;g0-ra 0-9ig növeli a számot,beírja egy saját regiszterbe.G1-re kijelzőt vált és ismétli

;a beírást egy saját regiszterbe.Ha végzett kijelzi az értékeket,majd ha ekkor g0-t

;nyomunk előlőről kezdi a folyamatot

.include "m128def.inc"

.CSEG

.def a=r16

.def seg=r17

.def b=r18

.def c=r19

.def bit=r20

.def d=r21

.def e=r22

.def f=r23

.def g=r24

.def del1=r25

.macro stack\_init

ldi a, HIGH(RAMEND)

out sph, a

ldi a, LOW(RAMEND)

out spl, a

.endmacro

.macro port\_init

ldi a, 0

sts DDRG, a

ldi a, 0xFF

out DDRA, a

ldi a, 0b00001100

out DDRE, a

ldi a, 0x80

out DDRC, a

.endmacro

.org 0x00

rjmp init

.org 0x100

init:

port\_init

stack\_init

ldi seg, 255

ldi bit, 0b10110000

ldi f, 255

rjmp start

start:

ldi a,0

lds a, PING

cpi a, 1

breq g0

cpi a, 2

breq g1

rjmp start

g0:

cpi f, 255

breq f0

cpi f, 0

breq f1

cpi f, 1

breq f2

cpi f, 2

breq f3

g00:

rjmp start

g1:

call delay

inc f

eor e,e

eor seg,seg

cpi f,3

breq nullazf

rjmp g0

f0:

cpi seg, 10

breq nullaz

mov e, seg

ori e, 0b10110000

mov r10,e

out PORTA, r10

call delay

inc seg

rjmp g00

f1:

cpi seg, 10

breq nullaz

mov e, seg

ori e, 0b10100000

mov r11,e

out PORTA, r11

call delay

inc seg

rjmp g00

f2:

cpi seg, 10

breq nullaz

mov e, seg

ori e, 0b10010000

mov r12,e

out PORTA, r12

call delay

inc seg

rjmp g00

f3:

cpi seg, 10

breq nullaz

mov e, seg

ori e, 0b10000000

mov r13,e

out PORTA, r13

call delay

inc seg

rjmp g00

delay:///Késleltetés beállítása

ldi d,10

d1:

ldi e,255

d2:

ldi b,255

d3:

dec b

brne d3

dec e

brne d2

dec d

brne d1

ret

nullaz:

ldi seg,0

rjmp g0

nullazf:

ldi f, 255

rjmp kijelez

rjmp g0

kijelez:

lds a,ping

cpi a,1

breq kilep

out porta,r13

call delay2

out porta,r12

call delay2

out porta,r11

call delay2

out porta,r10

call delay2

rjmp kijelez

delay2:

ldi del1,255

d31:

dec del1

brne d31

ret

kilep:

rjmp start