

Ενσωματωμένα Συστήματα Μικροεπεξεργαστών – ΗΡΥ411

Αναφορά Εργαστηρίου 2 – Εξοικείωση με την Οικογένεια Μικροελεγκτών Atmel AVR – Μια απλή οθόνη 7-segment LED

Ομάδα: LAB41145877

Κιούλος Ευάγγελος 2016030056

Εισαγωγή:

Σκοπός της δεύτερης εργαστηριακής άσκησης ήταν η περαιτέρω εξοικείωση με την οικογένεια μικροελεγκτών Atmel AVR. Κληθήκαμε να υλοποιήσουμε μια απλή οθόνη 7-segment LED σε Assembly AVR, χρησιμοποιώντας τον χρονιστή του πρώτου εργαστηρίου. Με την ενσωμάτωση του χρονιστή από το πρώτο εργαστήριο, δημιουργούμε την οθόνη με πολυπλεξία στο χρόνο.

Υλοποίηση Κώδικα:

Αρχικά, πρέπει να ορίσουμε το data segment του προγράμματος μας χρησιμοποιώντας την εντολή “.dseg” όπως φαίνεται στο παρακάτω απόσπασμα κώδικα.

```
.dseg                ; We specify that the data segment of our program
.org SRAM_START      ; begins in the start of sram (address 0x0060).
; We allocate 8 bytes of sram for storing the bcd digits we want to display.
; The label marks the address where data memory is allocated
bcd_num:
.byte 8              ; allocate 8 bytes of sram
```

Όπως φαίνεται παραπάνω, με την εντολή “.org SRAM_START” ορίζουμε ότι το data segment θα ξεκινά από την αρχή της μνήμης SRAM (στην συγκεκριμένη περίπτωση αυτή είναι η διεύθυνση 0x0060). Με το “.byte 8” ορίζουμε ότι θέλουμε να κάνουμε allocate 8 bytes μνήμης SRAM και με το label “bcd_num” ορίζουμε την διεύθυνση αυτής της allocated μνήμης. Σε αυτή την περίπτωση το label “bcd_num” δείχνει στην διεύθυνση 0x0060. Αυτός ο χώρος, γίνεται allocated για να αποθηκεύουμε τα ψηφία που θέλουμε να δείχνουμε στην οθόνη.

Στην συνέχεια, στο code segment του προγράμματος μας (το οποίο ορίζεται από την εντολή .cseg) αρχικοποιούμε τον καταχωρητή X ώστε να δείχνει στην πρώτη θέση της δεσμευμένης μνήμης και αρχικοποιούμε όλες τις δεσμευμένες θέσεις μνήμης με την τιμή 0xFF, έτσι ώστε να μην ανάβει κάποιο από τα segments της οθόνης στην αρχή του προγράμματος, όπως φαίνεται και παρακάτω. Παρατηρούμε ότι οι διευθύνσεις στις οποίες έχουμε αρχικοποιήσει τις τιμές της δεσμευμένης μνήμης είναι οι “0x0061” έως και “0x0068”.

```
; Initialize X register to point at the
; start of data we want to store in the
; allocated memory
ldi XL, low(bcd_num)
ldi XH, high(bcd_num)

; initialize memory for bcd numbers with value 0xff,
; so that every segment is off in the beginning.
ldi r18, 8
; This loops 8 times because we have 8 allocated bytes.
init:
ldi r16, 0xFF
st X+, r16      ; After each loop pointer X increments to the next address
dec r18         ; until the loop is over.
brne init
```

Με την ένδειξη “X+” στην εντολή store, ορίζουμε ότι η τιμή που θέλουμε να αποθηκεύσουμε θα αποθηκευτεί στην ζητούμενη διεύθυνση και έπειτα ο δείκτης X θα δείχνει στην επόμενη διεύθυνση. Επομένως, μετά την αρχικοποίηση των δεσμευμένων θέσεων, ο δείκτης X έγινε reset στην πρώτη θέση της δεσμευμένης μνήμης και εισάχθηκαν τα ψηφία που θέλουμε να προβάλλουμε στην οθόνη.

```
; Reset X register to point at the start of allocated space
ldi XL, low(bcd_num)
ldi XH, high(bcd_num)

;insert bcd digits we want to display
; "+" after register X increments pointer X after storing value
ldi r16, 0x05
st X+, r16                ; Store value 0x05 at address 0x0061
ldi r16, 0x03
st X+, r16                ; Store value 0x03 at address 0x0062
ldi r16, 0x07
st X+, r16                ; Store value 0x07 at address 0x0063
```

Στην συγκεκριμένη περίπτωση, θέλουμε να προβάλλουμε τον αριθμό “735”, επομένως εισάγουμε στην μνήμη τα BCD ψηφία με την σειρά που θέλουμε να εμφανιστούν στα anodes από τα αριστερά προς τα δεξιά(AN0 – AN7). Επομένως, εισάγουμε στην πρώτη θέση της μνήμης το ψηφίο “5”, στην δεύτερη θέση το ψηφίο “3” και στην τρίτη θέση το ψηφίο “7”. Στην συνέχεια, ορίζουμε ότι τα ports A και C θα είναι έξοδοι, αρχικοποιούμε τους καταχωρητές που χρειαζόμαστε για την χρήση του TIMER/COUNTER0 και του timer0 overflow interrupt, αρχικοποιούμε τον καταχωρητή r17, ο οποίος έχει την τιμή του ring counter, με την τιμή 0x00 και μεταβαίνουμε στο κύριο πρόγραμμα που αποτελείται από έναν ατέρμονο βρόγχο.

ΣΗΜΕΙΩΣΗ: Παρατηρήθηκε ότι η καθυστέρηση του 1msec από το προηγούμενο εργαστήριο ήταν επαρκή για την δημιουργία της οθόνης 7-segment με πολυπλεξία στο χρόνο και χρησιμοποιήθηκε αυτούσια.

Στην ρουτίνα εξυπηρέτησης του interrupt, υλοποιούμε τον ring counter, χρησιμοποιώντας την εντολή “rol r17” που ολισθαίνει ένα “1” στα bits του καταχωρητή r17 ενώ τα υπόλοιπα bits του είναι “0”. Μετά από κάθε ολίσθηση, εμφανίζουμε το περιεχόμενο του r17 στο port C. Όταν ο καταχωρητής r17 πάρει την τιμή “0x00” τότε φορτώνουμε στον καταχωρητή την τιμή “0x01” (0b00000001) και εμφανίζουμε τα περιεχόμενα του στο port C. Αυτό έχει ως αποτέλεσμα ότι, όταν η τιμή του r17 είναι “0b10000000”, η επόμενη τιμή του θα είναι το “0b00000001”, χωρίς να εμφανίζεται ποτέ η τιμή 0x00. Παρακάτω φαίνεται το απόσπασμα από την ρουτίνα εξυπηρέτησης του interrupt που υλοποιεί τον ring counter.

```
; shift an '1' through zeros
rol r17
; check if value is 0x00
cpi r17, 0
brne cont
; Reset ring counter to "00000001" if r17 is "00000000"
ldi r17, 0b00000001
; output the value of r17 to port C
cont:
out PORTC, r17
```

```
ld r16, X+           ; Load BCD value from the address that X points to
cpi XL, low(bcd_num+9) ; check if address is out of bounds
brne cont2
ldi XL, low(bcd_num)   ; reset pointer if address is out of bounds
ldi XH, high(bcd_num)
ld r16, X+             ; Load BCD value from the address that X points to

cont2:
rcall bcd_to_7_segment ; Decode BCD to 7-segment and display digit in port A
```

Αποτελέσματα προσομοίωσης:

Program Counter	0x00000019
Stack Pointer	0x045F
X Register	0x0060
Y Register	0x0000
Z Register	0x0000

[illegible]

Στην συνέχεια, αφού κάνουμε reset την τιμή του καταχωρητή X στην αρχική του θέση, εισάγουμε τα ψηφία που θέλουμε να εμφανίσουμε στην οθόνη:

Memory 4	
Memory: data IRAM	Address: 0x0060, data
data 0x0060	05 03 07 ff ff ff ff 00 00 00 00 00 00 00
data 0x0077	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Μετά την εισαγωγή των BCD ψηφίων στην μνήμη ο καταχωρητής X ξαναγίνεται reset έτσι ώστε να δείχνει στην διεύθυνση που υπάρχει το πρώτο ψηφίο που θέλουμε να εμφανίσουμε. Στην συνέχεια, τα ports A και C ορίζονται ως έξοδοι, αρχικοποιούνται οι καταχωρητές που χρειαζόμαστε για τον timer 0 και τον prescaler, όπως είδαμε και στο προηγούμενο εργαστήριο, η τιμή του r17, που αντιστοιχεί στον ring counter αρχικοποιείται στην τιμή “0x00” και το πρόγραμμα μεταβαίνει σε έναν ατέρμονο βρόγχο. Όταν συμβεί το interrupt, μεταβαίνουμε στον interrupt handler όπου αρχικά, ο ring counter ξεκινά την μέτρηση και εμφανίζει την τιμή ‘1’ στο πρώτο pin του port C και επομένως ανάβει το πρώτο anode:

Name	Address	Value	Bits
I/O PINC	0x33	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"/>
I/O DDRC	0x34	0xFF	<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 checked="" type="checkbox"/> <input checked="" type="checkbox"/>
I/O PORTC	0x35	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"/>

Στην συνέχεια διαβάζουμε την τιμή που υπάρχει στην διεύθυνση που δείχνει ο καταχωρητής X, την εισάγουμε στον καταχωρητή r16 και εμφανίζουμε τα περιεχόμενα της στο αντίστοιχο segment:

	Name	Address	Value	Bits
X Register	0x0061	R15	0x00	I/O PINA 0x39 0x92 <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Y Register	0x0000	R16	0x05	I/O DDRA 0x3A 0xFF <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 checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Z Register	0x0000	R17	0x01	I/O PORTA 0x3B 0x92 <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Παρατηρούμε ότι ο καταχωρητής X δείχνει στην πρώτη δεσμευμένη διεύθυνση η οποία έχει την τιμή “5” σε BCD. Όπως βλέπουμε, η τιμή BCD “5” αποκωδικοποιείται σε “0b10010010” που αντιστοιχεί στο “5” σε 7-segment. Τέλος, παρατηρούμε ότι ο r16 έχει την τιμή “0x05” την ίδια στιγμή που ο r17 έχει την τιμή “0x01”. Οπότε συμπεραίνουμε ότι το ψηφίο “5” εμφανίζεται στο πρώτο anode όπως ήταν αναμενόμενο. Αντίστοιχα, όπως βλέπουμε και στις παρακάτω εικόνες, εμφανίζονται και τα υπόλοιπα ψηφία που θέλουμε να εμφανίσουμε στα σωστά segments.

Εμφάνιση του “3” στο 2ο segment:

Name	Address	Value	Bits
I/O PINC	0x33	0x02	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>
I/O DDRC	0x34	0xFF	<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 checked="" type="checkbox"/> <input checked="" type="checkbox"/>
I/O PORTC	0x35	0x02	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>

	Name	Address	Value	Bits
X Register	0x0062	R15	0x00	I/O PINA 0x39 0xB0 <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Y Register	0x0000	R16	0x03	I/O DDRA 0x3A 0xFF <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 checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Z Register	0x0000	R17	0x02	I/O PORTA 0x3B 0xB0 <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Εμφάνιση του “7” στο 3ο segment:

Name	Address	Value	Bits
I/O PINC	0x33	0x04	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
I/O DDRC	0x34	0xFF	<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 checked="" type="checkbox"/> <input checked="" type="checkbox"/>
I/O PORTC	0x35	0x04	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Name	Address	Value	Bits
X Register	0x0063 R15	0x00	0x00
Y Register	0x0000 R16	0x07	0x07
Z Register	0x0000 R17	0x04	0x04

Οι υπόλοιπες θέσεις μνήμης έχουν την τιμή “0xFF”, επομένως τα υπόλοιπα 5 anodes (από το 4° έως και το 8°) θα μένουν σβηστά όπως παρατηρούμε στην παρακάτω εικόνα.

Name	Address	Value	Bits
PINC	0x33	0x08	0x08
DDRC	0x34	0xFF	0xFF
PORTC	0x35	0x08	0x08

Name	Address	Value	Bits
X Register	0x0064 R15	0x00	0x00
Y Register	0x0000 R16	0xFF	0xFF
Z Register	0x0000 R17	0x08	0x08

Όταν το πρόγραμμα μεταβεί στο όγδοο anode, και άρα ο ring counter έχει την τιμή “0b10000000”, ο καταχωρητής X δείχνει στην διεύθυνση “bcd_num+8” (δηλ. 0x0068) η οποία είναι η τελευταία διεύθυνση που έχουμε δεσμεύσει.

Name	Address	Value	Bits
PINC	0x33	0x80	0x80
DDRC	0x34	0xFF	0xFF
PORTC	0x35	0x80	0x80

Name	Address	Value	Bits
X Register	0x0068 R15	0x00	0x00
Y Register	0x0000 R16	0xFF	0xFF
Z Register	0x0000 R17	0x80	0x80

Στο επόμενο interrupt, παρατηρούμε ότι ο ring counter έχει ξανά την τιμή “0b00000001”, επομένως ανάβει πάλι το πρώτο ψηφίο και ο καταχωρητής X δείχνει πάλι στη πρώτη θέση της δεσμευμένης μνήμης και άρα εμφανίζεται ξανά το ψηφίο που θέλουμε, δηλαδή σε αυτή την περίπτωση το 5.

Name	Address	Value	Bits
PINC	0x33	0x01	0x01
DDRC	0x34	0xFF	0xFF
PORTC	0x35	0x01	0x01

Cycle Counter	89933
Frequency	10,000 MHz
Stop Watch	8.993,30 μs

Για να αλλάξει το κάθε ψηφίο χρειαζόμαστε 1msec. Αφού έχουμε 8 ψηφία, το πρώτο ψηφίο θα ανάψει για δεύτερη φορά μετά από περίπου 8msec, όπως βλέπουμε και παραπάνω.

Name	Address	Value	Bits
X Register	0x0061 R15	0x00	0x00
Y Register	0x0000 R16	0x05	0x05
Z Register	0x0000 R17	0x01	0x01

Το πρόγραμμα, συνεχίζει να εκτελείται κατά αυτό τον τρόπο, επομένως, συμπεραίνουμε ότι, ο αριθμός που θέλουμε να εμφανίσουμε στην οθόνη παραμένει στην οθόνη και στα ίδια ψηφία. Στην περίπτωση μας, το “735”, εμφανίζεται στα πρώτα 3 anodes και μένει εκεί.