



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Εργαστήριο Μικροϋπολογιστών
5^η Εργαστηριακή Άσκηση

Ομάδα: 64
Μέλη: Κωνσταντίνος Ιωάννου (ΑΜ: 03119840)
Γιώργος Μυτιληναίος (ΑΜ: 03119841)

Ημερομηνία εργαστηριακής εξέτασης: Τετάρτη 30 Νοεμβρίου 2022
Ημερομηνία υποβολής: Κυριακή 4 Δεκεμβρίου 2022

Ζήτημα 5.1

Αρχικά τονίζουμε ότι εδώ δεν θα δείξουμε τον κώδικα που μας δίνεται έτοιμος στις διαφάνειες για την αρχικοποίηση αλλά και την χρήση (read/write) του ολοκληρωμένου PCA955 .(Στο .zip υπάρχουν ολόκληροι οι κώδικες)

```
186 int main(void) {
187     unsigned char F0,F1,temp1,A,B,C,D;
188
189     twi_init();
190     DDRE=0xF0;
191     PCA9555_0_write(REG_CONFIGURATION_0, 0x00); //Set EXT_PORT0 as output
192     while(1)
193     {
194         temp1=PINB;
195         temp1=~temp1;
196         temp1=temp1&0xF //00001111
197         A=temp1&0x01;
198         B=temp1&0x02;
199         C=temp1&0x04;
200         D=temp1&0x08;
201         B=B>>1;
202         C=C>>2;
203         D=D>>3;
204         F1=(A&C)&(B|D);
205         F0=(~A&B)|(~B&C&D);
206         F0=~F0;
207         F0=F0&0x01;
208         F1=F1<<1;
209         F0=F0|F1;
210         F0=F0&0x03;
211         PCA9555_0_write(REG_OUTPUT_0, F0);
212     }
```

Οπότε ο μόνος κώδικας που γράφουμε εμείς αποτελεί την main(). Αρχικοποιούμε με την συνάρτηση **twi_init()** , και θέτουμε την θύρα B ως είσοδο .Με την εντολή **PCA9555_0_write(REG_CONFIGURATION_0, 0x00)** Θέτουμε τις θύρες IO0_0 έως IO0_7 ως έξοδο για να μπορούμε να πάρουμε το αποτέλεσμα μας από αυτές και μέσω βραχυκύκλωσης να το οδηγήσουμε στο PORTD. Στην συνέχεια παίρνουμε την τιμή από την είσοδο(PINB) την αναστρέφουμε λόγο της ανάστροφής λογικής της πλακέτας και απομονώνουμε τα τέσσερα MSB που μας ενδιαφέρουν. Με την ίδια λογική απομονώνουμε στις μεταβλητές A,B,C,D Τα MSB και προφανώς τα μεταφέρουμε με ολίσθηση στη σωστή θέση ώστε να έχουμε είτε λογικό 0 είτε λογικό 1. Τέλος αφού υπολογίσουμε τις τιμές F0 και F1 βάζουμε την F0 στο πρώτο MSB και την F1 στο δεύτερο MSB τα περνάμε και τα δύο στη F0 και μέσω της εντολής **PCA9555_0_write(REG_OUTPUT, F0)** εμφανίζουμε την τιμή του F0 στις θύρες IO0.

Ζήτημα 5.2

```
169 int main(void){
170
171     unsigned char x;
172     twi_init();
173     PCA9555_0_write(REG_CONFIGURATION_1, 0xF0); //11110000 θέτω IO1_0~IO1_3 output and IO1_4~IO1_7 input
174     PCA9555_0_write(REG_CONFIGURATION_0, 0x00); //IO0 as output
175     PCA9555_0_write(REG_OUTPUT_1, 0x0E); // 1110 θέτω το IO1_0 ως μηδεν για να το ενεργοποιήσω (δηλαδή ενεργοποιεί την πρώτη γραμμή)
176     while(1){
177         x=PCA9555_0_read(REG_INPUT_1); //Διαβάζω το IO1
178         x=x&0xF0; //x&11110000 => απομονώνω τα 4 MSB
179         x=x>>4; // τα ολιγοθένα δεξιά ώστε να γίνουν LSB
180         PCA9555_0_write(REG_OUTPUT_0, ~x); // τα βγάζω στην έξοδο IO0 ( ως 0 λόγω αναστροφής λογικής)
181
182
183     }
184 }
```

Όπως και πριν δεν σχολιάζουμε τον κώδικα που μας δόθηκε στις διαφάνειες του μαθήματος. Στα σχόλια στον κώδικα εξηγούμε τι κάνουμε με την κάθε εντολή οπότε εδώ θα δώσουμε μια πιο γενική ιδέα της περιγραφής. Θέτουμε ως έξοδο την IO_0 και τα 4 LSB της IO1 ενώ σαν είσοδο θέτουμε τα 4 MSB της IO1. Επιπλέον με χρήση της συνάρτησης write(REG_OUTPUT_1,...) ενεργοποιούμε μόνο το IO1_0 από την έξοδο ,δηλαδή δίνουμε τάση μόνο στην πρώτη γραμμή ώστε αν πατήσουμε κάποιο μπουτόν εκτός της πρώτης γραμμής να μην δώσουμε είσοδο .Συνεπώς ανάλογα με το ποια μπουτόν πιέζουμε από την πρώτη γραμμή δίνουμε κατάλληλη είσοδο ένα binary που φαίνεται στα bit IO1_4 έως IO1_7 οπότε απομονώνουμε αυτά τα τέσσερα MSB με ολίσθηση τα κάνουμε LSB και τα μεταφέρουμε ως έξοδο του IO0. Σημειώνουμε ότι όσα μπουτό πιέζουμε τόσα led ανοίγουν ταυτόχρονα στην PORTD.