

Μάθημα

Εργαστηριακή Άσκηση 3

Λουδάρος Ιωάννης (1067400) - Χριστίνα Κρατημένου (1067495)



Μπορείτε να δείτε την τελευταία έκδοση του Project [εδώ](#) ή σκανάροντας τον κωδικό QR που βρίσκεται στην επικεφαλίδα.

Περιγραφή Αναφοράς

Παρακάτω παραθέτουμε τις απαντήσεις μας στην “Εργαστηριακή Άσκηση 3” του μαθήματος “Προηγμένοι Μικροεπεξεργαστές” καθώς και σχόλια τα οποία προέκυψαν κατά την εκπόνηση της.

Σας ενημερώνουμε ότι λόγω θεμάτων συμβατότητας (Το Microchip Studio τρέχει μόνο σε intel επεξεργαστές), δεν ήμασταν σε θέση να ελέγξουμε κατά πόσο ο κώδικας μας είναι όντως λειτουργικός.

Περιεχόμενα

Σχεδιασμός	2
Ευθεία Κίνηση	2
Αριστερή Κίνηση	3
Δεξιά Κίνηση	3
Επιπλέον Σημειώσεις	3
Ζητούμενο 1.....	4
Ζητούμενο 2.....	5
Ζητούμενο 3.....	5

Απαντήσεις

Σχεδιασμός

Διαβάζοντας την εκφώνηση, αντιλαμβανόμαστε ότι πρέπει να υλοποιήσουμε τα παρακάτω:

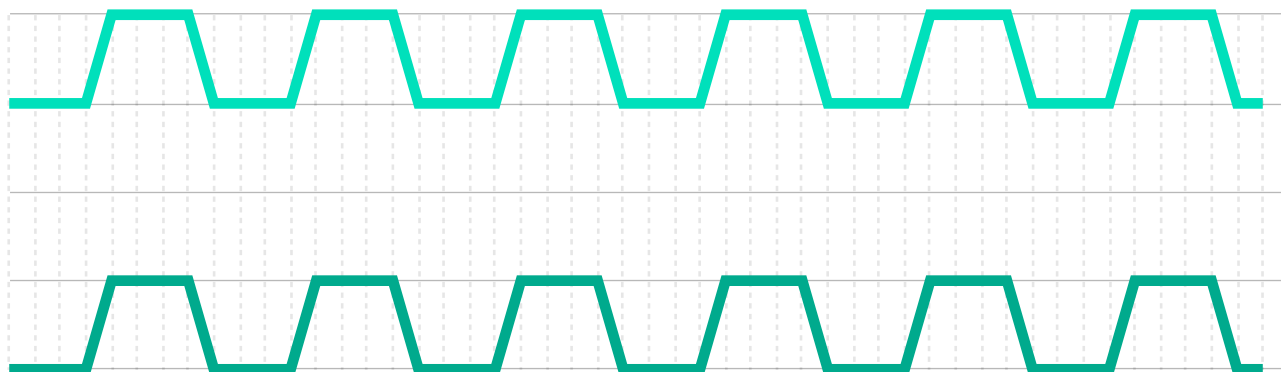
Οι κυματομορφές που δίνονται σε αυτό το βήμα, **δίνονται μόνο για να φανεί ξεκάθαρα η σχέση των μεταξύ τους συχνοτήτων**. Δεν αποτελούν ακριβή αναπαράσταση της εξόδου των Pins του PORTF που θα χρησιμοποιήσουμε.

Ευθεία Κίνηση

Η συσκευή κινείται ευθεία, συνεπώς οι δύο τροχοί κινούνται με την ίδια ταχύτητα. Έτσι τα LED αναβοσβήνουν στον ίδιο ρυθμό

— Αριστερή Ρόδα

— Δεξιά ρόδα



Σημείωση: Υπάρχει λάθος στην εκφώνηση. Όταν στρίβει ένα όχημα με 4 τροχούς, η ρόδες που είναι στην εσωτερική πλευρά της στροφής, κινούνται πιο αργά από τις ρόδες που είναι στην εξωτερική της πλευρά. Αυτό συμβαίνει, γιατί οι ρόδες στην εξωτερική πλευρά καλούνται να διανύσουν μεγαλύτερη απόσταση, από ότι αυτές στην εσωτερική πλευρά.

Συγκεκριμένα, για αυτό το παράδειγμα μάλιστα, για να καταφέρει το όχημα να κάνει στροφή 90°, χωρίς να χτυπήσει στον τοίχο, όπως απαιτείται, ο τροχός της εσωτερικής πλευράς, θα έπρεπε να μείνει τελείως ακίνητος.

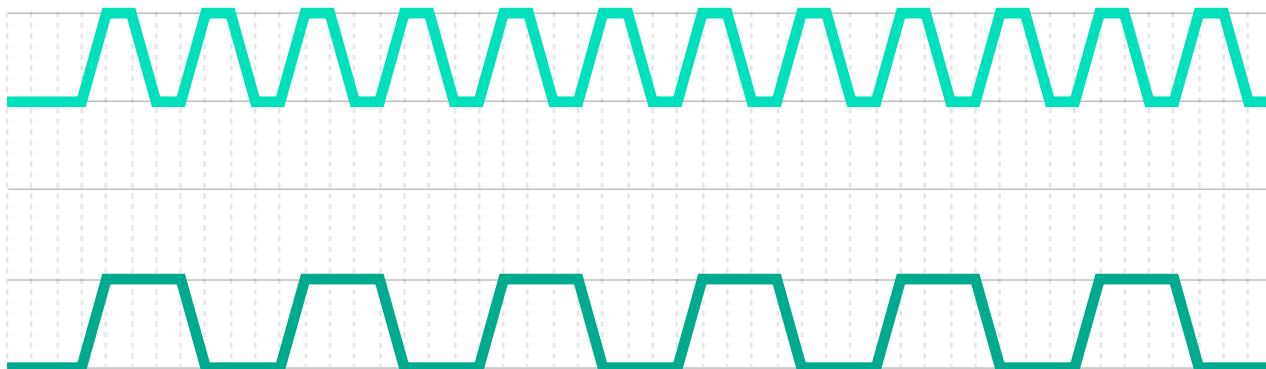
Οι παρακάτω απαντήσεις, παρόλα αυτά, θα δοθούν, όπως ζητούνται από την εκφώνηση. Τα παραπάνω σχόλια δίνονται απλά για να διορθωθεί μελλοντικά το Φυλλάδιο του Εργαστηρίου. Γράφονται σε φιλικό τόνο, και σε καμία περίπτωση δεν γράφτηκαν για να σας επιτεθούμε, (ούτε για να μας μειώσετε τον βαθμό) (πλακίτσα).

Αριστερή Κίνηση

Για να πραγματοποιηθεί αριστερή στροφή, χρειάζεται, σύμφωνα με την εκφώνηση, ο αριστερός τροχός να κινηθεί με τον διπλάσιο ρυθμό από ότι ο δεξιός. Έτσι έχουμε έναν παλμό της μορφής:

— Αριστερή Ρόδα

— Δεξιά ρόδα

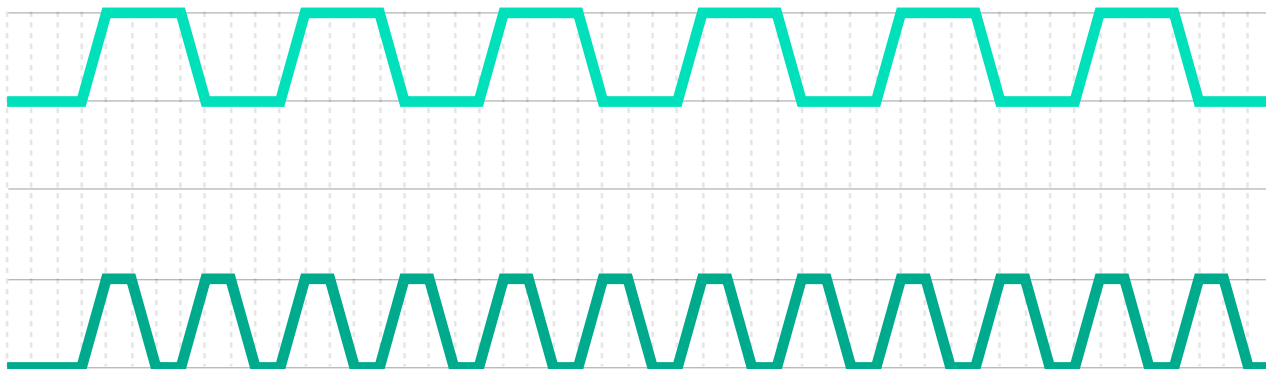


Δεξιά Κίνηση

Για να πραγματοποιηθεί δεξιά στροφή, χρειάζεται, σύμφωνα με την εκφώνηση πάλι, ο δεξιός τροχός να κινηθεί με τον διπλάσιο ρυθμό από ότι ο αριστερός. Έτσι έχουμε έναν παλμό της μορφής:

— Αριστερή Ρόδα

— Δεξιά ρόδα



Επιπλέον Σημειώσεις

Για την καλύτερη αναγνωσιμότητα του κώδικα επιλέξαμε να δημιουργήσουμε ένα header file το οποίο περιέχει όλες τις συναρτήσεις, τις global σταθερές, καθώς και τα include statements που χρησιμοποιούμε. Μπορείτε να το βρείτε στον ίδιο κατάλογο με το παρόν έγγραφο, ως "iheader.h".

Ζητούμενο 1

Όπως μας γίνεται ξεκάθαρο, το πρώτο ζητούμενο απαιτεί από εμάς να σετάρουμε τον TCA σε Split Mode και το αντιμετωπίζουμε σαν δύο ξεχωριστούς 8μπιτους timers. Καλούμε μια διακοπή κάθε φορά που ο timer με τα MSBs φτάνει στην θεμιτή τιμή (εμείς έχουμε θέσει να γίνονται διακοπές κάθε 8 κύκλους). Έπειτα, μέσα στην διακοπή ελέγχουμε τις καταστάσεις και των δύο timers και αποφασίζουμε ανάλογα αν είναι στιγμή να ανάψουμε τα αντίστοιχα led.

```
1#include "iheader.h"
2
3
4
5int main() {
6
7    initialise_pins();
8    initialise_TCA();
9    set_TCA();
10   start_TCA();
11
12   sei();
13
14   while(1) {
15   }
16}
17
```

```
1ISR(TCA0_HUNF_vect) {
2
3    int intflags = TCA0.SPLIT.INTFLAGS;
4    TCA0.SPLIT.INTFLAGS = intflags;
5
6    // hcount interrupt
7    if ((TCA0.SPLIT.INTFLAGS & 00000010)==1)
8    {
9        hcount++;
10       if (hcount!=2) {}
11       else {
12           PORTD.OUTTGL = 0x01;
13           hcount = 0;
14       }
15   }
16
17   // lcount interrupt
18   else if ((TCA0.SPLIT.INTFLAGS & 00000001)==1)
19   {
20       lcount++;
21       if (lcount!=2) {}
22       else {
23           PORTD.OUTTGL = 0x02;
24           lcount = 0;
25       }
26   }
27};
```

Ζητούμενο 2

Το μόνο που χρειάζεται να προσθέσουμε στον κώδικα μας είναι τις ρυθμίσεις του ADC και να αλλάξουμε την συνθήκη του ατέρμονος βρόγχου.

```
1#include "iheader.h"
2
3int main() {
4
5    initialise_pins();
6    initialise_TCA();
7    set_TCA();
8    start_TCA();
9    initialise_ADC();
10   free_running();
11   start_ADC();
12
13   sei();
14
15   while(wall==0) {
16   }
17}
18
```

```
1ISR(ADC0_WCOMP_vect)
2{
3
4    int intflags = ADC0.INTFLAGS;
5    ADC0.INTFLAGS = intflags;
6    wall = 1;
7    disable_TCA();
8    PORTD.OUT = 0x04; // only PIN2 LED on
9};
10
```

Ζητούμενο 3

Στο ζητούμενο 3 χρειάστηκε επιπλέον να σετάρουμε τα κατάλληλα switches, όπως επίσης και να προσθέσουμε κάποιους βρόγχους (στην λογική της προηγούμενης άσκησης, που ελέγχουν κάθε φορά το τρέχον είδος κίνησης).

```
1#include "iheader.h"
2
3int main() {
4
5    initialise_pins();
6    initialise_TCA();
7    set_TCA();
8    start_TCA();
9    initialise_ADC();
10   free_running();
11   start_ADC();
12   initialise_switches();
13
14   sei();
15
```

```
15
16   while(1) {
17
18       // straight movement
19       while(wall==0) {
20       }
21
22       // right movement
23       if (turning_right==1) {
24           right_movement();
25           while (turning_right==1) { /*re-enable SW5*/ }
26       }
27
28       // left movement
29       if (turning_left==1) {
30           left_movement();
31           while (turning_left==1) { /*re-enable SW6*/ }
32       }
33
34       initialise_TCA();
35       set_TCA();
36       start_TCA();
37
38   }
39
40   cli();
41
42   return 0;
43}
```