind, wurden nicht besonders kenntlich gemacht. Solche Namen sind Warenzeichen der jeweiligen Warenzeichen-Inhaber. Aus dem Fehlen der Markierung ® kann nicht geschlossen werden, daß diese Bezeichnungen freie Warennamen sind.

*PBasic*® ist ein Warenzeichen von Parallax, Inc.

**Hinweis** Herausgeber, Übersetzer und Autoren dieser Publikation haben mit größter Sorgfalt die Texte, Abbildungen und Programme erarbeitet. Dennoch können Fehler nicht völlig ausgeschlossen werden. Wilke Technology übernimmt daher weder eine Garantie noch eine juristische Verantwortung oder Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen. Mitteilungen über eventuelle Fehler werden jederzeit gerne entgegengenommen.

Alle Rechte vorbehalten.

## Einleitung

Die BASIC-Briefmarken® II F & G basiert auf dem BASIC-Briefmarken® II-Chip. Als zusätzliche Komponenten sind folgende Features vorhanden:

- LCD - Interface
- Keyboard - Interface 24 Tasten ( 8 x 3 Matrix )
- Real-Time-Clock mit Weckfunktion
- I<sup>2</sup>C-Bus Interface
- RS232 - Schnittstelle (1 x TxD)
- 12 Bit 8 Kanal Analog-Digital-Umsetzer ( nur bei G-Typ)
- 8 Bit 4 Kanal Digital-Analog-Umsetzer (nur bei G-Typ)

Mit diesem Paket und dem BASIC-Briefmarke® II - Grundpaket können Sie sofort mit der Entwicklung von Applikationen beginnen. Als Systemanforderung wird folgender PC benötigt:

- IBM PC oder 100% kompatibler
- 3,5" - Diskettenlaufwerk
- freien COM-Port
- 128K freien RAM-Speicher
- MS-DOS 2.0 oder größer

## Spannungsversorgung

Die BASIC-Briefmarken® II F & G werden mit einer stabilisierten 5V Gleichspannung versorgt. Diese wird an PIN 33 (GND) und PIN 34 (+5V) eingespeißt.

## Programmierung

Die BASIC-Briefmarken® II F & G werden über die serielle Schnittstelle (COM-Port) des PCs programmiert. Dazu ist die BASIC-Briefmarke® II F & G mit der Betriebsspannung zu versorgen. Zur Verbindung mit dem PC werden folgende Pins der BASIC-Briefmarke® II F & G benötigt:

Pin	Bedeutung	9-pol Sub-D-Buchse	
16	SIN	3	(TxD)
15	SOUT	2	(RxD)
14	ATN	4	(DTR)
13	GND	5	(GND)

Außer diesen Verbindungen ist es noch notwendig auf der PC-Seite die Pins 6 und 7 ( DSR und RTS) zu brücken. Wenn man den Kontakt S2 auf der Unterseite der Platine brückt, wird der Schreibschutz des EEPROMs aktiviert. Dadurch wird verhindert, daß man die Briefmarke versehentlich programmiert.

## LCD-Interface

Die BASIC-Briefmarken® II F & G besitzen ein LCD - Displayinterface, das die gesamte Steuerung des Displays übernimmt. Unterstützt werden LCD - Displays mit einem Hitachi 44780 - Controller und folgender Größe:

- 1 x 16 Zeichen mit 1:8 und 1:16 Multiplexrate
- 2 x 40 Zeichen
- 4 x 20 Zeichen

Angesprochen wird das LCD - Display direkt vom Coprozessor. Dieser empfängt über I/O-Pin 15 der BASIC-Briefmarke® die Befehle. Die Kommunikation läuft über den Befehl SEROUT

**SEROUT 15,813+\$8000,["BefehlParameter"]**

Es stehen folgende Befehle für die Displaysteuerung zur Verfügung:

Befehl	Parameter	Beschreibung
L	0	1 x 16 Zeichen (Multiplexrate 1:8)
L	1	1 x 16 Zeichen (Multiplexrate 1:16)
L	2	2 x 40 Zeichen-Display
L	4	4 x 20 Zeichen-Display
L	C	Display löschen
L	H	Home-Position
L	P,xx	Cursor auf Position 0<=xx<=79 positionieren
L	T/String+cr	Text auf Display ausgeben, muß mit cr abgeschlossen werden.

Nach dem initialisieren des Displays ist eine Pause von ca. 50ms einzuhalten bevor auf dem Display etwas ausgegeben werden kann.

# BASIC-Briefmarke® II F & G

Beispiele:

```
serout 15,813+$8000,["L0"]      'initialisiert
                                  '1x16 Zeichen
                                  'Display im 1 : 8
                                  'Modus
serout 15,813+$8000,["LH"]      'Cursor auf Home
                                  'positionieren

serout 15,813+$8000,["LP",40]   'Cursor auf
                                  'Position 40 posi-
                                  'tionieren

serout 15,813+$8000,["LTHALLO",cr]
                                  'HALLO auf dem
                                  'Display ausgeben

i var byte
serout 15,813+$8000,["L4"]      'initialisiert 4x20
                                  'Zeichen Display

pause 50
loop:
    i = i + 1
    serout 15,813+$8000,["LC"]   'Display löschen
    serout 15,813+$8000,["LTI = ",dec i,cr]
goto loop
```

## Tastatur

Die BASIC-Briefmarken® II F & G können direkt mit einer 24 Tasten (8 x 3) Matrixtastatur verbunden werden. Das Scannen der Tastatur wird dabei vom Coprozessor übernommen. Dieser wird über den SEROUT-Befehl angesprochen. Das Scannen der Tastatur erfolgt über die Pins 2 bis 12 der BASIC-Briefmarken® II F & G.

**SEROUT** 15,813+\$8000, ["K"]

**SERIN** 15,813+\$0000, [Variable]

Der vom Coprozessor gelieferte Wert liegt im Bereich 0 bis 24, wobei eine 0 bedeutet, daß keine Taste gedrückt wurde. Die Anschlußbelegung der Matrixtastatur ist aus folgender Tabelle ersichtlich.

	Spalte	0	1	2	3	4	5	6	7
Zeile									
0		1	2	3	4	5	6	7	8
1		9	10	11	12	13	14	15	16
2		17	18	19	20	21	22	23	24

Beispiel:

```
taste var byte
loop:
    serout 15,813+$8000, ["K"]
    serin 15,813+$0000, [taste]
    debug " Taste ",dec taste, cr
goto loop
```

## I<sup>2</sup>C-Bus-Interface

Die BASIC-Briefmarken® II F&G besitzen ein I<sup>2</sup>C-Bus-Interface mit der entsprechende Peripheriebausteine direkt angesprochen werden können. Auf den I<sup>2</sup>C-Bus kann mit folgenden Befehl zugegriffen werden:

```
SEROUT 15,813+$8000,["I",%BAdr,InAdr,Anz,{Variable}]  
SERIN 15,813+$0000,[Variable]
```

- **BAdr** ist die Adresse des Bausteins, diese wird als binär Wert angegeben. Ist das niederwertigste Bit "0" dann wird in diesen Baustein geschrieben, bei einer "1" wird aus diesen Baustein gelesen.
- **InAdr** ist die Speicheradresse innerhalb des Bausteins
- **Anz** gibt die Anzahl der zu lesenden/schreibenden Bytes an.
- **Variable** ist die zu lesende bzw. zu schreibende Variable, bei Serout ist sie nur anzugeben, wenn geschrieben wird. Soll gelesen werden, dann ist sie bei Serin zu verwenden.

Beispiel:

```
wert var byte  
i = 0  
loop:  
    i = i + 1  
    serout 15,813+$8000,["I",%10100000,i,1,85]  
        'schreibt ein Byte  
        '(85) in die interne  
        'Adresse i des ange-  
        'sprochenen Bausteins  
    pause 100          '100ms Pause  
    serout 15,813+$8000,["I",%10100001,i,1]  
    serin 15,813+$0000,[Wert]  
    debug "WERT = ", dec wert  
goto loop
```

## RS232 Schnittstelle

Die BASIC-Briefmarken® II F & G besitzen eine RS232 Schnittstelle mit einer Sendeleitung. Die RS232 Schnittstelle wird über den I/O-Pin 6 (TxD) der BASIC-Briefmarke® gesteuert. Um die RS232 Schnittstelle zu benutzen, muß die Brücke S1 auf **1 und 2** gebrückt werden (**Standardmäßig so geschaltet**). Die RS232-Schnittstelle steht am Pin 24 der BASIC-Briefmarke® II F&G zur Verfügung. Serielle Daten kann die BASIC-Briefmarke® über jeden I/O-Pin empfangen. Es ist nur ein 22 KOhm Widerstand zwischen dem I/O-Pin der BASIC-Briefmarke® und dem Sender zu schalten.

Um den I/O-Pin ohne den RS232-Treiber zu benutzen, muß S1 auf **1 und 3** gebrückt werden.

Beispiel:

```
Wert var byte
```

```
Wert = 0
```

```
loop:
```

```
    Wert = wert + 1
```

```
    serout 6,813+$0000,[Wert] 'sendet den Inhalt  
                               'von "Wert" seriell  
                               'über den 232-  
                               'Treiber mit der  
                               'Baudrate 1200 BPS
```

```
goto loop
```

## Analog-Digital-Umsetzer

Die BASIC-Briefmarke® II G verfügt über einen 8-Kanal 12-Bit Analog-Digital-Umsetzer mit einem Eingangsspannungsbereich von 0 bis 4.096V.

Der AD-Umsetzer wird über folgende Befehle angesteuert:

```
SEROUT 15,813+$8000,["Befehl",Parameter]
SERIN 15,813+$0000,[Highbyte,Lowbyte]
```

Befehl	Parameter	Beschreibung
A	0 bis 7	Kanal 0 bis 7 gegen GND
A	8	Kanal0 - Kanal1
A	9	Kanal2 - Kanal3
A	10	Kanal4 - Kanal5
A	11	Kanal6 - Kanal7
A	12	Kanal1 - Kanal0
A	13	Kanal3 - Kanal2
A	14	Kanal5 - Kanal4
A	15	Kanal7 - Kanal6

Beispiele:

```
' *** Dieses Programm setzt die an Kanal 0 anlie-
' *** gende analoge Spannung in einen digitalen
' *** Wert um, sowie die Differenzspannung zwi-
' *** schen Kanal4 und Kanal5.
```

```
Meßwert var word
```

```
loop:
serout 15,813+$8000,["A",0]
serin 15,813+$0000,[Meßwert.highbyte,Meßwert.lowbyte]
debug " Meßwert = ", dec Meßwert, cr
serout 15,813+$8000,["A",10]
serin 15,813+$0000,[Meßwert.highbyte,Meßwert.lowbyte]
debug " Kanal4 - Kanal5 = ", dec Meßwert, cr
goto loop
```

## Digital-Analog-Umsetzer

Die BASIC-Briefmarke® II G besitzt einen 4-Kanal 8-Bit Digital-Analog-Umsetzer mit einem Spannungsbereich von 0 bis 5 V.

```
SEROUT 15,813+$8000,["Befehl",Kanal,Wert]
```

Befehl	Kanal / Wert	Beschreibung
D	0 / 0-255	Kanal 0 mit Wert 0 bis 255
D	1 / 0-255	Kanal 1 mit Wert 0 bis 255
D	2 / 0-255	Kanal 2 mit Wert 0 bis 255
D	3 / 0-255	Kanal 3 mit Wert 0 bis 255
D	4 / a,b,c,d	alle Kanäle mit Wert a,b,c,d für $0 \leq a,b,c,d \leq 255$

Beispiel:

```
wert var byte
i var byte
ot1200 con 813+$8000

loop:
    serout 15,ot1200,["D",0,i]      'Gibt den Wert
                                    'von i auf dem
                                    'Kanal 0 aus
    i = i + 1
goto loop                          'neue Wandlung
                                    'starten
```

## Real-Time-Clock

Die BASIC-Briefmarken® II F & G besitzen eine Echtzeituhr mit Weckfunktion und 256 Byte RAM. Von diesem RAM-Speicher sind die ersten 16 Byte vor die Uhrzeit, Datum und Weckzeit reserviert, die Restlichen können mit Hilfe des I<sup>2</sup>C-Bus-Interface beschrieben oder gelesen werden, siehe dazu Beispiel beim I<sup>2</sup>C-Bus auf Seite 11. Die Daten der Uhr können über Pin 1 (BAT) durch Anschluß einer Batterie (Spannung DC 3-5 V) gesichert werden. Die Uhrzeit ist BCD und wird mit folgenden Befehlen eingelesen:

```
SEROUT 15,813+$8000,["U"]
SERIN 15,813+$0000,[h,S,M,H,D,M]
```

Das Setzen der Uhrzeit geschieht mit folgenden Befehlen:

```
SEROUT 15,813+$8000,["T", $h,$S,$M,$H,$D+$xx,$M]
PAUSE 50
SEROUT 15,813+$8000,["B"]
```

\$xx = \$0	für Schaltjahr
\$xx = \$C0	wenn nächstes Jahr Schaltjahr
\$xx = \$80	wenn Schaltjahr in zwei Jahren
\$xx = \$40	wenn Schaltjahr in drei Jahren

Die Weckzeit wird mit folgenden Befehlen programmiert:

```
SEROUT 15,813+$8000,["W", $h,$S,$M,$H,$D,$M]
```

```
SEROUT 15,813+$8000,["Xn"]
```

n = 0 '	Wecken deaktivieren
n = 1 '	täglich wecken
n = 2 '	am bestimmten Datum wecken

# BASIC-Briefmarke® II F & G

Beispiel:

```
ot1200 con 813+$8000
serout 15,ot1200,["T", $11, $11, $11, $11, $11+$00, $11]
pause 50
serout 15,ot1200,["B"]           'Uhrzeit auf 11. Nov.
                                   '11h 11min 11sec 11H
                                   'setzen und aktivieren
```

## BASIC-Briefmarken®-Pins

Die BASIC-Briefmarken® II F & G besitzen 15 universelle I/O-Pins, welche frei verfügbar sind und über die Pins 18 bis 32 herausgeführt sind.

Die Funktion der I/O-Pins und wie man sie programmiert ist im Handbuch "BASIC-Briefmarke® II" beschrieben. Zu beachten ist, daß der I/O-Pin6 auch als RS232 Ausgang benutzt werden kann, siehe dazu unter RS232-Schnittstelle.

## Stromsparmmodus

Die BASIC-Briefmarken® II F & G benötigen als Versorgungsstrom ca. 25mA. Durch den Stromsparmmodus sinkt der Stromverbrauch auf ca. 9mA. Um den Stromsparmmodus zu aktivieren, sind folgende Schritte der Reihe nach durchzuführen.

1. Pin2 und Pin3 von S4 (siehe Layoutabbildung) muß durch einen Lötkecks gebrückt werden. Wird Pin2 und Pin1 gebrückt, so kann über I/O-Pin14 der BASIC-Briefmarke® der Sekundentakt der Uhr eingelesen werden.
2. Dieses kleine Programm wird in die BASIC-Briefmarke downgeloadet:

```
OT1200 CON 813+$8000  
serout 15,OT1200,["X0"]
```

Dieses Programm verhindert, daß die BASIC-Briefmarke® im Sekundentakt einen Reset bekommt.

3. Nun können Sie Ihr Programm downloaden.
4. In Ihrem Programm wird der Sleepmodus durch folgende Befehle aktiviert. Weckzeit programmieren und aktivieren mit:

```
serout 15,813+$8000,["W", $h,$S,$M,$H,$D,$M]  
Pause 50  
serout 15,813+$8000,["Xn"]
```

Xn = X1 für täglich

Xn = x2 für am bestimmten Datum

5. Coprozessor in den Sleepmodus versetzen mit:

```
serout 15, 813+$8000,["S"]
```

# BASIC-Briefmarke® II F & G

## 5. BASIC-Briefmarke mit

**end**

in den Sleepmodus versetzen.

Beispiel:

```
' Folgendes Programm versetzt die BASIC-Briefmarken II F&G in
' den Sleepmodus und weckt sie alle 10 Sekunden wieder auf,
' um einen analogen Meßwert an Kanal 0 aufzunehmen und
' diesen auf einem 4x20 Zeichen Display anzuzeigen.
```

```
messwert      var      word

ot1200 con 813+$8000
t1200  con 813
serout 15,ot1200,["L4"]'4x20 Zeichen Display 'initialisieren
pause 100
serout 15,ot1200,["LC"]
serout 15,ot1200,["A",0]
serin15,t1200,[messwert.highbyte,messwert.lowbyte]
serout 15,ot1200,["LMesswert= ",dec messwert,cr]
serout 15,ot1200,["T", $00,$50,$00,$00,$01,$01]
pause 50
serout 15,ot1200,["B"]
serout 15,ot1200,["X0"]
serout 15,ot1200,["W", $00,$00,$01,$00,$01,$01]
pause 50
serout 15,ot1200,["X1"]
serout 15,ot1200,["S"]
end
```

Bei Benutzung des Stromsparmodes ist zu beachten, daß die BASIC-Briefmarken® II F&G durch einen Reset wieder aufgeweckt werden. Alle Daten des RAMs werden dadurch gelöscht. Daten die nach dem Sleepmodus noch benötigt werden, sollten vorher ins EEPROM der BASIC-Briefmarke® geschrieben werden, damit sie später noch verfügbar sind.

## Versionsnummer

Bei den BASIC-Briefmarken® II F&G besteht die Möglichkeit sich die Versionsnummer des Coprozessors anzeigen zu lassen  
Die Versionsnummer kann mit folgenden Befehlen aus dem Coprozessor gelesen werden:

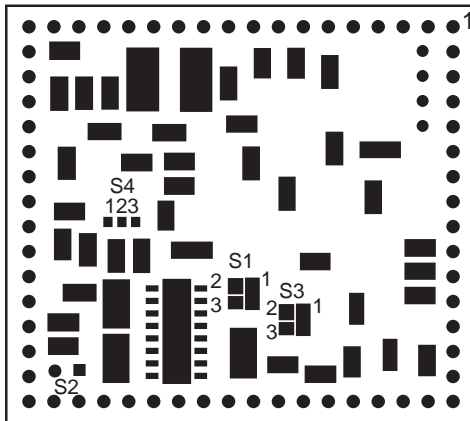
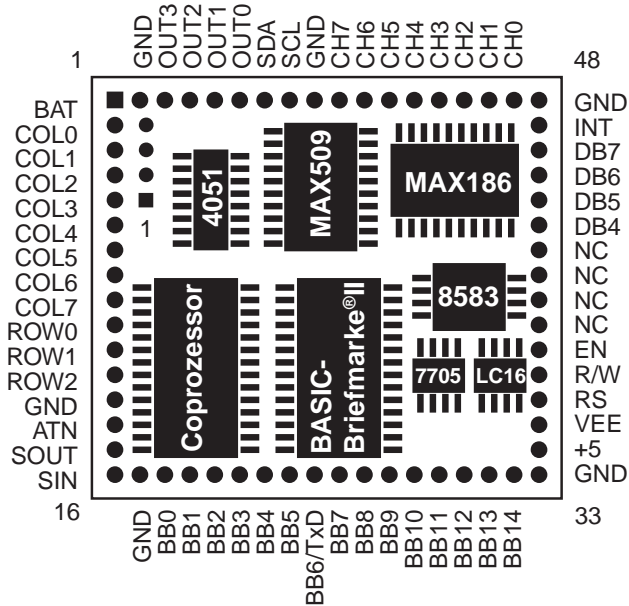
```
SEROUT 15,813+$8000,["F"]  
SERIN  15,813+$0000,[Version]
```

- **Version** ist eine als Byte zu deklarierende Variable, die die Versionsnummer (im Hexadezimal Format) des Coprozessors aufnimmt. Dabei beinhaltet das low Nibble von Version die Version des Coprozessors und das high Nibble die Art des Coprozessors, hier 3 für Coprozessor der BASIC-Briefmarken® II F&G.

# BASIC-Briefmarke® II F & G

## Layout

## Pinbelegung



# BASIC-Briefmarke® II F & G

<b>PIN</b>	<b>BEZEICHNUNG</b>	<b>ERKLÄRUNG</b>
1	BAT	Externe Batterie für Uhr
2	COL0	Matrixtastatur Spalte 0
3	COL1	Matrixtastatur Spalte 1
4	COL2	Matrixtastatur Spalte 2
5	COL3	Matrixtastatur Spalte 3
6	COL4	Matrixtastatur Spalte 4
7	COL5	Matrixtastatur Spalte 5
8	COL6	Matrixtastatur Spalte 6
9	COL7	Matrixtastatur Spalte 7
10	ROW0	Matrixtastatur Zeile 0
11	ROW1	Matrixtastatur Zeile 1
12	ROW2	Matrixtastatur Zeile 2
13	GND	Ground
14	ATN	Programmierschnittstelle
15	SOUT	Programmierschnittstelle
16	SIN	Programmierschnittstelle
17	GND	Ground
18	BB0	BASIC-Briefmarken I/O-Pin0
19	BB1	BASIC-Briefmarken I/O-Pin1
20	BB2	BASIC-Briefmarken I/O-Pin2
21	BB3	BASIC-Briefmarken I/O-Pin3
22	BB4	BASIC-Briefmarken I/O-Pin4
23	BB5	BASIC-Briefmarken I/O-Pin5
24	BB6/TxD	I/O-Pin6 sowie RS232 TxD
25	BB7	BASIC-Briefmarken I/O-Pin7
26	BB8	BASIC-Briefmarken I/O-Pin8
27	BB9	BASIC-Briefmarken I/O-Pin9
28	BB10	BASIC-Briefmarken I/O-Pin10
29	BB11	BASIC-Briefmarken I/O-Pin11
30	BB12	BASIC-Briefmarken I/O-Pin12
31	BB13	BASIC-Briefmarken I/O-Pin13
32	BB14	BASIC-Briefmarken I/O-Pin14
33	GND	Ground
34	+5	5V Versorgungsspannung
35	VEE	Kontrastspannung LC-Display
36	RS	Register Select LC-Display
37	R/W	LC-Display Read/Write

# BASIC-Briefmarke® II F & G

38	EN	Enable LC-Display
39	NC	Not Connected
40	NC	Not Connected
41	NC	Not Connected
42	NC	Not Connected
43	DB4	LC-Display Databit 4
44	DB5	LC-Display Databit 5
45	DB6	LC-Display Databit 6
46	DB7	LC-Display Databit 7
47	INT	Ausgang der Uhr
48	GND	Ground
49	CH0	ADC-Kanal0
50	CH1	ADC-Kanal1
51	CH2	ADC-Kanal2
52	CH3	ADC-Kanal3
53	CH4	ADC-Kanal4
54	CH5	ADC-Kanal5
55	CH6	ADC-Kanal6
56	CH7	ADC-Kanal7
57	GND	Ground
58	SCL	I <sup>2</sup> C-Bus-Clock
59	SDA	I <sup>2</sup> C-Bus-Data
60	OUT0	DAC-Kanal0
61	OUT1	DAC-Kanal1
62	OUT2	DAC-Kanal2
63	OUT3	DAC-Kanal3
64	GND	Ground

## Demo-Programm-Listing

```
OT1200 CON      813+$8000
T1200  CON      813+$0000

Key     var      byte
ASCIKEY var      byte
Kanal   var      byte
ADCIN   var      word
DACOUT  var      byte
I       var      byte
J       var      byte
Datum   var      byte(6)
Version var      byte

      serout 15,OT1200,["L4"]
      pause 100
      serout 15,OT1200,["F"]
      serin  15,T1200,[Version]
      serout 15,OT1200,["LTCoprozessor",cr]
      serout 15,OT1200,["LP",20]
      serout 15,OT1200,["LTTyp:",decVersion.highnib,cr]
      serout 15,OT1200,["LP",40]
      serout 15,OT1200,["LTVersion: ",dec Version.lownib,cr]
      pause 5000
menü:  serout 15,OT1200,["LC"]
      serout 15,OT1200,["LT1. ADC-Test",13]
      serout 15,OT1200,["LP",20]
      serout 15,OT1200,["LT2. DAC-Test",13]
      serout 15,OT1200,["LP",40]
      serout 15,OT1200,["LT3. Uhr-Test  ?",13]
loop:  gosub keyscan:
      if key > 3 then loop:
      branch Key,[loop,ADC,DAC,UHR]
      goto menü

ADC:   serout 15,OT1200,["LC"]
      pause 100
      serout 15,OT1200,["LTInput-0   Input-1  ",cr]
      serout 15,OT1200,["LTInput-2   Input-3  ",cr]
      serout 15,OT1200,["LTInput-4   Input-5  ",cr]
      serout 15,OT1200,["LTInput-6   Input-7  ?",cr]
Loop1: gosub keyscan
      if key > 7 then Loop1:
      Kanal = Key
      serout 15,OT1200,["A",Kanal]
      serin  15,T1200,[ADCIN.highbyte,ADCIN.lowbyte]
```

# BASIC-Briefmarke® II F & G

```
serout 15,OT1200,["LC"]
pause 100
serout 15,OT1200,["LP",23]
serout 15,OT1200,["LTKanal-",dec Kanal," : ",dec ADCIN,13]
gosub keyscan
goto menü:

DAC: serout 15,OT1200,["LC"]
pause 100
serout 15,OT1200,["LTOoutput-0           ",13]
serout 15,OT1200,["LTOoutput-1           ",13]
serout 15,OT1200,["LTOoutput-2           ",13]
serout 15,OT1200,["LTOoutput-4           ? ",13]
Loop2: gosub keyscan
if key > 3 then Loop2
Kanal = key
serout 15,OT1200,["LC"]
pause 100
serout 15,OT1200,["LTBitte Wert ueber Tastatur eingeben
(000-255): ",13]
DACOUT = 0
for I = 1 to 3
Loop3: gosub keyscan
ASCIKEY = key+$30
serout 15,OT1200,["LT",ASCIKEY,13]
pause 100
if key > 9 then Loop3
DACOUT = DACOUT*10
DACOUT = DACOUT+key
next
if DACOUT < 256 then OK:
DACOUT = 255
OK: serout 15,OT1200,["D",Kanal,DACOUT]
pause 150
gosub keyscan
goto menü:

UHR: serout 15,OT1200,["LC"]
pause 500
serout 15,OT1200,["LT1. Uhr setzen           ",cr]
serout 15,OT1200,["LT2. Alarm setzen        ",cr]
serout 15,OT1200,["LT3. Alarm aktivieren    ",cr]
serout 15,OT1200,["LT4. Uhr anzeigen       ? ",cr]
Loop4: gosub keyscan
if key > 4 then Loop4
branch key, [Menü,Uhr_Set,Alarm,Alarm_On,Uhr_Dis]

Uhr_Set:
serout 15,OT1200,["LC"]
```

# BASIC-Briefmarke® II F & G

```

pause 500
serout 15,OT1200,[„LTUhrzeit?:“,cr]
serout 15,OT1200,[„LP“,20]
serout 15,OT1200,[„LTMMDDHHMSSsh“,cr]
serout 15,OT1200,[„LP“,40]
for I = 5 to 0
Datum(I) = 0
for J = 0 to 1
Loop5: gosub keyscan
pause 150
if key > 9 then Loop5
Datum(I) = Datum(I)*16
Datum(I) = Datum(I)+key
serout 15,OT1200,[„LT“,dec key,13]
next
next
serout15,OT1200,[„T“,Datum(0),Datum(1),Datum(2),Datum(3),
Datum(4),Datum(5)]
pause 200
serout 15,OT1200,[„B“]
serout 15,OT1200,[„X0“]
pause 200
goto UHR:

Uhr_Dis:
serout 15,OT1200,[„LC“]
serout 15,OT1200,[„LTHEute ist der:“,cr]
serout 15,OT1200,[„LP“,40]
serout 15,OT1200,[„LTeS ist Uhr“,cr]
Loop6: serout 15,OT1200,[„U“]
serin 15,T1200,[Datum(0),Datum(1),Datum(2),Datum(3),
Datum(5),Datum(4)]
serout 15,OT1200,[„LP“,15]
for I = 5 to 1
serout 15,OT1200,[„LT“,hex2 Datum (I),cr]
if I >= 4 then SP:
if I = 5 then DP:
if I = 4 then LF:
if I = 1 then DP:
serout 15,OT1200,[„LT:“,cr]
goto DP:
SP: serout 15,OT1200,[„LT.“,cr]
goto L1:
LF: serout 15,OT1200,[„LP“,47]
DP: next
pause 500
serout 15,OT1200,[„K“]
serin 15,T1200,[Key]
if Key = 0 then Loop6:
```

# BASIC-Briefmarke® II F & G

```
goto UHR

Alarm: serout 15,OT1200,[„LC“]
       pause 500
       serout 15,OT1200,[„LTAlarmzeit?MDHMS:“,13]
       for I = 5 to 0
       Datum(I) = 0
       for J = 0 to 1
Loop7: gosub keyscan
       pause 150
       if key > 9 then Loop7
       Datum(I) = Datum(I)*16
       Datum(I) = Datum(I)+key
       serout 15,OT1200,[„LT“,dec key,13]
       next
       next
       serout 15,OT1200,[„W“,Datum(0),Datum(1),Datum(2),
                       Datum(3),Datum(4),Datum(5)]
       pause 100
       goto UHR:

Alarm_On:
       serout 15,OT1200,[„LC“]
       pause 200
       serout 15,OT1200,[„LT1. No Alarm“,cr]
       serout 15,OT1200,[„LP“,20]
       serout 15,OT1200,[„LT2. Daily Alarm      „,cr]
       serout 15,OT1200,[„LT3. Dated Alarm    ? „,cr]

Loop8: gosub keyscan
       if key > 3 then Loop8
       branch key,[UHR,NO,Daily,Dated]

NO:    serout 15,OT1200,[„X0“]
       pause 100
       goto UHR:

Daily: serout 15,OT1200,[„X1“]
       serout 15,OT1200,[„S“]
       end
       pause 100
       goto UHR:

Dated: serout 15,OT1200,[„X2“]
       pause 100
       goto UHR:

keyscan:
       serout 15,OT1200,[„K“]
       serin 15,T1200,[Key]
       if key = 0 then keyscan:
       lookdown Key,[10,1,2,3,4,5,6,7,8,9],Key
       return
```

## Technische Daten

<b>Speicher:</b>	2048 Byte EEPROM mehr als 100000 Schreibzyklen
<b>Geschwindigkeit:</b>	20 MHz 10000 BASIC-Befehle / sec
<b>Temperaturbereich:</b>	0..70°C
<b>PC-Schnittstelle:</b>	bidirektional über RS232
<b>I/O-Leitungen:</b>	15 frei verwendbare I/O-Pins Eingang/Ausgang seriell Eingang/Ausgang digital Eingang-Analog (RCTIME) Ausgang-Analog (PWM) Belastbarkeit: als Quelle 20 mA pro Pin als Senke 25 mA pro Pin bzw. Summe aller Belastungen pro Port als Quelle 40 mA pro Port als Senke 50 mA pro Port
<b>AD-Umsetzer:</b>	12-Bit, 8-Kanal, 0 bis 4.096V, ca. 25/sec
<b>DA-Umsetzer:</b>	8-Bit, 4-Kanal, 0 bis 5V, ca. 25/sec
<b>Tastaturinterface:</b>	24 Tasten ( 3 x 8 Matrix )
<b>LCD-Interface:</b>	1 x 16 Zeichen 2 x 40 Zeichen 4 x 20 Zeichen
<b>Uhr:</b>	Real-Time-Clock mit Weckfunktion
<b>RS232:</b>	1 x TxD

# BASIC-Briefmarke® II F & G

**I<sup>2</sup>C-Bus:** Protokoll wird vom Coprozessor erstellt

**Maße :** 48 x 43 mm

**Betriebsspannung:** 5V DC geregelt

**Stromaufnahme:** ca. 25 mA  
ca. 9 mA im Sleep-Modus

## Notizen

# Notizen



**Wilke Technology GmbH**  
Krefelder Str. 147  
52070 Aachen, Germany  
Phone: +49 (0) 241 / 918 900  
Fax: +49 (0) 241 / 918 9044  
Email: [info@wilke.de](mailto:info@wilke.de)  
Web: [www.wilke.de](http://www.wilke.de)