

ENGINE

Teaching online electronics, microcontrollers and programming in Higher Education

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

Practice leaflet: Module_2-3 push button

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.

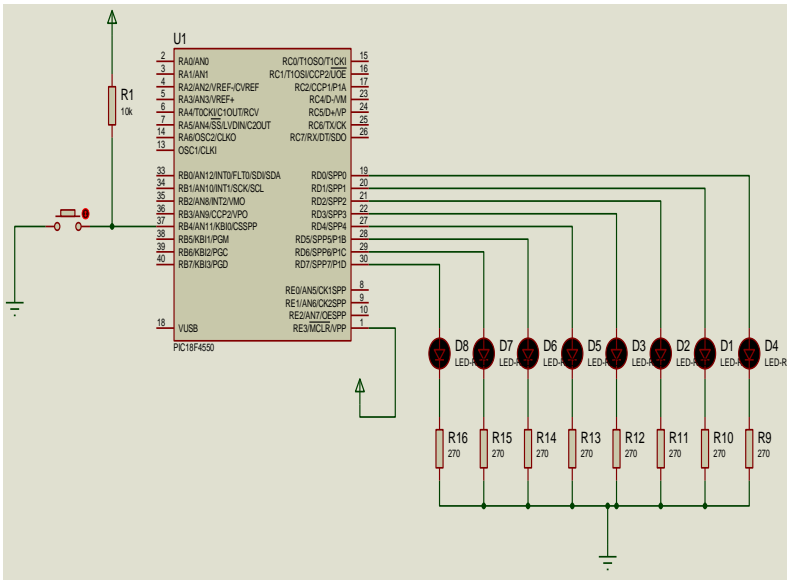
Πίνακας Περιεχομένων

| | |
|---|---|
| Δραστηριότητες..... | 4 |
| 1. Αύξηση μετρητή με χρήση push button | 4 |
| 2. Ανάγνωση πατήματος push button | 5 |
| 3. Ανάγνωση πατήματος push button με κλήση συνάρτησης | 7 |

Δραστηριότητες

1. Αύξηση μετρητή με χρήση push button

Στόχος της δραστηριότητας είναι κάθε φορά που πατάμε ένα μπουτόν το οποίο είναι συνδεδεμένο στον ακροδέκτη RB4 της πόρτας B το περιεχόμενο της πόρτας D να αυξάνεται κατά 1. Το αποτέλεσμα της αύξησης να φαίνεται σε 8 LEDs τα οποία είναι συνδεδεμένα στην πόρτα D.

| | |
|-------------------------------------|--|
| <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 push button με το RB4.</p>  <p>Figure 1. Διασύνδεση</p> |

| | |
|--|--|
| <p style="text-align: center;">Βήμα 2 (5 λεπτά)</p> | <p style="text-align: center;">Μελετήστε τον παρακάτω κώδικα</p> <pre> #include<main.h> #byte PORTD=0xF83 #byte PORTB=0xF81 //*****Από εδώ αρχίζει το κύριο πρόγραμμα***** void main() { //άνοιγμα αγκύλης της συνάρτησης main set_tris_d(0x00); //Η θύρα D γίνεται έξοδος set_tris_b(0xFF); //Η θύρα B γίνεται είσοδος PORTD=0x00; //Στην πόρτα D δίνεται η αρχική τιμή 00000000 while(TRUE) { //Βρόχος που δεν τελειώνει ποτέ (συνθήκη πάντα αληθής) while(input(PIN_B4) == 1) { // Αναμονή έως ότου πατηθεί το μπουτόν } // Στην κατάσταση αναμονής δεν εκτελείται καμιά εντολή // Όταν πατηθεί το μπουτόν γίνεται έξοδος από τον βρόχο // και εκτελείται η επόμενη εντολή delay_ms(50); // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων while(input(PIN_B4) == 0) { //Αναμονή έως ότου αφεθεί το μπουτόν } // Στην κατάσταση αναμονής δεν εκτελείται καμιά εντολή // Όταν πατηθεί το μπουτόν γίνεται έξοδος από τον βρόχο // και εκτελείται η επόμενη εντολή delay_ms(50); // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων PORTD=PORTD+1; //Αύξηση του περιεχομένου της πόρτας D +1 } //κλείσιμο της αγκύλης του while(TRUE) } // κλείσιμο της αγκύλης του main </pre> |
| <p style="text-align: center;">Βήμα 3 (3 λεπτά)</p> | <p style="text-align: center;">Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή</p> |
| <p style="text-align: center;">Βήμα 4 (2 λεπτά)</p> | <p style="text-align: center;">Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά</p> |

2. Ανάγνωση πατήματος push button

Σε αυτήν την δραστηριότητα θέλουμε κάθε φορά που πατιέται και αφήνεται ένα μπουτόν να αλλάζει η κατάσταση του κάθε ακροδέκτη της πόρτας D, δηλαδή τα 1 γίνονται 0 και τα 0 γίνονται 1. Στην πόρτα D θα δοθεί η αρχική τιμή 00001111.

| | |
|-----------------------------|--|
| <p>(10 λεπτά)</p> | <p style="text-align: center;">*** το κύκλωμα είναι ίδιο με πριν***</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> #byte PORTD=0xF83 #byte PORTB=0xF81 // *****Από εδώ αρχίζει το κύριο πρόγραμμα***** void main() { set_tris_d(0x00); //H θύρα D έξοδος set_tris_b(0x00); //H θύρα B έξοδος PORTD=0x0F; //Στην πόρτα D δίνεται η αρχική τιμή 00001111 while(TRUE) { //Βρόχος που δεν τελειώνει ποτέ while(input(PIN_B4) == 1) { // Αναμονή έως ότου πατηθεί το μπουτόν } delay_ms(50); // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων // Αναμονή έως ότου αφηθεί το μπουτόν // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων PORTD=PORTD^0b11111111; // γίνεται αναστροφή των τιμών των ακροδεκτών της πόρτας D // μέσω της λογικής πύλης XOR } } </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. Ανάγνωση πατήματος push button με κλήση συνάρτησης

Σε αυτήν την δραστηριότητα θέλουμε κάθε φορά που πατιέται και αφήνεται ένα μπουτόν να αλλάζει η κατάσταση του κάθε ακροδέκτη της πόρτας D, δηλαδή τα 1 γίνονται 0 και τα 0 γίνονται 1. Στην πόρτα D θα δοθεί η αρχική τιμή 00001111. Για την ανάγνωση του push button θα χρησιμοποιηθεί συνάρτηση.

| | |
|-------------------------------------|--|
| <p>(20 λεπτά)</p> | <p style="text-align: center;">*** το κύκλωμα είναι ίδιο με πριν***</p> <p>Βήμα 1. Ολοκλήρωση κώδικα</p> <p>Βήμα 2. Μεταφορά κώδικα στον μικροελεγκτή και έλεγχος λειτουργίας</p> |
| <p>Βήμα 1 (14 λεπτά)</p> | <p>Συμπληρώστε τον παρακάτω κώδικα</p> <pre>#include<main.h> #byte PORTD=0xF83 #byte PORTB=0xF81 void push_button(void); //Δήλωση της συνάρτησης πατήματος μπουτόν. //Προσοχή, πάνω από την main() //Με αυτή τη συνάρτηση εκτελούνται όλες οι εντολές //με τις οποίες //ο μικροελεγκτής αναγνωρίζει το πλήρες //πάτημα (πατώ+αφήνω) ενός μπουτόν //*****Από εδώ αρχίζει το κύριο πρόγραμμα***** void main() { set_tris_d(0x00); //Η θύρα D έξοδος set_tris_b(0xFF); //Η θύρα B είσοδος PORTD=0x0F; while(TRUE) { //συνθήκη πάντα αληθής //Γίνεται κλήση της συνάρτησης πατήματος //μπουτόν push_button(); //εναλλαγή καταστάσεων στα bit της PORTD ----- } } //Ορισμός της συνάρτησης push_button //Ο ορισμός της συνάρτησης γράφεται μετά το κύριο //πρόγραμμα //Με τη συνάρτηση αυτή ο μικροελεγκτής περιμένει //ένα πλήρες πάτημα του μπουτόν (πατώ+αφήνω) //και στη συνέχεια πάει στην επόμενη εντολή. void push_button(void) { while(input(PIN_B4) == 1) { // Αναμονή έως ότου πατηθεί το //μπουτόν }</pre> |

| | |
|---------------------|---|
| | <pre>// Στην κατάσταση αναμονής δεν εκτελείται καμιά εντολή // Όταν πατηθεί το μπουτόν γίνεται έξοδος από τον βρόχο // και εκτελείται η επόμενη εντολή delay_ms(50); // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων while(input(PIN_B4) == 0) { //Αναμονή έως ότου αφεθεί το μπουτόν } // Στην κατάσταση αναμονής δεν εκτελείται καμιά εντολή // Όταν αφεθεί το μπουτόν γίνεται έξοδος από τον βρόχο // και εκτελείται η επόμενη εντολή delay_ms(50); // Καθυστέρηση 50 ms για αποφυγή φαινόμενου αναπηδήσεων }</pre> |
| Βήμα 2 (6 λεπτά) | Δημιουργήστε το hex file και φορτώστε το στον μικροελεγκτή. Ελέγξτε ότι το κύκλωμα λειτουργεί σωστά |