

Συμπληρώνουμε το τμήμα δηλώσεων ως εξής:

```
1  .dseg
2  .ORG 0x0100
3      arrayA: .byte 10
4      arrayB: .byte 10
5      arrayC: .byte 10
6
7  .cseg
8      .ORG 0x0000
9
10 start:
11      ldi r16, 0b11101010 ; set PCn for STB and acknowledge
12      out DDRC, r16
13
14      ldi r16, 0b00000000 ; clear portC
15      out PORTC, r16
16
17      out DDRB, r16 ; PBn will be used for input
18      out DDRD, r16 ; PDn will be used for input
19
20      ldi r18, 30 ; initialize r18 for delay
21
22      ldi r27, high(arrayA) ; initialize X
23      ldi r26, low(arrayA)
24
25      ldi r29, high(arrayB) ; initialize Y
26      ldi r28, low(arrayB)
27
28      ldi r31, high(arrayC) ; initialize Z
29      ldi r30, low(arrayC)
```

Όπου τα array A,B,C χρησιμοποιούνται για να αποθηκεύσουν τα δεδομένα από τις συσκευές Σ1, Σ2, Σ3 αντίστοιχα και οι καταχωρητές X,Y,Z (οι οποίοι αποτελούνται από ζευγάρια καταχωρητών) χρησιμοποιούνται για τον προσδιορισμό της θέσης των πινάκων. Το PCn θα χρησιμοποιηθεί για το STB και το acknowledge. Για αυτό θέτουμε στις θέσεις PC0, PC2, PC4 (με την χρήση του καταχωρητή DDRC) την τιμή 0 για το STB και στις θέσεις PC1, PC2, PC3 την τιμή 1 για το acknowledge. Στη συνέχεια μηδενίζουμε το portC και θέτουμε τα PCn και PDn την τιμή 0 καθώς θα χρησιμοποιηθούν για είσοδο δεδομένων.

Στη συνέχεια το πρόγραμμα προχωράει στον βρόγχο polling.

```
31 polling:
32     sbic PORTC, 0 ; if portC[0] = 1
33     rcall devA    ; read input from devA
34
35     sbic PORTC, 2 ; if portC[2] = 1
36     rcall devB    ; read input from devB
37
38     sbic PORTC, 4 ; if portC[4] = 1
39     rcall devC    ; read input from devC
40
41     rjmp polling ; jmp to polling and check the devices again
```

Ο βρόγχος ελέγχει αν τα bit στις θέσεις PortC[0], PortC[2], PortC[4] έχουν πάρει την τιμή 1 και αν ναι καλεί την αντίστοιχη ρουτίνα. Στο τέλος δεν τερματίζει απλά καλεί ξανά τον εαυτό του.

Η ρουτίνα για την 1<sup>η</sup> συσκευή έχει ως εξής:

```
devA:
    in r17, PORTB    ; r17 = portB
    st X+, r17       ; arrayA[i] = r17
    ldi r16, 0b00000010 ; send acknowledge to portC[1]
    out PORTC, r16
    rcall delay      ; delay for about 10μsec
    ret              ; return from routine
```

- Αρχικά διαβάζει τα bit από την συσκευή.
- Τα αποθηκεύει στον πίνακα arrayA και αυξάνει τον μετρητή X κατά 1.
- Στέλνει το acknowledge στην συσκευή.
- Γίνεται ένα delay για περίπου 10 μsec.
- Γίνεται έξοδος από την ρουτίνα.

Name	Address	Value	Bits
I/O PINC	0x26	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRC	0x27	0x6A	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
I/O PORTC	0x28	0x01	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/>

Βάζουμε στο portB μερικά δεδομένα π.χ την τιμή ff στο δυαδικό

Βλέπουμε ότι στην 1<sup>η</sup> θέση του πίνακα arrayA μπήκε η τιμή ff,

και ότι στο portC[1] μπήκε η τιμή 1.

Βάζουμε αυτή τη φορά στο portB την τιμή f0.

Βλέπουμε πως μπήκε στην 2<sup>η</sup> θέση του πίνακα δίπλα στο 1<sup>ο</sup> στοιχείο.

Αντιστοίχως η ρουτίνα για την 2<sup>η</sup> συσκευή:

Διαβάζονται τα δεδομένα από τον καταχωρητή PortD στον καταχωρητή r17, και γίνεται and στον καταχωρητή r17 έτσι ώστε να κοπούν τα bit τα οποία δεν θέλουμε. Ακολουθούμε την ίδια διαδικασία όπως στην 1<sup>η</sup> συσκευή:

Θέτουμε στο portC[3] την τιμή 1:

Δίνουμε στο PortD την τιμή FF:

Βλέπουμε πως στον arrayB μπήκε η τιμή 0f καθώς τα πρώτα bit έχουν κοπεί

Και το acknowledge μπήκε στην θέση 4

Name	Address	Value	Bits
I/O PINC	0x26	0x00	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRC	0x27	0x6A	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
I/O PORTC	0x28	0x08	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Η ρουτίνα για την 3<sup>η</sup> συσκευή:

```
devC:
    in r17, PORTD
    andi r17, 0b11110000 ; r17 = portD[4-7]
    st Z+, r17           ; arrayC[i] = r17
    ldi r16, 0b00100000 ; send acknowledge to portC[5]
    out PORTC, r16
    rcall delay          ; delay for about 10µsec
    ret                  ; return from routine
```

Δίνουμε στο portC[4] την τιμή 1:

Name	Address	Value	Bits
I/O PINC	0x26	0x00	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O DDRC	0x27	0x6A	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O PORTC	0x28	0x10	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>

Και στο portD την τιμή FF:

Name	Address	Value	Bits
I/O PIND	0x29	0x00	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O DDRD	0x2A	0x00	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O PORTD	0x2B	0xFF	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>

Καθώς κόβονται τα τελευταία 4 bit στο arrayC γίνεται εισαγωγή της τιμής f0 στο arrayC:

Memory 4

Memory: data IRAM

data 0x0100

00 00 00 00 00 00 00 00 00 00

data 0x010A

00 00 00 00 00 00 00 00 00 00

data 0x0114

f0 00 00 00 00 00 00 00 00 00

Και το acknowledge μπαίνει στην 5<sup>η</sup> θέση του portC

Name	Address	Value	Bits
I/O PINC	0x26	0x00	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O DDRC	0x27	0x6A	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>
I/O PORTC	0x28	0x20	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>

Τέλος στην ρουτίνα delay γίνεται ένα nop και μειώνεται η τιμή r18 μέχρι να μηδενιστεί. Πριν την επιστροφή από την ρουτίνα δίνουμε στον καταχωρητή r18 την τιμή r18 έτσι ώστε να είναι έτοιμος για την επόμενη κλήση της ρουτίνας.

```
delay:
    nop
    dec r18      ; r18 = r18 - 1
    breq return ; if r18 == 0 return
    rjmp delay  ; else goto delay

return:
    ldi r18, 30 ; before return set r18 = 30
    ret
```