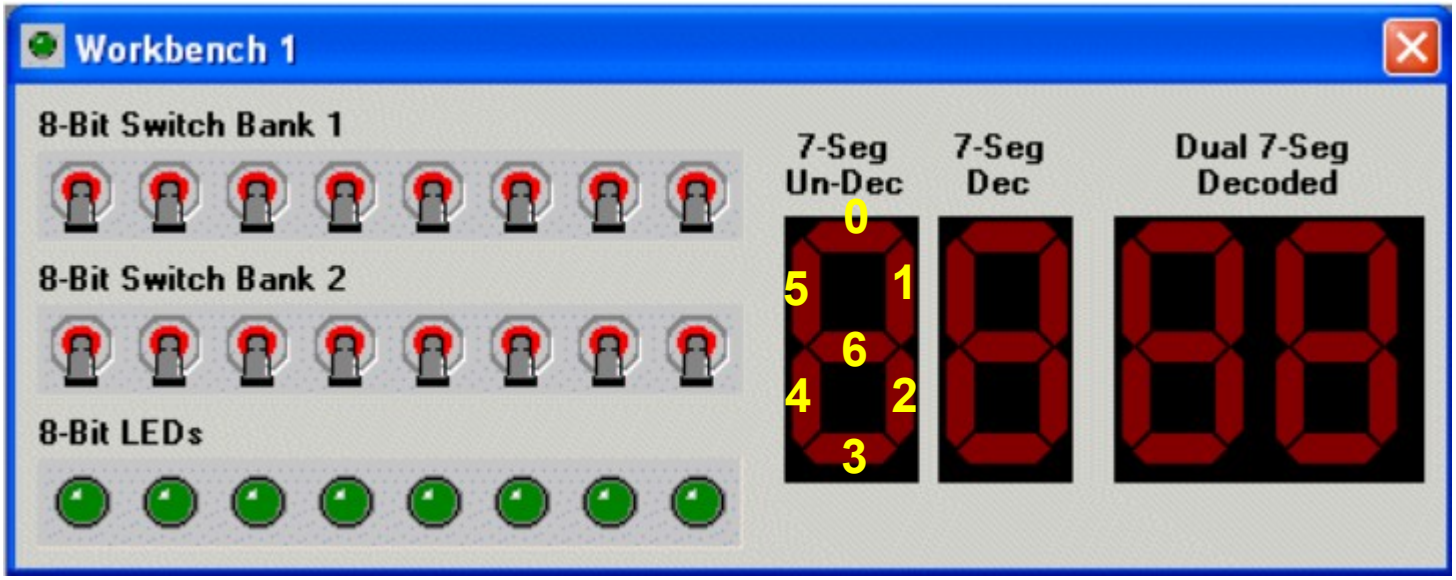
Hardverközeli programozás 1
7. gyakorlat

Kocsis Gergely

2019.04.01.

DIY Calculator Workbench

A Workbench további be és kimeneteket tartalmaz. Elérhető a tools menű Workbench #1 menüpontja alatt.



Port Típusa	**Port címe**	**I/O típusa**
Input	$F000	8-bites kapcsolósor #1
Input	$F001	8-bites kapcsolósor #2
Output	$F020	8-bites LED-sor
Output	$F021	Dekódolatlan 7szegmenses kijelző
Output	$F022	Dekódolt 7szegmenses kijelző
Output	$F023	Dupla dekódolt 7szegmenses kijelző

DIY Calculator Workbench

Írj olyan DIY Calculator assembly programot, ami a Workbench felső gombsoráról kapott nyolcbites értéket megjeleníti mind a dekódolt, mind a dekódolatlan 7szegmenses kijelzőn, mind a LEDsoron.

Módosítsd a programot úgy, hogy a felső gombsor olvasásával együtt az alsó gombsort is figyelje a program. Ezúttal az alsó gombsoron betáplált értéket mutassa a LEDsor.

Port Típusa	**Port címe**	**I/O típusa**
Input	$F000	8-bites kapcsolósor #1
Input	$F001	8-bites kapcsolósor #2
Output	$F020	8-bites LED-sor
Output	$F021	Dekódolatlan 7szegmenses kijelző
Output	$F022	Dekódolt 7szegmenses kijelző
Output	$F023	Dupla dekódolt 7szegmenses kijelző

DIY Calculator QWERTY

A kalulátor tartalmaz egy QWERTY billentyűzetet is. Ez a tools menű Terminal menüpontja alatt érhető el.

Port Típusa	**Port címe**	**I/O típusa**
Input	$F008	QWERTY billentyűzet

DIY Calculator QWERTY

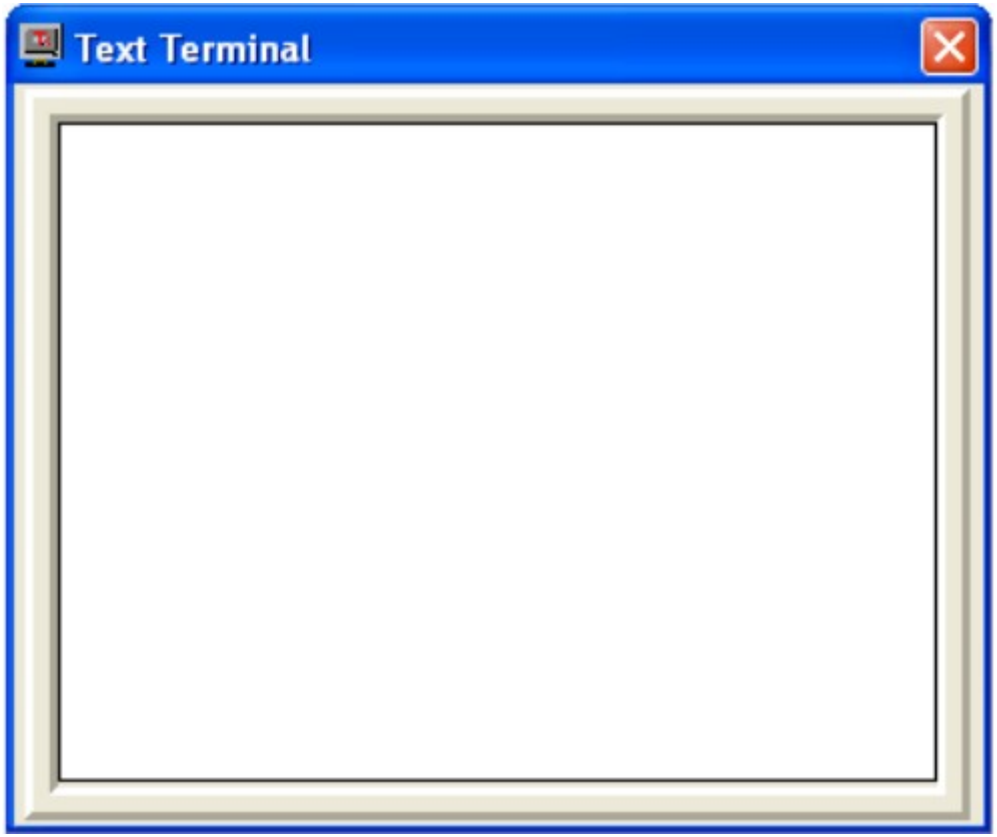
A billentyűzethez tartozó input porton mindaddig $00 jelenik meg, amíg gombnyomás nem történik.

Írj olyan DIY Calculator assembly programot, ami a bilentyűzetről karaktereket olvas, majd megjeleníti azokat a kijelzőn.
A program elején töröld a képernyőt.

Port Típusa	**Port címe**	**I/O típusa**
Input	$F008	QWERTY billentyűzet

DIY Calculator konzol

A kalulátor tartalmaz egy konzolt is. Ez szintén a tools menű terminal menüpontja alatt érhető el.



Port Típusa	**Port címe**	**I/O típusa**
Output	$F028	Konzol

DIY Calculator konzol

A konzol speciális kódjai

Kód	Leírás
$00	Null kód, nincs jelentése
$01	Későbbi használatra foglalt (egyelőre null)
$02	Konzol törlése
$03	Kurzor (0,0) pozícióba állítása
$04	Backspace
$05	Új sor
$06	Csengő megszólaltatása (beep)
$07	Kurzor Fel
$08	Kurzor Le
$09	Kurzor Jobbra
$0A	Kurzor Balra

Módosítsd az előző programot úgy, hogy a billentyűzetről bekért kódokat ne a kijelzőn, hanem a konzolon jelenítse meg.

DIY Calculator Megszakítások

IRQ: Interrupt ReQuest
IACK: Interrupt ACKnowledgement

IRQ mask: Engedélyezi, vagy tiltja a megszakítások figyelését
Beállítása: `SETIM`
Kikapcsolása: `CLRIM`

A maszk beállításakor az IRQ vektor törlődik. Onnantól kezdve, figyeljük a megszakításokat. Ha érkezik egy, akkor az addig tárolódik, amíg az aktuálisan futó utasítás be nem fejeződik. Utána ugrik egy speciális megszakításkezelő szubrutinra.

IV: Interrupt vektor: A megszakításkezető szubrutin címét tárolja.
Beállítása: BLDIV

DIY Calculator Megszakítások

```
## Konstansok beállítása
MAINDISP:       .EQU    $F031   # Kijelző
SIXLEDS:        .EQU    $F032   # LEDsor

## Program kezdete
                .ORG    $4000

## Az átmeneti tároló inicializálása
INITTEMP:       LDA     %01010101       # Érték az ACC-be
                STA     [TEMP]          # Érték az ACC-ből a TEMP-be

## További inicializálás
GETREADY:       LDA     $09             # A '9' karakterkódja az ACC-be
                BLDSP   $4FFF           # Veremmutató az utolsó címre
                BLDIV   SERVICE         # A megszakításkezelő a SERVICE
                                        # cimkén kezdődik
                SETIM                   # Figyeljük a megszakításokat

# 9876543210 megjelenítése újra és újra
MAINLOOP:       STA     [MAINDISP]      # Az ACC kiírása a képernyőre
                DECA                    # ACC csökkentése
                JNN     [MAINLOOP]      # Ugrás, ha ACC nem lett negatív
                LDA     $09             # Ha igen, akkor ACC legyen $09
                JMP     [MAINLOOP]      # ...és ugrás az elejére
```

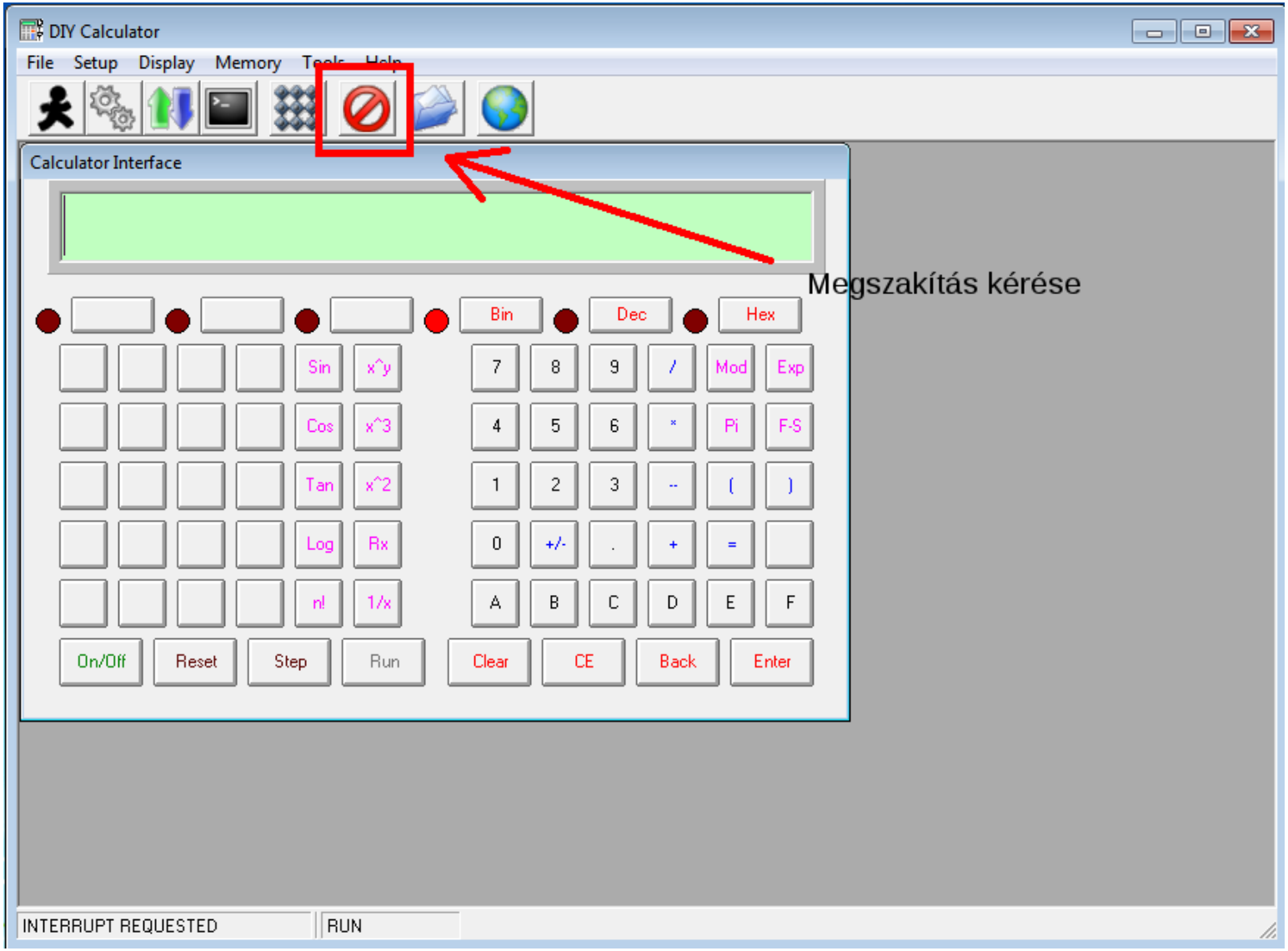
DIY Calculator Megszakítások

```
                . . .

## A megszakításkezelő szubrutin
SERVICE:        PUSHA                   # ACC elmentése a verembe
                LDA     [TEMP]          # TEMP beolvasása ACC-be
                STA     [SIXLEDS]       # ACC kiírása a LEDekre
                XOR     %11111111       # ACC invertálása
                STA     [TEMP]          # A módosított érték mentése TEMPbe
                POPA                    # A mentett tartalom visszaolvasása
                RTI                     # Visszatérés a megszakításból

## Helyfoglalás egybájtos tárolónak
TEMP:           .BYTE

## Program vége
                .END
```

XOR: logikai “kizáró vagy”. Azonos bitek esetén 0, különbözők esetén 1.

CLRIM A megszakításkezelő elején: Ne lehessen megszakításkezelés közben megszakítást kezdeményezni. A végére kell egy SETIM is.

DIY Calculator Megszakítások



DIY Calculator Megszakítások

Írj olyan, DIYC programot, mely beolvas a billentyűről egy pontosan két karakteres hexadecimális számot egy szubrutin segítségével.
A számot egyetlen bájton add vissza a veremben.
A szubrutin nem módosíthatja sem az ACC, sem a státuszregiszterek tartalmát.
Jelenítsd meg a beolvasott számot a kétkarakteres dekódolt kijelzőn ($F023).

PUSHSR: A státuszregiszter tartalmát a verem tetejére írja
POPSR: A státuszregiszterbe írja a verem tetején lévő bájtot.