PIC-Batterie overload protection

1. Systemanforderungen

Um die Batterien durch den Betrieb von hohen Lasten (z. b. Kühlschrank) über einen langen Zeitraum nicht zu überlasten ist es notwendig den Betriebszeitraum abhängig vom festgelegten Batterie Spannungsgrenzen einzuschränken.

Für den Betrieb eines Kühlschranks werden folgende Grenzen festgelegt:

- Einschalten bei Ub >12V
- Ausschalten bei Ub < 12V
- Wiedereinschalten bei Ub > 12,5V

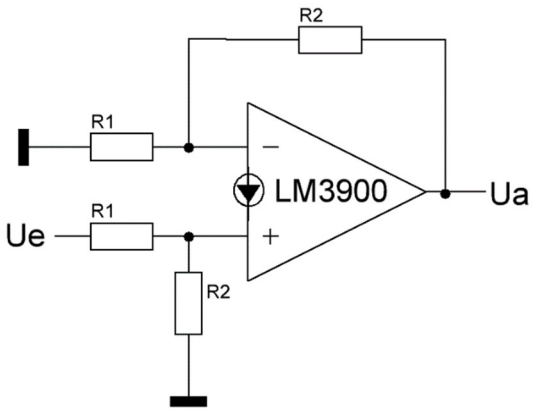
2. Systembeschreibung

Hierfür wird ein PIC 16F627A Mikrokontroller der Firma Microchip eingesetzt, der zur Batteriespannungsüberwachung einen AD-Wandler steuert, und ein Relais zu Ein- oder Abschaltung des Kühlschranks kontrolliert.

2.1 Eingangsspannungsverstärker

Als Eingangsverstärker wird der Norton Verstärker LM3900 wie nachfolgend beschrieben, verwendet.

Da die maximale Eingangsspannung am Norton Verstärker 15V nicht überschreite, wird er mit eine Versorgungsspannung 10<Vcc<15V betrieben. Unter Berücksichtigung der Tatsache, dass der nachfolgender AD- Wandler mit einer 5V Referenz betrieben wird, muss die Verstärkung des Norton Verstärkers **Ua/Ue** möglichst nahe an 1/3 eingestellt werden.



Gewählt werden deshalb:

R1 = 1M und **R2=330k**

Die Verstärkung beträgt: **Ua/Ue = 0,33**

Da die Verarbeitung von Eingangssignalen Ue<0,6V im vorliegenden Anwendungsfall nicht vorkommt, wird auf ein „Common mode biasing“ verzichtet. Schließlich, da der Ausgangspegel am Norton Verstärker Ausgang ebenfalls oberhalb von 0,6V liegen, wird auch auf die Einführung einer Ausgangsdiode verzichtet.

2.2 Ansteuerung des Kompressor Relais

Das Relais Rel1 besitzt eine 80 Ohm Spule und verursacht bei dem Treibertransistor T1 bei UD3=0,5V, UCEsat=0,5V und Uamax=12,75V einen Kollektorstrom von etwa 147 mA. Laut Datenblatt weist der Transistor 2N1711 bei Ic=147 mA eine hFE von etwa 130 und benötigt demnach einen IBmin von 1,13mA. Aus Sicherheitsgründen wird der Basisstrom um den Faktor 2,5 größer gewählt d.h. IB=2,825mA.

Damit erhält man einen Basiswiderstand: $R9 = \frac{RB0 - U_{BE} - UD3}{2{,}825} = 1{,}416k$

Gewählt wird: **R9 = 1,5k**

PIC-Batterie overload protection

3. PIC 16F627A

Der Microcontroller wird mit einem Quarz von 4,069 MHZ betrieben, welches zu einen instruction cycle von 0,976563µs führt.

Verwendet man als TMR0 Prescaler die Teilung 1/16, so erhält man ein Interrupt alle 0,976563 * 256 * 16 = 4msec.

3.1 Analog Digital Wandler

Eingesetzt wird der 8 bit AD- Wandler TLC 549 (Fmax = 1MHz, Min ADC recovery time 20µ), welcher mit eine Referenzspannung von 5V betrieben wird.

Dass die Eingangsspannung des Wandlers während der Messung sich nicht ändert, muss jede Messung in zwei aufeinander folgenden Routinen erfolgen.

4 Stückliste

C1 = 1 x 330n
C6 = 1 x 10µ
C2,C5 = 2 x 100n
C3,C4 = 2 x 15p

D1,D2,D3 = 3 x 1N4004

F1 = 1 x 1A
F2 = 1 x 15A

LED1,LED2 = 2 x

LM3900 = 1 x

OP1 = 1 x

Qz1 = 1 x 4,096 MHz

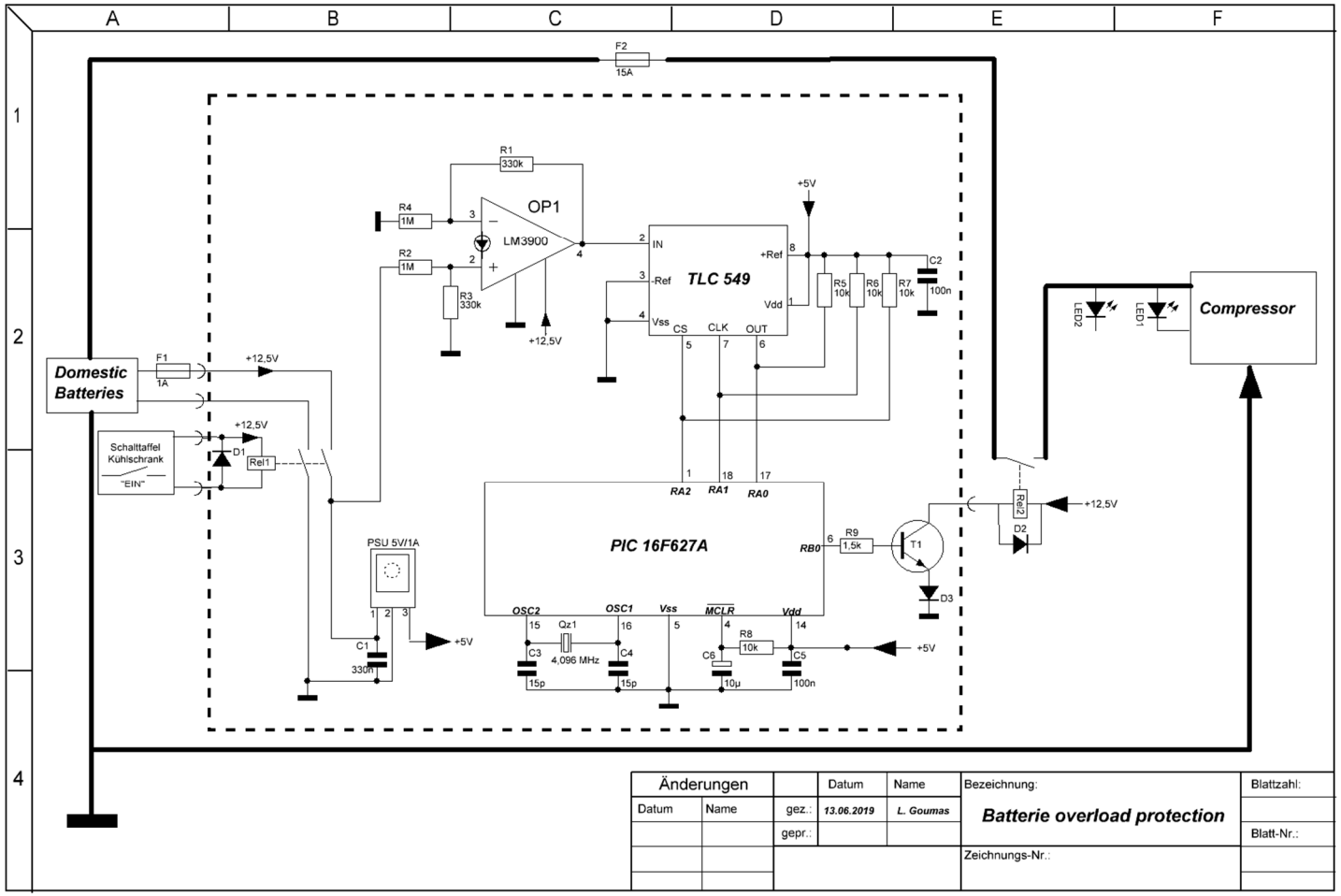
R9 = 1 x 1,5k
R1,R3 = 2 x 330k
R2,R4 = 2 x 1M
R5,R6,R7,R8 = 4 x 10k

Rel1 = 1 x 12V/960Ohm/Conrad505170
Rel2 = 1 x 12V/30A/80Ohm/conrad504209

T1 = 1 x 2N1711

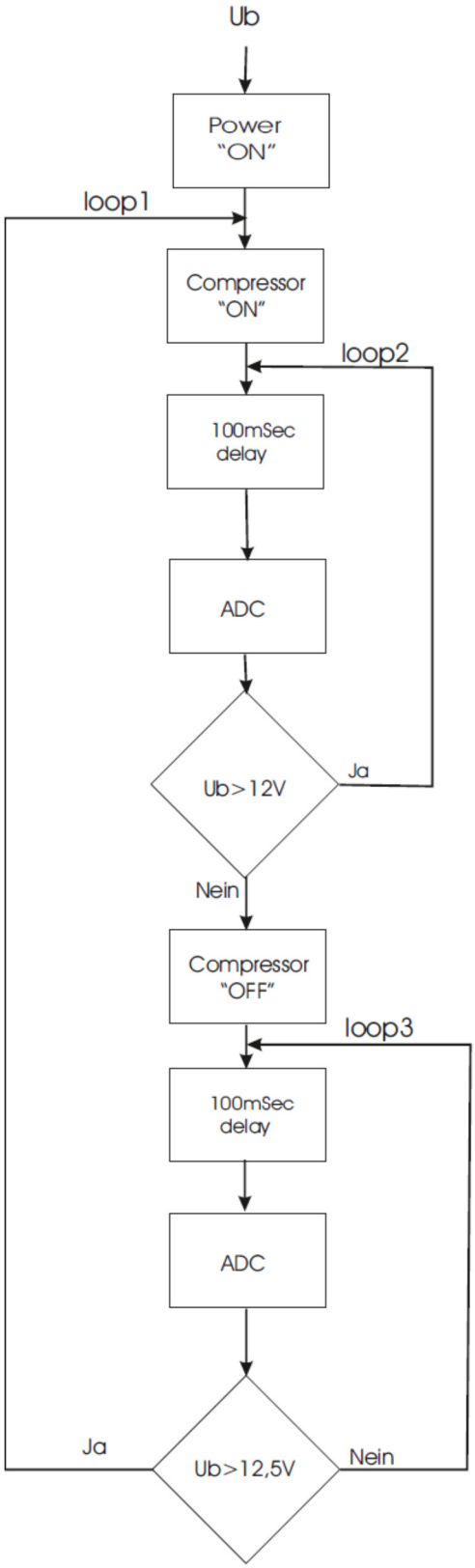
PIC-Batterie overload protection

5 Stromlaufplan



PIC-Batterie overload protection

6 Pseudo-Assembler Flussdiagramm



Batterie overload protection

PIC-Batterie overload protection

6.1 Assembler listing

```
;Die Betriebsdauer eines Kühlschranks soll abhängib von der Batteriespannung
;(Ub)überwacht werden. Der Kühlschrank soll nur eingeschaltet werden bei
;Ub > 12,5V und ausgeschaltet werden bei Ub <12V
;AD Wandler TLC549, Fmax = 1MHz, Min ADC recovery time 20µ
;Um Ub schwankungen abzudecken, AD-Wandlung zweimal nacheinander durchführen
;Eingang RA0, und Ausgänge RA1 und RA4 über Pull ups an VCC
;RA0 an "OUT", RA1 an CLK, RA4 an CS,
;RB0 Digital I/O
;Wenn RB4 als digitales I/O verwendet werden soll, dann: "LVP_OFF"
;f=4,096MHz / 0,976563µs,  (Osc > 500kHz d.h. XT_OSC), Prescaler 16
;PIC16F627A,
;L. Goumas / 15.06.2019
;*****************************************************************

;Assembler directives
   list,P16F627
   #include <p16F627A.inc>          ;Controller header file
   __CONFIG _CP_OFF & _PWRTE_OFF & _WDT_OFF & _XT_OSC  & _LVP_OFF
        ERRORLEVEL       -302            ;Supress BanK SELECTION MESSAGES
;*****************************************************************

;Zuordnung der verwendeten Register

w_copy              equ     0x20    ;saved w register
s_copy          equ     0x21    ;saved STATUS register
byte            equ     0x22    ;Allgemeines Zählregister
hundred         equ     0x23  ;100mSec register
count           equ     0x24    ;ADW 20µsec Pausenregister
adres           equ     0x25    ;ADW Ergebnis
bit             equ     0x26    ;Zählregister für 8 ADW Durchläufe
highref         equ     0x27    ;12,5V Referenz
lowref          equ     0x28    ;12,0V Referenz
;*****************************************************************

;Ziele der Register operationen

w               equ     0                       ;W-Zielregister
f               equ     1                       ;f-zielregister
;*****************************************************************

;Macro- Definitionen

bank0               macro
                    bcf             STATUS, 5
                    bcf             STATUS, 6
                    endm

bank1               macro
```

```
                bsf     STATUS, 5
                endm
;*****************************************************************************
*

                org     0x00    ;Programmbeginn bei Adresse 00h
                goto    start
;*****************************************************************************

        org     0x04            ;Interrupt service Adresse

;Interrupt service vector

                bcf     INTCON, 7       ;Interrupt disabled
                movwf   w_copy          ;save w in 0x20
                movf    STATUS, 0       ;Verschiebe STATUS in w
                movwf   s_copy          ;save STATUS in 0x21

;Interrupt routine start

                decfsz  byte, 1         ;Ist delay "0"?
                goto    $+02            ;Nein
                bsf     hundred, 0      ;Ja

;Interrupt routine ende

intrent         bcf     INTCON, 2       ;TMRO interrupt flag (TMROIF) löschen

;restore context and return to the main programm

                movf    s_copy, 0       ;Hole STATUS zu rück in w
                movwf   STATUS          ;Daten zurück in STATUS
                movf    w_copy, 0       ;Akku daten sichern
                retfie                  ;Interrupt ende und GIE enabled.
;*****************************************************************************

;Subroutine initialisierung

init            movlw .230
                movwf   highref         ;(5/255)x230/0,36=12,5V
                movlw .220
                movwf   lowref          ;(5/255)x220/0,36=12V
                movlw 0x07
                movwf   CMCON           ;törn off comparators CM<2:0>=111
                bank1                   ;Gehe zu Bank1
                movlw B'11111001'
                movwf   TRISA           ;RA0 Eingang: RA<2:1> Ausgänge
                movlw B'11111110'
                movwf   TRISB           ;RB0 Ausgang
                bank0                   ;zurück zu Bank0
                clrf    PORTA
```

PIC-Batterie overload protection

```
;TMRO voreinstellen und starten
;TMRO overflow alle 0,976563 x 16 x 256 = 4 ms

                bcf     INTCON,     7       ;global interrupt disabled
                bank1                       ;gehe zu bank 1
                bcf     OPTION_REG, 5       ;intern. instruct. cycle clock
                bcf     OPTION_REG, 2
                bsf     OPTION_REG, 1
                bsf     OPTION_REG, 0       ;PS2,PS1,PS0="011"/prescaler 1:16
                bcf     OPTION_REG, 3       ;prescaler assigned to TMRO
                bank0                       ;zurück zu bank 0
                return


;Subroutine für 100msec delay

delay       clrf    hundred                 ;100mSec register löschen
                movlw .25
            movwf byte              ;Zählregister auf "25"
            clrf  TMR0
                btfss   hundred, 0      ;100msec vorbei?
                goto    $-01            ;Nein
                return


;Subroutine für 20µsec ADW-pause

adwpause            movlw .5
                    movwf count         ;Register count mit 5 laden
                    decfsz  count, 1    ;Ist count "0"?
                    goto    $-01        ;Nein: Wiederholen
                  return


;Subroutine AD - Wandlung

adw                 clrf    adres       ;Wandlungsergebnis register löschen
                    call    adwpause    ;20µs pause
                    movlw .8
                    movwf bit           ;Zaelregister bit mit 8 laden
                    bcf     STATUS, 0   ;Carry bit löschen
                    bcf    PORTA, 2     ;Chip select enable/Wandler gibt MSB aus
                    bsf    PORTA, 1     ;Clock "High"
                    btfsc   PORTA, 0    ;Ist RA0 auf 0?
                    goto    $+03        ;nein
                    bcf    adres, 7 ;Wenn "0" MSB in adres löschen
                    goto    $+02
                    bsf    adres, 7 ;MSB in adres setzen
                    rlf    adres, f  ;Inhalt adres nach links schieben
                    bcf    PORTA, 1     ;Clock "Low"
```

```
                decfsz  bit, f          ;Ist bit register auf 0?
                goto    $-09            ;nein
                rlf     adres, f  ;letztes Nachschieben
                bsf     PORTA, 2        ;Chip select disable
                return
;*****************************************************************************

;Hauptprogramm

start           call    init
                movlw   B'10100000'
                movwf   INTCON              ;GIE&TOIE enabled/TOIF cleared

loop1           bsf     PORTB, 0        ;Compressor "ON"
loop2           call    delay
                call    adw
                call    adw
                bsf     STATUS, 0       ;Flag C setzen
                movf    adres, 0        ;Batteriespannung in w
                subwf   lowref
                btfsc   STATUS, 0       ;Ist Ub > 12V?
                goto    $+02            ;Nein.
                goto    loop2           ;Ja
                bcf     PORTB, 0        ;Kompressor "OFF"
loop3           call    delay
                call    adw
                call    adw
                bsf     STATUS, 0       ;Flag C setzen
                movf    adres, 0        ;Batteriespannung in w
                subwf   highref, 0
                btfsc   STATUS, 0       ;Ist Ub > 12,5V?
                goto    loop3           ;Nein.
                goto    loop1           ;Ja.

        end
*****************************************************************************
```