

Εργαστηριακή Άσκηση 4

MICROCHIP STUDIO

Διάρθρωση Παρουσίασης

- ▶ Σκοπός
- ▶ Περιγραφή και υλοποίηση παραδείγματος με Pulse-Width Modulation (PWM)
- ▶ Επεξήγηση εργαστηριακής άσκησης

ΣΚΟΠΟΣ

Ο σκοπός της άσκησης αυτής είναι η εξοικείωσή σας με:

- ▶ Εξοικείωση με το Pulse-Width Modulation (PWM).

Εφαρμογές PWM

- ▶ Το Pulse-Width Modulation μπορεί να χρησιμοποιηθεί με πολλά περιφερειακά και εφαρμογές, όπως:
 - Audio
 - Ρύθμιση έντασης LED
 - Analog signal generation
 - Ρύθμιση servos, DC motors, κτλ.
- ▶ Μέσω του Microchip Studio μπορούμε να δημιουργήσουμε ένα waveform το οποίο να συνδεθεί με το pin ενός περιφερειακού. Συγκεκριμένα, το waveform output είναι διαθέσιμο στο PORT που επιλέγουμε και μπορεί να χρησιμοποιηθεί ως έξοδος (αρκεί να οριστεί το PORT ως έξοδος).
- ▶ Ωστόσο, είναι δύσκολο να δουλέψουμε με περιφερειακά μέσω του Simulation.

Παράδειγμα

ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ PWM

Περιγραφή Παραδείγματος

- ▶ Στο παράδειγμα αυτό θέλουμε να ενεργοποιήσουμε το PWM ώστε να δημιουργήσουμε έναν παλμό που θα λειτουργεί σαν ρολόι. Με αυτό θα αναβοσβήνει ένα LED.
- ▶ Θέλουμε το LED να ανάβει όταν ο παλμός ανεβαίνει στην ψηλή στάθμη (rising edge) και να σβήνει όταν πέφτει στη χαμηλή στάθμη (falling edge).

Single-Slope PWM Generate

- ▶ Με τη βοήθεια του TCA timer μπορούμε να δημιουργήσουμε παλμούς.
- ▶ Για τη χρήση του Single-Slope PWM αρχικά πρέπει να ορίσουμε τον prescaler του timer (όπως και στην περίπτωση του απλού timer)
→ `TCA0.SINGLE.CTRLA=TCA_SINGLE_CLKSEL_DIV1024_gc`
- ▶ Έπειτα, ορίζουμε μέχρι ποια μέγιστη τιμή TOP θα μετρήσει ο timer →
`TCA0.SINGLE.PER = value`
- ▶ Θέτουμε το duty cycle του παλμού μέσω της χρήσης του καταχωρητή CMPx →
`TCA0.SINGLE.CMP0 = value`
- ▶ Επιλέγουμε το Waveform Generation Mode που θέλουμε, στην περίπτωσή μας το Single-Slope → `TCA0.SINGLE.CTRLB |= TCA_SINGLE_WGMODE_SINGLESLOPE_gc`
- ▶ Τέλος, θέτουμε στο enable το TCA →
`TCA0.SINGLE.CTRLA |= TCA_SINGLE_ENABLE_bm`

Διευκρινήσεις

- ▶ Με την επιλογή του prescaler και την εκχώρηση τιμής στον καταχωρητή PER μπορούμε να υπολογίσουμε την συχνότητα με τον παρακάτω τύπο.

$$f_{\text{PWM_SS}} = \frac{f_{\text{CLK_PER}}}{N(\text{PER}+1)}$$

- ▶ Ο καταχωρητής PER είναι 16-bit, επομένως το ελάχιστο resolution είναι TCA.PER = 0x0002 και το μέγιστο είναι TCA.PER = MAX-1.
- ▶ Η τιμή του CMPx καθορίζει το duty cycle. Για παράδειγμα, αν του δώσουμε την μισή τιμή του καταχωρητή PER τότε έχουμε 50% duty cycle.
- ▶ Υπάρχουν και άλλες λειτουργίες του PWM. Για περισσότερες λεπτομέρειες ανατρέξτε στο ATmega4808 DataSheet σελ.191-199.

Interrupt Enable

- ▶ Πρέπει να ενεργοποιήσουμε το Overflow Interrupt → όταν ο timer είναι ίσος με την BOTTOM τιμή (πηγαίνουμε στην ψηλή στάθμη) →
`TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm`
- ▶ Πρέπει να ενεργοποιήσουμε και το CMPx Interrupt → όταν ο timer είναι ίσος με την τιμή του καταχωρητή CMPx (πηγαίνουμε στην χαμηλή στάθμη) →
`TCA0.SINGLE.INTCTRL |= TCA_SINGLE_CMP0_bm`

Συμβουλή: Ανατρέξτε στο *ATmega4808 DataSheet* και διαβάστε καλά τις σελίδες σχετικά με τον TCA0. Αναγράφονται αναλυτικά όλες οι πιθανές λειτουργίες του PWM καθώς και υπάρχουν παραδείγματα για τον σχηματισμό παλμών. Επίσης, στην άσκηση μπορείτε να χρησιμοποιήσετε και τους timers TCB που έχουν λειτουργία 8-bit PWM (σελ.239-).

Υλοποίηση

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

int main(){
    PORTD.DIR |= PIN1_bm; //PIN is output
    //prescaler=1024
    TCA0.SINGLE.CTRLA=TCA_SINGLE_CLKSEL_DIV1024_gc;
    TCA0.SINGLE.PER = 254; //select the resolution
    TCA0.SINGLE.CMP0 = 127; //select the duty cycle
    //select Single_Slope_PWM
    TCA0.SINGLE.CTRLB |= TCA_SINGLE_WGMODE_SINGLESLOPE_gc;
    //enable interrupt Overflow
    TCA0.SINGLE.INTCTRL = TCA_SINGLE_OVF_bm;
    //enable interrupt COMP0
    TCA0.SINGLE.INTCTRL |= TCA_SINGLE_CMP0_bm;
    TCA0.SINGLE.CTRLA |= TCA_SINGLE_ENABLE_bm; //Enable
    sei();
    while (1){ }
}
```

```
ISR(TCA0_OVF_vect){
    //clear the interrupt flag
    int intflags = TCA0.SINGLE.INTFLAGS;
    TCA0.SINGLE.INTFLAGS = intflags;
    PORTD.OUT |= PIN1_bm; //PIN is off
}
```

```
ISR(TCA0_CMP0_vect){
    //clear the interrupt flag
    int intflags = TCA0.SINGLE.INTFLAGS;
    TCA0.SINGLE.INTFLAGS = intflags;
    PORTD.OUTCLR |= PIN1_bm; //PIN is off
}
```

- ▶ Πάρτε τον κώδικα και κάντε Simulation στο Microchip Studio.
- ▶ Παρατηρήστε πως λειτουργεί ο timer TCA.

Εργαστηριακή Άσκηση

ΥΛΟΠΟΙΗΣΗ ΜΝΗΜΗΣ RAM

Περιγραφή

- ▶ Σε αυτή την εργαστηριακή άσκηση θα προσομοιώσουμε τη λειτουργία μιας μνήμης RAM, η οποία θα έχει έναν ρολόι για READ και ένα ρολόι για WRITE.
- ▶ Η RAM αποτελείται από 32 θέσεις των 4-bit η κάθε μία.
- ▶ Για την εγγραφή μιας τιμής θα πατάμε το switch5 και όταν έρθει το rising edge ή το falling edge του αντίστοιχου παλμού θα εγγράφεται μια τυχαία τιμή σε μια διαθέσιμη θέση.
- ▶ Για την ανάγνωση μια τιμής θα πατάμε το switch6 και όταν έρθει το falling edge του αντίστοιχου παλμού, θα εμφανίζεται η τελευταία τιμή που αποθηκεύτηκε με τη βοήθεια των τεσσάρων LED που έχουμε στη διάθεσή μας.
- ▶ Θέλουμε οι ενέργειες αυτές να γίνονται ξεχωριστά. Συγκεκριμένα, φροντίστε οι παλμοί να μην είναι όμοιοι και να μην ταυτίζονται τα edges. Επίσης, σε περίπτωση που πατηθεί ένα switch ενώ εκτελείται μια άλλη λειτουργία, περιμένουμε να ολοκληρωθεί η προηγούμενη και μετά με τον επόμενο παλμό την εκτελούμε χωρίς να χρειαστεί να ξαναπατηθεί το switch.

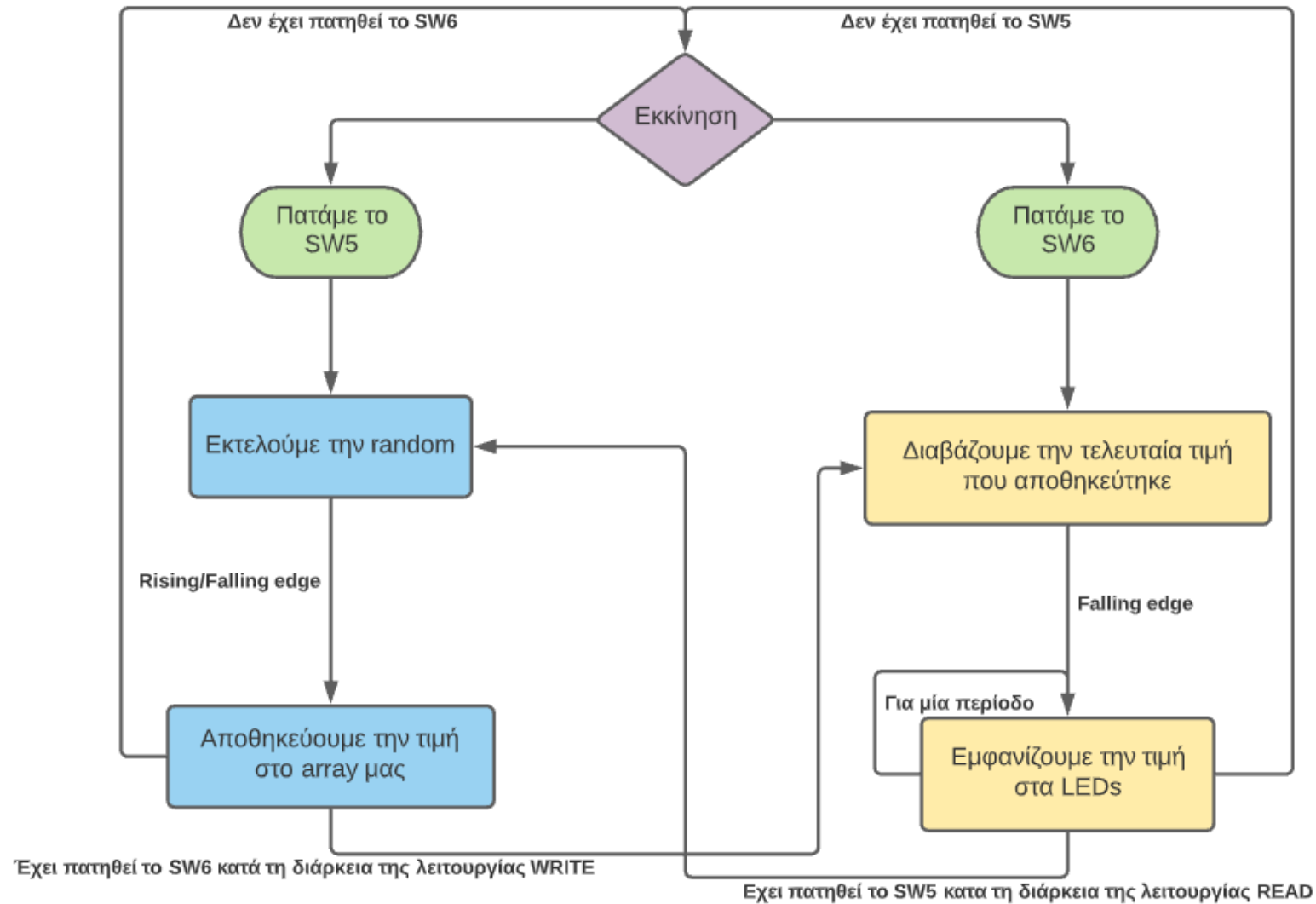
WRITE λειτουργία

- ▶ Εφόσον πατηθεί το switch, υπολογίζουμε την τυχαία τιμή μέσω μίας random και περιμένουμε να έρθει rising ή falling edge του παλμού, ώστε να την αποθηκεύσουμε στο array μας.
- ▶ BONUS:
 - ▶ Κάθε φορά που θα προστίθεται μια τιμή εκτελέστε έναν αλγόριθμο που τοποθετεί τους αριθμούς κατά αύξουσα σειρά.
 - ▶ Πόσους παλμούς του PWM χρειάζεται κατά μέσο όρο για να ολοκληρωθεί;

READ λειτουργία

- ▶ Εφόσον πατηθεί το switch, θέλουμε να διαβαστεί η τελευταία τιμή που καταχωρήθηκε στη μνήμη μας.
- ▶ Θέλουμε η τιμή να εμφανιστεί στα τέσσερα LEDs (PIN0, PIN1, PIN2, PIN3). Για παράδειγμα, εάν η τελευταία τιμή ήταν 5, θέλουμε τα LED να πάρουν την τιμή 1010 (θυμηθείτε πως με την τιμή 1 το LED σβήνει και με την τιμή 0 ανάβει).
- ▶ Επομένως, οι τιμές που θα έχουμε στη μνήμη μας θα είναι μέχρι 4 bit.
- ▶ Θέλουμε αυτήν την τιμή να μας την εμφανίζει για μία ολόκληρη περίοδο του αντίστοιχου παλμού.

Διάγραμμα ροής



Παραδοτέα

- ▶ Παραδίδεται αναλυτική αναφορά με τη λειτουργία του κώδικά σας.
- ▶ Επίσης, παραδίδεται τον κώδικά σας με αναλυτικά σχόλια.
- ▶ Τέλος, στην αναφορά σας σχεδιάστε τους παλμούς που έχετε δημιουργήσει, εξηγήστε πως προέκυψαν και γιατί επιλέξατε αυτόν τον τρόπο υλοποίησης.