



Τμήμα Μηχανικών Η/Υ και Πληροφορικής
Computer Engineering and Informatics Department (CEID)
www.ceid.upatras.gr

Εργαστήριο Προηγμένοι Μικροεπεξεργαστές

Εργαστηριακές Ασκήσεις

Πάτρα, 2024-25

1 Εισαγωγικά

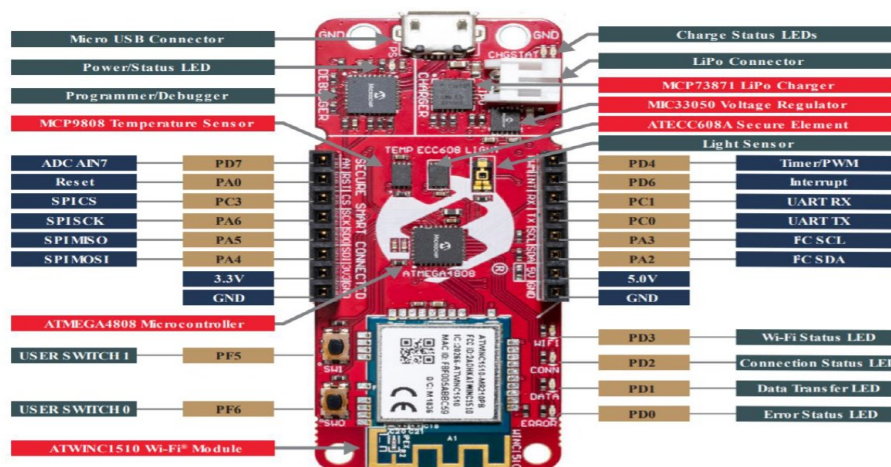
Η AVR είναι μια οικογένεια μικροεπεξεργαστών που αναπτύχθηκε αρχικά από την Atmel και μετέπειτα αγοράστηκε από την Microchip Technology. Βασίζεται σε μια τροποποιημένη Harvard αρχιτεκτονική, 8-bit RISC single-chip μικροεπεξεργαστή. Η AVR είναι μια από τις πρώτες οικογένειες μικροεπεξεργαστών που χρησιμοποίησαν on-chip flash memory, για αποθήκευση προγραμμάτων, σε αντίθεση με μια προγραμματιζόμενη ROM, EPROM ή EEPROM που χρησιμοποιούνταν από άλλους μικροελεγκτές. Οι μικροεπεξεργαστές της οικογένειας αυτής χρησιμοποιούνται σε πολλές εφαρμογές, ως πλήρη ενσωματωμένα συστήματα. Αξιοποιούνται ως εκπαιδευτικές ενσωματωμένες διατάξεις και έγιναν διάσημοι λόγω της ευρείας χρήσης τους σε πολλά open hardware development boards, της σειράς Arduino.

Το Microchip Studio (ή Atmel® Studio 7) είναι μια ολοκληρωμένη πλατφόρμα ανάπτυξης (Integrated Development Platform - IDP), για την δημιουργία εφαρμογών με AVR και SAM microcontrollers (MCU). Η πλατφόρμα προσφέρει εύχρηστο περιβάλλον για τη σύνταξη, την κατασκευή και τον εντοπισμό σφαλμάτων εφαρμογών που είναι γραμμένα σε C/C++ ή/και σε Assembly. Επίσης, παρέχεται η δυνατότητα εισαγωγής σχεδίων σε Arduino ως C/C++ projects και υποστηρίζονται 500+ AVR και SAM διατάξεις. Τέλος, το Microchip Studio περιλαμβάνει μια τεράστια βιβλιοθήκη πηγαίου κώδικα με 1600+ παραδείγματα εφαρμογών. Για περισσότερες πληροφορίες επισκεφτείτε την σελίδα της Microchip, στην οποία καταγράφονται αναλυτικά όλες οι δυνατότητες του Studio.

2 Το AVR-IoT Wx Development Board

Το AVR-IoT Wx Development Board είναι μια ευέλικτη και εύκολα επεκτάσιμη πλατφόρμα δημιουργίας και ανάπτυξης εφαρμογών. Βασίζεται στην αρχιτεκτονική μικροελεκτή AVR που χρησιμοποιεί τεχνολογία Wi-Fi. Συνοπτικά, μια αντίστοιχη εφαρμογή που αναπτύσσεται στο συγκεκριμένο σύστημα αποτελείται από τρία βασικά blocks:

- 1 ATmega4808 Microcontroller.
- 2 ATECC608A Secure Element.
- 3 ATWINC1510 Wi-Fi Controller Module.



Σχήμα 0.1 : AVR Development Board Overview

3 Εγκατάσταση Microchip Studio

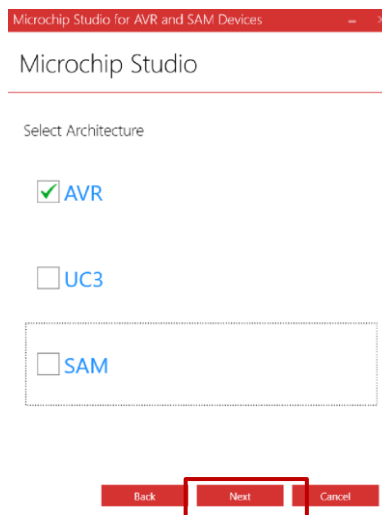
Το Microchip Studio μπορείτε να το βρείτε διαθέσιμο στον παρακάτω σύνδεσμο:

<https://www.microchip.com/mplab/microchip-studio>

Η εγκατάσταση του γίνεται εύκολα, ακολουθώντας τα παρακάτω βήματα:

- 1 Πατήστε Download στην επιλογή Microchip Studio for AVR and SAM Devices 7.0.2594 Web Installer.
- 2 Αφού ολοκληρωθεί η παραπάνω διαδικασία, ανοίξετε το αρχείο και περιμένετε.
 - Στο παράθυρο που θα εμφανιστεί επιλέξτε το path στο οποίο θα αποθηκευτεί το Studio και επιλέξτε το κουτί "I agree to the Terms and Conditions".
 - Στη συνέχεια επιλέξτε την επιλογή "Next".
- 3 Στο επόμενο παράθυρο, έχετε την επιλογή να εγκαταστήσετε τρεις διαφορετικές οικογένειες που υποστηρίζονται από το Studio:
 - Το πρώτο είναι το AVR, το οποίο θα πρέπει να παραμείνει επιλεγμένο καθώς με αυτό θα ασχοληθούμε.
 - Το UC3 και το SAM αποτελούν άλλα είδη αρχιτεκτονικών και δεν είναι απαραίτητο να τα επιλέξετε για εγκατάσταση.

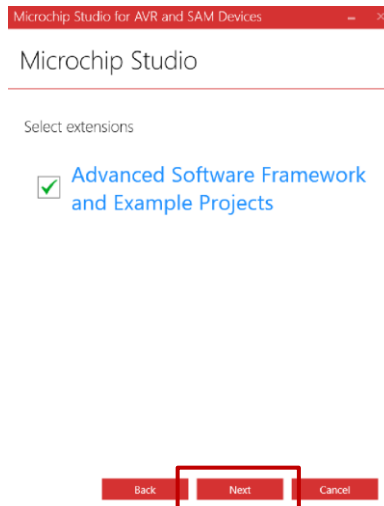
Αφού σιγουρευτείτε ότι επιλέξατε το AVR επιλέξτε "Next".



Σχήμα 0.2 : Παράθυρο Επιλογών AVR και SAM Devices

- 4 Advanced Software Framework (ASF):

Το ASF παρέχει ένα πλούσιο σύνολο λειτουργικών drivers και code modules που αναπτύχθηκαν από ειδικούς για τη μείωση του χρόνου σχεδιασμού. Το ASF είναι μια δωρεάν βιβλιοθήκη ανοιχτού κώδικα, που έχει σχεδιαστεί για να χρησιμοποιείται στις φάσεις αξιολόγησης, πρωτοτυπίας, σχεδιασμού και παραγωγής. Μπορείτε να το κρατήσετε επιλεγμένο και στον ελεύθερο χρόνο σας να δείτε τα παραδείγματα. Στη συνέχεια, μπορείτε να επιλέξετε "Next".



Σχήμα 0.3 : Παράθυρο Επιλογής ASF

- 5 Εφόσον όλες οι επιλογές του Validation είναι επιλεγμένες, είστε έτοιμοι στη συνέχεια να επιλέξετε “Next”. Στο επόμενο παράθυρο μπορείτε να επιλέξετε το “Install” και να περιμένετε να εγκατασταθεί το Microchip Studio και τα υπόλοιπα εργαλεία που χρειάζονται.



Σχήμα 0.4 : Παράθυρο Επιλογής System Validation

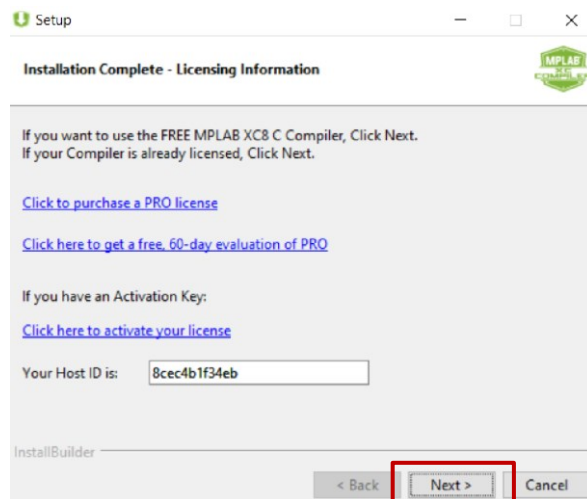
- 6 Κατά τη διάρκεια της εγκατάστασης, θα εμφανιστεί ένα επιπλέον παράθυρο επιλογών, για την εγκατάσταση του MPLAB XC8 C Compiler, με τον οποίο προσφέρεται η δυνατότητα βελτιστοποίησης του κώδικα.
- Επιλέξτε το “Next”, όπως φαίνεται στο παρακάτω σχήμα.
 - Έπειτα, επιλέξτε το «I accept the agreement» και στη συνέχεια επιλέξτε το “Next”.
 - Στα επόμενα παράθυρα επιλέξτε με τη σειρά το “Free” και μετά επιλέξτε το “Next”.
 - Μόλις τελειώσει η εγκατάσταση του, θα εμφανιστεί το αντίστοιχο παράθυρο στο οποίο μπορείτε να επιλέξετε το “Next”.
 - Στο επόμενο και τελευταίο για τον compiler παράθυρο μπορείτε να επιλέξετε το “Finish”, με το οποίο ολοκληρώνεται η εγκατάστασή του.

Εργαστήριο:

Προηγμένοι Μικροεπεξεργαστές



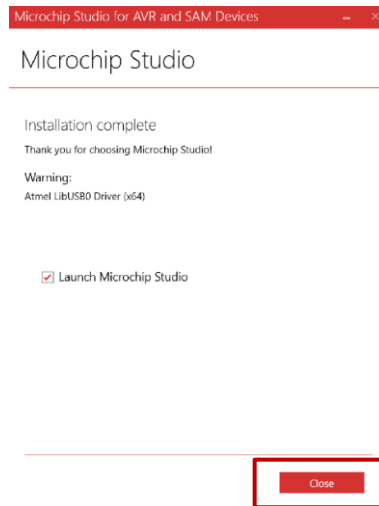
Σχήμα 0.5 : Παράθυρο Compiler Setup Wizard



Σχήμα 0.6 : Παράθυρο Licensing Information

- 7 Ένα άλλο εργαλείο που θα εγκατασταθεί είναι και το Microsoft Visual, το οποίο γίνεται αυτόματα, χωρίς να απαιτείται κάποια περαιτέρω ενέργεια.
- 8 Όταν ολοκληρωθεί η εγκατάσταση, θα εμφανιστεί το παρακάτω παράθυρο, στο οποίο μπορείτε να επιλέξετε "Close" και να ανοίξει αυτόματα το Microchip Studio.

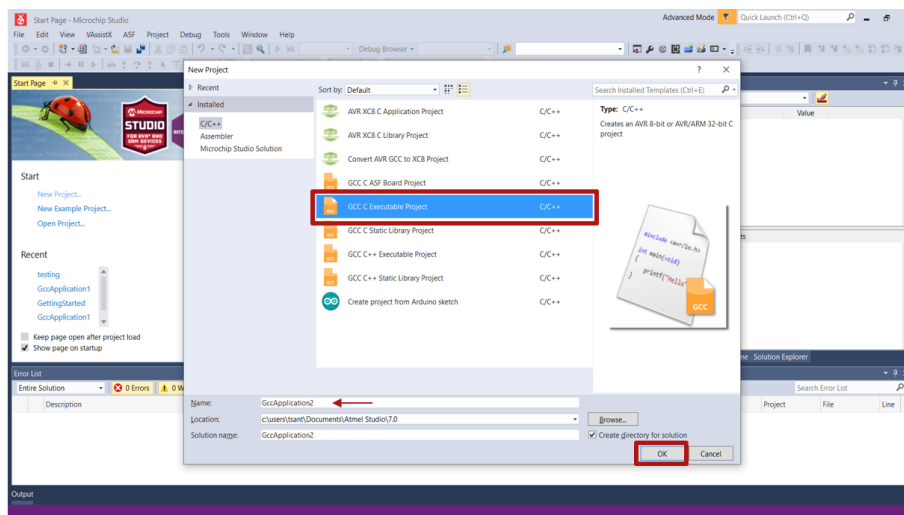
Εργαστήριο:
Προηγμένοι Μικροεπεξεργαστές



Σχήμα 0.7 : Παράθυρο Installation Complete

4 Δημιουργία Νέου Project

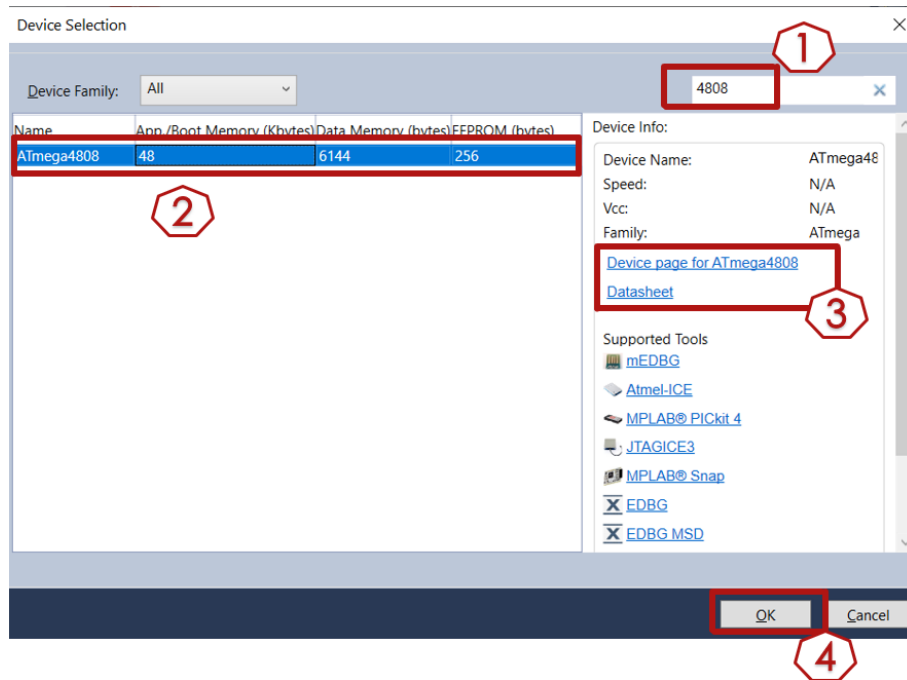
Στο εγκαταστημένο πλέον Microchip Studio, κατευθυνθείτε στην επιλογή “File → New → Project”. Τα παραδείγματα που θα υλοποιηθούν, θα είναι στη γλώσσα C και επομένως θα επιλέξουμε το “GCC C Executable Project”. Επιλέγουμε το όνομα της εφαρμογής μας και στη συνέχεια επιλέγουμε “OK”.



Σχήμα 0.8 : Παράθυρο Δημιουργίας “New Project”

Επιλέγουμε τον μικροελεγκτή “ATmega4808”, ακολουθώντας τα βήματα, που φαίνονται στην εικόνα του παρακάτω σχήματος. Μπορούμε να δούμε κατευθείαν το Datasheet και το Device Page του συγκεκριμένου μικροελεγκτή, για περισσότερες πληροφορίες σχετικά με τις δυνατότητές του (Βήμα 3). Αφού επιλέξουμε την συσκευή μας, στη συνέχεια επιλέγουμε το “OK” (Βήμα 4) και στη συνέχεια μπορούμε να αρχίσουμε να προγραμματίζουμε.

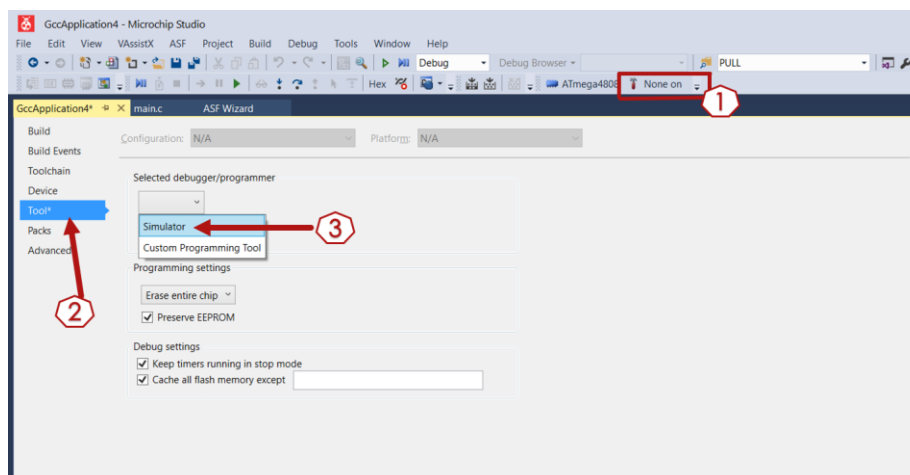
Εργαστήριο:
Προηγμένοι Μικροεπεξεργαστές



Σχήμα 0.9 : Παράθυρο Επιλογής “Device”

5 Simulation και Debugging

Είναι σημαντικό όταν κάνουμε πρώτη φορά Debug να έχουμε επιλέξει το “Simulator” ως “Debugger”, ακολουθώντας τα βήματα που φαίνονται στο παρακάτω σχήμα.



Σχήμα 0.10 : Παράθυρο Simulator – Debugger

6 Παράδειγμα 1: Λειτουργία ενός LED

Για να ενεργοποιηθεί/απενεργοποιηθεί ένα LED με κάποια συγκεκριμένη τιμή στην περίοδο, π.χ. $T=10\text{ms}$, θα πρέπει πρώτα να οριστεί το “Direction” του συγκεκριμένου Pin ως “Output”. Αρχικά, το PORT που θα χρησιμοποιηθεί είναι το PORTD καθώς οι τέσσερες πρώτοι ακροδέκτες (PINs) του συνδέονται με LEDs, πάνω στην πλακέτα (όπως φαίνεται και στο Board Overview). Για να οριστεί ένα PIN ως “Output” πρέπει να τεθεί το αντίστοιχο ψηφίο (bit) του Register DIR (Data Direction) με ‘1’. Τώρα μπορεί να δοθεί η τιμή ‘0’ ή ‘1’ στον Register Out (Output Value) και να “ενεργοποιείται” ή να “απενεργοποιείται”, το LED αντίστοιχα.

Σημείωση: Στη φάση του *Simulation* οι χρόνοι του “delay” όπως και των “timers”, (που θα δούμε σε επόμενη άσκηση) δεν είναι αντιπροσωπευτικοί του χρόνου που αναμένουμε. Μόνο με την εκτέλεση του κώδικα πάνω στην πλακέτα μπορούμε να αντιληφθούμε τους πραγματικούς χρόνους. Για το λόγο αυτό, μπορούμε να επιλέγουμε σχετικά μικρούς χρόνους, για να αποφύγουμε μεγάλες καθυστερήσεις αναμονής κατά την προσομοίωση.

Ο κώδικας της υλοποίησης του παραδείγματος παρουσιάζεται παρακάτω:

```
#include <avr/io.h>
#include <util/delay.h>
#define del 10
int main(void){

    PORTD.DIR |= 0b00000010; //PIN1_bm //PIN is output
    PORTD.OUT |= 0b00000010; //PIN1_bm //LED is off

    while (1) {

        PORTD.OUTCLR= 0b00000010; //PIN1_bm //LED is on
        _delay_ms(del);           //wait for 10ms

        PORTD.OUT |= 0b00000010; //PIN1_bm //LED is off
        _delay_ms(del);           //wait for 10ms

    }
}
```

Σχήμα 0.11 : Κώδικας Παραδείγματος 1

Μπορείτε να χρησιμοποιήσετε τον κώδικα του παραδείγματος, για να εκτελέσετε τη διαδικασία του “Simulation” στο “Microchip Studio”. Ανοίξτε το I/O παράθυρο (Debug ⇒ Windows ⇒ I/O) και ξεκινήστε το Debugging. Εκτελέστε βήμα-βήμα τις εντολές (η συνάρτηση delay δεν εκτελείται βηματικά) και παρατηρήστε τις τιμές που παίρνουν οι καταχωρητές (Registers) του PORTD.

Σημείωση: Για να δείτε τις τιμές των δηλωμένων σταθερών *PIN1_bm* και άλλων που θα δούμε στη συνέχεια, μπορείτε να ανατρέξετε στο header file *iom4808.h*, το οποίο βρίσκεται στον φάκελο που έχετε εγκαταστήσει το *Microchip Studio* :

π.χ. Path: “~\Atmel\Studio\7.0\packs\atmel\ATmega_DFP\1.7.374\include\avr\iom4808.h”

7 Παράδειγμα 2: Διακοπή (Interrupt)

Οι διακόπτες (switches) που μπορούν να χρησιμοποιηθούν βρίσκονται στο PORTF και είναι οι PIN5 και PIN6 (ανατρέξτε στο Board Overview). Για τη χρήση ενός διακόπτη (switch), το pullup enable bit που του αντιστοιχεί πρέπει να είναι ενεργοποιημένο (ανατρέξτε σελ. 158 στο ATmega4808 DataSheet). Επίσης, το σημείο του παλμού στο οποίο θα ενεργοποιηθεί η μονάδα διαχείρισης της διακοπής (interrupt) πρέπει να οριστεί. Εδώ η ενεργοποίησή της και στις δύο άκρες του παλμού επιλέγεται (ανατρέξτε σελ. 158 στο ATmega4808 DataSheet). Με την ενεργοποίηση των συγκεκριμένων ψηφίων (bits) του PIN5, το σύστημα μπορεί να δεχτεί διακοπές (interrupts) όταν πατηθεί ο διακόπτης (switch).

Όταν γίνει η διακοπή (interrupt), συγκεκριμένη συνάρτηση η οποία θα την διαχειριστεί, ενεργοποιείται. Αυτή η συνάρτηση είναι μια ρουτίνα διαχείρισης διακοπών (ISR routine) η οποία παίρνει ως όρισμα μια διεύθυνση (interrupt vector) η οποία αντιστοιχεί σε μία διακοπή. Για παράδειγμα, το interrupt vector για τη διακοπή του PORTF είναι το `PORTF_PORT_vect`. Στο αρχείο `iom4808.h` υπάρχουν όλα τα interrupt vectors που μπορούν να χρησιμοποιηθούν από το συγκεκριμένο board.

Ο κώδικας της υλοποίησης ενός παραδείγματος με διακοπή, παρουσιάζεται παρακάτω:

```
#include <avr/io.h>
#include <avr/interrupt.h>

int x=0; //logic flag

int main() {
    PORTD.DIR |= 0b00000010; //PIN is output
    PORTD.OUT |= 0b00000010; //LED is off
    //pullup enable and Interrupt enabled with sense on both edges
    PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc;
    sei(); //enable interrupts
    while (x==0){
        PORTD.OUTCLR= 0b00000010; // LED is on
    }

    PORTD.OUT |= 0b00000010; //LED is off
    cli(); //disable interrupts
}

ISR(PORTF_PORT_vect){
    int y = PORTF.INTFLAGS; //Procedure to
    PORTF.INTFLAGS=y; //clear the interrupt flag
    x=1; //change logic flag to get out of loop
}
```

Σχήμα 0.12 : Κώδικας Παραδείγματος 2

Η λειτουργία του κώδικα του παραδείγματος 2, είναι η εξής:

Στο σύστημα υπάρχει ένα LED και ένας διακόπτης (switch). Σύμφωνα με την κανονική λειτουργία του προγράμματος, το LED ενεργοποιείται μέσα σε ένα βρόχο (while loop). Όταν πατηθεί ο διακόπτης ενεργοποιείται η διακοπή (interrupt) και αλλάζει η τιμή μιας τυχαίας μεταβλητής (int x) ώστε να σταματήσει η λειτουργία του LED και του προγράμματος.

Για να ενεργοποιηθεί η διακοπή (interrupt) το ψηφίο (bit) 5 του register INTFLAGS στο PORTF πρέπει να πατηθεί, εφόσον το πρόγραμμα είναι σε κάποιο breakpoint. Μόλις το ψηφίο (bit) αλλάξει από '0' σε '1' και το πρόγραμμα συνεχίσει με την επόμενη εντολή (STEP OVER), θα ενεργοποιηθεί η ρουτίνα διαχείρισης διακοπών (interrupt routine).

Εργαστηριακή Άσκηση 01:

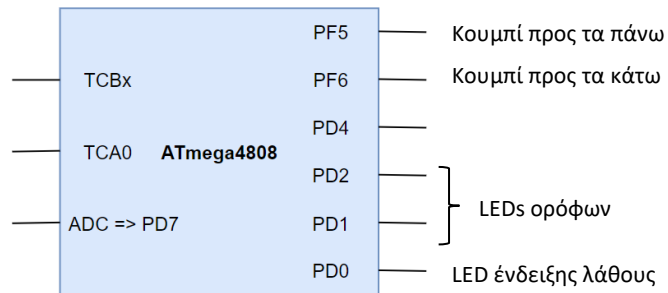
Ανελκυστήρας

1 Περιγραφή

Σκοπός της Εργαστηριακής Άσκησης είναι η προσομοίωση της λειτουργίας ενός ανελκυστήρα, το οποίο μπορεί να κινηθεί μεταξύ 3 επιπέδων. Το ασανσέρ διαθέτει δύο κουμπιά για την μετακίνηση στις δυο κατευθύνσεις: προς τα πάνω και προς τα κάτω. Η κίνηση πραγματοποιείται μεταξύ των ορόφων ισόγειο, 1^{ος} και 2^{ος} μετά από πίεση του αντίστοιχου κουμπιού (βέλος προς τα πάνω και βέλος προς τα κάτω - push button) και ενεργοποιείται ένας διαφορετικός συνδυασμός από μια σειρά LEDs για να υποδεικνύεται η θέση του ανελκυστήρα.

Για την υλοποίηση της Εργαστηριακής Άσκησης 1, θεωρήστε ότι:

- Ο ανελκυστήρας ξεκινάει από το ισόγειο (επίπεδο 0) και κάθε φορά που βρίσκεται σε αυτό, όλα τα διαθέσιμα LEDs είναι σβηστά (λογικό '1').
- Όταν πατηθεί το βέλος προς τα πάνω, μια διακοπή (interrupt) προκύπτει και ένα LED ανάβει (λογικό '0') υποδηλώνοντας ότι ο ανελκυστήρας ανεβαίνει ένα επίπεδο από αυτό που βρισκόταν προηγουμένως.
- Αντίστοιχα, κάθε φορά που πατιέται το βέλος προς τα κάτω, το LED που άναψε τελευταίο σβήνει (λογικό '1') υποδηλώνοντας ότι ο ανελκυστήρας κατεβαίνει ένα επίπεδο από αυτό που βρισκόταν προηγουμένως.
- Σε περίπτωση που είτε πατηθεί το πάνω βέλος ενώ ο ανελκυστήρας βρίσκεται στον 2^ο όροφο, είτε πατηθεί το κάτω βέλος ενώ ο ανελκυστήρας βρίσκεται στο ισόγειο, δεν εκτελείται καμία ενέργεια και τα LEDs παραμένουν στην κατάσταση που ήταν.
- Σε περίπτωση που πατηθούν ταυτόχρονα και τα δυο κουμπιά, το LED ένδειξης λάθους ενεργοποιείται για 10ms και έπειτα σβήνει, χωρίς να μεταβληθεί η κατάσταση των υπόλοιπων LEDs.
- Τα PIN του PORTD που χρησιμοποιούνται για την ενεργοποίηση των LEDs είναι τα PIN0 (ένδειξη λάθους), PIN1, και PIN2. Η ένδειξη του ορόφου στον οποίο βρίσκεται ο ανελκυστήρας, δίνεται από το πλήθος των LEDs που είναι αναμμένα. Έτσι, στον όροφο 0 όλα τα LEDs είναι σβηστά, στον 1^ο μόνο ένα LED είναι αναμμένο και στον 2^ο δυο LEDs.
- Για την υλοποίηση των διακοπών, χρησιμοποιούνται τα PIN5 και PIN6 του PORTF.
- Ο χρόνος (10ms) σε αυτή και μόνο την άσκηση θα υλοποιηθεί με τη χρήση delay.



Σχήμα 1.1 : Απεικόνιση ATmega4808, Εργαστηριακής Άσκησης 01

Χρήσιμη οδηγία: Δείτε τα παραπάνω παραδείγματα 1 και 2.

2 Ερωτήματα Εργαστηριακής Άσκησης 1

- 1 Υλοποιήστε τη λειτουργία κίνησης του ανελκυστήρα, χρησιμοποιώντας όλες τις διαδικασίες που εξηγήθηκαν.

3 Αναφορά Εργαστηριακής Άσκησης 1

- 1 Αναπτύξτε **τον κώδικά σας** συμπεριλαμβάνοντας τα απαραίτητα σχόλια επεξήγησης του κώδικα και βεβαιωθείτε πως όλες οι λειτουργίες εκτελούνται, όπως προβλέπεται.
- 2 Παραδίδετε **διάγραμμα ροής** καθώς και **αναλυτική περιγραφή** της λειτουργίας του κώδικά σας, στην οποία θα εξηγείτε τον τρόπο με τον οποίο εκτελούνται όλες οι ανωτέρω διαδικασίες.

*** Σημείωση: Παραδίδετε ένα συνολικό αρχείο με τον κώδικα, τα αναλυτικά σχόλια για τις εντολές που χρησιμοποιήσατε, το διάγραμμα και την περιγραφή ΑΝΑ ΟΜΑΔΑ (μόνο ο ένας εκ των δυο της ομάδας κάνει υποβολή).

*** ΑΝΑΓΡΑΦΕΤΕ **ΑΠΑΡΑΙΤΗΤΩΣ τα ονόματα και ΑΜ των μελών της ομάδας** εντός του αρχείου που παραδίδετε.

*** Το όνομα του αρχείου να είναι: AM1_AM2_EA1.docx

Βιβλιογραφία - Αναφορές

- 1 Microchip, Επίσημη ιστοσελίδα:
<https://www.microchip.com/mplab/microchip-studio>
- 2 AVR-IoT WxHardware User Guide:
<https://ww1.microchip.com/downloads/en/DeviceDoc/AVR-IoT-Wx-Hardware-User-Guide-DS50002805C.pdf>
- 3 ATmega4808/4809 Data Sheet:
<https://ww1.microchip.com/downloads/en/DeviceDoc/ATmega4808-09-DataSheet-DS40002173B.pdf>
- 4 Microchip, “Advanced Software Framework (ASF)”:
<https://www.microchip.com/mplab/avr-support/advanced-software-framework>
- 5 Microchip, “MPLAB® XC Compilers”:
<https://www.microchip.com/en-us/development-tools-tools-and-software/mplab-xc-compilers>