

2 Γενικά χαρακτηριστικά των μικροελεγκτών AVR

Περιεχόμενο-Στόχοι

Το κεφάλαιο αυτό παρουσιάζει την αρχιτεκτονική και τα χαρακτηριστικά των μικροελεγκτών AVR. Όλες αυτές οι πληροφορίες είναι απολύτως απαραίτητες προκειμένου να είναι δυνατή η αξιοποίηση του μικροελεγκτή μέσα από τον αντίστοιχο προγραμματισμό. Η αρχιτεκτονική που θα παρουσιαστεί είναι κοινή για όλα τα κλασικά μοντέλα AVR.

Περιεχόμενα Κεφαλαίου

2.1 Εισαγωγή

2.2 Αρχιτεκτονική

2.1 Εισαγωγή

Δεν υπάρχει καμία αμφιβολία ότι, η οικογένεια μικροελεγκτών AVR της Atmel είναι η δημοφιλέστερη μεταξύ αυτών που χρησιμοποιούνται παγκοσμίως. Λέγοντας AVR, **εννοούμε τους 8bit μικροελεγκτές της οικογένειας**, τους οποίους θα συναντήσουμε στην πλειοψηφία των εφαρμογών, όπως αυτές που βασίζονται στο Arduino, κλπ. Λόγω της φύσης των εφαρμογών που καλούνται να υποστηρίξουν, η αρχιτεκτονική των 8bit είναι ικανοποιητική. Πρόσφατα, η εταιρία Atmel έχει κυκλοφορήσει και τους 32bit μικροελεγκτές AVR. Επιπλέον, τα τελευταία χρόνια έχουν αναπτυχθεί ειδικές πλατφόρμες που επιτρέπουν σε κάθε ενδιαφερόμενο να αναπτύξει εύκολα και γρήγορα εφαρμογές βασισμένες σε μικροελεγκτή.

Αναφορικά τώρα με την αυτόνομη λειτουργία του μικροελεγκτή, αυτή εξασφαλίζεται κυρίως με την υποστήριξη των ακόλουθων λειτουργιών:

(α) Εκτέλεση των προγραμμάτων που είναι φορτωμένα στη μνήμη (μέσω της ΚΜΕ-MCPU)

Η ΚΜΕ διαθέτει Αριθμητική και Λογική Μονάδα (ΑΛΜ), καταχωρητές ειδικής και γενικής χρήσης, αποκωδικοποιητή, κλπ, προκειμένου να είναι δυνατή η εκτέλεση των εντολών του προγράμματος για κάθε εφαρμογή.

(β) Ανταλλαγή σημάτων με το εξωτερικό περιβάλλον (μέσω των θυρών/ακροδεκτών)

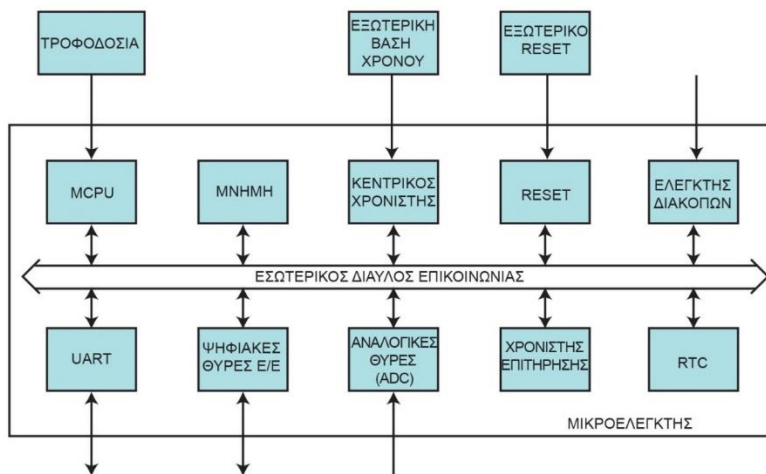
Οι ψηφιακές και αναλογικές θύρες εισόδου/εξόδου, υποστηρίζουν επικοινωνία με το εξωτερικό περιβάλλον. Ιδιαίτερη σημασία έχουν οι ψηφιακές έξοδοι όταν πρόκειται να ελέγξουμε μια εξωτερική συσκευή. Από την άλλη πλευρά, κάθε αναλογική είσοδος μπορεί να αναγνωρίσει αναλογικά σήματα στο διάστημα [0-5]V και να τα αποδώσει σε επίπεδο προγράμματος με 1024 (10 bit) διαφορετικές

στάθμες. Με τάση αναφοράς (AREF) τα 5V, υποστηρίζεται ανάλυση $5/1024=0.0048V$, ενώ αν καθοριστεί διαφορετικά, η ανάλυση μπορεί να προσαρμοστεί για να ανταποκρίνεται καλύτερα στις απαιτήσεις της εφαρμογής.

(γ) Αυτόνομη λειτουργία (τροφοδοσία και χρονισμός)

Η αυτόνομη λειτουργία, δεν αναφέρεται μόνο στην εξωτερική τροφοδοσία, αλλά και στον χρονισμό που εξασφαλίζει την εκτέλεση των εντολών και την ορθή λειτουργία του μικροελεγκτή. Ο χρονισμός αναφέρεται τόσο στο κεντρικό ρολόι λειτουργίας, όσο και στη διαδικασία της επιτήρησης όπως αναφέρθηκαν στο προηγούμενο κεφάλαιο. Στις περισσότερες περιπτώσεις, ο χρονισμός προέρχεται από ένα εξωτερικό κρύσταλλο με τυπική τιμή 16MHz.

Το σχήμα 2.1 δείχνει την εσωτερική οργάνωση ενός μικροελεγκτή όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο. Τα σημαντικότερα τμήματα σχετίζονται με την εκτέλεση των εντολών προγράμματος του χρήστη προκειμένου να καταστεί λειτουργικός ο μικροελεγκτής. Τα προγράμματα και τα δεδομένα φιλοξενούνται στο σύστημα μνήμης, ενώ οι εντολές εκτελούνται από την κεντρική μονάδα επεξεργασίας (ΚΜΕ-MCPU).



Σχήμα 2.1 Δομικά στοιχεία μικροελεγκτή

2.2 Αρχιτεκτονική

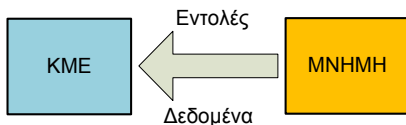
Το σχήμα 2.2 δείχνει την εσωτερική αρχιτεκτονική του μικροελεγκτή. Όπως εύκολα θα διαπιστώσετε, τα συστατικά της αρχιτεκτονικής που αφορούν στη μονάδα επεξεργασίας (MCPU) όπως οι καταχωρητές, η αριθμητική και λογική μονάδα, ο μετρητής προγράμματος, κλπ δεν διαφέρουν στην ουσία από αυτά που διαθέτουν οι κλασικοί μικροεπεξεργαστές. Ο μικροελεγκτής διαθέτει επιπλέον συστατικά (εκτός της MCU), που σκοπό έχουν την υποστήριξη των λειτουργιών που απαιτούνται όπως, η επικοινωνία με τις εξωτερικές συσκευές, η επιτήρηση λειτουργίας, η ανάγνωση αναλογικών σημάτων, κλπ.

Το σχήμα 2.2 είναι εστιασμένο στην εσωτερική αρχιτεκτονική του μικροελεγκτή, στην οποία φαίνονται κυρίως τα συστατικά που αποτελούν την κεντρική μονάδα επεξεργασίας (MCPU).

- **Καταχωρητές γενικής χρήσης (32 το πλήθος).** Πρόκειται για καταχωρητές που είναι διαθέσιμοι στο χρήστη προκειμένου να μπορεί να αναπτύσσει λειτουργικά προγράμματα που περιλαμβάνουν λογική ελέγχου, αριθμητικών και λογικών υπολογισμών, κλπ.
- **Αριθμητική και Λογική Μονάδα (ΑΛΜ).** Εκτέλεση αριθμητικών και λογικών πράξεων.
- **Μνήμη δεδομένων.** Αποθήκευση και ανάγνωση δεδομένων που διαχειρίζονται τα προγράμματα.
- **Μνήμη EEPROM.** Μόνιμη αποθήκευση δεδομένων (δυνατότητα σβησίματος με ηλεκτρικά σήματα)
- **Μονάδα διακοπών.** Πρόκειται για ειδικό κύκλωμα που αναγνωρίζει τις εξωτερικές διακοπές και βοηθά την ΚΜΕ να τις διαχειριστεί κατάλληλα.
- **Μονάδα SPI.** Υποστήριξη ειδικού πρωτοκόλλου επικοινωνίας με εξωτερικές ψηφιακές συσκευές όπως αισθητήρες που δίνουν δεδομένα σε ψηφιακή μορφή.
- **Χρονιστής επιτήρησης.** Ο βασικός χρονιστής εξασφαλίζει τη σωστή συνεργασία των συστατικών (εσωτερικών βαθμίδων), ώστε να είναι δυνατή η εκτέλεση των εντολών. Ο χρονιστής επιτήρησης, λειτουργεί βοηθητικά εξασφαλίζοντας την ομαλή λειτουργία της εφαρμογής από τυχόν δυσλειτουργίες.
- **Αναλογικός συγκριτής.** Υποστήριξη της λειτουργίας ανάγνωσης αναλογικών σημάτων.
- **Υπομονάδες υποστήριξης θυρών (μονάδες Εισόδου/Εξόδου).** Υπομονάδες που εξασφαλίζουν την ανταλλαγή δεδομένων μεταξύ του μικροελεγκτή και μονάδων (διατάξεων και κυκλωμάτων) που βρίσκονται στο εξωτερικό περιβάλλον.

Αρχιτεκτονική Princeton και Harvard

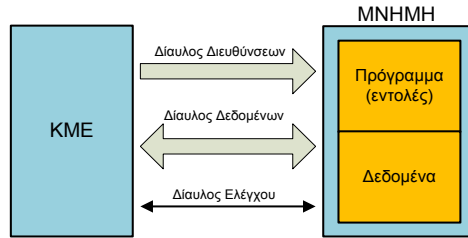
Όπως είναι γνωστό, σε οποιοδήποτε σύστημα βασισμένο σε μικροεπεξεργαστή ή μικροελεγκτή, το πρόγραμμα (εντολές) και τα δεδομένα φιλοξενούνται στο σύστημα μνήμης. Από εκεί μεταφέρονται στην ΚΜΕ προκειμένου να είναι δυνατή η εκτέλεση τους (σχήμα 2.3).



Σχήμα 2.3 Μεταφορά από τη μνήμη στην ΚΜΕ

Ο τρόπος οργάνωσης της μνήμης καθώς και ο τρόπος επικοινωνίας της με την ΚΜΕ επηρεάζει άμεσα την ταχύτητα εκτέλεσης των εντολών, αφού η μεταφορά απαιτεί χρόνο. Όπως είναι λογικό, η ύπαρξη πολλαπλών διαύλων επικοινωνίας (π.χ. διευθύνσεων και δεδομένων) βελτιώνει σε μεγάλο βαθμό αυτή τη διαδικασία, αφού υπάρχει δυνατότητα ταυτόχρονης μεταφοράς εντολών και δεδομένων. Από την πρώτη σχεδίαση τέτοιων συστημάτων, επικράτησαν δύο βασικές αρχιτεκτονικές οι οποίες αναπτύχθηκαν στο Πανεπιστήμιο του Princeton και του Harvard αντίστοιχα

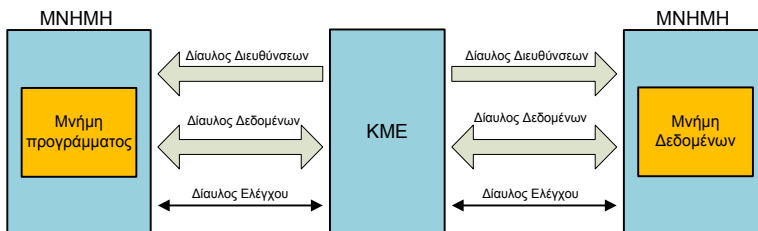
(από τα οποία πήραν και το όνομά τους). Η διαφορά των δύο αρχιτεκτονικών, βασίζεται στον τρόπο οργάνωσης της μνήμης ο οποίος υποστηρίζεται από τη μέθοδο διασύνδεσης με την ΚΜΕ. Στην αρχιτεκτονική Princeton (σχήμα 2.4), οι εντολές του προγράμματος και τα δεδομένα μοιράζονται την ίδια μνήμη, ενώ η επικοινωνία με την ΚΜΕ γίνεται από τους αντίστοιχους διαύλους.



Σχήμα 2.4 Αρχιτεκτονική Princeton

Λόγω αυτού του τρόπου διασύνδεσης, χρειάζονται δύο κύκλοι για την εκτέλεση μιας εντολής, αφού πρέπει να μεταφερθεί τόσο η εντολή, όσο και τα αντίστοιχα δεδομένα. Αυτή η αρχιτεκτονική, προτιμήθηκε την δεκαετία του '70 λόγω της απλότητας υλοποίησης σε συνδυασμό με τους τεχνολογικούς περιορισμούς της εποχής.

Με τη εξέλιξη της τεχνολογίας αλλά και τις νέες απαιτήσεις σε απόδοση, υιοθετήθηκε η αρχιτεκτονική Harvard (σχήμα 2.5) στην οποία χρησιμοποιείται ξεχωριστή μνήμη για πρόγραμμα και δεδομένα. Αυτό πρακτικά σημαίνει ότι, μπορούν να χρησιμοποιηθούν ξεχωριστοί δίαυλοι. Έτσι, μπορούν να υλοποιηθούν διαδικασίες ταυτόχρονα. Με άλλα λόγια, κατά τη διάρκεια της διαδικασίας εκτέλεσης μιας εντολής, μπορεί να γίνεται και η μεταφορά της επόμενης. Με αυτό τον τρόπο, η εκτέλεση μιας εντολής διαρκεί μόνο ένα κύκλο.

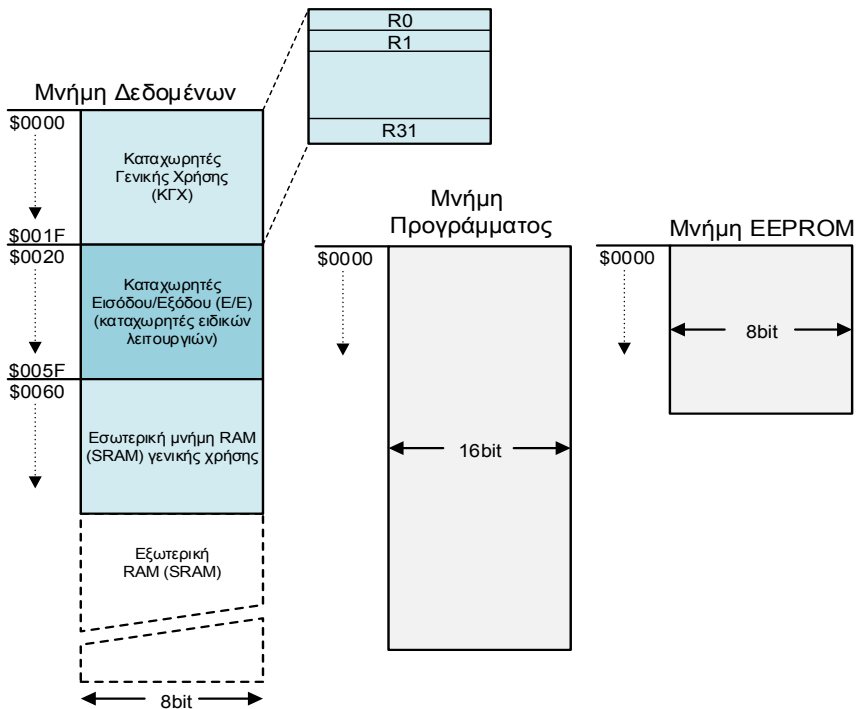


Σχήμα 2.5 Αρχιτεκτονική Harvard

Οι μικροελεγκτές AVR βασίζονται στην αρχιτεκτονική Harvard.

Το σύστημα μνήμης του AVR

Όπως αναφέρθηκε προηγουμένως, οι μικροελεγκτές AVR διαθέτουν ξεχωριστές περιοχές μνήμης τόσο για τα δεδομένα, όσο και για τα προγράμματα. Το σχήμα 2.6 δείχνει τη μνήμη δεδομένων που υλοποιείται με μια μνήμη τύπου SRAM και τη μνήμη προγράμματος για την οποία χρησιμοποιείται μνήμη Flash. Ταυτόχρονα, φαίνεται και η μνήμη EEPROM η οποία υπάρχει στην πλειοψηφία των μικροελεγκτών AVR και φιλοξενεί δεδομένα που πρέπει να διατηρηθούν και κατά την απουσία τροφοδοσίας. Ανάλογα το μοντέλο AVR, το μέγεθος των περιοχών μνήμης μπορεί να διαφέρει, ενώ υπάρχει η δυνατότητα χρήσης εξωτερικής μνήμης RAM (SRAM).



Σχήμα 2.6 Το σύστημα μνήμης του AVR

Πιο συγκεκριμένα, η δομή και η χρήση των περιοχών μνήμης που φαίνονται στο σχήμα 2.6, είναι:

Μνήμη δεδομένων

Φιλοξενεί τα δεδομένα όπως αυτά διαμορφώνονται κατά τη διάρκεια εκτέλεσης των προγραμμάτων, ενώ προσφέρει πρόσβαση και αξιοποίηση των δυνατοτήτων του μικροελεγκτή μέσα από τους καταχωρητές. Κάθε θέση έχει χωρητικότητα 8bit.

Καταχωρητές γενικής χρήσης

Πρόκειται για 32 καταχωρητές 8bit που είναι διαθέσιμοι στον προγραμματιστή για την ανάπτυξη του κώδικα των εφαρμογών (R0 έως R31)

Καταχωρητές Εισόδου/Εξόδου

Είναι 64 καταχωρητές με χωρητικότητα 8bit και προσφέρουν πρόσβαση στο υλικό και τις δυνατότητες του μικροελεγκτή. Για παράδειγμα, η αποστολή δεδομένων σε εξωτερικό κύκλωμα μέσω κάποιας θύρας επικοινωνίας, γίνεται με την εγγραφή στον αντίστοιχο καταχωρητή ελέγχου. Η ίδια φιλοσοφία χρησιμοποιείται για την πρόσβαση σε χρονοστάθμους, κλπ.

Εσωτερική μνήμη RAM

Χρησιμοποιείται για την αποθήκευση δεδομένων που χρησιμοποιεί το πρόγραμμα

Εξωτερική μνήμη RAM

Η εξωτερική μνήμη αποτελεί εξωτερικό κύκλωμα και χρησιμοποιείται για επέκταση της χωρητικότητας της μνήμης δεδομένων

Μνήμη προγράμματος

Φιλοξενεί τον κώδικα των εφαρμογών με χωρητικότητα που ποικίλει από μοντέλο σε μοντέλο, ενώ η χωρητικότητα κάθε θέσης είναι 16bit (2 byte)

Μνήμη EEPROM

Πρόκειται για μνήμη περιορισμένης χωρητικότητας η οποία φιλοξενεί δεδομένα που διατηρούνται και κατά την απουσία τροφοδοσίας. Αν για παράδειγμα ο χρήστης καταχωρήσει κάποιες ρυθμίσεις για την εφαρμογή, τότε αυτές αποθηκεύονται στη μνήμη δεδομένων με αποτέλεσμα να χάνονται όταν αφαιρείται η τροφοδοσία. Για να διατηρηθούν αυτές οι ρυθμίσεις, χρησιμοποιούμε τη μνήμη EEPROM, ώστε στην επόμενη εκκίνηση της εφαρμογής να μην χρειαστεί επανακαθορισμός των ρυθμίσεων.

Οι καταχωρητές εισόδου/εξόδου έχουν πρόσβαση απευθείας στο υλικό του μικροελεγκτή, ενώ σε μερικές περιπτώσεις είναι περισσότεροι από ότι στα πιο κλασικά μοντέλα AVR. Έτσι, στα μοντέλα που διαθέτουν περισσότερους από 32 ακροδέκτες (θύρες) εισόδου/εξόδου, υπάρχει στη μνήμη δεδομένων μια ακόμα περιοχή (εκτεταμένη μνήμη εισόδου/εξόδου).

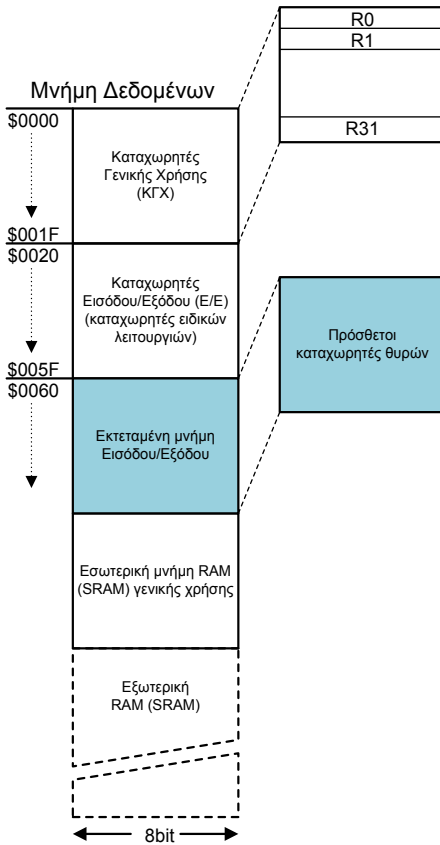
Το σχήμα 2.7 δείχνει τη μνήμη δεδομένων με την εκτεταμένη περιοχή που υποστηρίζει περισσότερους από 32 ακροδέκτες. Ο πίνακας 2.1 δείχνει το μέγεθος της μνήμης σε διάφορα μοντέλα μικροελεγκτών AVR.

Πίνακας 2.1 Μεγέθη μνήμης διαφόρων μοντέλων AVR

Μοντέλο AVR (ενδεικτικά)	Μνήμη RAM (κύρια μνήμη δεδομένων)	Μνήμη Flash (μνήμη προγράμματος)	Μνήμη EEPROM (μνήμη δεδομένων)
ATmega48	512byte	4Kbyte	256byte
ATmega88	1Kbyte	8Kbyte	512byte
ATmega168	1Kbyte	16Kbyte	512byte
ATmega328	2Kbyte	32Kbyte	1Kbyte
ATmega32	2Kbyte	32Kbyte	1Kbyte
ATmega640	8Kbyte	64Kbyte	4Kbyte
ATmega1280	8Kbyte	128Kbyte	4Kbyte
ATmega2560	8Kbyte	256Kbyte	4Kbyte

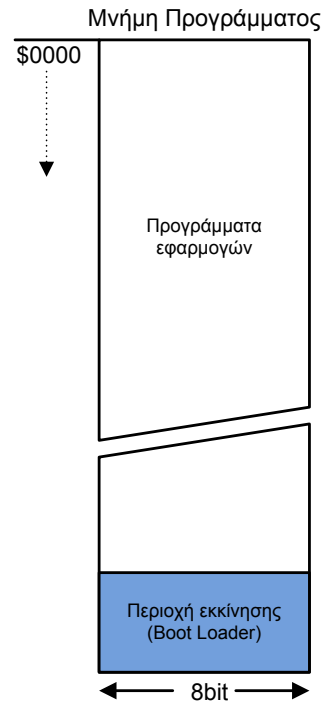
Σημείωση

Βάσει του προθέματος, 1Kbyte=1x10³ byte=1000byte. Επειδή όμως διευθύνσεις και δεδομένα εκφράζονται πάντα στο δυαδικό σύστημα, θα πρέπει τα αντίστοιχα byte (1 byte = 8bit) να αντιπροσωπεύουν αυτά τα ψηφία. Έτσι, 1Kbyte=1024 byte, αφού 2¹⁰=1024 (πλησιέστερα στο 1000). Στην πράξη, αναφέροντας για παράδειγμα μέγεθος μνήμης 8Kbyte, εννοούμε 8x1Kbyte=8x1024byte=8192 byte.



Σχήμα 2.7

Εκτεταμένη μνήμη εισόδου εξόδου



Σχήμα 2.8

Περιοχή εκκίνησης (Boot Loader)

Σε πολλά μοντέλα AVR, η μνήμη προγράμματος χωρίζεται σε δύο περιοχές (π.χ. ATmega328), (α) την περιοχή του προγράμματος των εφαρμογών και (β) την περιοχή εκκίνησης (Boot Loader-BL). Το σχήμα 2.8 δείχνει τις δυο περιοχές μνήμης που προαναφέρθηκαν.

Ο Boot Loader υποστηρίζει τη δυνατότητα εύκολου προγραμματισμού του μικροελεγκτή. Επιπλέον, κατά την ενεργοποίηση της λειτουργίας του μικροελεγκτή, η εκτέλεση των εντολών μπορεί να ξεκινήσει από αυτή την περιοχή. Ο προγραμματισμός αυτής της περιοχής παρουσιάζει εξαιρετικό ενδιαφέρον από την άποψη του πλήρους ελέγχου της λειτουργίας του μικροελεγκτή.

Καταχωρητές γενικής χρήσης

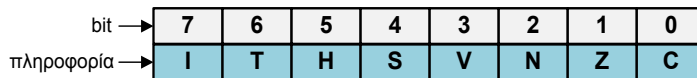
Οι μικροελεγκτές AVR 8bit διαθέτουν 32 καταχωρητές γενικής χρήσης (8 bit) οι οποίοι αντιστοιχούν σε συγκεκριμένες διευθύνσεις. Επιπλέον, μια ομάδα από αυτούς, αποτελούν τους καταχωρητές X,Y,Z (16bit) που προσφέρουν αυξημένες δυνατότητες σε σχέση με τους υπόλοιπους καταχωρητές. Το σχήμα 2.9 δείχνει τους καταχωρητές γενικής χρήσης με τις αντίστοιχες διευθύνσεις, αλλά και το σχηματισμό των καταχωρητών X,Y και Z.

		7	0	Διεύθυνση	
Καταχωρητές Γενικής Χρήσης		R0		\$00	
		R1		\$01	
		.			
		.			
		R15		\$0F	
		R16		\$10	
		.			
		.			
		R26		\$1A	X (χαμηλής τάξης byte)
		R27		\$1B	X (υψηλής τάξης byte)
		R28		\$1C	Y (χαμηλής τάξης byte)
		R29		\$1D	Y (υψηλής τάξης byte)
		R30		\$1E	Z (χαμηλής τάξης byte)
		R31		\$1F	Z (υψηλής τάξης byte)

Σχήμα 2.9 Καταχωρητές γενικής χρήσης

Καταχωρητής Κατάστασης (SREG-Status Register)

Από τους καταχωρητές ειδικής χρήσης, ο σημαντικότερος που ενδιαφέρει τον προγραμματιστή είναι ο καταχωρητής κατάστασης. Οι πληροφορίες που περιέχει είναι απολύτως απαραίτητες, αφού μέσα από αυτές μπορούμε να ανιχνεύσουμε τη μορφή ενός αποτελέσματος της αριθμητικής και λογικής μονάδας, κλπ. Το σχήμα 2.10 δείχνει τη δομή αυτού του καταχωρητή.



Σχήμα 2.10 Bit Καταχωρητή Κατάστασης (SREG)

Στον πίνακα 2.2 φαίνεται η πληροφορία που δίνει κάθε bit του καταχωρητή κατάστασης.

Πίνακας 2.2 Πληροφορίες Bit Καταχωρητή Κατάστασης (SREG)

Bit	Περιγραφή
I	Ενεργοποίηση και απενεργοποίηση των διακοπών που προέρχονται από το εξωτερικό περιβάλλον
T	Χρησιμοποιείται για χειρισμό bit μέσω των εντολών BLD και BST
H	Κρατούμενο που προέρχεται από το τρίτο στο τέταρτο bit ενός αριθμού (π.χ. σε μια εντολή ADD)
S	Bit προσήμου. Η τιμή του προέρχεται με αποκλειστικό 'H μεταξύ των bit N και V
V	Υπερχείλιση σε πράξη με συμπλήρωμα ως προς 2
N	Ενεργοποιείται όταν το περισσότερο σημαντικό ψηφίο από κάποιο αποτέλεσμα είναι 1 (στην ουσία πρόκειται για το αρνητικό πρόσημο)
Z	Δείχνει πότε το αποτέλεσμα μιας πράξης είναι μηδέν
C	Κρατούμενο (ή δανεικό σε αφαίρεση) που έχει προκύψει σε αποτέλεσμα κάποιας πράξης

Καταχωρητές E/E (I/O Registers)

Η πλήρης αξιοποίηση των αρχιτεκτονικών χαρακτηριστικών και των δυνατοτήτων του μικροελεγκτή (π.χ. χρονιστές, επικοινωνία με εξωτερικές συσκευές), γίνεται

κυρίως μέσα από ειδικούς καταχωρητές εισόδου/εξόδου. Η προσπέλαση των καταχωρητών μπορεί να γίνει είτε με το συμβολικό όνομα, είτε με τη διεύθυνση. Για παράδειγμα, μέσω του καταχωρητή DDRB (\$17) μπορούμε να καθορίσουμε την κατεύθυνση των δεδομένων των ακροδεκτών της ψηφιακής θύρας B, ενώ με τον PORTB μπορούμε να ενεργοποιήσουμε συγκεκριμένους ακροδέκτες (π.χ. να ανάψουμε ένα LED).

Ο πίνακας 2.3 δείχνει ενδεικτικά τους διαθέσιμους καταχωρητές E/E του μικροελεγκτή AVR.

Πίνακας 2.3 Καταχωρητές E/E

Συμβολικό όνομα	Περιγραφή
TWBR	Καθορισμός ρυθμού για το Serial Clock - SCL
TWSR	Κατάσταση διπλού σειριακού διαύλου
TWAR	Καταχωρητής διεύθυνσης για επικοινωνία με εξωτερική συσκευή-slave
TWDR	Data/Address Shift Register διεπαφής επικοινωνίας
ADCL	Low byte μέτρησης του αναλογικού σε ψηφιακό μετατροπέα (ADC)
ADCH	High byte μέτρησης του αναλογικού σε ψηφιακό μετατροπέα (ADC)
ADCSRA	Καταχωρητής ελέγχου και κατάστασης του ADC
ADMUX	Καταχωρητές επιλογέα πολυπλέκτη ADC
ACSR	Καταχωρητής κατάστασης αναλογικού συγκριτή
UBRR1	Low byte για τον καθορισμό του baud rate του UART
UCSRB	Καταχωρητής ελέγχου και κατάστασης B του UART
UCSRA	Καταχωρητής ελέγχου και κατάστασης A του UART
UDR	Καταχωρητής που δείχνει ότι το UART είναι έτοιμο για λήψη
SPCR	Καταχωρητής ελέγχου SPI
SPSR	Καταχωρητής κατάστασης SPI
SPDR	Καταχωρητής δεδομένων SPI
PIND	Ανάγνωση από τους ακροδέκτες της θύρας D
DDRD	Καθορισμός κατεύθυνσης δεδομένων της θύρας D
PORTD	Εγγραφή δεδομένων (έξοδος) στη θύρα D
PINC	Ανάγνωση από τους ακροδέκτες της θύρας C
DDRC	Καθορισμός κατεύθυνσης δεδομένων της θύρας C
PORTC	Εγγραφή δεδομένων (έξοδος) στη θύρα C
PINB	Ανάγνωση από τους ακροδέκτες της θύρας B
DDRB	Καθορισμός κατεύθυνσης δεδομένων της θύρας B
PORTB	Εγγραφή δεδομένων (έξοδος) στη θύρα B
PINA	Ανάγνωση από τους ακροδέκτες της θύρας A
DDRA	Καθορισμός κατεύθυνσης δεδομένων της θύρας A
PORTA	Εγγραφή δεδομένων (έξοδος) στη θύρα A
EEDR	Καταχωρητής ελέγχου EEPROM
EEDR	Καταχωρητής δεδομένων EEPROM
EEARL	Low byte καταχωρητή διευθύνσεων EEPROM
EEARH	High byte καταχωρητή διευθύνσεων EEPROM
UBRRH	High byte για τον καθορισμό του baud rate του UART
WDTCSR	Καταχωρητής κατάστασης του χρονιστή επιτήρησης (WatchDog)
ASSR	Ασύγχρονος καταχωρητής κατάστασης για τον χρονιστή/μετρητή 1/2
OCR2	Καταχωρητής για σύγκριση με την τιμή του χρονιστή/μετρητή 2
TCNT2	Καταχωρητής για προσπέλαση του χρονιστή/μετρητή 2
TCCR2	Καταχωρητής ελέγχου χρονιστή/μετρητή 2
ICR1L	Low byte καταχωρητή που ενημερώνεται από την τιμή του χρονιστή/μετρητή 1 όταν συμβεί συγκεκριμένο γεγονός

ICR1H	High byte καταχωρητή που ενημερώνεται από την τιμή του χρονιστή/μετρητή 1 όταν συμβεί συγκεκριμένο γεγονός
OCR1BL	Low byte καταχωρητή B του οποίου το περιεχόμενο συγκρίνεται συνεχώς με το περιεχόμενο του χρονιστή/μετρητή 1 ώστε να παραχθεί κάποιο συγκεκριμένο γεγονός
OCR1BH	High byte καταχωρητή B του οποίου το περιεχόμενο συγκρίνεται συνεχώς με το περιεχόμενο του χρονιστή/μετρητή 1 ώστε να παραχθεί κάποιο συγκεκριμένο γεγονός
OCR1AL	Low byte καταχωρητή A του οποίου το περιεχόμενο συγκρίνεται συνεχώς με το περιεχόμενο του χρονιστή/μετρητή 1 ώστε να παραχθεί κάποιο συγκεκριμένο γεγονός
OCR1AH	High byte καταχωρητή A του οποίου το περιεχόμενο συγκρίνεται συνεχώς με το περιεχόμενο του χρονιστή/μετρητή 1 ώστε να παραχθεί κάποιο συγκεκριμένο γεγονός
TCNT1L	Low byte Καταχωρητή για απευθείας πρόσβαση στον χρονιστή/μετρητή 1
TCNT1H	High byte Καταχωρητή για απευθείας πρόσβαση στον χρονιστή/μετρητή 1
TCCR1B	Καταχωρητής ελέγχου B χρονιστή/μετρητή 1
TCCR1A	Καταχωρητής ελέγχου A χρονιστή/μετρητή 1
SFIOR	Καταχωρητής ειδικών συναρτήσεων (π.χ. ρύθμιση pull-up αντιστάσεων)
OCDR	Καταχωρητής που χρησιμοποιείται για αποσφαλμάτωση (debugging) σε συνδυασμό με το αντίστοιχο λογισμικό προγραμματισμού
OSCCAL	Καταχωρητής συχνότητας για calibration του ταλαντωτή χρονισμού
TCNTO	Καταχωρητής για απευθείας πρόσβαση στον χρονιστή/μετρητή 0
TCCRO	Καταχωρητής ελέγχου χρονιστή/μετρητή 0
MCUCSR	MCPU καταχωρητής κατάστασης και ελέγχου (πληροφορίες για την πηγή που προκάλεσε το πιο πρόσφατο reset)
MCUCR	MCPU καταχωρητής κατάστασης (π.χ. γεγονός διακοπής)
TWCR	Καταχωρητής κατάστασης σειριακού διαύλου
SPMCR	Καταχωρητής ελέγχου για την αποθήκευση στη μνήμη προγράμματος
TIFR	Καταχωρητής κατάστασης διακοπής για τον μετρητή/χρονιστή
TIMSK	Καταχωρητής μάσκας διακοπής χρονιστή
GIFR	Καταχωρητής κατάστασης διακοπών (π.χ. για τους ακροδέκτες που υποστηρίζουν διακοπές)
GICR	Καταχωρητής ελέγχου διακοπών (π.χ. για τους ακροδέκτες που υποστηρίζουν διακοπές)
OCRO	καταχωρητής του οποίου το περιεχόμενο συγκρίνεται συνεχώς με το περιεχόμενο του χρονιστή/μετρητή 0 ώστε να παραχθεί κάποιο συγκεκριμένο γεγονός
SPL	Low byte του δείκτη σωρού
SPH	High byte του δείκτη σωρού
SREG	Καταχωρητής κατάστασης

Σημείωση 1

Κάθε καταχωρητής E/E αντιστοιχεί σε μια διεύθυνση μνήμης και σε μια διεύθυνση θύρας. Αυτό σημαίνει όμως ότι, οι διευθύνσεις αυτές δεν είναι ίδιες σε όλα τα μοντέλα των μικροελεγκτών AVR. Για παράδειγμα, ο καταχωρητής ελέγχου κατεύθυνσης DDRB της θύρας B, αντιστοιχεί σε διεύθυνση θύρας \$04 για τον ATmega328 και στην \$17 για τον ATmega8515. Έτσι, για να εξασφαλίσουμε τη λειτουργικότητα ενός προγράμματος (να τρέχει σε διαφορετικούς μικροελεγκτές) που χρησιμοποιεί για παράδειγμα τη θύρα B, δεν χρησιμοποιούμε τη διεύθυνση θύρας αλλά απλά το συμβολικό όνομα DDRB (π.χ. για τον έλεγχο κατεύθυνσης δεδομένων). Βέβαια, για την ενεργοποίηση της αντιστοίχισης μεταξύ διευθύνσεων

και συμβολικών ονομάτων καταχωρητών, ενσωματώνουμε αρχεία τύπου .inc στον αντίστοιχο κώδικα ανάλογα με τον τύπο του μικροελεγκτή που χρησιμοποιούμε.

Σημείωση 2

Σε όλα τα σχήματα που παρουσιάστηκαν, χρησιμοποιήθηκαν αριθμοί του δεκαεξαδικού συστήματος. Οι αριθμοί στο δεκαεξαδικό σύστημα γράφονται με το πρόθεμα '\$' ή '0x'. Για παράδειγμα, ο αριθμός 0F μπορεί να γραφεί \$0F ή 0x0F. Το σύμβολο '\$' χρησιμοποιείται συνήθως για διευθύνσεις, ενώ το '0x' για σταθερές ποσότητες (αριθμούς).

ΦΥΛΛΑ ΑΣΚΗΣΕΩΝ

1. Σημειώστε για κάθε πρόταση, αν είναι σωστή ή λάθος

Πρόταση	Σωστή	Λάθος
Ένας μικροελεγκτής υποστηρίζει εφαρμογές γενικού σκοπού	<input type="checkbox"/>	<input type="checkbox"/>
Μια αναλογική είσοδος αναγνωρίζει τάσεις στο διάστημα [0-5]V	<input type="checkbox"/>	<input type="checkbox"/>
Η μονάδα MCPU δεν έχει κοινά στοιχεία με την CPU των συμβατικών υπολογιστών	<input type="checkbox"/>	<input type="checkbox"/>
Η μνήμη Flash του μικροελεγκτή χάνει τα περιεχόμενά της κατά την απουσία τροφοδοσίας	<input type="checkbox"/>	<input type="checkbox"/>
Η αρχιτεκτονική Harvard προσφέρει υψηλότερες επιδόσεις στον μικροελεγκτή από ότι η Princeton	<input type="checkbox"/>	<input type="checkbox"/>
Όλες οι εντολές φιλοξενούνται στην ΚΜΕ	<input type="checkbox"/>	<input type="checkbox"/>
Η MCPU αναφέρεται στην ΚΜΕ	<input type="checkbox"/>	<input type="checkbox"/>
Στην αρχιτεκτονική Harvard έχουμε διπλό δίαυλο δεδομένων και μονό δίαυλο διευθύνσεων	<input type="checkbox"/>	<input type="checkbox"/>
Η μνήμη EEPROM διατηρεί τα περιεχόμενά της κατά την απουσία της τροφοδοσίας	<input type="checkbox"/>	<input type="checkbox"/>

2. Συμπληρώστε τις λέξεις ή το κείμενο που λείπει

2.1 Ο Μετρητής Προγράμματος δείχνει τη _____ της επόμενης εντολής που πρόκειται να _____ από την ΚΜΕ.