*εισαγωγή στους μικροϋπολογιστες*

**Πρόσθετες Γνώσεις Θεωρίας**

* **τι είναι ο μικροϋπολογιστής?**

*Ο καλύτερος τρόπος να περιγραφεί ένας μΥπολογιστής ή μικροελεγκτής είναι να συγκριθεί με έναν κανονικό υπολογιστή (πχ PC ή MAC).*

*Ο υπολογιστής (desktop ή laptop) αποτελείται από μια σειρά από εξαρτήματα και μονάδες, όπως η CPU (πχ Pentium, i5, κτλ), η μνήμη RAM, ο σκληρός δίσκος και οι συσκευές εισόδου και εξόδου (keyboard, mouse, οθόνη). Τα προγράμματα αποθηκεύονται στο σκληρό δίσκο, εκτελούνται από τη CPU με τα προσωρινά δεδομένα να αποθηκεύονται στη RAM και όλα αυτά συντονίζονται από το κυρίως πρόγραμμα που αποκαλείται ‘λειτουργικό σύστημα’.*

*Αντίστοιχα σε έναν μΥπολογιστή υπάρχει η CPU (καλείται και μικροεπεξεργαστής), κάποια ποσότητα μνήμης flash, μνήμη RAM, την EEPROM και ανάλογα τον τύπο υπάρχουν μια σειρά από άλλα στοιχεία (όπως αναλογικό-ψηφιακός μετατροπέας – ADC), όλα αυτά σε ένα chip! Φυσικά οι δυνατότητες ενός μΥπολογιστή απέχουν αρκετά από τις αντίστοιχες ενός PC, αλλά σε γενικές γραμμές η λειτουργία τους είναι αντίστοιχη. Έτσι ο μΥπολογιστής τρέχει όποιο πρόγραμμα είναι αποθηκευμένο στη flash, χρησιμοποιεί την RAM για προσωρινή αποθήκευση δεδομένων και την EEPROM για πιο μακροπρόθεσμη αποθήκευση.*

* **τι καλείται AVR?**

*Οι μικροελεγκτές AVR χρησιμοποιούν τροποποιημένη Αρχιτεκτονική Χάρβαρντ 8-bit RISC (Reduced Instruction Set Computers) και αναπτύχθηκαν από την Atmel για πρώτη φορά το 1996. Η AVR ήταν μια από τις οικογένειες μΥπολογιστών που έκαναν χρήση της μνήμης flash για την αποθήκευση του προγράμματος, σε αντίθεση με τα Programmable ROM, EPROM ή EEPROM που χρησιμοποιούνται από άλλους μΥπολογιστές. Η βασική αρχιτεκτονική των AVR επινοήθηκε από δύο φοιτητές στο Νορβηγικό Ινστιτούτο Τεχνολογίας, τους Alf-Bogen EGIL και Vegard Wollan.*

* **επισκόπηση των μΥπολογιστών AVR**

*Οι AVR κατατάσσονται κυρίως στους:*

***tinyAVR — σειρά ATtiny***

*1-8 kB πρόγραμμα μνήμης
8–32 Ακροδέκτες
Περιορισμένο σετ εντολών χειρισμού περιφερειακών*

***megaAVR — σειρά ATmega***

*4-256 kB πρόγραμμα μνήμης
28-100 Ακροδέκτες
Εκτεταμένο σετ εντολών (Εντολές Πολλαπλασιασμού και μεγαλύτερο χώρος για το πρόγραμμα μνήμης)
Εκτεταμένο σετ εντολών χειρισμού περιφερειακών*

***XMEGA —  σειρά ATxmeg***

*16-384 kB πρόγραμμα μνήμης
44-64-100 Ακροδέκτες
Εκτεταμένα χαρακτηριστικά επιδόσεων, όπως η DMA, "Event System", καθώς και υποστήριξη κρυπτογράφησης.
Εκτεταμένο σετ εντολών χειρισμού περιφερειακών και DAC*

***Ειδικών Εφαρμογών AVR***

*megaAVR με ειδικά χαρακτηριστικά, όπως ελεγκτή LCD , ελεγκτή USB, PWM, CAN*

* **τι είναι τα ενσωματωμένα συστήματα?**

*Ένα ενσωματωμένο σύστημα είναι ένα σύστημα υπολογιστή σχεδιασμένο να πραγματοποιεί ένα ή περισσότερα ειδικά καθήκοντα συχνά με πραγματικούς περιορισμούς υπολογιστών. Είναι ενσωματωμένο καθώς αποτελούν μέρος μιας ολοκληρωμένης συσκευής, και συχνά περιλαμβάνει hardware και μηχανικά μέρη.*

*Συστήματα μΥπολογιστών βρίσκονται ενσωματωμένα σε μια σειρά από συσκευές που χρησιμοποιούνται καθημερινά, όπως είναι η τηλεόραση, ο φούρνος μικροκυμάτων, το πλυντήριο ρούχων, κ.ά. Στη πραγματικότητα, σχεδόν κάθε σύγχρονη οικιακή συσκευή περιέχει έναν μικροεπεξεργαστή για τη διαχείριση των λειτουργιών της, με αποτέλεσμα το 98% των συστημάτων υπολογιστών που διατίθενται σήμερα παγκοσμίως, να είναι ενσωματωμένα.*

* **λειτουργία ενός συστήματος μΥπολογιστή**

*Για να χρησιμοποιηθεί ένα σύστημα μΥπολογιστή θα πρέπει:*

 *α. Να γραφτεί ή να υπάρχει το πρόγραμμα που θα καθορίσει τις λειτουργίες του συστήματος.*

 *β. Θα πρέπει το πρόγραμμα αυτό να μεταγλωττιστεί (compile) σε γλώσσα μηχανής, ώστε να γίνει καταληπτό από τη CPU και τα περιφερειακά στοιχεία της.*

 *γ. Χρησιμοποιώντας κάποιον programmer μεταφέρεται η γλώσσα μηχανής στη μνήμη προγράμματος του μΥπολογιστή.*

 *δ. Στη συνέχεια γίνεται δοκιμή του προγράμματος και εάν χρειαστεί, debugging και νέα δοκιμή.*

* **χαρακτηριστικά του χρησιμοποιούμενου μικροελεγκτή Mega32:**

*8-bit AVR μικροϋπολογιστής*

*Advanced RISC (Reduced Instruction Set Computers) αρχιτεκτονική:*

 *131 εντολές*

 *32x8 καταχωρητές γενικής χρήσης*

*Τμήματα μνήμης αυξημένης αντοχής:*

 *32K bytes In-system Flash μνήμη προγράμματος*

 *1024 (1Κ) bytes μνήμη EEPROM*

 *2048 (2K) bytes εσωτερική SRAM*

*32 προγραμματιζόμενες γραμμές εισόδου/εξόδου (I/O lines)*

*Πολλαπλές περιφερειακές συσκευές:*

 *Δυο 8-bit Timers/απαριθμητές με ξεχωριστούς prescalers*

 *Ένας 16-bit Timer/Counter (απαριθμητής) με ξεχωριστούς prescalers*

 *Analog to Digital Converter (ADC) 8 καναλιών, 10-bit*

 *Προγραμματιζόμενη σειριακή θύρα USART*

 *SPI (Serial Programming Interface)*

 *Προγραμματιζόμενος Watchdog Timer*

* **αναπτυξιακό σύστημα μΥπολογιστών, STK500**

*Το starter kit STK500 είναι ένα σύστημα ανάπτυξης για πολλά μοντέλα του μικροϋπολογιστή AVR της Atmel. Επικοινωνεί με τον υπολογιστή μέσω της σειριακής θύρας RS232 και τροφοδοτείται με DC τάση από 10 έως 15 V. Διαθέτει βάσεις για μοντέλα μικροελεγκτών με 8, 20, 28 και 40 ακροδέκτες, ενώ επιτρέπει είτε παράλληλο είτε σειριακό προγραμματισμό. Περιλαμβάνει επιπλέον 8 switches και 8 leds γενικής χρήσης που μπορούν να λειτουργήσουν σε οποιαδήποτε εφαρμογή. Επιτρέπει την εύκολη πρόσβαση σε όλες τις θύρες εισόδου/εξόδου του εκάστοτε AVR μέσω συνδέσεων ακροδεκτών. Διαθέτει επίσης δεύτερη σειριακή θύρα, για την είσοδο ή έξοδο δεδομένων.*

* **το AVR Studio**

*Το AVR Studio είναι το ιδανικό πρόγραμμα για κάθε εφαρμογή πάνω σε AVR. Περιλαμβάνει έναν συντάκτη/διορθωτή (editor), έναν συμβολομεταφραστή (assembler) και έναν αποσφαλματωτή (debugger) και επιτρέπει την αξιοποίηση κάθε προσομοιωτή AVR, του STK500 και κάθε πλακέτας που βασίζεται στον AVR και έχει σχεδιαστεί κατάλληλα.*

*Το AVR Studio διατίθεται δωρεάν στο site της Atmel (www.atmel.com) και εγκαθίσταται στον υπολογιστή με μια πολύ απλή διαδικασία.*

*Μέσω του περιβάλλοντος του AVR Studio γίνονται όλες οι επιθυμητές ρυθμίσεις σχετικά με το μοντέλο του AVR, τον τρόπο προγραμματισμού, τη μνήμη όπου θα αποθηκευτεί το πρόγραμμα, τις ασφάλειες, τα bits κλειδώματος, την τάση λειτουργίας, τη συχνότητα λειτουργίας κ.λ.π..*

*Αφού όλες οι προεπιλογές ρυθμιστούν, ο μΥπολογιστής AVR που βρίσκεται στην πλακέτα STK500 προγραμματίζεται βάσει της assembly που έχει φτιαχτεί στο AVR Studio.*

* **Παράδειγμα εφαρμογής : LEDs on**

*Το παράδειγμα αυτό έχει ως σκοπό να δημιουργηθεί το πλέον απλό πρόγραμμα, το οποίο θα έχει τη δυνατότητα να ενεργοποιεί τα LEDs του STK500.*

*Ο Mega32 έχει 4 Ports (PortA, PortB, PortC, PortD) το καθένα από τα οποία διαθέτουν 8 γραμμές (ΡΑ0, ΡΑ1, ΡΑ2…ΡΑ7 / ΡΒ0, ΡΒ1…ΡΒ7, κοκ), και οι οποίες μπορούν να χρησιμοποιηθούν είτε ως είσοδοι σημάτων και δεδομένων, είτε ως έξοδοι. Για να καθοριστεί κάποιο port αν θα λειτουργήσει ως είσοδος ή ως έξοδος, θα πρέπει να επιλεγεί η κατάλληλη τιμή που θα τοποθετηθεί στον αντίστοιχο καταχωρητή εισόδου/εξόδου του port (DDRA, DDRB, DDRC, DDRD αντίστοιχα για κάθε port).*

*Έτσι, εάν γραφτεί η 8-bit τιμή ‘****00000000****’ στον καταχωρητή* ***DDRA****, οι γραμμές (ΡΑ0…ΡΑ7) του* ***PORTA*** *καθορίζονται ως* ***είσοδοι*** *του μΥπολογιστή.*

*Αντιστοίχως, εάν γραφτεί η 8-bit τιμή ‘****11111111****’ στον καταχωρητή* ***DDRA****, οι γραμμές (ΡΑ0…ΡΑ7) του* ***PORTA*** *καθορίζονται ως* ***έξοδοι*** *του μΥπολογιστή.*

*Για να εκτελεστεί πχ η εντολή «****θέσε το PORTB (PΒ0,ΡΒ1…ΡΒ7) ως έξοδο του μΥπολογιστή****» , θα πρέπει να γραφτεί σε assembly:*

 ldi R21, 0b11111111

 out DDRB, R21 ;PΒ0,ΡΒ1…ΡΒ7 as outputs

*όπου ο ‘****R21****’ είναι ένας τυχαίος από τους 32 καταχωρητές γενικής χρήσης (R0,R1,R2….R31) που μπορούν να χρησιμοποιηθούν για να αποθηκευουν μια σειρά από 8-bit δεδομένων, το ‘****0b11111111****’ είναι τα δεδομένα που θα γραφτούν (όπου το ‘0b’είναι για να ορίσουμε ότι τα δεδομένα αυτά είναι binary, και το ‘11111111’ τα κυρίως δεδομένα), και* ***ldi*** *και* ***out*** *οι εντολές που χρησιμοποιεί η assembly για να φορτώσει δεδομένα σε έναν* ***καταχωρητή γενικής χρήσης*** *και σε έναν* ***ειδικής χρήσης*** *(όπως είναι ο DDRx), αντίστοιχα.*

*Αφού ορίστηκε λοιπόν το* ***PORTB*** *ως έξοδος δεδομένων από τον μΥ με τη παραπάνω διαδικασία, είναι δυνατόν πλέον να στείλουμε τα δεδομένα που θέλουμε στο port αυτό.*

*Έστω ότι τα δεδομένα που επιθυμούμε να σταλούν στην έξοδο του μΥ είναι μια* ***σειρά από ‘1’.*** *Για να πραγματοποιηθεί η αποστολή αυτή, ακολουθείται η ίδια λογική με αυτή που χρησιμοποιήθηκε παραπάνω για τον ορισμό του port:*

 ldi R21,0b11111111

 out PORTB,R21 ;send ‘1’ to all PORTB outputs

*Με αυτό τον τρόπο ορίστηκε το πλέον απλό σετ τεσσάρων εντολών, ενός προγράμματος το οποίο είναι θα είναι σε θέση να ορίσει το ένα port του μΥ ως έξοδο και να στείλει σε αυτό μια σειρά από άσσους.*

*Για να μπορεί όμως αυτό το σετ εντολών να γίνει ολοκληρωμένο πρόγραμμα, θα πρέπει καταρχήν να οριστεί η αρχή και το τέλος του προγράμματος. Χρησιμοποιούμε δυο εντολές για να θέσουμε τα «άκρα»:*

reset: ;Main program entry point on reset

 *το reset χρησιμοποιείται για να ορίσει την αρχή του προγράμματος (program entry point), και:*

forever:

 rjmp forever ;endless loop

 *Δημιουργείται έτσι ένα loop στο τέλος του προγράμματος, το οποίο λόγω της σχεδίασής του αναγκάζει τον μΥ σε ατέρμονη επανάληψή, ορίζοντας με αυτό τον τρόπο το τέλος της διαδικασίας. Η εντολή* rjmp *οδηγεί τον μΥ σε αναζήτηση της επικεφαλίδας που ακολουθεί την εντολή. Στην προκειμένη περίπτωση δηλαδή ο μΥ διαβάζει το* forever:*σαν επικεφαλίδα, καταχωρεί το σημείο στο οποίο βρίσκεται και όταν ακολουθήσει η εντολή* rjmp forever , *πηγαίνει ξανά στο σημείο που είχε βρει την επικεφαλίδα αυτή και επαναλαμβάνει το loop έως ότου τεθεί εκτός λειτουργίας.*

* **βασικές παράμετροι σχεδίασης :**
1. *Για να μπορέσει ο μΥπολογιστής που χρησιμοποιείται στη συγκεκριμένη περίπτωση (Mega32) να διαβάσει ένα πρόγραμμα που έχει γραφτεί σε assembly, θα πρέπει να οριστεί ο τρόπος που αυτό θα μετατραπεί στη γλώσσα του συγκεκριμένου μΥ. Γι’αυτό τον λόγο,* ***πριν*** *από τον ορισμό της αρχής του προγράμματος* ***μπαίνει πάντοτε η γραμμή:***

.include "m32def.inc"

*Πραγματοποιείται με αυτό τον τρόπο μέσω του AVR Studio, ένα «πάντρεμα» των εντολών του προγράμματος σε assembly, με τη γλώσσα λειτουργίας του Mega32, βάσει του πίνακα* ***m32def.inc*** *που βρίσκεται καταχωρημένος στο AVR Studio.*

1. *Αναφέρθηκε ότι οι καταχωρητές γενικής χρήσης (****General Purpose Working Registers****) είναι συνολικά 32 – από τον R0 έως τον R31. Οι πρώτοι 16 όμως registers (R0,R1,R2…R14,R15) είναι καταχωρητές στους οποίους δεν είναι δυνατόν να φορτωθεί απευθείας κάποια τιμή (όπως στο παράδειγμα, με την εντολή ldi -* ***load immediately****), γι'αυτό και δεν χρησιμοποιούνται τόσο συχνά. Επίσης, οι 6 τελευταίοι καταχωρητές (R26,R27…R31) είναι καταχωρητές που μπορούν να χρησιμοποιηθούν ανά ζεύγη, ώστε να ορίσουν τρεις* ***16-bit registers****, τους* ***Χ****,* ***Υ*** *και* ***Ζ*** *(βλ. σελ. 11, AVR Mega32 manual).*

*Επομένως, όταν χρειάζεται κάποιος καταχωρητής γενικής χρήσης για να «φορτώσει» δεδομένα, συνήθως χρησιμοποιούνται οι ενδιάμεσοι registers,* ***από τον R16 έως τον R25****.*

1. *Λόγω της σχεδίασης της αναπτυξιακής πλακέτας STK500 εάν σταλεί πχ στα led’s του board τα δεδομένα ‘****11111111****’ (που σημαίνει ότι στο επιλεγμένο port εξόδου* ***όλα τα lines θα έχουν τάση +5V****), αυτά δεν θα ενεργοποιηθούν, ενώ αντίθετα όταν σταλούν τα δεδομένα ‘****00000000****’ αυτά θα ανάψουν (βλέπε σχηματικό πλακέτας STK500).*

***Η σύνθεση όλων των παραπάνω δεδομένων και εντολών για την σχεδίαση ενός προγράμματος που θα ενεργοποιεί τα LEDs της STK500, δίνει το ακόλουθο αποτέλεσμα:***

;First lab program : LED’s ON

.include "m32def.inc"

reset: ;Main program entry point on reset

 ldi R21, 0b11111111 ;PA0,1,2,3,4,5,6,7 outputs (STK500 LEDs)

 out DDRB, R21

 ldi R21, 0b11111111 ;write all STK500 LEDs OFF

 out PORTB, R21

forever:

 in R21, 0b00000000 ;set STK500 LEDs ON

 out PORTB, R21

 rjmp forever ;repeat forever