

Ich habe ein Problem:

anbei ein kleines Problem, bei dem ich wieder Deine Hilfe benötige:

Es gelingt mir nicht eine LED einzuschalten (das funktioniert noch) und mit der gleichen Taste wieder auszuschalten (Test-anaus-1.bas).

Ein abgewandeltes Programm (test-anaus.bas) funktioniert aber, da eine andere LED für das erneute Betätigen des Tasters verwendet wird.

Somit scheinen auch alle Bedingungen und Schleifen zu funktionieren.

Was mache ich falsch ? Übrigens, im Debug-Modus funktioniert es fehlerfrei - im "normalen" nicht!

hier das Programm Test-anaus-1.bas:

```
*****
'          LICHTAN-LICHTAUS
'          23.09.06
'
'          (c) Frank S.
*****
'Beim Programmstart leuchtet RB,5
'RC7 soll durch Betätigen von RC,5 eingeschaltet werden
'Wird RC,5 erneut betätigt, geht RB,5 aus sofern RC,7 high ist

        DEFINE device iL_TROLL

        INP RC,4
        LOW RC,7

Loop:
        IF RC,4 = 0 THEN GOTO Lichten
        IF RC,7 = 1 THEN GOTO Lichtaus
        GOTO Loop

Lichten:
        SET RC,7          'schaltet LED ein
        GOTO Loop

Lichtaus:
        IF RC,4 = 0 THEN GOTO Shutdown
        GOTO Loop

Shutdown:
        RES RC,7
        GOTO Loop
        END
```

Dein Programm könnte funktionieren, wenn der Troll-Chip nur nicht so schnell wäre. Jeder Programmschritt dauert ca. 1µs und ein Sprung dauert ca. 2µs. Die LED RC,7 ist demnach nach circa.....

```
IF RC,4 = 0 THEN GOTO Lichtan           3µs
    IF RC,7 = 1 THEN GOTO Lichtaus
    GOTO Loop
```

```
Lichtan:
    SET RC,7           'schaltet LED ein    1µs

    GOTO Loop           2µs
```

```
Summe:           6µs
```

6µs eingeschaltet und da die Taste noch gedrückt ist (die Taste müsstest Du nach 2 µs wieder loslassen!!!!), wird die LED auch gleich wieder ausgeschaltet. Das menschliche Auge ist nicht geeignet diese schnellen Schaltvorgänge zu registrieren. Deshalb ist es nötig, das Programm erst nach Loslassen der Taste nach „loop“ springen zu lassen. Um das Programm dann noch perfekter zu schreiben, habe ich auch noch die Tasten entprellt (siehe im Buch PIC-Microcontroller Programmierung Seite 179-181.

hier die ergänzte Programmversion Test-anaus-1a.bas:

```
*****
'          LICHTAN-LICHTAUS
'          Test-anaus-1a.bas
'          23.09.06
'
'          (c) Frank S. , Wolfram H.
*****
```

```
'Beim Programmstart leuchtet RB,5
'RC7 soll durch Betätigen von RC,5 eingeschaltet werden
'Wird RC,5 erneut betätigt, geht RC,7 aus
```

```
*****
' 1.)
' Du siehst in diesem Programm einige Programmzeilen, die mehrfach identisch
' im Programm vorkommen. Diese redundanten Programmzeilen lassen sich
' in Unterprogrammen (Subroutinen) zusammenfassen. Die Verwendung von Sub-
' routinen spart kostbaren Programmspeicher. Versuch mal dieses Programm durch
' Subroutinen zu verkürzen.
'
' 2.)
' Mit dem Basicbefehl "toggle" könnte das Programm weiter vereinfacht werden.
' Versuche eine Programmversion mit dem "toggle"-Befehl zu schreiben.
'
' 3.)
```

' In der Hauptschleife "LOOP" habe ich die Reihenfolge der IF-Abfragen
' geändert, warum? Kannst Du das herausfinden? Tipp, es hat mit dem Kontaktprellen
' zu tun.

```
DEFINE device iL_TROLL
```

```
INP RC,4  
LOW RC,7  
HIGH RB,5
```

Loop:

```
IF RC,7 = 1 THEN GOTO Lichtaus  
IF RC,4 = 0 THEN GOTO Lichtan
```

```
GOTO Loop
```

Lichtan:

```
*****
```

```
'Entprellen der Taste RC,4 *
```

```
wait 10 'warte 10ms *
```

```
IF RC,4 = 1 THEN GOTO loop 'Taste noch gedrückt?, wenn nicht weiter loopen
```

```
*****
```

```
SET RC,7 'schaltet LED ein
```

```
*****
```

```
'Warte bis Taste wieder losgelassen wurde *
```

Lichtan1: ' *

```
wait 10 'warte 10ms *
```

```
if RC,4 = 0 THEN GOTO Lichtan1 ' loope, wenn Taste noch gedrückt
```

```
*****
```

```
GOTO Loop 'fertig, goto loop
```

Lichtaus:

```
IF RC,4 = 0 THEN GOTO Shutdown
```

```
GOTO Loop
```

Shutdown:

```
*****
```

```
'Entprellen der Taste RC,4 *
```

```
wait 10 'warte 10ms *
```

```
IF RC,4 = 1 THEN GOTO loop 'Taste noch gedrückt?, wenn nicht weiter loopen
```

```
*****
```

```
RES RC,7
```

```

*****
'Warte bis Taste wieder losgelassen wurde      *
Shutdown1:      '
wait 10         'warte 10ms      *
if RC,4 = 0 THEN GOTO Shutdown1 ' loope, wenn Taste noch gedrückt
*****

GOTO Loop
END

```

vielen Dank für Deinen Einsatz !!

Anbei die bezgl. Punkt 1 Deiner Anmerkungen geänderte Version zur freundlichen Prüfung.
Die Version mit dem Toggle-Befehl muß ich mir erst noch erarbeiten. Falls es keine Mühe macht, kannst Du mal Deine Version schicken ??

hier die geänderte Programmversion Licht-an-aus.bas:

```

*****
'          LICHTAN-LICHTAUS
'          licht-an-aus.bas
'          24.09.06
'
'          (c) Frank S. , Wolfram H.
*****

'Beim Programmstart leuchtet RB,5
'RC7 soll durch Betätigen von RC,5 ein- und ausgeschaltet werden
*****
' 1.)
'  Du siehst in diesem Programm einige Programmzeilen, die mehrfach identisch
'  im Programm vorkommen. Diese redundanten Programmzeilen lassen sich
'  in Unterprogramme (Subroutinen) zusammenfassen. Die Verwendung von Sub-
'  routinen spart kostbaren Programmspeicher. Versuch mal dieses Programm durch
'  Subroutinen zu verkürzen.
'
'
'          DAS SOLLTE DIE LÖSUNG SEIN
'
' 2.)
'  Mit dem Basicbefehl "toggle" könnte das Programm weiter vereinfacht werden.
'  Versuche eine Programmversion mit dem "toggle"-Befehl zu schreiben.
'
' 3.)
'  In der Hauptschleife "LOOP" habe ich die Reihenfolge der IF-Abfragen
'  geändert, warum? Kannst Du das herausfinden?
'
*****

```

```
DEFINE device iL_TROLL
```

```
INP RC,4  
LOW RC,7  
HIGH RB,5
```

Loop:

```
IF RC,7 = 1 THEN GOTO Lichtaus  
IF RC,4 = 0 THEN GOTO Lichtan  
GOTO Loop
```

Entprellen:

```
wait 10          'warte 10ms      *  
IF RC,4 = 1 THEN GOTO loop  'Taste noch gedrckt?, wenn nicht weiter loopen  
Return
```

Lichtan:

```
Gosub Entprellen  
SET RC,7        'schaltet LED ein
```

Lichtan1:

```
wait 10          'warte 10ms      *  
if RC,4 = 0 THEN GOTO Lichtan1 'loope, wenn Taste noch gedrckt
```

```
Goto Loop      'Ende der Routinen LICHTAN und LICHTAN1
```

Lichtaus:

```
IF RC,4 = 0 THEN GOTO Shutdown  
GOTO Loop
```

Shutdown:

```
Gosub Entprellen  
RES RC,7
```

Shutdown1:

```
wait 10          'warte 10ms      *  
if RC,4 = 0 THEN GOTO Shutdown1 'loope, wenn Taste noch gedrckt
```

```
GOTO Loop      'Ende der Routinen SHUTDOWN und SHUTDOWN1
```

```
END
```

Natürlich sende ich Dir gern „meine“ Lösung mit dem TOGGLE-Befehl zu.

hier „meine“ TOGGLE-Lösung Test-anaus2a.bas:

```
*****
'          LICHTAN-LICHTAUS
'          Test-anaus-2a.bas
'          23.09.06
'
'          (c) Wolfram H.
*****
```

'RC7 soll durch Betätigen von RC,4 eingeschaltet werden
'Wird RC,4 erneut betätigt, wird LED RC,7 ausgeschaltet.

```
DEFINE device iL_TROLL
```

```
INP RC,4
LOW RC,7
```

```
define S51=rc,4
define led=rc,7
```

start:

```
IF S51 = 0 THEN GOTO schalten
```

```
GOTO start
```

schalten:

```
*****
```

```
'Entprellen der Taste RC,4          *
```

```
wait 10          'warte 10ms      *
```

```
IF S51 = 1 THEN GOTO start 'Taste noch gedrückt?, wenn nicht weiter loopen
```

```
*****
```

```
TOGGLE led          'schaltet LED ein/aus
```

```
*****
```

```
'Warte bis Taste wieder losgelassen wurde      *
```

```
schalten1:          '          *
```

```
wait 10          'warte 10ms      *
```

```
IF S51 = 0 THEN GOTO schalten1 ' warte, wenn Taste noch gedrückt
```

```
*****
```

```
GOTO start          'fertig, goto start
```

```
END
```
