



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 411 – ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021

Καθ. Α. Δόλλας

Εργαστήριο 1

ΕΞΟΙΚΕΙΩΣΗ ΜΕ ΤΗΝ ΟΙΚΟΓΕΝΕΙΑ ΜΙΚΡΟΕΛΕΓΚΤΩΝ ATMEL AVR

ΕΚΔΟΣΗ : 2.0

**Προθεσμία: Παρασκευή 15 Οκτωβρίου 2021, έως τα μεσάνυχτα
Ηλεκτρονική υποβολή στο eClass**

**Όλα τα Εργαστήρια είναι ΑΤΟΜΙΚΑ και όχι κατά ομάδες (προς
το παρόν)**

Σκοπός - Βήματα

Σκοπός του εργαστηρίου είναι η εξοικείωση με το περιβάλλον ανάπτυξης για μικροελεγκτή AVR, με την δημιουργία απλού προγράμματος (δύο παραλλαγές, καμιά 10αριά γραμμές η κάθε παραλλαγή). Η εξοικείωση περιλαμβάνει εγκατάσταση του εργαλείου, παρακολούθηση των tutorials και δημιουργία ενός απλού προγράμματος για ένα μικροελεγκτή της οικογενείας ATMEL AVR. Το πρόγραμμα, **πρέπει να είναι σε Assembly** (αλλά όχι σε C ακόμη), και **πρέπει να υλοποιεί ένα απλό χρονιστή (ενδεικτικά, 20msec)**. Υλοποιείστε το με δύο τρόπους: με βρόχο κατάλληλου αριθμού επαναλήψεων (κυρίως πρόγραμμα), και, με χρήση χρονομετρητών-χρονιστών/μετρητών (TIMER/COUNTER), όπου το κυρίως πρόγραμμα στην περίπτωση αυτή είναι ένας ατέρμονας βρόχος και τον χρόνο τον μετράμε με το Interrupt του χρονιστή. Κατόπιν, υπολογίσετε στο χαρτί πόσες εντολές εκτελέσατε σε κάθε περίπτωση για το μέτρημα μόνο (δηλαδή όχι οι εντολές του ατέρμονα βρόχου στην δεύτερη περίπτωση) για κάποια ενδεικτική περίοδο, π.χ.

1sec. Επειδή για να βρει κανείς την σωστή τιμή για τον χρονιστή περιλαμβάνει κάποιους υπολογισμούς, θεωρήσετε ότι το ρολόϊ του μικροελεγκτή σας είναι 10MHz – εσείς κάνετε υπολογισμούς για κάτι που να είναι «κοντά» στην επιθυμητή τιμή – κάποιες φορές η ακρίβεια υπολογισμών επηρεάζεται από τις τιμές που μπορούν να μπουν σε prescaler, κλπ. Επίσης, για να υπολογίσετε σωστά τις εντολές στην υλοποίηση του χρονιστή με βρόχο, θεωρήσετε ότι κάθε εντολή του AVR χρειάζεται ένα κύκλο ενώ η εντολή διακλάδωσης χρειάζεται δύο κύκλους.

Γιατί τα κάνουμε όλα αυτά; Για εξοικείωση με την πλατφόρμα και τα εργαλεία μας, και για να καταλάβουμε γιατί το Arduino αν δεν γνωρίζουμε καλά από μικροελεγκτές είναι χρήσιμη πλατφόρμα μόνο για χρήστες που δεν έχουν σχέση με το αντικείμενο. Όντως, η συμβατική λύση με το Arduino είναι η υλοποίηση βρόχου με πρόγραμμα και όχι με χρήση χρονιστή.

Επίτηδες η εκφώνηση αφήνει ανοιχτό το κατά πόσον ο βρόχος θα επαναλαμβάνεται ή θα είναι μόνο μία φορά, αν με τον τερματισμό κάποιος ακροδέκτης εξόδου θα γίνεται «1» (ή αν επαναλαμβάνεται ο βρόχος κατά πόσον ανά ένα μέτρημα του χρονιστή θα γίνεται εναλλάξ 1,0,1, κλπ.), ποιος TIMER θα χρησιμοποιηθεί, κλπ. Αυτά όλα πρέπει να απαντηθούν από εσάς, αλλά οποιαδήποτε απάντηση οδηγεί σε στοιχειωδώς σωστή λύση είναι αποδεκτή. Οι κωδικές σας πρέπει να έχουν επαρκή σχόλια στα Αγγλικά (όχι Greeklish ή Ελληνικά), με τον ενδεδειγμένο τρόπο τεκμηρίωσης, δηλ. Block Comments στην αρχή (είδος προγράμματος, όνομα φοιτητή/φοιτήτριας, ημερομηνία, περιβάλλον ανάπτυξης, λειτουργικότητα), και Block Comments στα βασικά δομικά στοιχεία (π.χ. το αντίστοιχο του `main()` και υπορουτίνες), κατόπιν δε επαρκή Line Comments. Τα labels **πρέπει** να έχουν κατάλληλα ονόματα.

Τα βήματα περιλαμβάνουν:

1. Εγκατάσταση ATMEL Studio 7.0 και εξοικείωση με αυτό, από τον ιστότοπο (που έχει και tutorials):
<https://www.microchip.com/mplab/avr-support/atmel-studio-7>
2. Εξοικείωση κατ' αρχήν με την οικογένεια ATMEL AVR και το ATMEL Studio.
3. Σχεδίαση των δύο προγραμμάτων στο χαρτί – δηλαδή, δική σας επιλογή του ποιον TIMER θα έχετε για το πρόγραμμα που τον χρησιμοποιεί, επιλογή Prescaler για τον TIMER, επιλογή του αν θα είναι ατέρμων βρόχος που θα επαναλαμβάνεται (δηλ. ο TIMER θα ξανατίθεται σε λειτουργία) ή θα τρέχει μία φορά μόνο, κλπ. Ομοίως, για το δεύτερο πρόγραμμα δική σας επιλογή καταχωρητών

- που θα χρησιμοποιηθούν για τον βρόχο που να τον υλοποιεί με κώδικα, και αντίστοιχη λειτουργικότητα με το πέρας του ενός δευτερολέπτου. Χρησιμοποιήσετε τον **AT90S8515** ή κάποιο παρόμοιο μικροελεγκτή υλοποίησης (υπάρχουν διάφορες παραλλαγές, ο προτεινόμενος είναι αυτός που έχουν αρχικά τα αναπτυξιακά συστήματα **STK500** του Εργαστηρίου). Αποφύγετε πάντως τους υπερβολικά μικρούς μικροελεγκτές που δεν θα έχουν αρκετούς πόρους, ή τους υπερβολικά μεγάλους που ενδεχόμενα θα τρέχουν αργά στον προσομοιωτή και θα αντιστοιχούν σε εξωτικά ολοκληρωμένα κυκλώματα που δεν έχουμε στο εργαστήριο. Οποιαδήποτε συσκευή με 2+1 timers, με 40 ακροδέκτες DIP, και 8-bit αρχιτεκτονική είναι επαρκέστατη (γενικά κάτι από **ATMega16**).
4. Υλοποίηση των προγραμμάτων στο ATME Studio, προσομοίωση και επιβεβαίωση λειτουργίας.
 5. Ανάλυση: στο διάστημα του ενός sec πόσες εντολές έτρεξε κάθε πρόγραμμα χρονιστή; Περιλάβετε στην ανάλυση και τις εντολές για να αρχικοποιήσετε τον TIMER αλλά και για να θέσετε σε κάποιο ακροδέκτη την σωστή τιμή όταν ολοκληρωθεί το δευτερόλεπτο.
 6. Γράψετε και υποβάλετε μαζί με τους κώδικες μαζί με την τελική σας αναφορά. Οικώδικες που θα υποβληθούν στο eClass, μαζί με στοιχεία για την δοκιμή λειτουργίας του προγράμματος, πρέπει να είναι σε μορφή που να μπορούν να δοκιμαστούν (όχι απλά pdf).
 7. Η αναφορά πρέπει να είναι κειμενογραφημένη, και το πολύ δύο-τρεις σελίδες. Δεν χρειάζεται να βάλετε όλους τους κώδικες πηγής, μόνο κάποια ψύγματα, υπολογισμούς για το πως θέσατε τις παραμέτρους στον χρονιστή, και την βασική δομή του κώδικά σας (π.χ. τι κάνει όταν τελειώσει; - Αυτό έχει διαφορετική απάντηση ανά φοιτητή). Επίσης λίγες γραμμές για το πως δοκιμάσατε ότι δουλεύει είναι απαραίτητες.

Παρουσίαση /Εξέταση

Επειδή το μάθημα γίνεται όπως έχει εξηγηθεί στην τάξη, η κατάθεση κωδίκων κλπ. στο eClass θα κατοχυρώνει την δουλειά που κάνετε, μαζί με την **σύντομη** αναφορά, ενώ **η εξέταση θα γίνει με όποιο τρόπο είναι εφικτός** (ενδεικτικά: για πολλά εργαστήρια μαζί στο εργαστήριο, με On-line εξέταση στο εργαστήριο ή Off-line όποτε μπορώ να τρέξω μόνος μου τους κώδικες), κλπ. Προφανώς προβλήματα κατά οποιαδήποτε εξέταση (ενδεικτικά - να έχετε κατεβάσει κώδικες από κάπου και να μην κατανοείτε πως λειτουργούν) θα συμπαρασύρει και την βαθμολογία όσων εργαστηρίων είχαν εξεταστεί Off-line, ακόμη και αν δεν πρόκειται για αντιγραφή μεταξύ φοιτητών.

Η αναφορά εκτός από την περιγραφή των κωδίκων πρέπει να περιλαμβάνει και το πως υπολογίσατε τον Prescaler (αν το κάνατε μέσα από το ATME Studio είναι μια χαρά αλλά πρέπει να πείτε πως π.χ. λάβατε υπόψη την συχνότητα του ρολογιού). Η αναφορά δεν χρειάζεται να είναι πάνω από 2-3 περιεκτικές σελίδες.

ΒΟΗΘΗΜΑ 1: Αν σκοπεύετε να ανάψετε κάποιο «λαμπάκι» - δηλαδή να κάνετε κάποιο ακροδέκτη «1», πρέπει στην αρχικοποίηση του προγράμματος να ορίσετε την κατεύθυνση του αντίστοιχου ακροδέκτη σαν έξοδο, να κάνετε τα περιεχόμενα «0» και με το τέλος του προγράμματος να κάνετε τα περιεχόμενα «1».

ΒΟΗΘΗΜΑ 2: Διαβάστε τι είναι και πως χρησιμοποιούνται τα breakpoints - είναι εξαιρετικά χρήσιμο εργαλείο για την αποσφαλμάτωση κώδικα.

ΠΡΟΣΟΧΗ (τα ξέρετε, αλλά τα ξαναθυμίζουμε)!

1) Η προεργασία να είναι σε ηλεκτρονική μορφή και μαζί με αρχεία με κώδικες που να μπορούμε να εκτελέσουμε. Το αρχείο πρέπει να το υποβάλλετε στο eClass.

2) Η έλλειψη προετοιμασίας ή επαρκούς τεκμηρίωσης οδηγεί σε απόρριψη.

3) Η διαπίστωση αντιγραφής σε οποιοδήποτε σκέλος της άσκησης οδηγεί στην απόρριψη όλων των εμπλεκομένων από το σύνολο των εργαστηριακών ασκήσεων, άρα και του μαθήματος. Αυτό γίνεται οποιαδήποτε στιγμή στη διάρκεια του εξαμήνου. Ως αντιγραφή νοείται και μέρος της αναφοράς, π.χ. σχήματα.

ΚΑΛΗ ΕΠΙΤΥΧΙΑ! ☺