

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

**Output 2: Online Course for Microcontrollers:
syllabus, open educational resources**

Practice leaflet: Module_2-4 INT0, INT1, INT2

Lead Partner: International Hellenic University (IHU)

Δήλωση

Αυτό το αρχείο συντάχθηκε στο πλαίσιο του έργου ENGINE. Όπου έχουν χρησιμοποιηθεί άλλα δημοσιευμένα και αδημοσίευτα υλικά, αυτά έχουν αναγνωρισθεί.

Πνευματική ιδιοκτησία

© Copyright 2021 - 2023 the [ENGINE](#) Consortium

Warsaw University of Technology (Poland)

International Hellenic University (IHU) (Greece)

European Lab for Educational Technology- EDUMOTIVA (Greece)

University of Padova (Italy)

University of Applied Sciences in Tarnow (Poland)

Όλα τα δικαιώματα διατηρούνται.



Αυτό το έγγραφο έχει άδεια Creative Commons Attribution-NonCommercial- NoDerivatives 4.0 International License.

Αυτό το έργο έχει χρηματοδοτηθεί με την υποστήριξη της Ευρωπαϊκής Επιτροπής. Αυτή η έκθεση αντικατοπτρίζει μόνο τις απόψεις του συγγραφέα και η Επιτροπή δεν μπορεί να θεωρηθεί υπεύθυνη για οποιαδήποτε χρήση των πληροφοριών που περιέχονται σε αυτήν.

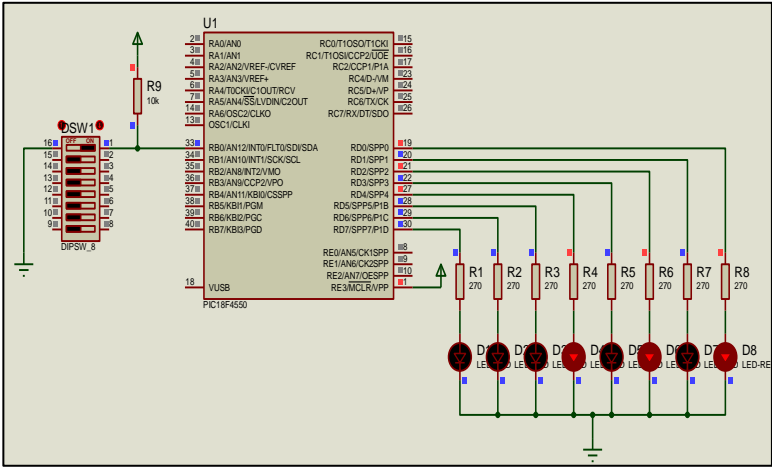
Πίνακας Περιεχομένων

Δραστηριότητες.....	4
1. Αύξηση μετρητή με χρήση INT0.....	4
2. Flash – INT1.....	5
3. Flash και κινούμενη τελεία.....	6

Δραστηριότητες

1. Αύξηση μετρητή με χρήση INTO

Στόχος της δραστηριότητας είναι κάθε φορά που αλλάζει η κατάσταση στο RB0 από λογικό 1 σε λογικό 0 να αυξάνεται ένας μετρητής μέσω πραγματοποίησης interrupt. Το περιεχόμενο του μετρητή θα εμφανίζεται σε 8 LEDs. Στο κυρίως πρόγραμμα δεν θα εκτελείται κάτι.

<p>(20 λεπτά)</p>	<p>Βήμα 1. Υλοποίηση του κυκλώματος</p> <p>Βήμα 2. Ολοκλήρωση κώδικα</p> <p>Βήμα 3. Μεταφορά κώδικα στον μικροελεγκτή</p> <p>Βήμα 4. Έλεγχος λειτουργίας</p>
<p>Βήμα 1 (10 λεπτά)</p>	<p>Υλοποίηση του κυκλώματος. Διασύνδεση PORTD με 8 LEDs, και διακόπτη με το RB0.</p>  <p><i>Figure 1. Διασύνδεση</i></p>

<p>Βήμα 2 (5 λεπτά)</p>	<p>Μελετήστε και συμπληρώστε τον παρακάτω κώδικα</p> <pre> #include <main.h> #define PORTD =0xF83 //δήλωση συναρτήσεων void init(void); void ext_int(void); //η μεταβλητή i θα χρησιμοποιηθεί ως μετρητής int i=0; void main(){ init(); //καλώ την συνάρτηση που κάνει αρχικοποιήσεις while(TRUE){;} //για πάντα } void init(){ set_tris_d(0x00); //η PORTD γίνεται έξοδος PORTD = 0b00000000; //αρχικοποίηση πόρτας ext_int_edge(0, H_TO_L); //ορισμός interrupt από το RB0 //και σε μετάβαση από "1" σε "0" enable_interrupts(GLOBAL); //ενεργοποίηση "γενικού διακόπτη" enable_interrupts(INT_EXT); //ενεργοποίηση διακοπής από το RB0 } #define INT_EXT HIGH //Ρουτίνα εξυπηρέτησης διακοπής από το RB0 void ext_int1(){ ----- //αυξάνεται ο μετρητής κατά 1 ----- //εμφανίζεται στην PORTD το περιεχόμενο του μετρητή } </pre>
<p>Βήμα 3 (3 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p>
<p>Βήμα 4 (2 λεπτά)</p>	<p>Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>

2. Flash – INT1

Σε αυτήν την δραστηριότητα θέλουμε κάθε φορά που αλλάζει κατάσταση στο RB1 από λογικό 0 σε λογικό 1, ένα LED που είναι συνδεδεμένο στο RD7 να αναβοσβήνει 2 φορές μέσω της αντίστοιχης ρουτίνας εξυπηρέτησης της διακοπής. Στο κυρίως πρόγραμμα δεν θα εκτελείται κάτι.

<p>(10 λεπτά)</p>	<p>*** Συνδέστε κατάλληλα το κύκλωμα ώστε να μπορεί να δημιουργηθεί interrupt από το RB1 ***</p> <p>Βήμα 1. Συμπλήρωση κώδικα</p>
-------------------	---

	<p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή</p> <p>Βήμα 3. Έλεγχος λειτουργίας</p>
<p>Βήμα 1 (4 λεπτά)</p>	<p>Συμπληρώστε τον παρακάτω κώδικα</p> <pre> #include <main.h> #define PORTD =0xF83 //δήλωση συναρτήσεων void init(void); void ext_int1(void); void main(){ init(); //καλώ την συνάρτηση που κάνει αρχικοποιήσεις while(TRUE){;} //για πάντα } void init(){ set_tris_d(0x00); //η PORTD γίνεται έξοδος PORTD = 0b00000000; //αρχικοποίηση πόρτας ext_int_edge(1, L_TO_H); //ορισμός interrupt από το RB1 //και σε μετάβαση από "0" σε "1" enable_interrupts(GLOBAL); //ενεργοποίηση "γενικού διακόπτη" enable_interrupts(INT_EXT1); //ενεργοποίηση διακοπής από το RB0 } #define INT_EXT1 HIGH //Ρουτίνα εξυπηρέτησης διακοπής από το RB1 void ext_int1(){ int i; for(i=1;i<3; i++){ //εκτελείται 2 φορές output_high(PIN_D7); //LED ανάβει ----- //αναμονή για 0.5s ----- //LED σβήνει ----- //αναμονή για 0.5s } } </pre>
<p>Βήμα 2 (4 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p>
<p>Βήμα 3 (2 λεπτά)</p>	<p>Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>

3. Flash και κινούμενη τελεία

Σε αυτήν την δραστηριότητα το κυρίως πρόγραμμα αναβοσβήνει 8 LEDs που είναι συνδεδεμένα στην PORTD. Όταν εκτελεστεί μία εξωτερική διακοπή από το RB2, τα 8 LEDs σχηματίζουν μία κινούμενη τελεία και μετά επιστρέφουν στην εκτέλεση του κυρίως προγράμματος.

<p>(20 λεπτά)</p>	<p>*** Συνδέστε κατάλληλα το κύκλωμα ώστε να μπορεί να δημιουργηθεί interrupt από το RB2 ***</p> <p>Βήμα 1. Ολοκλήρωση κώδικα</p> <p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή και έλεγχος λειτουργίας</p>
-------------------	---

<p style="text-align: center;">Βήμα 1 (14 λεπτά)</p>	<p style="text-align: center;">Συμπληρώστε τον παρακάτω κώδικα</p> <pre> #include <main.h> #define PORTD =0xF83 //δήλωση συναρτήσεων void init(void); void ext_int2(void); void main(){ init(); //καλώ την συνάρτηση που κάνει αρχικοποιήσεις while(TRUE){ //flash 8 LEDs PORTD=0b11111111; delay_ms(300); PORTD=0b00000000; delay_ms(300); } } //ρουτίνα αρχικοποιήσεων void init(){ set_tris_d(0x00); //η PORTD γίνεται έξοδος PORTD = 0b00000000; //αρχικοποίηση πόρτας ----- //ορισμός διακοπής από το RB2 //και σε μετάβαση από λογικό 1 σε λογικό 0 ----- //ενεργοποίηση "γενικού διακόπτη" ----- //ενεργοποίηση διακοπής από το RB0 } //Ρουτίνα εξυπηρέτησης διακοπής από το RB2 #INT_EXT2 void ext_int2() { //δημιουργία κινούμενης τελείας ----- } </pre>
<p style="text-align: center;">Βήμα 2 (6 λεπτά)</p>	<p>Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή. Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p>