

# HPY411- Ενσωματωμένα Συστήματα Μικροεπεξεργαστών

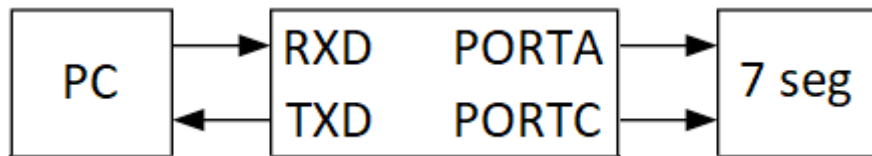
## Εργαστήριο 6

LAB41145851

19/11/2020

Εμμανουήλ Πετράκος AM 2014030009

Στο 6ο εργαστήριο επεκτείνεται το σύστημα με software reset μετά την λήψη μηνυμάτων στο USART. Η είσοδος παραμένει ο receiver του και τα μηνύματα έχουν την ίδια μορφή αλλά αλλάζει η αντιμετώπιση τους. Ως έξοδος του συστήματος παραμένουν οι θύρες A και C όπου οδηγούν τα 7 segment LED καθώς και ο transmitter του USART όπου στέλνει μηνύματα επιβεβαίωσης λήψης και επαναφοράς.

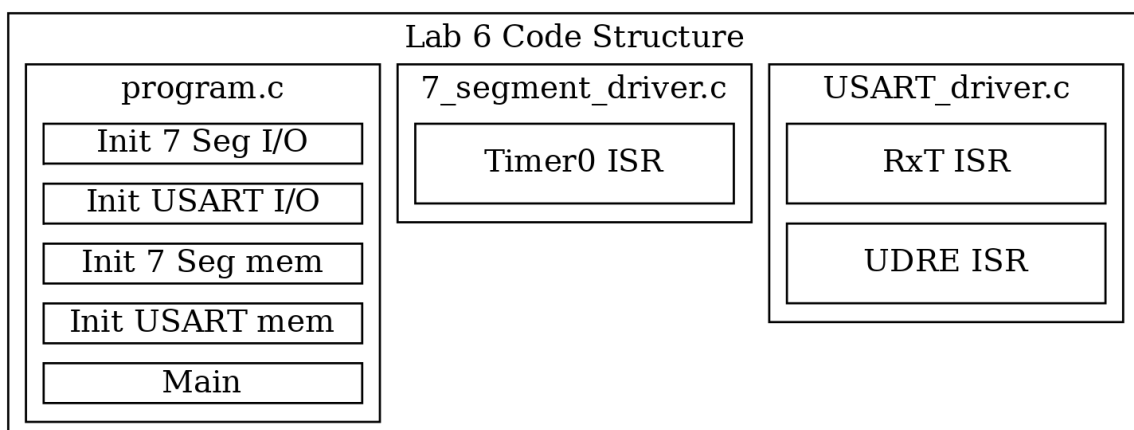


Top level schematic

### Επεξήγηση προσέγγισης

Πριν την υλοποίηση του reset, πρέπει να προετοιμαστεί ο προϋπάρχων κώδικας για την ενσωμάτωση του. Σύμφωνα με τον manual, το reset του watchdog timer καθαρίζει αυτόματα όλους τους I/O καταχωρητές αλλά δεν επηρεάζει η μνήμη. Τότε, στην περίπτωση του warm start πρέπει να επαναρχικοποιηθούν μόνο αυτοί και να μην γίνει κάποια αλλαγή στην μνήμη. Για αυτό, οι συναρτήσεις αρχικοποίησης σπάνε σε δύο κομμάτια η καθεμία, ώστε να γίνονται ξεχωριστά το σετάρισμα των I/O καταχωρητών και της SRAM. Η νέα δομή του κώδικα φαίνεται στο παρακάτω διάγραμμα.

Έχουν υλοποιηθεί warm και cold reset. Στο αρχείο program.h γίνεται define η σταθερά `warm_start_enable` η οποία δείχνει ποιο από τα δύο χρησιμοποιείται.



Η χρήση του watchdog timer απαιτεί τρεις αποφάσεις:

- Που και πως ενεργοποιείται/απενεργοποιείται/γίνεται reset.
- Τι γίνεται όταν ξυπνήσει και πως αντιδράει το σύστημα.
- Τι κάνει το σύστημα μετά την επαναρχικοποίηση.

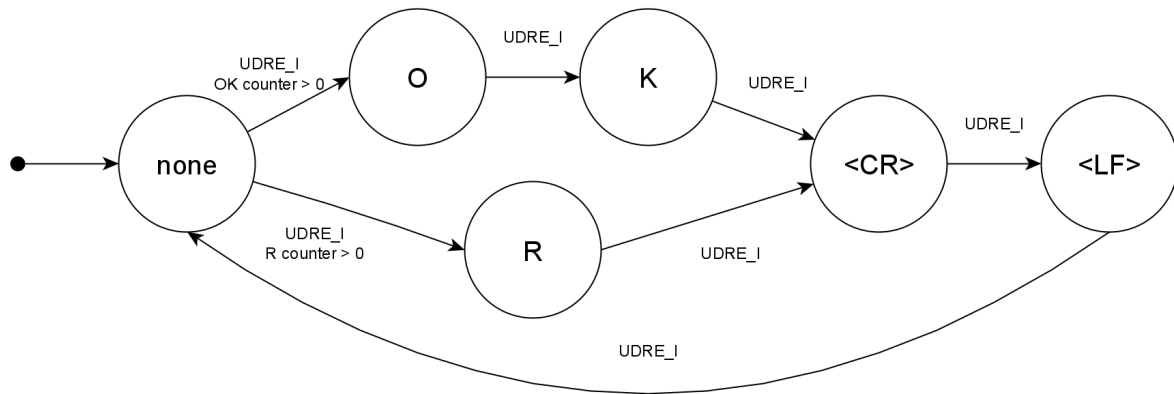
Σύμφωνα με τις προδιαγραφές του συστήματος, το reset πρέπει να ενεργοποιείται με οποιαδήποτε μετάδοση προς τον μικροελεγκτή. Για αυτό, στο τέλος της ISR του receiver, ο watchdog timer γίνεται reset. Ο simulator εμφανίζει ένα bug όπου ο watchdog timer τρέχει με το ρολόι του επεξεργαστή και όχι με τον δικό του ταλαντωτή του 1MHz. Για αυτό, κατά την προσομοίωση με ρολόι 10MHz πρέπει να χρησιμοποιείται 10 φορές μεγαλύτερη καθυστέρηση από την επιθυμητή. Επίσης, όπως έχει αναλυθεί στα προηγούμενα εργαστήρια, η αποστολή μιας απάντησης διαρκεί από 2 με 4 millisecond ανάλογα το φόρτο του buffer στον transmitter. Για να δοθεί αρκετός χρόνος για απάντηση προς το PC, η ενεργοποίηση του watchdog timer γίνεται με το macro `“wdt_enable(WDTO_60MS)”`.

Η εκκίνηση του συστήματος γίνεται από την διεύθυνση \$000 ανεξαρτήτου πηγής. Για να γίνουν οι σωστές αρχικοποιήσεις πρέπει να είναι γνωστό από που προέρχεται. Ο καταχωρητής MCUCSR, στο bit PORF δείχνει ότι έγινε λόγο power-on ενώ στο bit WDRF δείχνει ότι προέρχεται από τον watchdog timer.

- Στην περίπτωση του power-on και του cold reset πρέπει να αρχικοποιηθεί όλο το σύστημα. Αυτό γίνεται μέσω του παραγόμενου από τον compiler κώδικα για τον stack pointer και τις συναρτήσεις αρχικοποίησης μνήμης και I/O καταχωρητών.
- Αντίθετα, στην περίπτωση του warm start, πρέπει να αρχικοποιηθεί μόνο ο stack pointer και οι I/O καταχωρητές καθώς καθαρίζονται από το reset. Αυτό γίνεται με τον ίδιο τρόπο με την προηγούμενη περίπτωση. Λαμβάνοντας υπόψιν ότι οι global μεταβλητές είναι δηλωμένες έτσι ώστε ο compiler να μην δημιουργεί κώδικα αρχικοποίησης για αυτές (έχει αναλυθεί στο 4ο εργαστήριο), αποφεύγοντας τις συναρτήσεις αρχικοποίησης μνήμης τα δεδομένα μένουν ως έχει και επιτυγχάνεται το warm start.

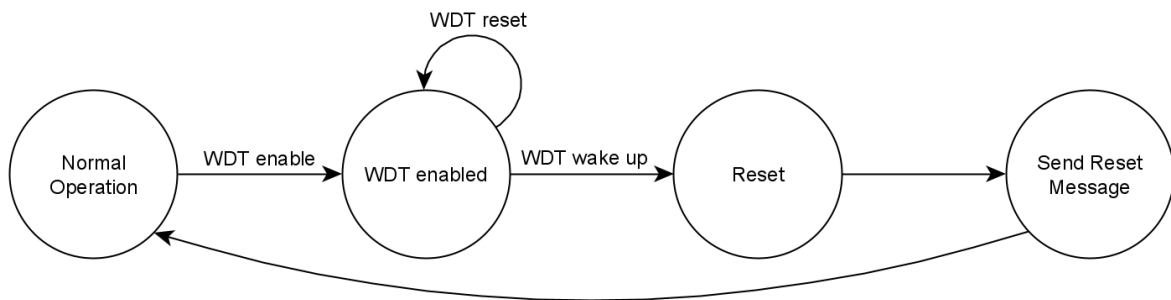
Τέλος, και τις δύο περιπτώσεις software reset, ενεργοποιείται το global interrupt flag και γίνεται αποστολή μηνύματος reset από την σειριακή θύρα ενεργοποιώντας το interrupt του UDRE και αυξάνοντας τον μετρητή εκκρεμών απαντήσεων reset.

Για την αποστολή της απάντησης reset (R<CR><LF>) έχει επεκταθεί η ρουτίνα εξυπηρέτησης UDRE Interrupts με μια νέα κατάσταση για την αποστολή του χαρακτήρα O. Επίσης, υπάρχουν δύο μετρητές εκκρεμών απαντήσεων, ένας για τα OK και ένας για τα R. Η λειτουργία τους είναι ίδια με αυτόν του 4ου εργαστηρίου. Τώρα, επειδή μετά από κάθε μήνυμα μπορεί να χρειαστεί να σταλούν 7 χαρακτήρες, αυξάνεται η πιθανότητα να ζητηθεί η μετάδοση απάντησης ενώ δεν έχει τελειώσει η προηγούμενη και μεγαλώνει η ανάγκη το σύστημα να ξέρει πόσες απαντήσεις περιμένουν. Στο παρακάτω διάγραμμα φαίνεται η νέα FSM της ISR του UDRE. Μετά από αυτό το σύστημα συνεχίζει κανονικά.



*FSM αποστολής απαντήσεων*

Η νέα λειτουργικότητα περιγράφεται σχηματικά με το παρακάτω διάγραμμα.



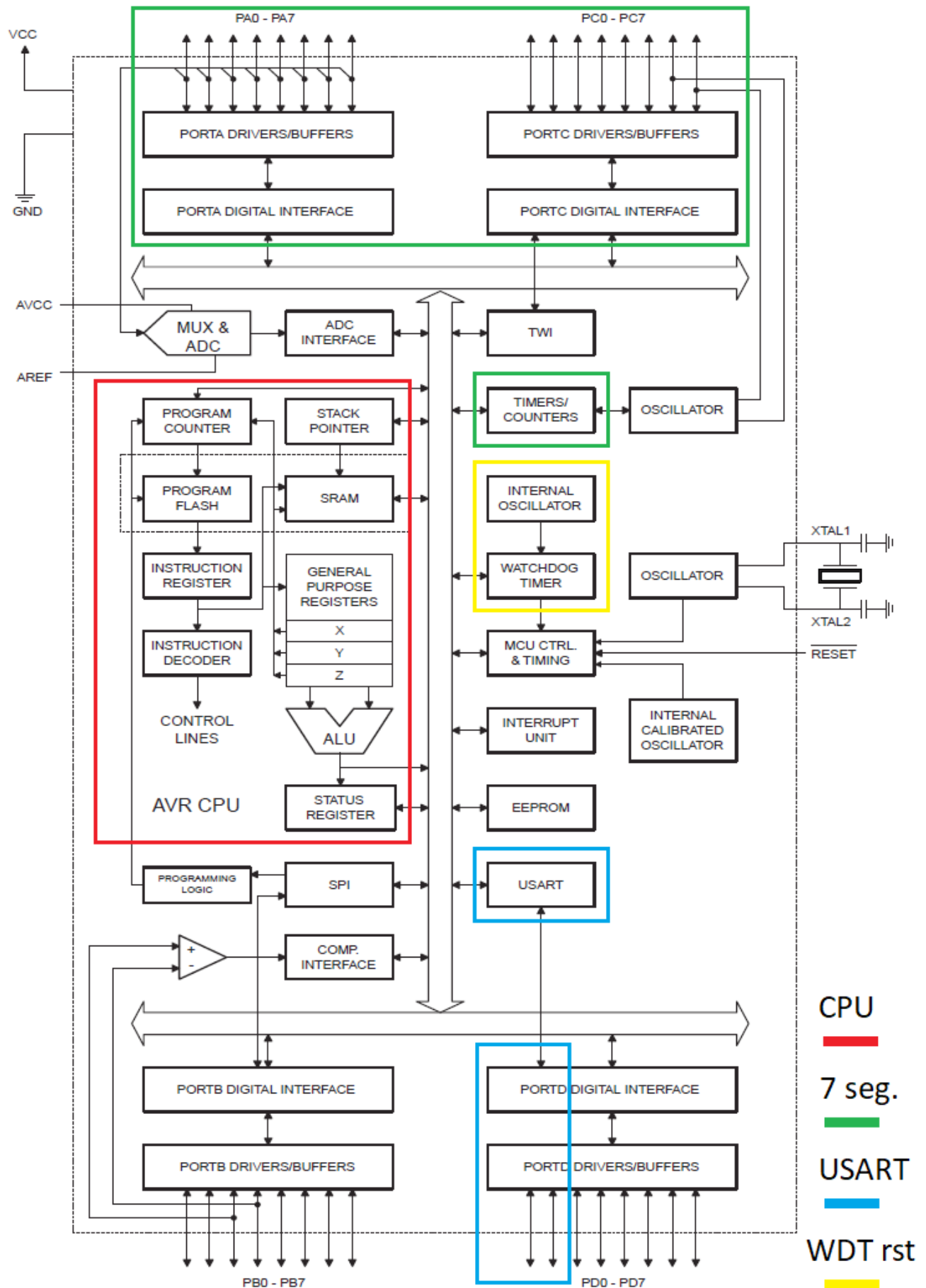
*Διάγραμμα λειτουργίας Software Reset με Watchdog timer*

Στο παρακάτω πίνακα φαίνεται πως χαρτογραφεί την μνήμη ο compiler και με γκρι χρώμα υπογραμμίζονται οι αλλαγές του παρόντος εργαστηρίου.

0x0000	
...	Register address space
0x005F	
0x0060	OK transmits left
0x0061	R transmits left
0x0062	Transmitter state
0x0063	7 segment format 0
...	...
0x006d	7 segment format A
0x006e	Data LSB
...	...
0x0075	Data MSB
...	Unused memory
...	
...	Stack
0x045f	

*Χάρτης μνήμης*

Τέλος, παρουσιάζεται ο χάρτης χρησιμοποιούμενων πόρων.



## Πειραματική Διαδικασία

Για τον έλεγχο της λειτουργίας του συστήματος στέλνεται μια εντολή μέσω του USART και παρακολουθούνται η μνήμη, ο stack pointer και οι I/O καταχωρητές πριν και μετά την διαδικασία του reset. Αυτό γίνεται χρησιμοποιώντας ένα breakpoint στην αντιμετώπιση των <LF> frames, δηλαδή στο τέλος των μηνυμάτων, και ένα breakpoint μετά το τέλος των αρχικοποιήσεων. Επίσης, με ένα breakpoint στο τέλος της ISR του receiver φαίνεται η ανανέωση του watchdog timer.

Warm Start:

Εντολή	Κατάσταση μνήμης δεδομένων, 0x006e – 0x0075 (LSB → MSB)							
-	0a	0a	0a	0a	0a	0a	0a	0a
N1234<CR><LF>	04	03	02	01	0a	0a	0a	0a
Wdt Reset	04	03	02	01	0a	0a	0a	0a

Cold Start:

Εντολή	Κατάσταση μνήμης δεδομένων, 0x006e – 0x0075 (LSB → MSB)							
-	0a	0a	0a	0a	0a	0a	0a	0a
N1234<CR><LF>	04	03	02	01	0a	0a	0a	0a
Wdt Reset	0a	0a	0a	0a	0a	0a	0a	0a

Και στις δύο περιπτώσεις, από το παράθυρο Processor Status φαίνεται ότι ο Stack Pointer αρχικοποιείται με την τιμή 0x045f. Επίσης, από το παράθυρο I/O φαίνεται ότι οι εξωτερικοί καταχωρητές αρχικοποιούνται σύμφωνα με τις συναρτήσεις αρχικοποίησης I/O.

Τέλος, οι απαντήσεις του USART είναι ίδιες σε κάθε περίπτωση και παρουσιάζονται παρακάτω. Φαίνεται ότι πέρα από την αναγνώριση λήψης του μηνύματος, δείχνει και ότι έκανε reset. Επίσης, το καθάρισμα του καταχωρητή κατά το reset δεν φαίνεται στο PC γιατί δεν ενεργοποιεί interrupt.

```
1 #61121
2 TCNT2 = 0x4f  O
3 #45
4 TCNT2 = 0x4b  K
5 #10401
6 TCNT2 = 0x0d  <CR>
7 #10404
8 TCNT2 = 0x0a  <LF>
9 #44628
10 TCNT2 = 0x00  Reset clears I/O regs
11 #116
12 TCNT2 = 0x52  R
13 #48
14 TCNT2 = 0x0d  <CR>
15 #10396
16 TCNT2 = 0x0a  <LF>
```

## Ανάλυση & Παρατηρήσεις

Παρατηρείται ότι από το reset του wdt που γίνεται ακριβώς πριν την αποστολή του “Ο” μέχρι και το καθάρισμα του συστήματος περνάνε περίπου 65000 κύκλοι, δηλαδή 6,5 ms. Σύμφωνα με το manual, σε αυτή τη ρύθμιση η καθυστέρηση έπρεπε να είναι 65ms. Δηλαδή επιβεβαιώνεται το bug του simulator και σε περίπτωση που ο κώδικας θα εφαρμοστεί σε πραγματικό μικροελεγκτή η ρύθμιση πρέπει να αλλάξει.

Ένας χαρακτήρας ενός εισερχόμενου μηνύματος βρίσκεται σε μια από τις τρεις καταστάσεις:

- Έχει διαβαστεί και βρίσκεται στην μνήμη.
- Βρίσκεται στον buffer του receiver ή είναι στην διαδικασία του διαβάσματος.
- Δεν έχει σταλεί ακόμα.

Παρατηρήται ότι μπορεί ένα εισερχόμενο μήνυμα να κοπεί αν για κάποιο λόγο καθυστερήσει το διάβασμα ενός frame τόσο ώστε να ξυπνήσει ο watchdog timer. Όταν το reset απενεργοποιεί το interrupt του receiver γίνεται flush ο buffer του.

- Στην περίπτωση του warm start, τα δεδομένα που έχουν προλάβει να γραφτούν στην μνήμη παραμένουν. Επίσης, οι χαρακτήρες που θα σταλθούν αφού επανενεργοποιηθούν τα interrupt θα γραφτούν. Όμως θα λείπουν οι χαρακτήρες που χάθηκαν στον buffer και το μήνυμα είναι ελλιπές.
- Στην περίπτωση του cold start, τα δεδομένα στην μνήμη σβήνονται μαζί με τα δεδομένα στον buffer και θα εμφανίζονται μόνο οι χαρακτήρες που στάλθηκαν μετά το reset.

Και στις δύο περιπτώσεις, αυτή η συμπεριφορά είναι ανεπιθύμητη και μπορεί να αντιμετωπιστεί με δύο αλλαγές στο κώδικα.

- Η πρώτη είναι οι αλλαγές στη μνήμη να γίνεται με μέθοδο double buffering. Δηλαδή να γράφεται ένας buffer όταν έρχεται ο χαρακτήρας N ή ένας αριθμός και μόλις το μήνυμα τελειώσει με <LF> αυτός να αντιγράφεται στην μνήμη που εμφανίζουν τα 7 segments LED.
- Η δεύτερη αλλαγή είναι να χρησιμοποιηθεί μια global μεταβλητή που δείχνει ότι το σύστημα έχει ξεκινήσει την ανάγνωση ενός μηνύματος. Παρατηρώντας το πίνακα των μηνυμάτων της εκφώνησης του εργαστηρίου 3, φαίνεται ότι όλα ξεκινάνε με A, C ή N και ότι αυτοί οι χαρακτήρες δεν εμφανίζονται πουθενά αλλού. Αν εμφανιστεί ένας από αυτούς του χαρακτήρες στο USART, το σύστημα ξέρει ότι ξεκίνησε ένα μήνυμα και θα κάνει την μεταβλητή 1. Όταν τελειώσει ένα μήνυμα ή γίνει ένα reset η μεταβλητή γίνεται 0. Αν έρθει ένας οποιοσδήποτε άλλος χαρακτήρας ενώ είναι 0, το σύστημα ξέρει ότι το μήνυμα έχει κοπεί και τον αγνοεί.

Συνδυάζοντας τις δύο παραπάνω αλλαγές, αν κοπεί ένα μήνυμα, το σύστημα θα αγνοήσει το τέλος του και ο buffer δεν θα γραφτεί ποτέ στην μνήμη. Επίσης, δεν στέλνεται απάντηση OK<CR><LF> και το PC γνωρίζει ότι η εντολή του μηνύματος δεν εκτελέστηκε.

Η υλοποίηση του παραπάνω φαίνεται να είναι εύκολη, αλλά δεν εφαρμόστηκε για να παραμείνει απλό το σύστημα και προσαρμόσιμο για τα επόμενα εργαστήρια. Επίσης, δεν υπάρχει ακόμα κάποια ένδειξη ότι το σύστημα μπορεί να βρεθεί σε τέτοια κατάσταση, καθώς τα μηνύματα αποσκοπούν σε αλλαγή των 7 segments και δεν έχει νόημα να γίνονται αλλαγές τόσο γρήγορα.

## Πηγές

ATmega16 manual

<http://ww1.microchip.com/downloads/en/devicedoc/doc2466.pdf>

AVR Libc User Manual

<http://savannah.nongnu.org/download/avr-libc/avr-libc-user-manual-2.0.0.pdf.bz2>