

# PC táp átalakító C64-hez

Ez az áramkör egy **precíziós pontosságú inverter**, ami a **PC táp 12V feszültségét alakítja át 9V/5A váltófeszültséggé** amire szüksége van a C64-nek az 5V-on kívül a megfelelő működéshez. Ez azért jó, mert a PC táp sokkal jobban terhelhető mint a C64 eredeti tápja, így ha külső perifériákat használunk mint az IDE64, REU, egyéb bővítő, garantáltan bírni fogja az eszközök energiaigényét. Ezen kívül mehet még róla a floppy drive, és a merevlemezek, háttértárak amik még a géphez vannak kapcsolva, vagyis 1 tápról mehet minden, nem kell 4-5 tápegységgel vesződni mindenhez.

Működése egyszerű, egy darlington tranzisztor hiddal állítja elő a váltófeszültséget polaritásváltással. A hidat egy mikrokontroller vezérli, aminek a szoftvere biztosítja a nagy pontosságú vezérlést, és a frekvenciaváltást. A firmware esetleges cseréjéhez van rajta ICSP programozó csatlakozó is. Az áramkör 50 és 60Hz között átkapcsolható egy jumper/kapcsoló segítségével, így biztosítható hogy mind PAL, mind NTSC gépekkel is jól működjön. Az üzemmódot a visszajelző LED is szemlélteti: ha lassan villog akkor 50Hz, ha gyorsan villog 60Hz-es módban van. Átkapcsolni üzem közben is bármikor lehet. Pontossága nagyon jó, pontosan megy róla a belső óra is. Az áramköri lap (*kis kép a sarokban*) felépítése univerzális, ugyanis be lehet építeni a táp belsejébe is a rögzítő furatok segítségével. Ha nem akarjuk a tápot megbontani, akkor a panelt rá lehet tűzni egy szabvány Molex tápcsatlakozóra, a kimenetet egy 4 erű kábellel kivezelve egy DIN8-as csatlakozó véggel a C64-re csatlakoztatható. A komplett készre szerelt PC táp (képen) semmilyen szerelést nem igényel, az áramkör be van építve (hűtéssel), és rendelkezik a C64 és a floppy DIN dugóival.

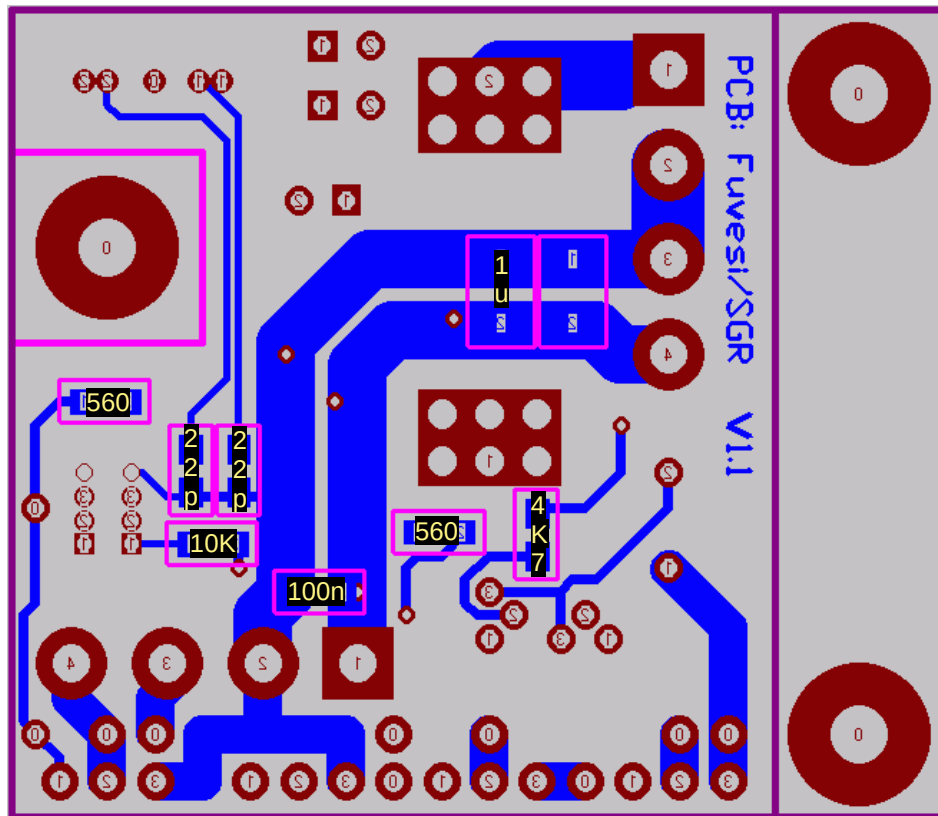
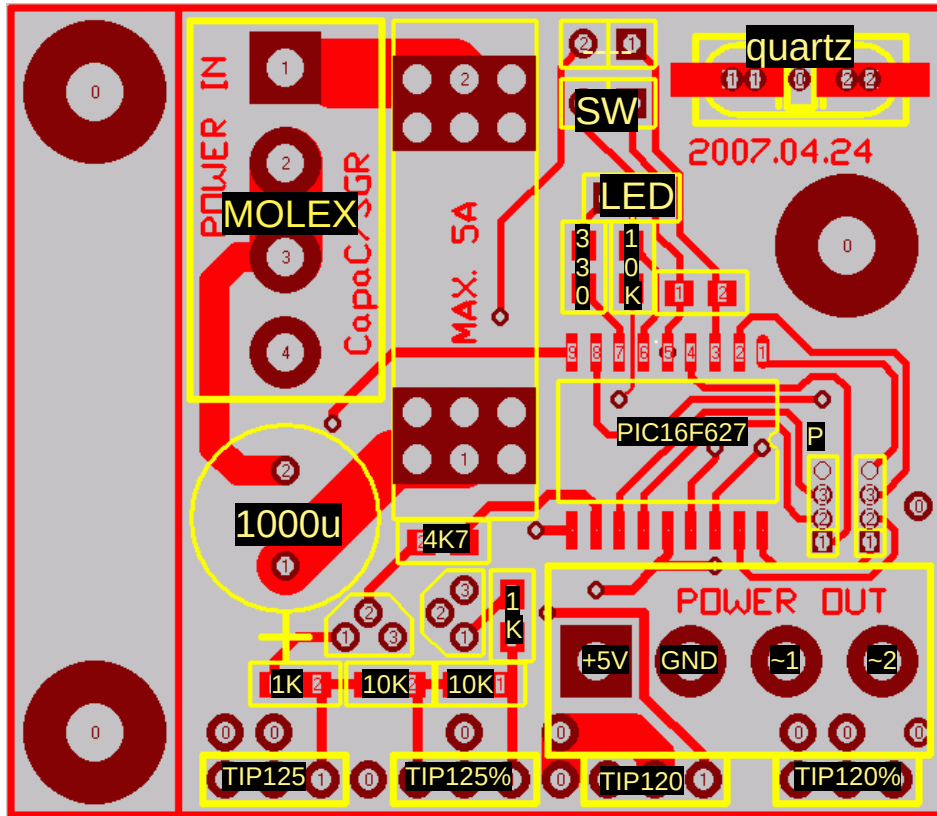


Az áramkör maximum 5A áramerősséggel terhelhető. Egy normál C64 0.5A terhelést jelent, magnóval 1A. Mérések szerint 1,5-2A terhelésig nem kell semmilyen hűtés az áramkörre, nem melegszik túl. Azaz 1 géphez nem kell hűtés, ha több gépet üzemeltetünk egyszerre róla akkor pedig kell. Az IDE64, és REU bővítők nem ezt a tápot hanem az 5V-ot használják, tehát ez nem terheli. Nagyobb terhelésnél **szigetelve** hűtőbordára kell szerelni, vagy úgy beépíteni a PC tápba, hogy **szigetelve** a kasztnihoz csavarozzuk a tranzisztorokat csillámmal és szigetelőgyűrűvel! Ne érintsük össze a hűtőzászlókat, mert ez zárlatot okoz! Ettől nem megy tönkre, de biztosítékot kell cserélni.

## Figyelem!

Mivel az előállított váltakozófeszültség **nem földfüggetlen** a +5V tápfeszültségtől ezért a C64-be épített gyári bekapcsolót szigorúan csak a BE állapotban szabad hagyni! Ezzel a gépet kikapcsolni nem szabad, mert a gép belső tápjának kialakítása olyan lehet hogy ettől tönkremehet! (eddig próbák során ilyen **nem** fordult elő) Csak a PC tápot szabad ki/be kapcsolni ha erről üzemel a gép! Az esetleges nem rendeltetésszerű használatból eredő károkért felelősséget nem vállalok!

A tápegység életveszélyes feszültségről üzemel (230V), ezért a tápot szétszedni, tárgyakat beledugni nem szabad! **Kizárólag védőföldeléssel ellátott konnektorral** használható!





aaa

```
*****
;
;
; Fajlnev: - *
; Datum: 2007.05.14 *
; Fajl verzio: 0.03 *
;
;
; Keszitette: Meszaros Arpad *
;
;
; Ceg: - *
;
;
;
;
; Szukseges fajok: p16f627.inc *
; Xtal: 3.2768MHz *
;
;
;
;
; Megjegyzes: *
; 9V inverter vezerloprogram *
; statusz LED es F0 kimenet 50/60Hz-en uzemel, freki atkapcsolas *
; hidvezerles mukodik, 16f628-ra forditva! *
; -MPLAB IDE alatt forditva, programozva ICD *
; -WatchdogTimer bekapcsolva, lefagy $\diamond$ sv $\diamond$ delem, t $\diamond$ pv $\diamond$ delem BOD *
;
*****

LIST P=16F628 ; processzor tipus megadasa
IFDEF __16F877
#include <P16F877.INC> ; deffinicios fajl hozzafuzes
;
; __CONFIG _CP_OFF & _DEBUG_OFF & _WRT_ENABLE_ON & _CPD_OFF &
_LVP_OFF & _BODEN_OFF & _PWRTE_ON & _WDT_OFF & _HS_OSC
;
; __IDLOCS 0x0000
ENDIF
IFDEF __16F628
#include <P16F628.INC> ; deffinicios fajl hozzafuzes
;
; __CONFIG _CP_OFF & _DATA_CP_OFF & _WDT_ON & _BODEN_ON & _PWRTE_ON
& _LVP_OFF & _MCLRE_ON & _HS_OSC
;
; __IDLOCS 0x0003 ; ID verzio
ENDIF

;***** KONSTANSOK DEFFINIALLASA *****
;
;#DEFINE DEVEL ; probapanelon kell!
#DEFINE RAMBASE 0x20 ; RAM kezdocime minden bankban
#DEFINE RAMBASE_COMM 0x70 ; Kozos RAM kezdocime
#DEFINE STAT_LED PORTB,1 ; st $\diamond$ tusz LED
#DEFINE F0_OUT PORTA,0 ; freki kimenet 50/60Hz
#DEFINE BRIDGE_1P PORTB,5 ; Inverter hid vezerles
#DEFINE BRIDGE_1N PORTB,4
#DEFINE BRIDGE_2P PORTB,3
#DEFINE BRIDGE_2N PORTB,2
#DEFINE FREQ_CH PORTB,0 ; GND - 60Hz kapcsolo

;***** VALTOZOK DEFFINIALLASA *****
w_temp EQU RAMBASE_COMM+0x0 ; Megszakitas W mentes
status_temp EQU RAMBASE_COMM+0x1 ; Megszakitas STATUS mentes
timer_status EQU RAMBASE+0x0 ; idozites jelzobitjei
clk50 EQU 0 ; bitflag - 50/sec utem beallitva
delay01 EQU RAMBASE+0x1 ; altalanos szamlalo

;***** FOPROGRAM KEZDETE *****
;
IFDEF DEVEL
ORG 0x003 ; Probapanel program kezdocim
```

```

        goto    MAIN_PROGRAM      ; Program kezdetere ugrik
ELSE
    ORG        0x000              ; processzor reset vektor
    goto    MAIN_PROGRAM      ; Program kezdetere ugrik
ENDIF
    ORG        0x004              ; megszakitas vektor helye
    goto    INT_SERVICE        ; megszakitas kiszolgalas

;***** FELHASZNALOI PROGRAM KEZDETE *****
MAIN_PROGRAM    clrwdt            ; 18ms-omnkent nullazni kell!
INIT_PIC        call    bank0
                IFDEF __16F877    ; init - portok beallitasa
                    clrf    PORTA    ; PORTA
                    clrf    PORTB
                    clrf    PORTE
                    CLRF    PORTD
                    clrf    PORTC
                    call    bank1
                    movlw   0x06      ; PORTA-PORTE digitalis
                    movwf   ADCON1
                    movlw   B'11111111'
                    movwf   TRISA
                    movlw   B'11111111'    ; PORTB
                    movwf   TRISB
                    movlw   B'11000000'    ; PORTC
                    movwf   TRISC
                    movlw   B'11111111'    ; PORTD
                    movwf   TRISD
                    movlw   B'11111111'    ; PORTE
                    movwf   TRISE
                ENDIF
                IFDEF __16F628
                    clrf    PORTA    ; PORTA
                    clrf    PORTB
                    movlw   0x07      ; PORTA digitalis
                    movwf   CMCON
                    call    bank1
                    movlw   B'11111110'
                    movwf   TRISA
                    movlw   B'11000001'    ; PORTB
                    movwf   TRISB
                ENDIF
                bcf    OPTION_REG,PS0    ; periferiak beallitasa
                bcf    OPTION_REG,PS1
                bsf    OPTION_REG,PS2    ; 0-2 eloosztó értékek 100 - :32
                bcf    OPTION_REG,PSA    ; Eloosztó Timer0-hoz
                bsf    OPTION_REG,T0SE    ; Lefuto elre számol
                bcf    OPTION_REG,T0CS    ; Belso utemado
                bcf    OPTION_REG,INTEDG ; RB0/INT lefuto elre
                bsf    OPTION_REG,NOT_RBPU ; RBPU ki
                call    bank0
                bsf    INTCON,T0IE    ; Megszakitasok bekapcsolasa
                bsf    INTCON,GIE    ; INIT vege!

                clrf    timer_status    ; init - változők beallitasa
                movlw   D'50'
                movwf   delay01

START_LOOP    clrwdt            ; 18ms-omnkent nullazni kell!
                btfss   timer_status,clk50 ; utem volt? Foprogramba a nem
idokritikus dolgot
                goto    START_LOOP    ; ha nem, vissza az elejere
                bcf    timer_status,clk50 ; utem volt, torles
                decfsz delay01,F
                goto    START_LOOP    ; ha a LED szamlalo nem erte el
vissza

```

```

        btfsc   FREQ_CH           ; ha Clear akkor 60Hz
        goto   LOAD_D50
        movlw  D'20'             ; gyors villogas
        goto   SET_DELAY
LOAD_D50  movlw  D'50'
SET_DELAY movwf  delay01         ; LED szamlalo beallitasa
        btfsc  STAT_LED         ; LED invertalo
        goto   CLEAR_LED
SET_LED   bsf    STAT_LED
        goto   END_LED
CLEAR_LED bcf    STAT_LED
END_LED   nop

        goto   START_LOOP       ; programmag vege, ujrakezdes

;***** MEGSZAKITAS KISZOLGALAS *****
INT_SERVICE movwf  w_temp       ; W regiszter mentese
        movf   STATUS,W
        movwf  status_temp     ; STATUS regiszter mentese
        ;felhasznaoi megszakitas kiszolgalas
        btfss  INTCON,T0IF     ; timer0 megszakitas volt?
        goto   TIMER0_END      ; ha nem, akkor kovi megszakitorutin
        btfsc  FREQ_CH         ; ha Clear akkor 60Hz beallit!
        goto   SET_STATUS      ; 50Hz-en nem kell beallitani semmit!
        movlw  D'42'           ; 60Hz elotoltes a szamlaloba!
        movwf  TMR0
SET_STATUS bsf    timer_status,clk50 ;statusz beallitasa
        btfsc  F0_OUT          ; F0 invertalo az INT rutinba mert
fontos az idozites!
        goto   CLEAR_F0
SET_F0     nop                 ; 2 ciklus kesleltetes 50% kitoltesi
tenyezoert

        nop
        bsf   F0_OUT
        bcf   BRIDGE_1N       ; hid alapallapot 1p-2n be
        bcf   BRIDGE_2P
        bsf   BRIDGE_1P
        bsf   BRIDGE_2N
        goto  END_F0
CLEAR_F0   bcf   F0_OUT
        bcf   BRIDGE_1P       ; hid invertalt 1n-2p be
        bcf   BRIDGE_2N
        bsf   BRIDGE_1N
        bsf   BRIDGE_2P
END_F0     nop
TIMER0_END bcf   INTCON,T0IF
        ;felhasznaoi megszakitas kiszolgalas vege, visszateres
        movf  status_temp,W
        movwf STATUS         ; STATUS visszatele
        swapf w_temp,F
        swapf w_temp,W       ; W regiszter helyreallitasa
        retfie               ; visszateres a megszakitasbol

;***** SZUBRUTINOK KEZDETE *****
bank0      bcf   STATUS,RP0    ;RAM bankvalto rutinok
        bcf   STATUS,RP1
        return

bank1      bsf   STATUS,RP0
        bcf   STATUS,RP1
        return

bank2      bcf   STATUS,RP0
        bsf   STATUS,RP1
        return

```

bank3

```
bsf STATUS,RP0  
bsf STATUS,RP1  
return
```

END

; directive 'end of program'