



ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΧΟΛΗ ΗΜΜΥ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ & ΥΛΙΚΟΥ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ ΓΙΑ ΤΟ ΜΑΘΗΜΑ:

ΗΡΥ 411 – ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2021

Καθ. Α. Δόλλας

Εργαστήριο 5

ΚΩΔΙΚΑΣ C ΣΤΟΝ ATMEΛ AVR

ΕΚΔΟΣΗ : 1.1

**Προθεσμία: Κυριακή 14 Νοεμβρίου 2021, έως τα μεσάνυχτα
Ηλεκτρονική υποβολή στο eClass**

**Το παρόν εργαστήριο είναι το τελευταίο που γίνεται ΑΤΟΜΙΚΑ
και όχι κατά ομάδες, οπότε φροντίστε να δημιουργήσετε
ομάδες των δύο φοιτητών για το Project που έπεται**

Σκοπός - Βήματα

Σκοπός του εργαστηρίου είναι να αρχίσουμε να γράφουμε κώδικα σε C αλλά έχοντας υπόψη την χρήση των πόρων του AVR σε χαμηλό επίπεδο.

Περιγραφή του Εργαστηρίου

Το εργαστήριο είναι λειτουργικά ισοδύναμο με το Εργαστήριο 4, εκτός του ότι **όλοι οι κώδικες θα είναι πλέον σε C**, περιλαμβανομένων και των Interrupt/Timer Service Routines. Θα πρέπει όμως όταν γράφουμε κώδικα να έχουμε κατά νου το πως τρέχει στους πόρους του AVR γιατί πάνω στο Εργαστήριο 5 θα χτίσουμε κατά το υπόλοιπο εξάμηνο εφαρμογές όλο και πιο απαιτητικές. Οι κώδικες πρέπει να είναι καθαροί και καλά δομημένοι, με καλή τεκμηρίωση και εύκολο να εξελιχθούν.

Εκτέλεση του Εργαστηρίου - Αλλαγές στον Κώδικα του Εργαστηρίου 4

Τα προφανή: κρατάμε το μέρος `main()` χωρίς αλλαγές και γράφουμε σε C **όλες πλέον** τις ρουτίνες για `TIMER`, σειριακή θύρα, κλπ. Από πλευράς κώδικα το εργαστήριο είναι τελείως τετριμμένο, και μάλιστα είναι πιο εύκολο από το Εργαστήριο 4. Ο λόγος που ξοδεύουμε μία ολόκληρη εβδομάδα σε αυτό είναι για να μελετήσετε σε βάθος την διεπαφή των διαφορετικών μερών του κώδικα, την χρήση της μνήμης, την αρχικοποίηση από τον `compiler της στοίβας για πέρασμα ορισμάτων` (που δεν είχαμε μέχρι τώρα), αλλά και την αποτελεσματικότητα του κώδικα που παράγεται. Επί πλέον, θέλουμε και τα εξής:

(α) Αν δεν έχετε δεσμεύσει/χρησιμοποιήσει μνήμη για την υλοποίηση των αποκωδικοποιητών `BCD-to-7-Segment-LED` πρέπει να το κάνετε στο εργαστήριο αυτό, φροντίζοντας η μνήμη που χρησιμοποιείτε να μην «πέφτει επάνω» στην στοίβα.

(β) Αν χρησιμοποιείτε `global` μεταβλητές μην τις συνδέσετε με συγκεκριμένο καταχωρητή. Να είσαστε φειδωλοί γιατί στο μέλλον θα πρέπει να κάνετε πολύ περισσότερους υπολογισμούς με τους ίδιους πόρους – όσο πιο προσεκτικά γραμμένος είναι ο κώδικάς σας τόσο λιγότερες αλλαγές θα χρειαστεί στο μέλλον.

(γ) Να έχετε κατανόηση του τι δομές `Assembly` δημιουργούνται.

Όσον αφορά την εκτέλεση του κώδικα, μάθετε (αν δεν έχετε μάθει ήδη) πως να κάνετε αποσφαλμάτωση με `Breakpoints` και να παρακολουθείτε το `Call Stack` (και γενικότερα την μνήμη). Θυμίζουμε ότι στην φιλοσοφία των μικροελεγκτών είναι κάποιο μέρος του κώδικα να «πειράζει» κάποιο μέρος της μνήμης (ή κάποιο καταχωρητή) και κάποιο άλλο μέρος του κώδικα ή του `hardware` να ενεργεί επί των περιεχομένων. Το έχουμε δει σε μικρή κλίμακα στην διεπαφή λογισμικού με την θύρα `USART` και σε κάπως μεγαλύτερη κλίμακα με τους αριθμούς που γράφουμε για να φανούν στην οθόνη, ενώ άλλος κώδικας κάνει πολυπλεξία στον χρόνο για την εμφάνιση. Για να αποσφαλματώσει κανείς τέτοιο κώδικα πρέπει να μπορεί να ελέγχει περιεχόμενα μνήμης, καταχωρητών (με την ευρεία έννοια – όχι μόνο καταχωρητές τύπου `rx`), ακροδεκτών, κλπ.

Παρουσίαση / Εξέταση / Αναφορά / Θέματα Βαθμολογίας

Ισχύει ότι και για τα προηγούμενα εργαστήρια. Φυσικά πρέπει να βάλετε στο `eClass` αρχεία εισόδου (`stimulus files`), κλπ. Πρακτικά, όταν εξεταστείτε επί των κωδίκων σας που υποβάλατε στο `eClass` και σας ρωτήσω (στην τελική εξέταση)

τι δοκιμάσατε ώστε να ξέρετε ότι δουλεύει ένα εργαστήριο, θα πρέπει να μπορείτε με τα αρχεία που έχετε αναρτήσει στο eClass να μου το δείξετε. Όπως και στα προηγούμενα εργαστήρια, αν διορθώνετε σε τρέχοντα εργαστήρια ελλείψεις προηγούμενων εργαστηρίων αυτό θα είναι προς όφελός σας, αλλά αν δεν το κάνετε θα βρείτε τις ελλείψεις μπροστά σας...

Ισχύει ότι τα εργαστήρια θα διορθωθούν όλα μαζί, στο τέλος, αλλά μην νομίζετε πως δεν βλέπω δειγματοληπτικά τι γίνεται. Τυπικά έχουμε και υποβολές με ελλείψεις στην αναφορά μέχρι «τσαπατσούλικους» κώδικες με τελείως επιφανειακή σχέση με τα ζητούμενα, ή κώδικες χωρίς stimulus file και είτε με γενικότητες στις αναφορές είτε με αναφορές που είναι απλό αναμάρτημα του κώδικα. Χωρίς να μπαίνω σε λεπτομέρειες, θέλω να πάρετε όλοι/όλες καλό βαθμό, αλλά αυτός θα πρέπει να αντιστοιχεί στην ποιότητα της δουλειάς σας. Με δεδομένο ότι βελτιώσεις σε μελλοντικά εργαστήρια συμπαρασύρουν βαθμό και παλαιότερων εργαστηρίων, αυτό σημαίνει ότι υπάρχει καλή πιθανότητα όλοι/όλες να βελτιώσετε τον βαθμό σας, αλλά εφόσον υπάρχει σταθερή προσπάθεια και πρόοδος, αν κάποιος δώσει 8-9 εργαστήρια του «5» και 1-2 του «10», και με καλή τελική εξέταση δεν θα μείνει στο 5,5 αλλά θα πάει στο 6 - 6,5, και σίγουρα όχι στο 10... Το ζητούμενο είναι να μάθετε, και αυτό προκύπτει από μία διαρκή στοχευμένη προσπάθεια (σε λογικό χρόνο όμως).

Επίσης, θα ήθελα να ενθαρρύνω τον σημαντικό αριθμό φοιτητών/φοιτητριών που πραγματικά βλέπω από τα εργαστήριά σας να προοδεύετε, να αφομοιώνετε την ύλη, και να εξελίξεστε - νομίζω και εσείς οι ίδιοι/ίδιες βλέπετε την πρόοδό σας που φαίνεται ακόμη και στο «στιλ» του κώδικα και την αυτοπεποίθηση που αποκτάτε με την τεχνολογία αυτή.

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 εμπλεκόμενων 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! ☺