



C.E.I.D.

Εργαστήριο Προηγμένων Μικροεπεξεργαστών

Αναφορά 1ης Εργαστηριακής Άσκησης

ΒΑΖΑΙΟΣ ΣΤΥΛΙΑΝΟΣ Α.Μ. 1054284

ΚΑΨΑΛΗΣ ΡΩΜΑΝΟΣ Α.Μ. 1056289

Ομάδα Α - Τμήμα 3 (Τρίτη 16:00-18:00)

2020

Περιεχόμενα

1. Αρχικός Κώδικας Άσκησης 1	3
2. Διορθώσεις Και Αιτιολόγηση	7
2.1 Λογικά λάθη.....	7
2.2. Συντακτικά λάθη.....	9
3. Τελικός Κώδικας Με Διορθώσεις	10

1. Αρχικός Κώδικας Άσκησης 1

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <unistd.h>
#include <sys/mman.h>
#include <stdio.h>
#include <stdlib.h>
#include <header.h>
```

```
#define PIOA_ID 2
#define TCO_ID 17
#define BUT_IDLE 0
#define BUT_PRESSED 1
#define BUT_RELEASED 2
#define LED_IDLE 0
#define LED_FLASHING 1
```

```
void FIQ_handler(void);
```

```
PIO* pioa = NULL;
```

```
AIC* aic = NULL;
```

```
TC* tc = NULL;
```

```
unsigned int button_state = BUT_IDLE;
```

```
unsigned int led_state = LED_IDLE;
```

```
int main( int argc, const char* argv[] ){
```

```
    unsigned int gen;
```

```
    STARTUP; //ΑΡΧΙΚΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ
```

```
    tc->Channel_0.RC = 8192; //ΠΕΡΙΟΔΟΣ 1 ΔΕΥΤΕΡΟΛΕΠΤΟ
```

```
    tc->Channel_0.CMR = 0x2084; // SLOW CLOCK , WAVEFORM , DISABLE CLK ON RC  
COMPARE
```

```
    tc->Channel_0.IDR = 0xFF; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ ΔΙΑΚΟΠΩΝ
```

```
    tc->Channel_0.IER = 0x10; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΜΟΝΟ ΤΟΥ RC COMPARE
```

```
    aic->FFER = (1<<PIOA_ID) | (1<<TC0_ID); //ΟΙ ΔΙΑΚΟΠΕΣ 2 ,17 ΕΙΝΑΙ ΤΥΠΟΥ FIQ
```

```
    aic->IECR = (1<<PIOA_ID) | (1<<TC0_ID); //ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΑΚΟΠΩΝ : PIOA & TC0
```

```
    pioa->PUER = 0x01; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΣΤΗ ΓΡΑΜΜΗ 0 : PULL-UP
```

```
    pioa->ODR = 0x01; //ΓΡΑΜΜΗ 0 : ΛΕΙΤΟΥΡΓΙΑ ΕΙΣΟΔΟΥ
```

```
    pioa->CODR = 0x02; //ΓΡΑΜΜΗ 1 : ΔΥΝΑΜΙΚΟ ΕΞΟΔΟΥ LOW
```

```
    pioa->OER = 0x02; //ΓΡΑΜΜΗ 1 : ΛΕΙΤΟΥΡΓΙΑ ΕΞΟΔΟΥ
```

```
    gen = pioa->ISR; // PIOA : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    pioa->PER = 0x03; //ΓΡΑΜΜΕΣ 0 , 1 : ΓΕΝΙΚΟΥ ΣΚΟΠΟΥ
```

```
    gen = tc->Channel_0.SR; //TC0 : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    aic->ICCR = (1<<PIOA_ID)|(1<<TC0_ID); // AIC : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    pioa->IER = 0x01; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΑΚΟΠΩΝ ΣΤΗ ΓΡΑΜΜΗ 0
```

```
while( (tmp = getchar()) != 'e'){  
}
```

```
aic->>IDCR = (1<<PIOA_ID) | (1<<TC0_ID); // ΔΙΑΚΟΠΗ ΤΩΝ AIC interrupts
```

```
tc->Channel_0.CCR = 0x02; // ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Timer
```

```
CLEANUP;
```

```
return 0;
```

```
}
```

```
void FIQ_handler(void)
```

```
{
```

```
    unsigned int data_in = 0;
```

```
    unsigned int fiq = 0;
```

```
    unsigned int data_out;
```

```
    fiq = aic->IPR; //ΕΝΤΟΠΙΣΜΟΣ ΠΕΡΙΦΕΡΕΙΑΚΟΥ ΠΟΥ ΠΡΟΚΑΛΕΣΕ ΤΗ ΔΙΑΚΟΠΗ
```

```
    if( fiq & (1<<PIOA_ID) ) //ΕΛΕΓΧΟΣ ΓΙΑ ΡΙΟΑ
```

```
    {
```

```
        data_in = pioa->ISR; //ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΠΗΓΗΣ ΤΗΣ ΔΙΑΚΟΠΗΣ
```

```
        aic->ICCR = (1<<PIOA_ID); //ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΔΙΑΚΟΠΗΣ ΑΠΟ AIC
```

```
        data_in = pioa->PDSR; //ΑΝΑΓΝΩΣΗ ΤΙΜΩΝ ΕΙΣΟΔΟΥ
```

```
        if( data_in & 0x01 ) //ΔΙΑΚΟΠΤΗΣ ΠΑΤΗΜΕΝΟΣ;
```

```
        {
```

```
            if(button_state == BUT_IDLE)
```

```
            {
```

```
                button_state = BUT_PRESSED;
```

```

        if( led_state == LED_IDLE ) //ΑΝ ΔΕΝ ΑΝΑΒΟΣΒΗΝΕΙ
        {
            tc->Channel_0.CCR = 0x05; //ΕΝΑΡΞΗ ΜΕΤΡΗΤΗ
            led_state = LED_FLASHING;
        }else{
            tc->Channel_0.CCR = 0x02; //ΔΙΑΚΟΠΗ ΜΕΤΡΗΤΗ
            led_state = LED_IDLE;
        }
    }
}

}

else{
    if(button_state == BUT_PRESSED) button_state = BUT_IDLE;
}
}

if( fiq & (1<<TC0_ID) )
{
    data_out = tc->Channel_0.SR; //ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΠΗΓΗΣ ΤΗΣ ΔΙΑΚΟΠΗΣ
    aic->ICCR = (1<<TC0_ID); //ΕΚΚΑΘΑΡΙΣΗ ΔΙΑΚΟΠΗΣ ΚΑΙ ΑΠΟ AIC
    data_out = pioa->ODSR; //ΑΝΑΓΝΩΣΗ ΤΙΜΩΝ ΕΞΟΔΟΥ
    pioa->SODR = data_out & 0x02;
    pioa->CODR = data_out & 0x02;
    tc->Channel_0.CCR = 0x05; //ΕΝΑΡΞΗ ΜΕΤΡΗΤΗ
}
}
}

```

2. Διορθώσεις Και Αιτιολόγηση

Στον κώδικα εντοπίσαμε 2 συντακτικά και 2 λογικά λάθη τα οποία εξηγούνται παρακάτω.

2.1 Λογικά λάθη

➤ 1° Λάθος: `if(data_in & 0x01)`

Διόρθωση: `if(!(data_in & 0x01))`

Αιτιολόγηση:

Σε αυτή την συνθήκη `if` χρειάζεται να συμπληρώσουμε τον τελεστή `!` (`not`) διότι όταν πατηθεί ο διακόπτης (`BUT_PRESSED`) έχουμε χαμηλό δυναμικό, οπότε `data_in = 0`. Άρα η συνθήκη της `if` είναι μη αληθής και επομένως δεν θα εκτελεστεί ο κώδικας εντός της `if`. Επομένως το `led` δεν θα αλλάζει κατάσταση. Αντιθέτως, θα αλλάξει κατάσταση μόνο όταν έχουμε αφήσει τον διακόπτη (`BUT_RELEASED`), διότι αλλάζει το δυναμικό, `data_in = 1`, και έτσι, η συνθήκη γίνεται αληθής. Εμείς, θέλουμε όταν πατάμε τον διακόπτη να αλλάζει κατάσταση άρα βάζουμε `!`.

➤ 2° Λάθος: `pioa->SODR = data_out & 0x02;`

Διόρθωση: `pioa->SODR = data_out | 0x02;`

Αιτιολόγηση:

Με βάση την θεωρία: Τα bits του καταχωρητή `PIO_ODSR` ενεργοποιούνται με την εγγραφή της τιμής 1 στην αντίστοιχη θέση του `PIO_SODR` και απενεργοποιούνται με την εγγραφή της τιμής 1 στην αντίστοιχη θέση του `PIO_CODR`. Για παράδειγμα, οι εντολές `pioa->PIO_SODR = 0x100;` και `pioa->PIO_CODR = 0x1;` θα θέσουν την έξοδο της 8ης γραμμής σε υψηλό δυναμικό και την έξοδο της γραμμής 0 σε χαμηλό δυναμικό.

Η εκφώνηση αναφέρει: «Ο ακροδέκτης 1 πρέπει να ρυθμιστεί σε λειτουργία εξόδου. Χαμηλό δυναμικό σε αυτόν τον ακροδέκτη συνεπάγεται σβήσιμο του LED. Υψηλό

δυναμικό αντίθετα συνεπάγεται διαφορά τάσης στα άκρα της φωτοδιόδου μεγαλύτερη των 0.7V και συνεπώς ενεργοποίηση της πηγής φωτός. Πέρα από το πάτημα του διακόπτη, θα προγραμματίσουμε το AT91 ώστε διακοπή να μπορεί να προκληθεί και από την εξάντληση της επιθυμητής μέτρησης από το μετρητή του συστήματος. Αν προκληθεί συνεπώς διακοπή από την ολοκλήρωση της μέτρησης του ενός δευτερολέπτου, η υπομονάδα εξόδου που ελέγχει τον ακροδέκτη στον οποίο είναι συνδεδεμένο το LED θα προγραμματιστεί έτσι ώστε να αντιστραφεί η κατάσταση του LED (από ενεργό να γίνει ανενεργό ή το αντίστροφο)».

Στην περίπτωση του αρχικού κώδικα με $\text{pioa} \rightarrow \text{SODR} = \text{data_out} \& 0x02;$

Έχουμε $\text{data_out} = \text{pioa} \rightarrow \text{ODSR}$ άρα διαβάζω τιμές εξόδου

Για το **SODR** $\text{data_out} = X3 X2 X1 X0 \& 0010 = 0 0 X1 0$

Άρα διατηρεί την κατάσταση της 1^{ης} γραμμής .

Για το **CODR** $\text{data_out} = X3 X2 X1 X0 \& 0010 = 0 0 X1 0$

Αν ήταν σβηστό ($X1=0$) παραμένει σβηστό, αν ήταν αναμμένο ($X1=0$) παραμένει αναμμένο. Όπως γίνεται αντιληπτό δεν είναι αυτό που ζητείται, επειδή θέλουμε το led να αναβοσβήνει.

Στην περίπτωση του διορθωμένου κώδικα με $\text{pioa} \rightarrow \text{SODR} = \text{data_out} | 0x02;$

Για το **SODR** $\text{data_out} = X3 X2 X1 X0 | 0010 = 0 0 1 0$

Άρα θέτει την κατάσταση της 1^{ης} γραμμής σε υψηλό δυναμικό (ανάβει το led)

Για το **CODR** $\text{data_out} = X3 X2 X1 X0 \& 0010 = 0 0 X1 0$

Άρα αν $X1=1$ σβήνει το led => 0010 και αν $X1=0$ => 0000 σε συνδυασμό και με την τιμή του SODR το ανάβει

Τελικά έχουμε: αν το led ήταν αναμμένο, σβήνει και αν ήταν σβηστό το led ανάβει! Δηλαδή αυτό που ζητείται.

2.2. Συντακτικά λάθη

Στη συνθήκη while:

```
while( (tmp = getchar()) != 'e'){  
    }
```

- **1° Λάθος:** Γίνεται χρήση της μεταβλητής της tmp χωρίς να έχει δηλωθεί προηγουμένως.

Διόρθωση: Η λύση είναι να την δηλώσουμε πριν το κάλεσμα της ως εξής:
char tmp;

- **2° Λάθος:** Ακόμα ο compiler του AT91 για κάποιο λόγο μας βγάζει error στην πρώτη αγκύλη της συνθήκης της while.

Διόρθωση: Η λύση του συγκεκριμένου λάθους είναι να τοποθετήσουμε αυτή την αγκύλη σε νέα γραμμή του κώδικα. Έτσι μετά την διόρθωση αυτών των λαθών ο κώδικας θα είναι:

```
char tmp;  
  
while( (tmp = getchar()) != 'e')  
{  
    }
```

3. Τελικός Κώδικας Με Διορθώσεις

Άσκηση 1

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <unistd.h>
#include <sys/mman.h>
#include <stdio.h>
#include <stdlib.h>
#include <header.h>
```

```
#define PIOA_ID 2
#define TCO_ID 17
#define BUT_IDLE 0
#define BUT_PRESSED 1
#define BUT_RELEASED 2
#define LED_IDLE 0
#define LED_FLASHING 1
void FIQ_handler(void);
PIO* pioa = NULL;
AIC* aic = NULL;
TC* tc = NULL;
```

```
unsigned int button_state = BUT_IDLE;
```

```
unsigned int led_state = LED_IDLE;
```

```
int main( int argc, const char* argv[] ){
```

```
    char tmp;
```

```
    unsigned int gen;
```

```
    STARTUP; //ΑΡΧΙΚΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ
```

```
    tc->Channel_0.RC = 8192; //ΠΕΡΙΟΔΟΣ 1 ΔΕΥΤΕΡΟΛΕΠΤΟ
```

```
    tc->Channel_0.CMR = 0x2084; // SLOW CLOCK , WAVEFORM , DISABLE CLK ON RC  
    COMPARE
```

```
    tc->Channel_0.IDR = 0xFF; //ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΟΛΩΝ ΤΩΝ ΔΙΑΚΟΠΩΝ
```

```
    tc->Channel_0.IER = 0x10; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΜΟΝΟ ΤΟΥ RC COMPARE
```

```
    aic->FFER = (1<<PIOA_ID) | (1<<TC0_ID); //ΟΙ ΔΙΑΚΟΠΕΣ 2 ,17 ΕΙΝΑΙ ΤΥΠΟΥ FIQ
```

```
    aic->IECR = (1<<PIOA_ID) | (1<<TC0_ID); //ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΑΚΟΠΩΝ : PIOA & TC0
```

```
    pioa->PUER = 0x01; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΣΤΗ ΓΡΑΜΜΗ 0 : PULL-UP
```

```
    pioa->ODR = 0x01; //ΓΡΑΜΜΗ 0 : ΛΕΙΤΟΥΡΓΙΑ ΕΙΣΟΔΟΥ
```

```
    pioa->CODR = 0x02; //ΓΡΑΜΜΗ 1 : ΔΥΝΑΜΙΚΟ ΕΞΟΔΟΥ LOW
```

```
    pioa->OER = 0x02; //ΓΡΑΜΜΗ 1 : ΛΕΙΤΟΥΡΓΙΑ ΕΞΟΔΟΥ
```

```
    gen = pioa->ISR; // PIOA : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    pioa->PER = 0x03; //ΓΡΑΜΜΕΣ 0 , 1 : ΓΕΝΙΚΟΥ ΣΚΟΠΟΥ
```

```
    gen = tc->Channel_0.SR; //TC0 : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    aic->ICCR = (1<<PIOA_ID)|(1<<TC0_ID); // AIC : ΕΚΚΑΘΑΡΙΣΗ ΑΠΟ ΤΥΧΟΝ ΔΙΑΚΟΠΕΣ
```

```
    pioa->IER = 0x01; //ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΑΚΟΠΩΝ ΣΤΗ ΓΡΑΜΜΗ 0
```

```
    while( (tmp = getchar()) != 'e')
```

```
    {
```

```
}
```

```
aic->>IDCR = (1<<PIOA_ID) | (1<<TC0_ID); // ΔΙΑΚΟΠΗ ΤΩΝ AIC interrupts
```

```
tc->Channel_0.CCR = 0x02; // ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Timer
```

```
CLEANUP;
```

```
return 0;
```

```
}
```

```
void FIQ_handler(void)
```

```
{
```

```
    unsigned int data_in = 0;
```

```
    unsigned int fiq = 0;
```

```
    unsigned int data_out;
```

```
    fiq = aic->IPR; //ΕΝΤΟΠΙΣΜΟΣ ΠΕΡΙΦΕΡΕΙΑΚΟΥ ΠΟΥ ΠΡΟΚΑΛΕΣΕ ΤΗ ΔΙΑΚΟΠΗ
```

```
    if( fiq & (1<<PIOA_ID) ) //ΕΛΕΓΧΟΣ ΓΙΑ ΡΙΟΑ
```

```
    {
```

```
        data_in = pioa->ISR; //ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΠΗΓΗΣ ΤΗΣ ΔΙΑΚΟΠΗΣ
```

```
        aic->ICCR = (1<<PIOA_ID); //ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΔΙΑΚΟΠΗΣ ΑΠΟ AIC
```

```
        data_in = pioa->PDSR; //ΑΝΑΓΝΩΣΗ ΤΙΜΩΝ ΕΙΣΟΔΟΥ
```

```
        if( !(data_in & 0x01) ) //ΔΙΑΚΟΠΤΗΣ ΠΑΤΗΜΕΝΟΣ;
```

```
        {
```

```
            if(button_state == BUT_IDLE)
```

```
            {
```

```
                button_state = BUT_PRESSED;
```

```
                if( led_state == LED_IDLE ) //ΑΝ ΔΕΝ ΑΝΑΒΟΣΒΗΝΕΙ
```

```
                {
```

```
                    tc->Channel_0.CCR = 0x05; //ΕΝΑΡΞΗ ΜΕΤΡΗΤΗ
```

```

        led_state = LED_FLASHING;
    }else{
        tc->Channel_0.CCR = 0x02; //ΔΙΑΚΟΠΗ ΜΕΤΡΗΤΗ
        led_state = LED_IDLE;
    }
}
}else{
    if(button_state == BUT_PRESSED) button_state = BUT_IDLE;
}
}

if( fiq & (1<<TC0_ID) )
{
    data_out = tc->Channel_0.SR;//ΕΚΚΑΘΑΡΙΣΗ ΤΗΣ ΠΗΓΗΣ ΤΗΣ ΔΙΑΚΟΠΗΣ
    aic->ICCR = (1<<TC0_ID); //ΕΚΚΑΘΑΡΙΣΗ ΔΙΑΚΟΠΗΣ ΚΑΙ ΑΠΟ AIC
    data_out = pioa->ODSR; //ΑΝΑΓΝΩΣΗ ΤΙΜΩΝ ΕΞΟΔΟΥ
    pioa->SODR = data_out | 0x02;
    pioa->CODR = data_out & 0x02;
    tc->Channel_0.CCR = 0x05; //ΕΝΑΡΞΗ ΜΕΤΡΗΤΗ
}
}

```

Τέλος Αναφοράς Πρώτης Εργαστηριακής Άσκησης

ΒΑΖΑΙΟΣ ΣΤΥΛΙΑΝΟΣ Α.Μ. 1054284

ΚΑΨΑΛΗΣ ΡΩΜΑΝΟΣ Α.Μ. 1056289

Τμήμα Α-3