**Εργαστήριο Προηγμένοι Μικροεπεξεργαστές**

Εργαστηριακή Άσκηση 01:

Ανελκυστήρας

2024-25

**Περιεχόμενα**

[Σχεδιασμός / Διάγραμμα Ροής 3](#__RefHeading___Toc30_1975538019)

[Περιγραφή Κώδικα 4](#__RefHeading___Toc30_1975538019_Copy_1)

[Main: 4](#__RefHeading___Toc666_1975538019)

[ISR: 5](#__RefHeading___Toc666_1975538019_Copy_1)

[Κώδικας 6](#__RefHeading___Toc666_1975538019_Copy_1_)

# Σχεδιασμός / Διάγραμμα Ροής

*Το διάγραμμα δημιουργήθηκε μεσω mermaid.js*

# Περιγραφή Κώδικα

Ολόκληρος ο κώδικας ειναι διαθέσιμος στο τέλος της αναφοράς.

## Main:

Αρχικοποιείται το σύστημα θέτωντας τα bits 0,1,2 του PORTD σαν έξοδο (εφόσον αυτά τα LED θα χρησιμοποιήσουμε) και τα θέτουμε 1 χωρίς να επηρεάζουμε τα υπόλοπα bit (bitwise OR)

PORTD.DIR |= 0b00000111; //BITS 0-2 OUTPUT (0 = ERR)

PORTD.OUT |= 0b00000111; //LED's are off

Θέτουμε τα PINCTRL REGISTERS των pin 5,6 του PORTF ως εξής: Default = HIGH και το interrupt καλείται οταν πάει LOW → HIGH (όταν αφήνει το κουμπί ο χρήστης).

PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_RISING\_gc;

PORTF.PIN6CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_RISING\_gc;

Ενεργοποιούμε τα interrupts και μπαίνουμε σε έναν ατέρμονο βρόγχο while, περιμένοντας interrupt.

sei(); //enable interrupts

while (1) {

}

## ISR:

Αρχικά αποθηκεύουμε τα INTFLAGS του PORTF στην μεταβλητή y.

int y = PORTF.INTFLAGS;

Εάν εχουν πατηθεί και τα δύο κουμπία, καθαρίζουμε το bit 0 του PORTD.OUT (με OUTCLR για γρήγορο και ασφαλή καθαρισμό), ενεργοποιώντας τα active low LED. Μετά απο ενα delay 10ms και μέσω bitwise OR απενεργοποιούμε το LED 0.

// Both buttons pressed (error)

if(y == 0b01100000) {

PORTD.OUTCLR = 0b00000001;

\_delay\_ms(10);

PORTD.OUT |= 0b00000001;

}

Εάν έχει πατηθεί το “κάτω” κουμπί (PF5), και δεν βρισκόμαστε ήδη στο ισόγειο, ανάλογα σε ποιόν όροφο βρισκόμαστε σβήνουμε το αντίστοιχο LED.

    // Down button (PF5)

    else if (y == 0b00100000) {

        // if not at ground floor

        if (PORTD.OUT != 0b00000111) {

            // 2nd floor ? LED2 OFF : LED1 && LED2 OFF

            PORTD.OUT |= (PORTD.OUT == 0b00000001)? 0b00000100 : 0b00000110;

        }

    }

Αντίστοιχη λειτουργία υπάρχει και για το “πάνω” κουμπί (PF6).

    // Up button (PF6)

    else if (y == 0b01000000) {

        // if not at 2nd floor

        if (PORTD.OUT != 0b00000001) {

            // ground floor ? LED1 ON : LED1 && LED2 ON

            PORTD.OUTCLR = (PORTD.OUT == 0b00000111)? 0b00000010 : 0b00000110;

        }

    }

Τέλος, καθαρίζουμε το interrupt flag.

PORTF.INTFLAGS = y; //clear the interrupt flag

Το πρόγραμμα λειτουργεί όπως αναμένεται.

# Κώδικας

#include <avr/io.h>

#include <avr/interrupt.h>

#include <util/delay.h>

int main() {

    PORTD.DIR |= 0b00000111; //BITS 0-2 OUTPUT (0 = ERR)

    PORTD.OUT |= 0b00000111; //LED's are off

    // pullup enable and Interrupt enabled with sense on rising edge (button press)

    PORTF.PIN5CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_RISING\_gc;

    PORTF.PIN6CTRL |= PORT\_PULLUPEN\_bm | PORT\_ISC\_RISING\_gc;

    sei(); //enable interrupts

    while (1) {

         // printf("."); // print statement for easy debugging (breakpoint at each cycle)

    }

}

ISR(PORTF\_PORT\_vect){

    int y = PORTF.INTFLAGS;

    // Both buttons pressed (error)

    if(y == 0b01100000) {

        PORTD.OUTCLR = 0b00000001;

        \_delay\_ms(10);

        PORTD.OUT |= 0b00000001;

    }

    // Down button (PF5)

    else if (y == 0b00100000) {

        // if not at ground floor

        if (PORTD.OUT != 0b00000111) {

            // 2nd floor ? LED2 OFF : LED1 && LED2 OFF

            PORTD.OUT |= (PORTD.OUT == 0b00000001)? 0b00000100 : 0b00000110;

        }

    }

    // Up button (PF6)

    else if (y == 0b01000000) {

        // if not at 2nd floor

        if (PORTD.OUT != 0b00000001) {

            // ground floor ? LED1 ON : LED1 && LED2 ON

            PORTD.OUTCLR = (PORTD.OUT == 0b00000111)? 0b00000010 : 0b00000110;

        }

    }

    PORTF.INTFLAGS = y; //clear the interrupt flag

}