



ΜΑΘΗΜΑ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΎΠΟΛΟΓΙΣΤΩΝ (ΡΟΗ Υ)

ΕΡΓΑΣΙΑ

5^Η ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ

ΣΠΟΥΔΑΣΤΕΣ

ΚΑΡΑΜ ΚΩΝΣΤΑΝΤΙΝΟΣ

ΚΟΛΙΟΣ ΑΠΟΣΤΟΛΟΣ

2022-23

Οι πλήρεις κώδικες όλων των ασκήσεων περιέχονται στο *zipped* αρχείο.

Ζήτημα 5.1 (Lab5_1.c)

Το κομμάτι κώδικα που προστέθηκε φαίνεται παρακάτω:

```
uint8_t a, b, c, d, a_xor, b_xor, f0_xor, f0, f1 ,out_final;

int main(void)
{
    twi_init();
    DDRB = 0x00; // Initialize PORTB
    as input
    PCA9555_0_write(REG_CONFIGURATION_0, 0x00); // Set EXT_PORT0 as
    output

    while (1)
    {
        uint8_t inp = PINB;
        inp = ~inp;

        a = inp & 0x01;
        b = (inp & 0x02) >> 1;
        c = (inp & 0x04) >> 2;
        d = (inp & 0x08) >> 3;
        a_xor = a^0x01;
        b_xor = b^0x01;
        // F0 = (A'B + B'CD)'
        f0 = ((a_xor & b) | (b_xor & (c & d)));
    }
}
```

```

    f0_xor = f0^0x01;
    // F1 = (AC)(B+D)
    f1 = ((a & c) & (b | d));

    f1 = f1 << 1;
    out_final = f0_xor + f1;
    PCA9555_0_write(REG_OUTPUT_0, out_final);

}
}

```

Θα εξηγήσουμε την λειτουργία του προγράμματος. Αφού αντιστρέψουμε την είσοδο από το PINB ώστε το πατημένο κουμπί να αντιστοιχεί στο λογικό 1, διαβάζουμε ένα ένα τα πρώτα τέσσερα bits, και τα τοποθετούμε στο LSB των μεταβλητών a, b, c και d. Όπου χρειάζεται αντιστροφή μεταβλητής γίνεται μέσω της πράξης XOR 1, ώστε να επηρεάζεται μόνο το LSB των μεταβλητών και να μην αλλοιώνεται το αποτέλεσμα. Αφού υπολογίσουμε τις συναρτήσεις f0 και f1, μετακινούμε μία θέση αριστερά την f1, τις προσθέτουμε λογικά και αντιγράφουμε το τελικό αποτέλεσμα στην μεταβλητή out_final. Τέλος, το αποτέλεσμα εμφανίζεται στα δύο πρώτα led της PORTD μέσω της κλήσης της συνάρτησης *PCA9555_0_write*, με ορίσματα το REG_OUTPUT_0 για να εμφανιστεί στο αποτέλεσμα στους ακροδέκτες IO0_0 και IO0_1 του κονέκτορα και το out_final που είναι το αποτέλεσμα των λογικών πράξεων.

Ζήτημα 5.2 (Lab5_2.c)

Το κομμάτι κώδικα που προστέθηκε φαίνεται παρακάτω:

```

int main(void)
{
    twi_init();
    DDRB = 0x00;    // Initialize PORTB as input
    PCA9555_0_write(REG_CONFIGURATION_0, 0x00);    // Set EXT_PORT0 as
output
    PCA9555_0_write(REG_CONFIGURATION_1, 0xFE);    // Set IO1_0 as
output and others as input

    while(1)
    {
        PCA9555_0_write(REG_OUTPUT_1, 0xFE);
        PCA9555_0_write(REG_OUTPUT_0, 0x00);    // Turn off LEDs
        uint8_t inp = PCA9555_0_read(REG_INPUT_1);

        if (inp == 0b11101110)

```

```

        PCA9555_0_write(REG_OUTPUT_0, 0x01);

    if (inp == 0b11011110)
        PCA9555_0_write(REG_OUTPUT_0, 0x02);

    if (inp == 0b10111110)
        PCA9555_0_write(REG_OUTPUT_0, 0x04);

    if (inp == 0b01111110)
        PCA9555_0_write(REG_OUTPUT_0, 0x08);
}
}

```

Θα εξηγήσουμε την λειτουργία του προγράμματος. Αρχικά θέτουμε όπως ζητάει εκφώνηση τον ακροδέκτη IO1_0 της θύρας επέκτασης 1 ως έξοδο και τους υπόλοιπους ως εισόδους. Έπειτα διαβάζουμε την είσοδο από το keypad μέσω της εντολής *PCA9555_0_read*, η οποία έχει ως όρισμα το REG_INPUT_1, αφού θέλουμε να διαβάσουμε από την θύρα επέκτασης 1. Επειδή έχουμε θέσει το IO1_0 ως έξοδο, είναι πάντα μηδέν και άρα ελέγχουμε κάθε φορά το bit που αντιστοιχεί στην στήλη που βρίσκεται το αντίστοιχο κουμπί, από IO1_4 (για το “*”) έως IO1_7 (για το “D”) και εμφανίζουμε την αντίστοιχη έξοδο μέσω της εντολής *PCA9555_0_write*.