



6^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"
(Ημερομηνία Παράδοσης: 27/6/2017)

Αφορούν στο Μικροελεγκτή AVR: Να δοθεί ο κώδικας και η θεωρητική παρουσίαση χωρίς να γίνει προσομοίωση. Μόνο όσοι επιθυμούν για δικό τους λόγο (επιβεβαίωσης ορθότητας του προγράμματος) μπορούν να κάνουν χρήση του περιβάλλοντος προσομοίωσης AVR Studio: <http://www.atmel.com/tools/ATMELSTUDIO.aspx>

1^η ΑΣΚΗΣΗ: Να γραφεί πρόγραμμα σε assembly για ένα σύστημα μικροελεγκτή AVR (mega16) το οποίο θα αναβοσβήνει και τα 8 leds (που θεωρούμε ότι είναι συνδεδεμένα με τα bit της θύρας *PortB*) με περίοδο περίπου 2 sec (θα τα κρατάει αναμμένα 0.5 sec και σβηστά για περίπου 1.5 sec). Υποθέτουμε απεικόνιση με θετική λογική - αναμμένο λογικό 1, σβηστό λογικό 0. Οι χρονοκαθυστερήσεις να υπολογιστούν με την αξιοποίηση μιας ρουτίνας *Delay10* που προκαλεί καθυστέρηση 10msec. Τα παραπάνω με την προϋπόθεση ότι το MSB της θύρας εισόδου *PortD* είναι στο λογικό '1'. Αλλιώς οι χρονικές διάρκειες να είναι ανάποδα (1.5sec αναμμένα και 0.5sec σβηστά). Το πρόγραμμα να είναι συνεχούς λειτουργίας.

2^η ΑΣΚΗΣΗ: Να υλοποιηθούν σε ένα σύστημα μικροελεγκτή AVR οι παρακάτω τρεις (3) λογικές συναρτήσεις:

$$X_0 = AB + CD \cdot E \cdot F, \quad X_1 = ABC \cdot D + D \cdot EF', \quad X_2 = X_0 + X_1$$

και να εμφανιστεί η τιμή τους στα τρία LSB της θύρας εξόδου *PortA* (0-2) με την ίδια σειρά. Οι μεταβλητές εισόδου (A, B, C, D, E, F) υποθέτουμε ότι αντιστοιχούν στα 6 MSB της θύρας εισόδου *PortC* (2-7) με την ίδια σειρά. Το πρόγραμμα να δοθεί σε assembly και γλώσσα C.

3^η ΑΣΚΗΣΗ: Να γραφεί πρόγραμμα σε assembly για ένα σύστημα μικροελεγκτή AVR το οποίο αρχικά να ανάβει το led0 που είναι συνδεδεμένο στο bit0 της θύρας εξόδου *PortB* (απεικόνιση με θετική λογική - αναμμένο λογικό 1, σβηστό λογικό 0 - αντίστοιχα και για τα υπόλοιπα ledx => bitx *PortB*).

Στην συνέχεια με το πάτημα των διακοπών (Push-buttons) SW0-4 που υποθέτουμε ότι είναι συνδεδεμένα στα αντίστοιχα bit της θύρας εισόδου *PortD* να συμβαίνουν τα εξής:

- SW0: άναμμα των led4-7 και σβήσιμο των led0-3.
- SW1: άναμμα των led0-3 και σβήσιμο των led4-7.
- SW2: συνεχής αριστερά περιστροφή (κυκλικά) της τρέχουσας κατάστασης των led με χρόνο ~0,1sec (δίνεται ρουτίνα *Delay100* που προκαλεί καθυστέρηση 0,1sec). Όταν επανέλθει ο διακόπτης να σταματάει η περιστροφή και τα led να μένουν στην κατάσταση που ήταν όταν έγινε OFF ο SW2.
- SW3: συνεχής δεξιά περιστροφή (κυκλικά) της τρέχουσας κατάστασης των led με χρόνο ~0,1sec. Όταν επανέλθει ο διακόπτης να σταματάει η περιστροφή και τα led να μένουν στην κατάσταση που ήταν όταν έγινε OFF ο SW3.
- SW4: επαναφορά στην αρχική κατάσταση (άναμμα του led0).

Όλες οι αλλαγές θα γίνονται στο πάτημα των Push-buttons SWx (bitx *PortD*), και η λειτουργία θα συνεχίζεται μέχρι να επανέλθει ο αντίστοιχος διακόπτης. Οι εντολές έχουν προτεραιότητα με μεγαλύτερη αυτή του SW0 και μικρότερη αυτή του SW4. Έτσι αν είναι πατημένο το SW4 και το SW1 τότε θα πραγματοποιηθεί η εντολή που αντιστοιχεί στο SW1. Υποθέτουμε συνεχή λειτουργία και ότι οι διακόπτες είναι συνδεδεμένοι με θετική λογική (για πάτημα δίνουν λογικό '1').

4^η ΑΣΚΗΣΗ: Να γραφεί πρόγραμμα σε Assembly και γλώσσα C, για ένα σύστημα μικροελεγκτή AVR που όταν προκαλείται διακοπή τύπου *INT1* (αντιστοιχεί στο pin *PD3*) να ανάβει τα 8 LED που θεωρούμε ότι είναι συνδεδεμένα στα bit της θύρας εξόδου *PortA* (απεικόνιση με θετική λογική - αναμμένο λογικό 1, σβηστό λογικό 0). Ορίζουμε η διακοπή *INT1* να προκαλείται στην ακμή καθόδου του σήματος. Τα LED να παραμένουν αναμμένα για περίπου 30 sec και μετά να σβήνουν. Υποθέτουμε ότι δίνεται μια ρουτίνα *Delay* που προκαλεί χρονοκαθυστερήση 1 sec. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί η διακοπή να ανανεώνεται ο χρόνος των 30 sec.