

ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ ΚΑΙ ΠΕΡΙΦΕΡΕΙΑΚΑ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ – ΑΣΚΗΣΗ 2

Ονοματεπώνυμο: Παπαγγέλης Άρης Ελευθέριος

ΑΕΜ:8883

Ονοματεπώνυμο: Παπαδάμ Στέφανος

ΑΕΜ:8885

5/6/2019

Το ζητούμενο σε αυτή την εργαστηριακή άσκηση είναι ο έλεγχος των στροφών ενός κινητήρα, τόσο μέσω ενός ποτενσιομέτρου, όσο και μέσω PWM που παράγεται από τον AVR Atmega16, καθώς και η μέτρηση κάποιων παραμέτρων της λειτουργίας του κινητήρα και η εμφάνισή τους στα LEDS.

Η ανάπτυξη του κώδικα πραγματοποιήθηκε στην αναπτυξιακή πλακέτα STK500.

ΣΗΜΕΙΩΣΗ: Ο AVR έχει οριστεί να τρέχει στα 4 MHz, επομένως όλοι οι timers και τα delays είναι υπολογισμένα με βάση αυτό.

Περιγραφή του κώδικα της άσκησης και σχεδιαστικές αποφάσεις

Α. Οδήγηση του κινητήρα μέσω του ποτενσιομέτρου

Ερώτημα Α1: Το ερώτημα δεν περιλαμβάνει συγγραφή κώδικα.

Ερώτημα Α2: Το ερώτημα δεν περιλαμβάνει συγγραφή κώδικα.

Ερώτημα A3: Το ποτενσιόμετρο θα είναι σε συγκεκριμένη θέση για κάθε γύρο εκτέλεσης του προγράμματος. Η ανάλυση του προγράμματος έχει ως εξής:

- I. Αρχικά γίνονται κάποιες αρχικοποιήσεις. Συγκεκριμένα, έχουμε αρχικοποίηση stack pointer, Port B σαν LEDS, Port C σαν πλήκτρα, falling edge interrupts στο external interrupt 0, timer 1 (16 bit) output compare interrupt με τιμή OCR1A = 15625, που είναι το 1 δευτερόλεπτο με τον prescaler 256 που χρησιμοποιήσαμε. Θέτουμε τον timer 1 σε CTC mode, τον μηδενίζουμε και ενεργοποιούμε τα global interrupts.
- II. Με τον πρώτο τρόπο, περιμένουμε μέχρι να περάσει 1 δευτερόλεπτο ώστε να κληθεί το output compare interrupt. Παράλληλα, καλείται ενδιάμεσα τακτικά το external interrupt 0 κάθε φορά που εντοπίζεται falling edge στο PIND2 (προερχόμενο από τον οπτικό κωδικοποιητή), και αυτό που κάνει είναι να αυξάνει τον καταχωρητή X κατά 1 (ο X χρησιμοποιείται ως μετρητής των παλμών).
- III. Μόλις περάσει το 1 δευτερόλεπτο, καλείται το output compare interrupt και απενεργοποιεί τον timer 1, το external interrupt 0 και το output compare interrupt. Σταματάμε να περιμένουμε και εμφανίζονται στα LED το low και το high byte του καταχωρητή X, πατώντας κάθε φορά το πλήκτρο SW7. Δηλαδή, ο συνολικός αριθμός των παλμών σε 1 sec. Διαιρώντας αυτόν τον αριθμό διά 32 (αριθμός ζευγών τομέων OK) θα πάρουμε τον αριθμό RPS του κινητήρα, και πολλαπλασιάζοντας επί 60 τον αριθμό RPM.
- IV. Με τον δεύτερο τρόπο, αφού ξανά αρχικοποιήσουμε όπως στο I. όποιους καταχωρητές χρειάζεται, αυτή τη φορά θέτοντας τον timer 1 σε normal mode και όχι σε CTC λειτουργία, θα περιμένουμε μέχρι να μετρηθούν 4 παλμοί από το external interrupt 0, κάνοντας τον αντίστοιχο έλεγχο στο κυρίως πρόγραμμα. Μόλις αυτό συμβεί, απενεργοποιούμε τα interrupts, βλέπουμε μέχρι πού έχει μετρήσει ο timer 1, και έτσι βρίσκουμε τη διάρκεια 4 παλμών την οποία και εμφανίζουμε στα LEDS πατώντας το SW7 (low και high byte του timer 1). Η διάρκεια αυτή είναι για 4 παλμούς, όμως ο οπτικός κωδικοποιητής έχει 32 ζεύγη τομέων και χρησιμοποιούμε prescaler=256, επομένως μία περιστροφή θα έχει διάρκεια: $32/4 * \text{διάρκεια}_4 \text{ παλμών} * 256 / 4\text{MHz}$. Επομένως, βρίσκουμε τη διάρκεια μιας περιστροφής, και κάνοντας αναγωγή στο 1 s και ύστερα στο 1 min βρίσκουμε τα RPM του κινητήρα.
- V. Ύστερα, θέλουμε να αποθηκεύσουμε την τάση τροφοδοσίας κινητήρα (TTK) για αυτόν το γύρο εκτέλεσης. Αυτό θα το επιτύχουμε με χρήση του A/D converter. Αρχικά, θέτουμε το Port A ως είσοδο του ADC, και συγκεκριμένα χρησιμοποιούμε το PINA0, αφού αφήνουμε τα MUX bits ίσα με 0. Ύστερα, ενεργοποιούμε τον ADC μέσω του ADEN, θέτουμε prescaler = $ck/32$, τάση αναφοράς VCC = 5V, και ενεργοποιούμε το bit ADSC για αρχή της μετατροπής. Κάνουμε rolling μέχρι το ADSC να γίνει ξανά low, οπότε και θα έχει ολοκληρωθεί η μετατροπή. Τέλος, εμφανίζουμε πρώτα τα 8 μικρότερα bits, και με το SW7 εμφανίζουμε ύστερα και τα 2 μεγαλύτερα bits και καθαρίζουμε το flag ADIF γράφοντας το με 1.
- VI. Έχουμε αποθηκεύσει όλες τις τιμές που θέλουμε για τη συγκεκριμένη θέση του ποτενσιόμετρου, οπότε γυρνάμε το ποτενσιόμετρο σε νέα θέση, και πατάμε SW0 για να κάνουμε Reset του προγράμματος. Μετά απο υποδείξεις των βοηθών του εργαστηρίου, δεν

πήραμε 10 μετρήσεις όπως λέει η εκφώνηση, αλλά 5.

- VII. Πέρα από τα 2 interrupts που περιγράφηκαν παραπάνω, ορίστηκαν επίσης οι βοηθητικές ρουτίνες SWITCH7, που περιμένει για το πάτημα του πλήκτρου 7 για την εμφάνιση της επόμενης τιμής, και η ρουτίνα DEBOUNCE, η οποία είναι ένα delay προκειμένου να προλάβει να επανέλθει στην αρχική του θέση το πλήκτρο μετά το πάτημα.
- VIII. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε θέση του ποτενσιόμετρου για την οποία θέλουμε να πάρουμε μέτρηση.

B. Οδήγηση του κινητήρα μέσω του AVR

Ερώτημα B1: Αυτή τη φορά δε χρησιμοποιούμε το ποτενσιόμετρο, αλλά η οδήγηση του κινητήρα γίνεται μέσω του AVR με χρήση PWM. Το μεγαλύτερο μέρος του κώδικα για το ερώτημα B1 είναι κοινό με τον κώδικα του ερωτήματος A3. Συγκεκριμένα, τα II. έως IV. είναι ακριβώς ίδια, οπότε δε θα ξανά αναλυθούν εδώ. Επιπλέον, δε ζητείται αποθήκευση της TTK αυτή τη φορά, οπότε το V. του A3 θα αφαιρεθεί. Θα αναλύσουμε μόνο τα κομμάτια που διαφέρουν, καθώς και όσα προστίθενται:

- I. Αρχικά γίνονται κάποιες αρχικοποιήσεις. Συγκεκριμένα, έχουμε αρχικοποίηση stack pointer, Port A σαν LEDs, Port C σαν πλήκτρα, PORTB3 (OC0) σαν output για το PWM, και θέτουμε το αρχικό duty cycle ίσο με 51 ($255 \cdot 0,2 = 51$), γράφοντας αυτή την τιμή στον OCR0. Χρησιμοποιείται δηλαδή ο timer 0 για το PWM, μιας και δε χρειαζόμαστε μεγάλη ακρίβεια στον έλεγχο της ταχύτητας ενός κινητήρα. Το mode που χρησιμοποιούμε είναι phase correct PWM, non inverting (ώστε όσο μεγαλώνει το duty cycle να αυξάνεται η ταχύτητα) και τίθεται μέσω του TCCR0.
Τέλος, έχουμε falling edge interrupts στο external interrupt 0, timer 1 (16 bit) output compare interrupt με τιμή OCR1A = 15625, που είναι το 1 δευτερόλεπτο με τον prescaler 256 που χρησιμοποιήσαμε. Θέτουμε τον timer 1 σε CTC mode, τον μηδενίζουμε και ενεργοποιούμε τα global interrupts.
- II. Ίδιο με A3.
- III. Ίδιο με A3.
- IV. Ίδιο με A3.
- V. Ύστερα, έχουμε δύο επιλογές, είτε να πατήσουμε το SW0 για αύξηση του duty cycle κατά 20%, είτε το SW1 για μείωση του κατά 20%. (Η εκφώνηση έλεγε σε βήματα του 5%, αλλά όπως και παραπάνω, μετά απο υποδείξεις των βοηθών του εργαστηρίου, πάρθηκαν λιγότερες μετρήσεις) Η αύξηση ή μείωση αυτή μεταφράζεται σε αύξηση ή μείωση του καταχωρητή dc κατά 51, και στην εγγραφή της νέας τιμής ξανά στον OCR0.

- VI. Τέλος, η εκτέλεση του προγράμματος επανέρχεται στο σημείο μετά τις αρχικές αρχικοποιήσεις, όπου παίρνονται ξανά μετρήσεις για το νέο duty cycle αυτή τη φορά.
- VII. Πέρα από τα 2 interrupts που περιγράφηκαν παραπάνω, ορίστηκαν επίσης οι βοηθητικές ρουτίνες SWITCH7, που περιμένει για το πάτημα του πλήκτρου 7 για την εμφάνιση της επόμενης τιμής, και η ρουτίνα DEBOUNCE, η οποία είναι ένα delay προκειμένου να προλάβει να επανέλθει στην αρχική του θέση το πλήκτρο μετά το πάτημα.
- VIII. Η παραπάνω διαδικασία επαναλαμβάνεται για κάθε τιμή του duty cycle για την οποία θέλουμε να πάρουμε μετρήσεις (σε βήματα 20% πάντα).

Ερώτημα B2: Ακολουθείται η ίδια διαδικασία με το ερώτημα B1, απλά αυτή τη φορά γίνεται αυτόματα η αλλαγή του duty cycle και όχι με το πάτημα του SW0 ή SW1. Επιπλέον, εφόσον δεν ζητείται να αποθηκευτούν μετρήσεις σε αυτό το ερώτημα, αποφασίσαμε να το υλοποιήσουμε κατα βήματα 5% όπως λέει η εκφώνηση.

- I. Αρχικά γίνονται κάποιες αρχικοποιήσεις. Συγκεκριμένα, έχουμε αρχικοποίηση stack pointer και PORTB3 (OC0) σαν output για το PWM, και θέτουμε το αρχικό duty cycle ίσο με 52 ($255 * 0,2 = 51 \sim 52$), γράφοντας αυτή την τιμή στον OCR0. Ο λόγος που χρησιμοποιούμε το 52 αντί για το 51, είναι διότι βολέυει περισσότερο για τις αυξήσεις και μειώσεις. Πρακτικά δεν έχει κάποια διαφορά.
- Χρησιμοποιείται δηλαδή ο timer 0 για το PWM, μιας και δε χρειαζόμαστε μεγάλη ακρίβεια στον έλεγχο της ταχύτητας ενός κινητήρα. Το mode που χρησιμοποιούμε είναι phase correct PWM, non inverting (ώστε όσο μεγαλώνει το duty cycle να αυξάνεται η ταχύτητα) και τίθεται μέσω του TCCR0. Δεν έχουμε χρήση interrupts ή άλλων timers αυτή τη φορά.
- Επιπλέον, ο καταχωρητής flag έχει φορτωθεί με 0. (θα καταστεί σαφές γιατί παρακάτω)
- II. Ελέγχουμε για το αν το flag είναι 0. Αν είναι, σημαίνει πως βρισκόμαστε σε φάση αύξησης του duty cycle, οπότε κάνουμε jump στον κώδικα αύξησης του duty cycle κατα 5% (= 13). Ύστερα ελέγχουμε αν έχει γίνει ήδη αύξηση 10 φορές, οπότε το duty cycle θα ήταν 182. Αν δεν έχει γίνει 182 ακόμα, τότε τερματίζουμε γράφοντας το νέο duty cycle στον OCR0, κάνουμε ένα delay 400 msec (δείτε στην ενότητα των προβλημάτων που αντιμετωπίσαμε γιατί το κάνουμε αυτό) και η ροή του προγράμματος επιστρέφει ξανά στο σημείο αμέσως μετά τις αρχικές αρχικοποιήσεις, ώστε να γίνει το επόμενο βήμα μεταβολής του duty cycle. Αν το duty cycle έχει γίνει 182, τότε σημαίνει πως έχει ήδη γίνει αύξηση 10 φορές, οπότε εισάγουμε το delay 10 sec που αναφέρεται στην εκφώνηση, και ύστερα κάνουμε τον καταχωρητή flag ίσο με 1, για να μεταβούμε στη φάση μείωσης του duty cycle.
- III. Αν στον παραπάνω έλεγχο για το flag, το flag είναι ίσο με 1, τότε σημαίνει πως βρισκόμαστε σε φάση μείωσης του duty cycle, οπότε κάνουμε jump στον κώδικα μείωσης του duty cycle κατα 5% (= 13). Ύστερα, ελέγχουμε αν έχει γίνει ήδη μείωση 10 φορές, οπότε το duty cycle θα ήταν ξανά 52 (αρχική τιμή). Αν δεν έχει γίνει 52 ακόμη, τότε τερματίζουμε γράφοντας το νέο duty cycle στον OCR0, κάνουμε ένα delay 400 msec (δείτε στην ενότητα των προβλημάτων που αντιμετωπίσαμε γιατί το κάνουμε αυτό) και η ροή του προγράμματος επιστρέφει ξανά στο σημείο αμέσως

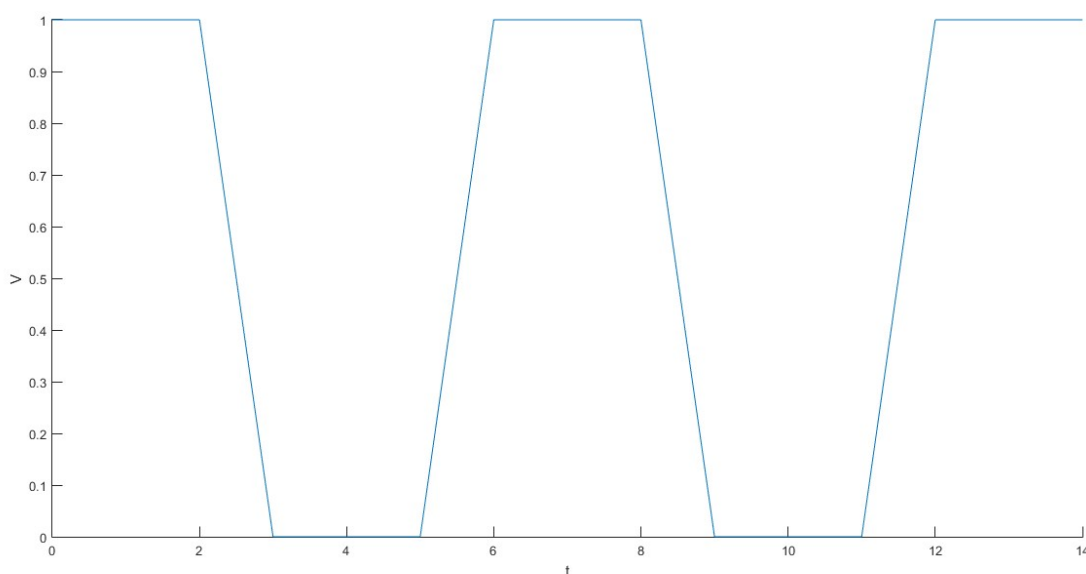
μετά τις αρχικές αρχικοποιήσεις, ώστε να γίνει το επόμενο βήμα μεταβολής του duty cycle. Αν το duty cycle έχει γίνει 52, τότε πριν γράψουμε το νέο duty cycle, θέτουμε τον καταχωρητή flag ίσο με 0 ώστε να μεταβούμε πάλι σε φάση αύξησης του duty cycle. Επομένως, στην επόμενη εκτέλεση θα εκτελεστεί πάλι η ροή II.

IV. Τα II. και IV. θα εναλλάσσονται διαρκώς, καθ' όλη τη διάρκεια εκτέλεσης του προγράμματος.

Απαντήσεις στα ερωτήματα της άσκησης

Ερώτημα A1

Το γράφημα για όλες τις μετρήσεις είχε την εξής μορφή και ανάλογα μεταβαλλόταν ο χρόνος t:

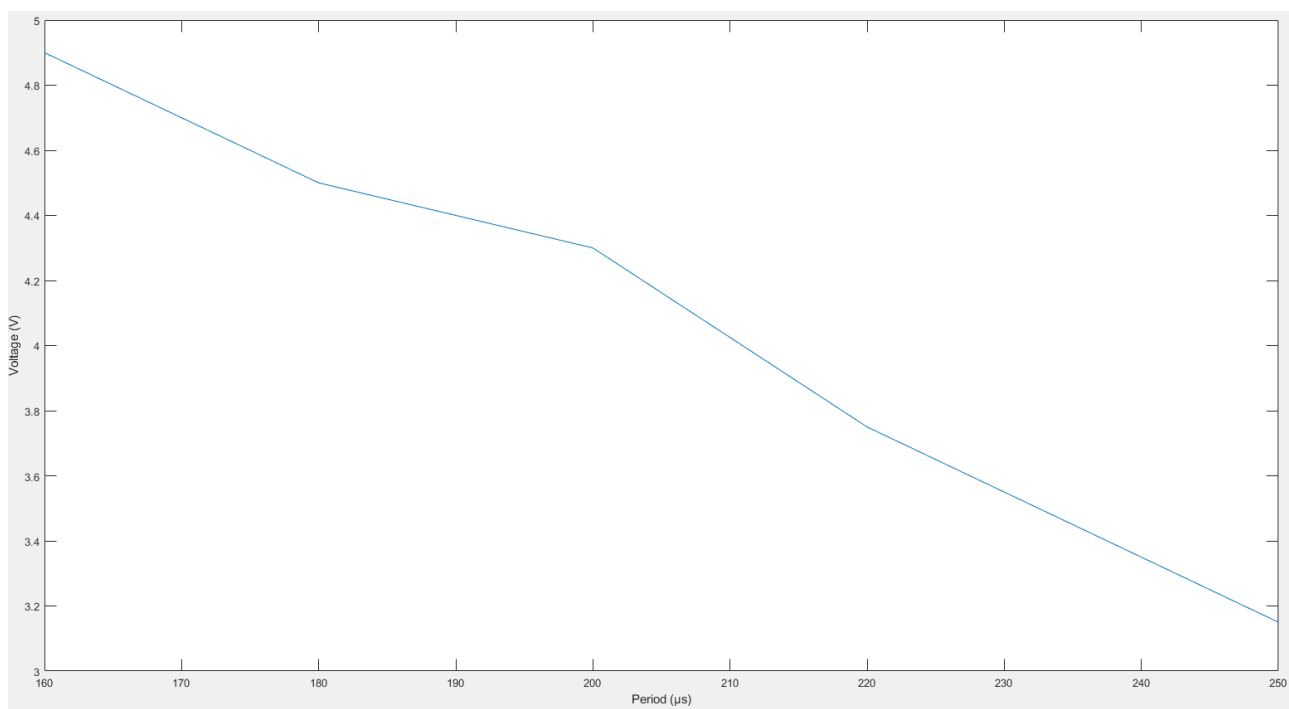


Ερώτημα A2

	V (Τάση σε Volt)	T(μισή περίοδος σε μs)	T(περίοδος σε μs)	Ταχύτητα (RPM)
1η μέτρηση	3,15	125	250	7500
2η μέτρηση	3,75	110	220	8522
3η μέτρηση	4,3	100	200	9375
4η μέτρηση	4,5	90	180	10416
5η μέτρηση	4,9	80	160	11718

Η ταχύτητα σε RPM θα είναι $60 / (32 * T * 10^{(-6)})$.

Η γραφική παράσταση της περιόδου σε σχέση με την τάση τροφοδοσίας φαίνεται παρακάτω:



Ερώτημα Α3

	Παλμοί/sec	Ταχύτητα (RPM) 1ος τρόπος	Διάρκεια 4 παλμών (prescaler=256)	Ταχύτητα (RPM) 2ος τρόπος	ΤΤΚ
1η μέτρηση	2539	4760	43	2725	3,44
2η μέτρηση	2726	5111	39	3004	3,82
3η μέτρηση	2572	4822	30	3906	4,44
4η μέτρηση	2721	5101	21	5580	4,41
5η μέτρηση	3127	5863	9	13020	4,99

Παρατηρούμε πως, ειδικά για τον 2ο τρόπο, τα αποτελέσματα που αφορούν την ταχύτητα του κινητήρα (το τελευταίο αποτέλεσμα ειδικά) δεν είναι και τόσο λογικά. Ίσως ο τύπος που χρησιμοποιήσαμε (αναφέρεται παραπάνω, στο ερώτημα Α3 IV.) να μην είναι σωστός, ή πολύ πιθανόν να μην σημειώσαμε σωστά τις ενδείξεις των LED λόγω της πίεσης χρόνου. Παρόλα αυτά, έχουμε την πεποίθηση πως ο κώδικας μας είναι σωστός, μιας και οι τιμές και των 2 τρόπων δεν διαφέρουν σε υπερβολικό βαθμό μεταξύ τους (με εξαίρεση την τελευταία τιμή όπως προαναφέραμε).

Και οι 2 τρόποι έχουν εντελώς διαφορετικά αποτελέσματα σε σχέση με το ερώτημα Α2, αλλά για αυτό ευθύνεται και το γεγονός πως τα δύο ερωτήματα έγιναν σε διαφορετικές μέρες και με διαφορετικές θέσεις του ποτενσιομέτρου, οπότε εκ των πραγμάτων θα ήταν διαφορετικό το αποτέλεσμα.

Πάντως, το αποτέλεσμα του ερωτήματος Α2, που εξήχθει με χρήση παλμογράφου, μπορεί να θεωρηθεί πιο ακριβές.

Ερώτημα B1

Τα αποτελέσματα σε αυτό το ερώτημα έβγαιναν παρόμοια με του ερωτήματος A3 (τα λάθη που υπάρχουν εκεί υπάρχουν και εδώ δυστυχώς).

Η μόνη διαφορά σε αυτή την περίπτωση είναι πως η οδήγηση του κινητήρα, αντί να γίνεται μέσω ποντεσιόμετρου, γίνεται μέσω παραγωγής παλμού PWM από τον AVR.

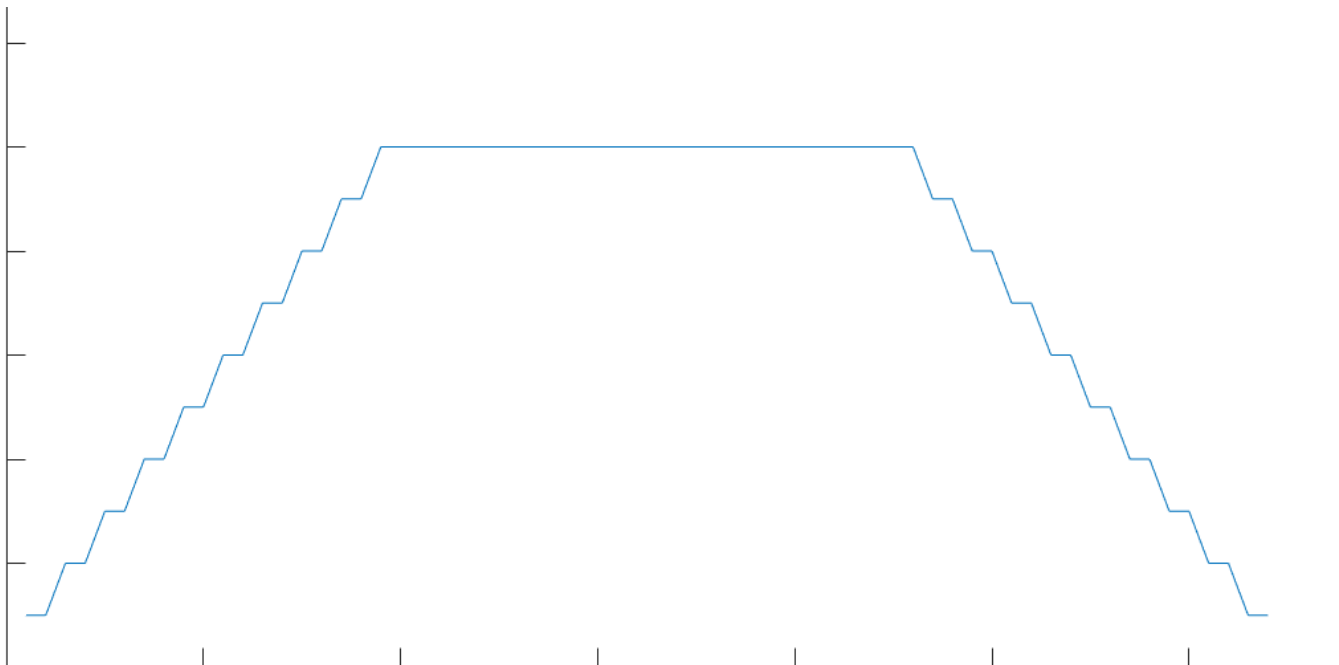
Για τον λόγο αυτό, κρίθηκε σκόπιμο να μην παρουσιαστεί η γραφική παράσταση duty cycle και RPM, διότι θα είναι λανθασμένη.

Το σωστό αποτέλεσμα θα ήταν παρόμοιο με την γραφική παράσταση του ερωτήματος A2.

Ερώτημα B2

Στο ερώτημα αυτό, παρατηρήσαμε πως ο κινητήρας αύξανε σταδιακά τις στροφές του, παρέμενε σε μεγάλη σχετικά ταχύτητα για 10 δευτερόλεπτα, και μετά η ταχύτητά του μειωνόταν ξανά. Αυτή η διαδικασία επαναλαμβανόταν.

Δηλαδή, η ταχύτητα του κινητήρα σχημάτιζε μια τραπεζοειδή κυματομορφή παρόμοια με την εξής:



(Το παραπάνω σχήμα δεν είναι ακριβές, είναι μία αναπαράσταση)

Όσον αφορά τη σύνδεση του PWM με LED, δυστυχώς αυτό το κομμάτι δεν προλάβουμε να το υλοποιήσουμε στο εργαστήριο.

Παρόλα αυτά, εικάζουμε πως θα συμβεί ό,τι και με τον κινητήρα, δηλαδή θα αυξάνεται σταδιακά η φωτεινότητα του LED έως ότου φτάσει σε μια μέγιστη τιμή, θα παραμείνει σε αυτή την τιμή για 10 δευτερόλεπτα, και ύστερα θα αρχίσει να μειώνεται ξανά η φωτεινότητα του μέχρι το αρχικό επίπεδο. Η παραπάνω διαδικασία θα επαναλαμβάνεται, όπως και με τον κινητήρα.

Προβλήματα που αντιμετωπίστηκαν

Τα προβλήματα που αντιμετωπίστηκαν αφορούσαν κυρίως την κατανόηση της εκφώνησης.

Συγκεκριμένα, στο ερώτημα B2 δε μπορούσαμε να καταλάβουμε τι εννοεί λέγοντας: “Παρατηρώντας την απόκριση του κινητήρα δοκιμάστε για πόσες περιόδους θα πρέπει να είναι σταθερό το duty cycle στην κάθε αύξηση και μείωση της ταχύτητας”. Επομένως, βάλαμε ένα delay 400 msec ανάμεσα σε κάθε αλλαγή του duty cycle, δηλαδή το κάθε duty cycle θα παραμένει ίδιο για 0,4 δευτερόλεπτα.

Άλλα προβλήματα που αντιμετωπίστηκαν αφορούσαν τον κατάλληλο ορισμό των Ports.

Συγκεκριμένα, δε μας είχε καταστεί αρκετά σαφές πως το Port C χρησιμοποιείται από το JTAG και έτσι έπρεπε να το χρησιμοποιήσουμε μόνο σαν switches. Στην αρχή είχαμε ορίσει τα LEDS στο Port C, οπότε έπρεπε να αλλάξουμε τις συνδέσεις στην πλακέτα και τον κώδικά μας για να τρέξει επιτυχώς το πρόγραμμα.

Στο ερώτημα A3 με τον ADC, αρχικά νομίζαμε πως θα δοθεί εξωτερική τάση αναφοράς στο pin AREF, οπότε δε χρησιμοποιούσαμε τον καταχωρητή ADMUX. Μετά μας είπαν οι βοηθοί πως ως τάση αναφοράς θα βάλουμε το VCC του AVR, δηλαδή 5 V. Επομένως, ορίσαμε μετά από το bit REFS0 του ADMUX ίσο με 1.

Κατά τα άλλα, ο κώδικας της άσκησης 2 δεν περιλάμβανε την υλοποίηση σύνθετης λογικής όπως οι άλλες δύο ασκήσεις, επομένως η ανάπτυξη των προγραμμάτων αυτών καθαυτών ήταν σχετικά απροβλημάτιστη.

Κώδικας

Ο κώδικας επισυνάπτεται στο αρχείο .zip που βρίσκεται και η αναφορά. Συγκεκριμένα, για το κάθε ερώτημα (A3, B1 και B2) υπάρχει ξεχωριστό project AVR Studio. Επομένως, για την εκτέλεση των προγραμμάτων το μόνο που έχει να κάνει κάποιος είναι να ανοίξει το αντίστοιχο project στο AVR Studio μέσω του αρχείου .aps που βρίσκεται στον κάθε φάκελο.

Δεν υπάρχει compilation πολλών αρχείων για το κάθε ερώτημα, επομένως κρίνεται μή – απαραίτητη η προσθήκη Readme αρχείου.