

AVR-ChipBasic2 - Bibliotheken

V1.43 (c) 2006-2012 Jörg Wolfram

1 Die Festkomma-Bibliothek FIXLIB (Bibliothekscod 0x10)

1.1 Allgemeines

Diese Bibliothek erweitert den BASIC-Computer um Festkomma-Berechnungen. Festkomma bedeutet, daß die Stellen vor und nach dem Komma fest vorgegeben sind. Bei dieser Bibliothek lässt sich die Stellenzahl per Initialisierungskommando einstellen. Intern ist eine Darstellung gewählt, die jeweils 2 Stellen (Wertebereich 0...99) in einem Byte zusammenfasst. Somit sind Vor- und Nachkommastellen auch nur in 2-er Schritten einstellbar.

- 2...32 Vorkomma- und 0...30 Nachkommastellen wählbar
- Anzahl der verfügbaren Variablen von der Anzahl der Stellen abhängig
- Belegt den Array-Bereich 256...511
- Formatierte Textausgabe
- Eingabe über Text im Array möglich
- Integer- Datenaustausch mit dem BASIC-System
- 4 Grundrechenarten, Vergleiche und weitere Funktionen

Systembedingt haben nur Parameter die auch angegeben wurden einen definierten Wert. Bei einem **CALL L,2,0** ist nur der Parameter 1 definiert, der für die Funktion notwendige Parameter 2 aber nicht. Somit wird der Variable 0 der Wert zugewiesen, der sich zu diesem Zeitpunkt gerade an dieser Stelle im RAM befindet.

1.2 Funktionsübersicht

Funktionsnummer	Funktion
0	Dummy-Funktion, falls die Bibliothek aus dem Hauptmenu aufgerufen wird
1	Initialisierung
2	Integer-Zuweisung Variable=Wert
3	Zuweisung über Textstring im Array
4	liefert Integerwert der Variablen
5	Textausgabe der Variablen
6	Absolutwert einer Variablen
7	Invers-Wert (0-Variable)
8	Addition $V3 = V1 + V2$
9	Subtraktion $V3 = V1 - V2$
10	Multiplikation $V3 = V1 * V2$
11	Division $V3 = V1 / V2$
12	Konstantenzuweisung mit vordefinierten Konstanten
13	Vorkomma-Anteil einer Variablen (INT)
14	Nachkomma-Anteil einer Variablen (FRAC)
15	Vergleich zweier Variablen
16	Variable kopieren
17	Variable / 2
18	Variable * 2
19	Script im Arrayspeicher starten

1.2.1 Funktion 0 : Dummy

Dies ist eine Dummy-Funktion, die lediglich 0 zurückliefert falls die Bibliothek aus dem Hauptmenu aufgerufen wird.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	0

1.2.2 Funktion 1 : Initialisierung

Diese Funktion muss vor allen anderen Funktionen aufgerufen werden, um die notwendigen Initialisierungen vorzunehmen. Der Rückgabewert entspricht der Anzahl der Variablen, die eingerichtet wurden und hängt von der Zahl der Vor- und Nachkommastellen ab.

Parameter	Bedeutung
Parameter 1	Anzahl der Vorkomma-Bytes - 1 (0...15)
Parameter 2	Anzahl der Nachkomma-Bytes
Parameter 3	—
Rückgabewert	Anzahl der initialisierten Variablen

1.2.3 Funktion 2 : Wertzuweisung von Integerwert

Setzt eine Variable auf einen Integerwert. Die Nachkommastellen werden auf 0 gesetzt

Parameter	Bedeutung
Parameter 1	Variablennummer
Parameter 2	Variablenwert
Parameter 3	—
Rückgabewert	0

1.2.4 Funktion 3 : Wertzuweisung von Textstring

Setzt eine Variable auf einen durch einen Textstring im Array definierten Wert. Als Trennzeichen zwischen Vorkomma- und Nachkommastellen kann sowohl das Komma als auch der Punkt genutzt werden. Beim ersten nicht-numerischen Zeichen im String wird die Konvertierung abgebrochen.

Parameter	Bedeutung
Parameter 1	Variablennummer
Parameter 2	Textstart im Array
Parameter 3	—
Rückgabewert	0

1.2.5 Funktion 4 : Integerwert der Variablen

Gibt den Integerwert der Variablen aus. Dabei wird entsprechend der Nachkommastellen gerundet.

Parameter	Bedeutung
Parameter 1	Variablennummer
Parameter 2	—
Parameter 3	—
Rückgabewert	Integer-Wert der Variablen

1.2.6 Funktion 5 : Textausgabe der Variablen

Gibt den Inhalt der Variablen als Textstring aus. Mit der Angabe von minimalen Vorkommabytes kann eine rechtsbündige Darstellung bewirkt werden. Ist die Anzahl der maximalen Nachkommabytes kleiner als die der Variablen, wird entsprechend gerundet.

Parameter	Bedeutung
Parameter 1	Variablennummer
Parameter 2	Bit 7...4: minimale Vorkommabytes -1 Bit 3...0: Nachkommabytes
Parameter 3	(optional Ausgabekanal)
Rückgabewert	0

1.2.7 Funktion 6 : Absolutwert einer Variablen

Der Absolutwert der Variable 1 wird in die Variable 2 gespeichert. $V2 = ABS(V1)$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.8 Funktion 7 : Invers-Wert einer Variablen

Der Invers-Wert von Variable 1 wird in die Variable 2 gespeichert. $V2 = -V1$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.9 Funktion 8 : Addition

Die Summe von Variable 1 und Variable 2 wird in die Variable 3 gespeichert. $V3 = V1 + V2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	Variablennummer von Variable 3
Rückgabewert	0

1.2.10 Funktion 9 : Subtraktion

Die Differenz von Variable 1 und Variable 2 wird in die Variable 3 gespeichert. $V3 = V1 - V2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	Variablennummer von Variable 3
Rückgabewert	0

1.2.11 Funktion 10 : Multiplikation

Das Produkt von Variable 1 und Variable 2 wird in die Variable 3 gespeichert. $V3 = V1 * V2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	Variablennummer von Variable 3
Rückgabewert	0

1.2.12 Funktion 11 : Division

Der Quotient von Variable 1 und Variable 2 wird in die Variable 3 gespeichert. $V3 = V1 / V2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	Variablennummer von Variable 3
Rückgabewert	0

1.2.13 Funktion 12 : Konstantenzuweisung

Für 8 vordefinierte Konstanten lässt sich die Wertzuweisung über diese Funktion vereinfachen.

Parameter	Bedeutung
Parameter 1	Variablennummer
Parameter 2	Konstantennummer (0...7)
Parameter 3	—
Rückgabewert	0

Die 8 Konstanten haben 15 gültige Nachkommastellen und sind wie folgt definiert:

Konstantennummer	Wert
0	0
1	1
2	PI
3	e (Eulersche Zahl)
4	ln(2)
5	ln(10)
6	SQRT(2)
7	SQRT(3)

1.2.14 Funktion 13 : Vorkomma-Anteil einer Variablen

Der Vorkomma-Anteil der Variable 1 wird in die Variable 2 gespeichert. Genau genommen werden nur die Nachkomma-Bytes auf 0 gesetzt $V2 = INT(V1)$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.15 Funktion 14 : Nachkomma-Anteil einer Variablen

Der Vorkomma-Anteil der Variable 1 wird in die Variable 2 gespeichert. Genaugenommen werden nur die Vorkomma-Bytes auf 0 gesetzt $V2 = \text{FRAC}(V1)$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.16 Funktion 15 : Vergleich zweier Variablen

Variable 1 wird mit Variable 2 verglichen. Ist Variable 1 größer als Variable 2, dann ist der Rückgabewert 1, ist Variable 1 kleiner als Variable 2, ist das Ergebnis 2 und bei Gleichheit beider Variablen wird 0 zurückgeliefert.

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	Vergleichsergebnis (1,0,2)

1.2.17 Funktion 16 : Variable kopieren

Variable 1 wird in die Variable 2 kopiert.

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.18 Funktion 17 : Multiplikation mit 2

Variable 1 wird durch Bitschieben mit 2 multipliziert und das Ergebnis in die Variable 2 gespeichert. Diese Funktion ist wesentlich schneller als eine entsprechende Multiplikation. $V2 = V1 * 2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.19 Funktion 18 : Division durch 2

Variable 1 wird durch Bitschieben durch 2 dividiert und das Ergebnis in die Variable 2 gespeichert. Diese Funktion ist wesentlich schneller als eine entsprechende Division. $V2 = V1 / 2$

Parameter	Bedeutung
Parameter 1	Variablennummer von Variable 1
Parameter 2	Variablennummer von Variable 2
Parameter 3	—
Rückgabewert	0

1.2.20 Funktion 19 : Starten eines Scripts im Array

Da die Bibliothek im Wesentlichen nur die Grundrechenarten bereitstellt, gibt es eine kleine Script-Engine mit der auch komplexere Funktionen relativ einfach berechnet werden können. Dazu muss der Code im ersten Drittel des Arrays (Zellen 0...255) stehen.

Parameter	Bedeutung
Parameter 1	Startadresse im Array
Parameter 2	opt. Parameter 1
Parameter 3	opt. Parameter 2
Parameter 4	opt. Parameter 3
Rückgabewert	Rückgabewert, je nach Exit-Befehl

Für Schleifen etc. stehen vier 8-Bit Countervariablen zur Verfügung Jeder Befehl besteht aus 2 Bytes, die noch in Halbbytes (Nibbles) unterteilt sind. Ein x steht dafür, dass der Wert des entsprechenden Nibbles nicht ausgewertet wird. Trotzdem ist es sinnvoll diese Nibbles auf 0 zu setzen. Für Schleifen etc. stehen vier 8-Bit Countervariablen zur Verfügung, beim Start eines Scripts werden die ersten drei mit den LOW-Bytes der Parameter 1...3 vorbelegt, Countervariable 4 wird auf 0 gesetzt. Systembedingt haben nur Parameter die angegeben wurden einen definierten Wert. Bei einem **CALL 7,19,17** ist nur der Parameter 1 auf 17 gesetzt, Parameter 2 und 3 sind undefiniert.

Code	Bedeutung
0 0 0 V	Script beenden, Variablenwert (Integer) von Variable v wird zurückgegeben
0 1 x M	Script beenden, der Wert von Counter M wird zurückgegeben
0 2 n n	E wird zur Script-Position nn gesprungen
0 3 C V	Variable V wird mit der Konstanten C initialisiert
0 4 x V	Variable V wird mit dem Wert des optionalen Parameters 1 initialisiert
0 5 x V	Variable V wird mit dem Wert des optionalen Parameters 2 initialisiert
0 6 x V	Variable V wird mit dem Wert des optionalen Parameters 3 initialisiert
1 0 W V	$W = V$
1 1 W V	$W = \text{ABS}(V)$
1 2 W V	$W = -V$
1 3 W V	$W = \text{INT}(V)$
1 4 W V	$W = \text{FRAC}(V)$
1 5 W V	$W = V * 2$
1 6 W V	$W = V / 2$
1 8 W V	überspringt die nächste Anweisung, wenn V gleich W ist
1 9 W V	überspringt die nächste Anweisung, wenn V ungleich W ist
1 A W V	überspringt die nächste Anweisung, wenn V größer W ist
1 B W V	überspringt die nächste Anweisung, wenn V kleiner W ist
4 U W V	$W = U + V$
5 U W V	$W = U - V$
6 U W V	$W = U * V$
7 U W V	$W = U / V$
8 M n n	der Counter M wird auf nn gesetzt
9 M n n	zum Counter M wird nn addiert
A M n n	überspringt die nächste Anweisung, wenn der Counter M gleich nn ist
B M n n	überspringt die nächste Anweisung, wenn der Counter M ungleich nn ist
C M x n	der Counter M wird mit Parameter N initialisiert
andere Codes	Keine Funktion (NOP)

1.3 Ein Fraktalprogramm als Beispiel

In den Zeilen 7 bis 10 werden die Koordinaten für X1, X2, Y1 und Y2 eingetragen.



```
PROGRAM :Fractal 2
01 LFIND L,16:A=120:B=76
02 IF L=0 ? "keine Matlib":END
03 A=120:B=76:C=4:VM 2:VID 0
04 Z=80:PAL 0,0,1,5,7
05 CALL L,1,1,3:CALL L,2,16,4
06 'Koordinatenvorgaben
07 DA 0,"-0.95#":CALL L,3,0,0
08 DA 0,"-0.72#":CALL L,3,1,0
09 DA 0,"0.17#":CALL L,3,2,0
10 DA 0,"0.33#":CALL L,3,3,0
11
12 'DX(V4) und DY(V5) berechnen
13 CALL L,2,6,A
14 CALL L,9,1,0,7:CALL L,11,7,6,4
15 CALL L,2,6,B
16 CALL L,9,3,2,7:CALL L,11,7,6,5
17 CALL L,16,0,15
18 TSET 0
19 FOR Y=0 TO B-1:FOR X=0 TO A-1
20 CALL L,12,8,0:CALL L,12,9,0
21 I=0
22 CALL L,10,8,8,10:'X^2
23 CALL L,10,9,9,11:'Y^2
24 CALL L,8,10,11,6:CALL L,15,6,16
25 R=~R:IF R=1 GOTO 34
26 CALL L,9,10,11,13:'XT
27 CALL L,8,0,13,13
28 CALL L,10,8,9,14:'YT
29 CALL L,17,14,14:CALL L,8,2,14,14
30 CALL L,16,13,8:CALL L,16,14,9
31 'Iteration zu ende
32 I=I+1:IF I<Z GOTO 22
33 PLOT Y,X,0:GOTO 35
34 D=I%(C-1):PLOT Y,X,D+1
35 IF KEY(0)=$F5 THEN VID 0
36 IF KEY(0)=$F6 THEN VID 1
37 CALL L,8,0,4,0:NEXT
38 CALL L,8,2,5,2
39 CALL L,16,15,0:NEXT
40 TGET S
41 ? @0,0;(S-T)/10;"s ":VID 1
42 GOTO 41
#
```

2 Die Messroutinen-Bibliothek MEASLIB (Bibliothekscode 0x12)

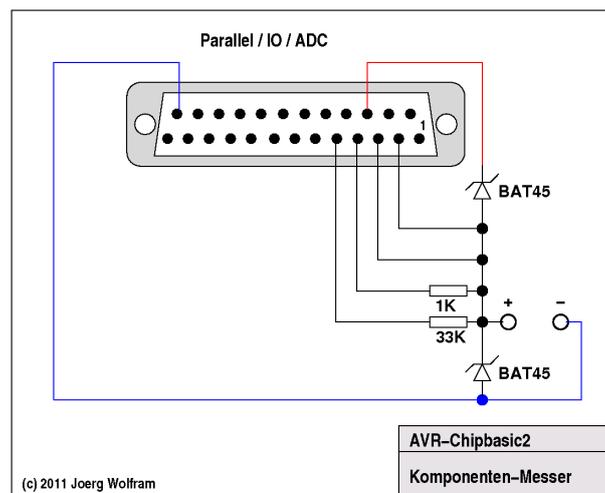
2.1 Allgemeines

Diese Bibliothek erweitert den BASIC-Computer um Messroutinen für Widerstände, Dioden, Kondensatoren und Induktivitäten in verschiedenen Messbereichen. Mit der unten angegebenen Schaltung werden folgende Messbereiche erreicht:

- Widerstandsmessbereich von 5 Ohm bis 1 MOhm
- Dioden-Flussspannung bis 4,88V, maximaler Strom 5mA
- Kapazitäts-Messbereich von 100pF bis 1000uF
- Induktivitäts-Messbereich von 20uH bis 800mH
- Platz für eigenes Messprogramm

2.2 Benötigte Hardware

Auch wenn die benötigte Hardware eher minimalistisch ist, ganz ohne geht es leider nicht. Benutzt werden nur die vier unteren Pins von Port A, so kann auch gleichzeitig ein LCD Modul zur Anzeige angeschlossen werden. Die notwendige Anschlussbeschaltung beschränkt sich auf zwei Widerstände und zwei Schottky-Dioden zum Schutz der Portpins. Im Folgenden werden die die Pins als PA.0 bis PA.3 bezeichnet. Wenn das Ganze direkt (ohne ChipBasic-Board) aufgebaut werden soll, sollten an die vier Portpins zusätzlich 180 Ohm Widerstände in Reihe geschaltet werden.



2.3 Funktionsübersicht

Funktionsnummer	Funktion
0	Start des Demonstrationsprogramms (aktuell nicht genutzt)
1	PA.1 auf HIGH, Spannungsmessung an PA.0 und PA.1
2	PA.2 auf HIGH, Spannungsmessung an PA.0 und PA.2
3	PA.3 auf HIGH, Spannungsmessung an PA.0 und PA.3
4	PA.3 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) > U_{\text{ref}}$ (Auflösung 300ns)
5	PA.3 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) > U_{\text{ref}}$ (Auflösung 500ns)
6	PA.2 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) > U_{\text{ref}}$ (Auflösung 300ns)
7	PA.2 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) > U_{\text{ref}}$ (Auflösung 64us)
8	PA.1 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 300ns)
9	PA.1 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 500ns)
10	PA.2 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 500ns)
11	PA.3 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 300ns)

2.3.1 Funktion 0 : Demo

Startet das eingebaute Demonstrationsprogramm. In der aktuellen Version besteht dieses nur aus einem "RET"

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	0

2.3.2 Funktion 1 : PA.1 auf HIGH, Spannungs-Messung an PA.0 und PA.1

Setzt die Portpins von Port A und führt Messungen mit dem ADC an den Pins PA.0 und PA.1 durch. Zur Verbesserung der Genauigkeit werden die Ergebnisse von jeweils 32 Einzelmessungen aufsummiert. Mit der Beispielschaltung lassen sich so Widerstände von ca. 5 Ohm bis 5 KOhm mit akzeptabler Genauigkeit messen. Der Ablauf ist dabei Folgender:

- Die Pins PA.0, PA.2 und PA.3 werden als Eingang ohne Pullup geschaltet, PA.1 als Ausgang mit HIGH Pegel
- Spannung an Pin PA.1 mit AVCC als Referenz 32 mal messen und die Werte aufsummieren
- Spannung an Pin PA.0 mit AVCC als Referenz 32 mal messen und die Werte aufsummieren
- PA.1 auf LOW Pegel setzen

Rückgabewert ist der Messwert von PA.0, der Messwert von PA.1 wird in die Arrayzelle 1024 abgelegt.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	32 x Messwert von PA.0, Messwert von PA.1 in AR(1024)

2.3.3 Funktion 2 : PA.2 auf HIGH, Spannungs-Messung an PA.0 und PA.2

Diese Funktion entspricht der Funktion 1 mit dem Unterschied, dass anstelle Pin PA.1 hier der Pin PA.2 als Quelle verwendet wird und in AR(1024) der Messwert von Pin PA.2 steht.

2.3.4 Funktion 3 : PA.3 auf HIGH, Spannungs-Messung an PA.0 und PA.3

Diese Funktion entspricht der Funktion 1 mit dem Unterschied, dass anstelle Pin PA.2 hier der Pin PA.3 als Quelle verwendet wird und in AR(1024) der Messwert von Pin PA.3 steht.

2.3.5 Funktion 4 : PA.3 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) > U_{\text{ref}}$ (Auflösung 300ns)

Pin PA.0 wird mit dem Analog-Komparator verbunden, als Vergleichsspannung dient die interne Bandgap-Referenzquelle mit ca. 1,1 Volt. Der Pin PA.3 wird von 0 auf 1 geschaltet und die Zeit gemessen, bis die Spannung an PA.0 größer als die Spannung der Referenzspannungsquelle ist. Die zeitliche Auflösung beträgt 300ns, der maximale Messbereich 60 Mikrosekunden. Es werden 16 Messungen im Abstand von 20ms durchgeführt und die Werte aufaddiert. Der maximal zurückgelieferte Wert beträgt somit 3200. Der Ablauf ist dabei Folgender:

- Die Pins PA.0, PA.1 und PA.2 werden als Eingang ohne Pullup geschaltet, PA.3 als Ausgang mit LOW Pegel
- PA.3 AUF HIGH schalten und die Zeit bestimmen, bis die Spannung an PA.0 $> U_{\text{ref}}$ ist.
- Nach 60 Mikrosekunden die Messung abbrechen und PA.3 auf LOW Pegel setzen
- Das Ganze 15 mal wiederholen

Rückgabewert ist die Summe der einzelnen Zeitmessungen.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	16 x Messwert (in 300 Nanosekunden-Schritten)

2.3.6 Funktion 5 : PA.3 von LOW nach HIGH, Zeitmessung bis U(PA.0) > Uref (Auflösung 500ns)

Diese Funktion entspricht der Funktion 4, allerdings mit einer anderen Zeitauflösung und einem größeren Messbereich. Die zeitliche Auflösung beträgt 500ns, der maximale Messbereich 6,4 Millisekunden. Es werden 16 Messungen im Abstand von 20ms durchgeführt und die Werte aufaddiert. Zum Schluss wird das Ergebnis durch 8 geteilt. Der maximal zurückgelieferte Wert beträgt somit 25600. Der Ablauf ist dabei folgender:

- Die Pins PA.0, PA.1 und PA.2 werden als Eingang ohne Pullup geschaltet, PA.3 als Ausgang mit LOW Pegel
- PA.3 AUF HIGH schalten und die Zeit bestimmen, bis die Spannung an PA.0 > Uref ist.
- Nach 6,4 Millisekunden die Messung abbrechen und PA.3 auf LOW Pegel setzen
- Das Ganze 15 mal wiederholen, die Messwerte aufaddieren
- Die Summe der Messwerte durch 8 teilen.

Rückgabewert ist die Summe der einzelnen Zeitmessungen geteilt durch 8.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	2 x Messwert (in 500 Nanosekunden-Schritten)

2.3.7 Funktion 6 : PA.2 von LOW nach HIGH, Zeitmessung bis U(PA.0) > Uref (Auflösung 500ns)

Diese Funktion entspricht der Funktion 5, der LOW-HIGH Übergang findet aber an Pin PA.2 statt.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	2 x Messwert (in 500 Nanosekunden-Schritten)

2.3.8 Funktion 7 : PA.2 von LOW nach HIGH, Zeitmessung bis U(PA.0) > Uref (Auflösung 64us)

Diese Funktion entspricht der Funktion 5, der LOW-HIGH Übergang findet an Pin PA.2 statt. Die zeitliche Auflösung beträgt 64us (Zeilenfrequenz), der maximale Messbereich 262 Millisekunden. Es werden 2 Messungen im Abstand von 20ms durchgeführt und die Werte aufaddiert. Der maximal zurückgelieferte Wert beträgt somit 8192. Der Ablauf ist dabei folgender:

- Die Pins PA.0, PA.1 und PA.3 werden als Eingang ohne Pullup geschaltet, PA.2 als Ausgang mit LOW Pegel
- PA.2 AUF HIGH schalten und die Zeit bestimmen, bis die Spannung an PA.0 > Uref ist.
- Nach 262 Millisekunden die Messung abbrechen und PA.3 auf LOW Pegel setzen
- Das Ganze 1 mal wiederholen

Rückgabewert ist die Summe der einzelnen Zeitmessungen.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	2 x Messwert (in 64 Mikrosekunden-Schritten)

2.3.9 Funktion 8 : PA.1 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 300ns)

Pin PA.0 wird mit dem Analog-Komparator verbunden, als Vergleichsspannung dient die interne Bandgap-Referenzquelle mit ca. 1,1 Volt. Der Pin PA.1 wird von 0 auf 1 geschaltet und die Zeit gemessen, bis die Spannung an PA.0 wieder kleiner als die Spannung der Referenzspannungsquelle ist. Die zeitliche Auflösung beträgt 300ns, der maximale Messbereich 60 Mikrosekunden. Es werden 16 Messungen im Abstand von 20ms durchgeführt und die Werte aufaddiert. Der maximal zurückgelieferte Wert beträgt somit 3200. Der Ablauf ist dabei folgender:

- Die Pins PA.0, PA.2 und PA.3 werden als Eingang ohne Pullup geschaltet, PA.1 als Ausgang mit LOW Pegel
- PA.1 AUF HIGH schalten und 300ns warten
- Die Zeit bestimmen, bis die Spannung an PA.0 $< U_{\text{ref}}$ ist.
- Nach 60 Mikrosekunden die Messung abbrechen und PA.1 auf LOW Pegel setzen
- Das Ganze 15 mal wiederholen

Rückgabewert ist die Summe der einzelnen Zeitmessungen.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	16 x Messwert (in 300 Nanosekunden-Schritten)

2.3.10 Funktion 9 : PA.1 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 500ns)

Diese Funktion entspricht der Funktion 4, allerdings mit einer anderen Zeitauflösung und einem größeren Messbereich. Die zeitliche Auflösung beträgt 500ns, der maximale Messbereich 6,4 Millisekunden. Es werden 16 Messungen im Abstand von 20ms durchgeführt und die Werte aufaddiert. Zum Schluss wird das Ergebnis durch 8 geteilt. Der maximal zurückgelieferte Wert beträgt somit 25600. Der Ablauf ist dabei folgender:

- Die Pins PA.0, PA.2 und PA.3 werden als Eingang ohne Pullup geschaltet, PA.1 als Ausgang mit LOW Pegel
- PA.1 AUF HIGH schalten und 300ns warten
- Die Zeit bestimmen, bis die Spannung an PA.0 $< U_{\text{ref}}$ ist.
- Nach 6,4 Millisekunden die Messung abbrechen und PA.1 auf LOW Pegel setzen
- Das Ganze 15 mal wiederholen, die Messwerte aufaddieren
- Die Summe der Messwerte durch 8 teilen.

Rückgabewert ist die Summe der einzelnen Zeitmessungen geteilt durch 8.

Parameter	Bedeutung
Parameter 1	—
Parameter 2	—
Parameter 3	—
Rückgabewert	2 x Messwert (in 500 Nanosekunden-Schritten)

2.3.11 Funktion 10 : PA.2 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 500ns)

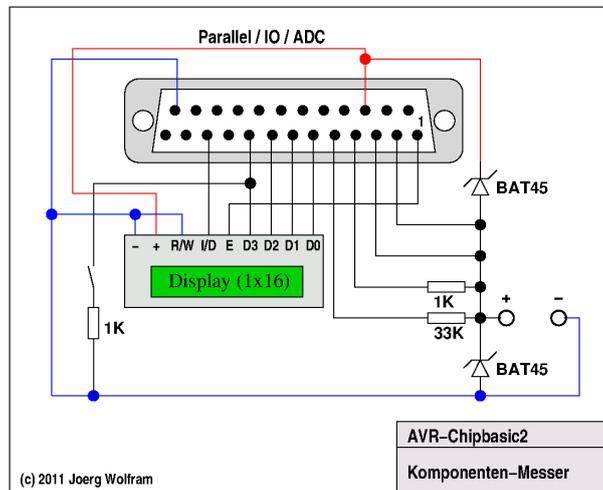
Diese Funktion entspricht der Funktion 9, der LOW-HIGH Übergang findet aber an Pin PA.2 statt.

2.3.12 Funktion 11 : PA.3 von LOW nach HIGH, Zeitmessung bis $U(\text{PA.0}) < U_{\text{ref}}$ (Auflösung 500ns)

Diese Funktion entspricht der Funktion 9, der LOW-HIGH Übergang findet aber an Pin PA.3 statt.

2.4 Ein Anwendungsbeispiel

Das nachfolgende Beispiel zeigt die komplette Beschaltung inclusive LCD und einem Taster zur Messbereichsumschaltung. Dieses könnte man natürlich auch standalone aufbauen und das Programm System mittels des Clone-Programmes oder den Loader übertragen. Das gäbe dann ein kompaktes Messgerät, natürlich nur wenn die Genauigkeit ausreichend ist. Neben der Mess-Bibliothek wird noch ein LCD-Treiber benötigt, das LCD selbst muss nicht unbedingt angeschlossen sein da parallel die Ausgabe über TV erfolgt.



Mit dem nachfolgenden Programm ergeben sich dann folgende Messbereiche, die mittels des Tasters gewählt werden können.

Messart	Messbereich
Widerstand	10 Ω ... 1M Ω
Diode	0 ... 4,8V
Kapazität	100pF ... 1000 μ F
Induktivität	10 uH ... 100mH

In den Zeilen 9 bis 13 sind die Skalierungsfaktoren abgelegt. Diese wurden empirisch mit Referenzbauteilen ermittelt, hängen aber zu einem guten Teil von Fertigungstoleranzen bei den eingesetzten Bauteilen (inclusive Controller) ab. Die folgende Tabelle gibt eine Zuordnung über die Skalierungswerte zu den Messbereichen, zum Kalibrieren sollte ein Bauteil genommen werden, welches wertemäßig ungefähr in der Mitte des Messbereiches liegt. Die angegebenen Messbereiche sind nur als Ungefährwerte anzusehen:

Adresse	Zeile	Messbereich
1026	9	Widerstände $6,5\text{K}\Omega < R < 1\text{M}\Omega$
1027	9	Widerstände $200\Omega < R < 6,5\text{K}\Omega$
1028	9	Widerstände $10\Omega < R < 200\Omega$
1029	10	Diodenspannung $0\text{V} < U < 4,8\text{V}$
1030,1031 (Endwert,Offset)	11	Kondensatoren $0,1\text{nF} < C < 72\text{nF}$
1032,1033 (Endwert,Offset)	11	Kondensatoren $72\text{nF} < C < 740\text{nF}$
1034,1035 (Endwert,Offset)	12	Kondensatoren $740\text{nF} < C < 22\mu\text{F}$
1036,1037 (Endwert,Offset)	12	Kondensatoren $22\mu\text{F} < C < 1000\mu\text{F}$
1038	13	Induktivitäten $50\mu\text{H} < L < 6,7\text{mH}$
1038	13	Induktivitäten $6,7\text{mH} < L < 830\text{mH}$

PROGRAM: Mess4

```

01 E=30:F=31:LFIND V,160
02 IF V=0 GOTO ~L+2
03 VM 7:COLOR 1,5:E=8:F=9
04 LFIND L,18:IF L>0 GOTO ~L+2
05 ? "Keine Messlib ";:END
06 GOSUB 89
07 ? "CB2 KOMPO-Meter":WAIT 20
08 'Korrektur-Faktoren
09 DATA 1026,3320,12000,2150
10 DATA 1029,5000
11 DATA 1030,740,96,7660,6
12 DATA 1034,2190,0,1043,0
13 DATA 1038,6710,8300
14 OUT $47E,15:'disable INPUT
15 GOSUB 25:GOSUB 80:'R
16 IF T=0 GOTO ~L-1
17 GOSUB 41:GOSUB 80:'D
18 IF T=0 GOTO ~L-1
19 GOSUB 48:GOSUB 80:'C
20 IF T=0 GOTO ~L-1
21 GOSUB 70:GOSUB 80:'L
22 IF T=0 GOTO ~L-1
23 GOTO ~L-8
24 'R Messung
25 CALL L,3
26 X=~R:N=AR(1024)-X:Y=AR(1026)
27 IF X>31000 GOTO ~L+4
28 IF X<200 GOTO ~L+4
29 X=X/10:SCALE R,0,Y,0,X,N:CLS
30 ? "(R) ";!$1D;R;" K";%F:RET
31 CLS :? "(R) > 1 M";%F:RET
32 CALL L,2
33 X=~R:N=AR(1024)-X:Y=AR(1027)
34 IF X<200 GOTO ~L+3
35 X=X/10:SCALE R,0,Y,0,X,N:CLS
36 ? "(R) ";!$0D;R;" ";%F:RET
37 CALL L,1
38 X=~R:N=AR(1024)-X:Y=AR(1028)
39 SCALE R,0,Y,0,X,N:CLS
40 ? "(R) ";!$1D;R;" ";%F:RET
41 'D-Messung
42 CALL L,2:Y=AR(1029)
43 X=~R:IF X>32000 GOTO ~L+3
44 SCALE R,0,Y,0,X,32767:CLS
45 ? "(D) ";!$3D;R;" V":RET
46 CLS :? "(D) > 4.88 V":RET

```

```

47 'C-Messung
48 CALL L,4
49 Y=AR(1030):O=AR(1031)
50 X=~R:IF X>3199 GOTO ~L+3
51 SCALE R,0,Y,0,X-0,32000:CLS
52 ? "(C) ";!$1D;R;" nF":RET
53 CALL L,5
54 Y=AR(1032):O=AR(1033)
55 X=~R:IF X>25000 GOTO ~L+3
56 SCALE R,0,Y,0,X-0,25600:CLS
57 ? "(C) ";!$1D;R;" nF":RET
58 CALL L,6
59 Y=AR(1034):O=AR(1035)
60 X=~R:IF X>25000 GOTO ~L+3
61 SCALE R,0,Y,0,X-0,25600:CLS
62 ? "(C) ";!$2D;R;" ";%E;"F"
63 CALL L,7
64 Y=AR(1036):O=AR(1037)
65 X=~R:IF X>4090 GOTO ~L+3
66 SCALE R,0,Y,0,X-0,4096:CLS
67 ? "(C) ";!$0D;R;" ";%E;"F":RET
68 CLS :? "(C) OVERFLOW ":RET
69 'L-Messung
70 ? @0,0;:CALL L,8:Y=AR(1038)
71 X=~R:IF X>3199 GOTO ~L+3
72 SCALE R,0,Y,0,X,3200:CLS
73 ? "(L) ";!$0D;R;" ";%E;"H":RET
74 ? @0,0;:CALL L,9:Y=AR(1039)
75 X=~R:IF X>25599 GOTO ~L+3
76 SCALE R,0,Y,0,X,25600:CLS
77 ? "(L) ";!$1D;R;" mH":RET
78 CLS :? "(L) OVERFLOW ":RET
79 'Test auf Tastendruck
80 I=0:T=0
81 'IF IN($807)=0 GOTO ~L-2
82 IF KEY(0)>0 GOTO ~L+3
83 IF IN($807)=0 GOTO ~L+2
84 RETURN
85 SYNC 1:I=I+1:IF I<50 GOTO ~L-3
86 T=1
87 RETURN
88 'my and omega symbols
89 DA 100,0,0,9,9,9,14,8,16
90 DA 108,0,0,14,17,17,10,27,0
91 FOR I=0 TO 15:J=I+100
92 OUT $A00+I,AR(J):NEXT
93 RETURN
#

```

3 Eine RTC Befehlsenerweiterung (Bibliothekscod 0x1F)

3.1 Allgemeines

Diese Demo-Bibliothek zeigt, wie der Befehlsumfang des BASIC-Computer erweitert werden kann. Dazu sind folgende Befehle implementiert:

3.2 Funktionsübersicht

Befehl	Funktion
_SETSEC n	setzt den Sekunden-Wert auf den Parameter n, dabei werden die Grenzen überwacht.
_SETMIN n	setzt den Minuten-Wert auf den Parameter n, dabei werden die Grenzen überwacht.
_SETHOUR n	setzt den Stunden-Wert auf den Parameter n, dabei werden die Grenzen überwacht.
_SETDAY n	setzt den Tage-Wert auf den Parameter n, dabei werden die Grenzen überwacht.
_SETMONTH n	setzt den Monats-Wert auf den Parameter n, dabei werden die Grenzen überwacht.
_SETYEAR n	setzt den Jahres-Wert auf den Parameter n (0..99)
_GETSEC V	liest den Sekunden-Wert in die Variable V ein
_GETMIN V	liest den Minuten-Wert in die Variable V ein
_GETHOUR V	liest den Stunden-Wert in die Variable V ein
_GETDAY V	liest den Tage-Wert in die Variable V ein
_GETMONTH V	liest den Monats-Wert in die Variable V ein
_GETYEAR V	liest den Jahres-Wert in die Variable V ein

Sämtliche Befehle haben nur eine Dummy-Funktion, eine Anbindung an eine konkrete RTC (z.B. über I2C) sollte aber kein Problem sein.