Στοιχεία Φοιτητών Ομάδας:

Δημήτριος Κωστορρίζος ΑΜ: 1054419

Λάμπρος Παπαδόπουλος ΑΜ: 1054433

Header.h File

```
typedef volatile struct _PIO
{
        unsigned int PER; /** PIO Enable r e g i s t e r ( Wr i te−only ) */
        unsigned int PDR; /** PIO Di sabl e r e g i s t e r ( Wr i te−only ) */
        unsigned int PSR; /** PIO Status r e g i s t e r ( Read−only ) */
        unsigned int Reserved0; /** Unused r e g i s t e r */
        unsigned int OER; /** Output Enable r e g i s t e r ( Wr i te−only ) */
        unsigned int ODR; /** Output Di sabl e r e g i s t e r ( Wr i te−only ) */
        unsigned int OSR; /** Output Status r e g i s t e r ( Read−only ) */
        unsigned int Reserved1; /** Unused r e g i s t e r */
        unsigned int IFER; /** Gl i t c h Input f i l t e r Enable ( Wr i te−only ) */
        unsigned int IFDR; /** Gl i t c h Input f i l t e r Di sabl e ( Wr i te−only ) */
        unsigned int IFSR; /** Gl i t c h Input f i l t e r Status ( Read−only ) */
        unsigned int Reserved2; /** Unused r e g i s t e r */
        unsigned int SODR; /** Set Output Data r e g i s t e r ( Wr i te−only ) */
        unsigned int CODR; /** Cl ear Output Data r e g i s t e r ( Wr i te−only ) */
        unsigned int ODSR; /** Output Data Status r e g i s t e r ( Read−only ) */
        unsigned int PDSR; /** Pin Data Status r e g i s t e r ( Read−only ) */
        unsigned int IER; /** I n t e r r u p t Enable r e g i s t e r ( Wr i te−only ) */
        unsigned int IDR; /** I n t e r r u p t Di sabl e r e g i s t e r ( Wr i te−only ) */
        unsigned int IMR; /** I n t e r r u p t Mask r e g i s t e r ( Wr i te−only ) */
        unsigned int ISR; /** I n t e r r u p t Status r e g i s t e r ( Read−only ) */
        unsigned int MDER; /** Mul t i−d r i v e r Enable ( Wr i te−only ) */
```

```
    unsigned int MDDR; /** Mul t i−d r i v e r Di sabl e ( Wr i te−only ) */
    unsigned int MDSR; /** Mul t i−d r i v e r Status ( Read−only ) */
    unsigned int Reserved3; /** Unused r e g i s t e r */
    unsigned int PUDR; /** Pul l−up Di sabl e r e g i s t e r ( Wr i te−only ) */
    unsigned int PUER; /** Pul l−up Enable r e g i s t e r ( Wr i te−only ) */
    unsigned int PUSR; /** Pul l−up Status r e g i s t e r ( Read−only ) */
    unsigned int Reserved4; /** Unused r e g i s t e r */
    unsigned int ASR; /** P e r i p h e r a l A s e l e c t ( Wr i te−only ) */
    unsigned int BSR; /** P e r i p h e r a l B s e l e c t ( Wr i te−only ) */
    unsigned int ABSR; /** P e r i p h e r a l AB Status ( Read−only ) */
    unsigned int Reserved5[9]; /** Unused r e g i s t e r */
    unsigned int OWER; /** Output wr i t e enable ( Wr i te−only ) */
    unsigned int OWDR; /** Output wr i t e di sabl e ( Wr i te−only ) */
    unsigned int OWSR; /** Output wr i t e Status ( Read−only ) */
}PIO;

typedef volatile struct _AIC
{
    unsigned int SMR[32]; /** Source mode r e g i s t e r ( Read−Wr i t e ) */
    unsigned int SVR[32]; /** Source v e c t o r r e g i s t e r ( Read−Wr i t e ) */
    unsigned int IVR; /** I n t e r r u p t v e c t o r r e g i s t e r ( Read−only ) */
    unsigned int FVR; /** Fast I n t e r r u p t v e c t o r r e g i s t e r ( Read−only ) */
    unsigned int ISR; /** I n t e r r u p t s tatus r e g i s t e r ( Read−only ) */
    unsigned int IPR; /** I n t e r r u p t pending r e g i s t e r ( Read−only ) */
    unsigned int IMR; /** I n t e r r u p t mask r e g i s t e r ( Read−only ) */
    unsigned int CISR; /** Core I n t e r r u p t s tatus r e g i s t e r ( Read−only ) */
    unsigned int Reserved1[2]; /** Unused r e g i s t e r */
    unsigned int IECR; /** I n t e r r u p t enable command r e g i s t e r ( Wr i te−only ) */
    unsigned int IDCR; /** I n t e r r u p t di sabl e command r e g i s t e r ( Wr i te−only ) */
```

```
        unsigned int ICCR; /** I n t e r r u p t c l e a r command r e g i s t e r ( Wr i te−only ) */
        unsigned int ISCR; /** I n t e r r u p t s e t command r e g i s t e r ( Wr i te−only ) */
        unsigned int EICR; /** End o f I n t e r r u p t command r e g i s t e r ( Wr i te−only ) */
        unsigned int SPUR; /** Spur ious I n t e r r u p t v e c t o r r e g i s t e r ( Read−Wr i t e ) */
        unsigned int DCR; /** Debug c o n t r o l r e g i s t e r ( Read−Wr i t e ) */
        unsigned int Reserved2; /** Unused r e g i s t e r */
        unsigned int FFER; /** Fast Fo r c ing enable r e g i s t e r ( Wr i te−only ) */
        unsigned int FFDR; /** Fast Fo r c ing di sabl e r e g i s t e r ( Wr i te−only ) */
        unsigned int FFSR; /** Fast Fo r c ing s tatus r e g i s t e r ( Wr i te−only ) */
}AIC;

typedef volatile struct _TCCHAN
{
        unsigned int CCR; /** Channel Cont r o l Re g i s t e r ( Wr i te−only ) */
        unsigned int CMR; /** Channel Mode Re g i s t e r ( Read−Wr i t e ) */
        unsigned int Reserved1[2]; /** Unused r e g i s t e r */
        unsigned int CV; /** Counter Value ( Read−only ) */
        unsigned int RA; /** Re g i s t e r A ( Read−Wr i t e ) */
        unsigned int RB; /** Re g i s t e r B ( Read−Wr i t e ) */
        unsigned int RC; /** Re g i s t e r C ( Read−Wr i t e ) */
        unsigned int SR; /** Status Re g i s t e r ( Read−only ) */
        unsigned int IER; /** I n t e r r u p t Enable Re g i s t e r ( Wr i te−only ) */
        unsigned int IDR; /** I n t e r r u p t Di sabl e Re g i s t e r ( Wr i te−only ) */
        unsigned int IMR; /** I n t e r r u p t Mask Re g i s t e r ( Read−only ) */
        unsigned int Reserved2[4]; /** Unused r e g i s t e r */
}TCCHAN;

typedef volatile struct _TC
{
```

```
        TCCHAN Channel_0;
        TCCHAN Channel_1;
        TCCHAN Channel_2;
        unsigned int BCR; /** Block Cont r o l Re g i s t e r ( Wr i te−only ) */
        unsigned int BMR; /** Block Mode Re g i s t e r ( Read−Wr i t e ) */
}TC;

/// ΕΝΑΡΞΗ ΑΡΧΙΚΟΠΟΙΗΣΗΣ ΣΥΣΤΗΜΑΤΟΣ
/// ΠΡΕΠΕΙ ΝΑ ΓΙΝΕΤΑΙ ΠΑΝΤΑ ΣΤΗΝ ΑΡΧΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ
#define STARTUP \
unsigned int _FIQtmp; \
/* ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΠΡΟΣΠΕΛΑΣΗ ΜΝΗΜΗΣ ΣΥΣΤΗΜΑΤΟΣ */ \
int fd = open("/dev/mem", O_RDWR | O_SYNC); \
/* ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΕΓΓΡΑΦΗ ΡΟΥΤΙΝΑΣ ΕΞΥΠΗΡΕΣΗΣ ΔΙΑΚΟΠΩΝ */ \
int fd2 = open("/proc/FIQ", O_RDWR | O_SYNC); \
/* ΕΝΕΡΓΟΠΟΙΗΣΗ ΔΙΚΑΙΩΜΑΤΩΝ ΠΡΟΣΠΕΛΑΣΗΣ ΣΤΗΝ*/ /** PIOA & AIC*/ \
char* mptr = mmap(0, 0x1000, PROT_READ | PROT_WRITE, \
MAP_SHARED, fd, 0xFFFFF000); \
/* ΑΠΑΡΑΙΤΗΤΟ ΓΙΑ ΠΡΟΣΠΕΛΑΣΗ ΤΩΝ*/ /**TIMERS*/ \
char* pptr = mmap(0, 0x1000, PROT_READ | PROT_WRITE, \
MAP_SHARED, fd, 0xFFFA0000); \
/* ΕΛΕΓΧΟΣ ΑΠΟΚΤΗΣΗΣ ΔΙΚΑΙΩΜΑΤΩΝ */ \
if(mptr == MAP_FAILED || pptr == MAP_FAILED){ \
close(fd); \
close(fd2); \
exit(1); \
} \
/* ΟΡΙΖΟΥΜΕ ΔΙΕΥΘΥΝΣΗ ΤΗΣ*/ /**PIOA*/ \
pioa = (PIO*)(mptr + 0x400); \
```

```
/* ΟΡΙΖΟΥΜΕ ΔΙΕΥΘΥΝΣΗ ΤΗΣ*/ /**AIC*/ \
aic = (AIC*)(mptr); \
/* ΟΡΙΖΟΥΜΕ ΔΙΕΥΘΥΝΣΗ ΤΟΥ*/ /**TC*/ \
tc = (TC*)(pptr); \
_FIQtmp = (unsigned int)FIQ_handler; \
write(fd2, &_FIQtmp, sizeof(unsigned int)); \
_FIQtmp = fcntl(STDIN_FILENO, F_GETFL, 0); \
_FIQtmp |= O_NONBLOCK; \
fcntl(STDIN_FILENO, F_SETFL, _FIQtmp)
/// ΛΗΞΗ ΑΡΧΙΚΟΠΟΙΗΣΗΣ
/// ΕΝΑΡΞΗ ΕΠΑΝΑΦΟΡΑΣ ΣΥΣΤΗΜΑΤΟΣ
/// ΠΡΕΠΕΙ ΝΑ ΓΙΝΕΤΑΙ ΠΑΝΤΑ ΣΤΟ ΤΕΛΟΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ
#define CLEANUP \
_FIQtmp = fcntl(STDIN_FILENO, F_GETFL, 0); \
_FIQtmp &= O_NONBLOCK; \
fcntl(STDIN_FILENO, F_SETFL, _FIQtmp); \
/* ΕΠΙΣΤΡΟΦΗ ΔΕΣΜΕΥΜΕΝΗΣ ΜΝΗΜΗΣ */ \
munmap(mptr, 0x1000); \
munmap(pptr, 0x1000); \
close(fd); \
close(fd2)
/// ΛΗΞΗ ΕΠΑΝΑΦΟΡΑΣ ΣΥΣΤΗΜΑΤΟΣ
#define DISABLE_FIQ \
{ \
unsigned int __Reg_save; \
asm("mrs %0, cpsr;" \
"orr %0, %0, #0x40;" \
"msr cpsr_c, %0;" \
:"=r" (__Reg_save) \
```

```
:"r"(__Reg_save) \
);}
#define ENABLE_FIQ \
{ \
unsigned int __Reg_save; \
asm("mrs %0, cpsr;" \
"bic %0, %0, #0x40;" \
"msr cpsr_c, %0;" \
:"=r" (__Reg_save) \
:"r"(__Reg_save) \
);}
```

Askisi1.c File

```
#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 TC0_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;
// Θέσε την αρχική κατάσταση του κουμπιού σε μη πατημένο
unsigned int button_state = BUT_IDLE;
// Θέσε την αρχική κατάσταση του λαμπακίου σε σβηστό
unsigned int led_state = LED_IDLE;
// Όρισε μια μη απαιτούμενη μεταβλητή tmp
unsigned int tmp;
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 = 2084; // 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
	// Διάβασε χαρακτήρες από το πληκτρολόγιο μέχρι να εισαχθεί ο χαρακτήρας e...
	while( (tmp = getchar()) != 'e')
	{
	}
	// Εκτέλεσε την διαδικασία τερματισμού όταν εισαχθεί ο χαρακτήρας e
	aic->IDCR = (1<<PIOA_ID) | (1<<TC0_ID); // ΔΙΑΚΟΠΗ ΤΩΝ AIC i n t e r r u p t s
	tc->Channel_0.CCR = 0x02; // ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ ΤΟΥ Time r
	CLEANUP;
	return 0;
}

void FIQ_handler(void)
{
	// Θέσε τον καταχωρητή δεδομένων εισόδου σε 0
	unsigned int data_in = 0;
	// Θέσε την κατάσταση διακοπής σε 0
	unsigned int fiq = 0;
	// Όρισε ένα καταχωρητή δεδομένων  εξ΄δου
	unsigned int data_out;
	fiq = aic->IPR; //ΕΝΤΟΠΙΣΜΟΣ ΠΕΡΙΦΕΡΕΙΑΚΟΥ ΠΟΥ ΠΡΟΚΑΛΕΣΕ ΤΗ ΔΙΑΚΟΠΗ
	if( fiq & (1<<PIOA_ID) )
	{ //ΕΛΕΓΧΟΣ ΓΙΑ PIOA
		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)
                {
                        // Θέσε την κατάσταση του κουμπιού ως πατημένο
                        button_state = BUT_PRESSED;
                        // Αν το λαμπάκι είναι σβηστό
                        if( led_state == LED_IDLE )
                        { //ΑΝ ΔΕΝ ΑΝΑΒΟΣΒΗΝΕΙ
                                tc->Channel_0.CCR = 0x05; //ΕΝΑΡΞΗ ΜΕΤΡΗΤΗ

                                // Θέσε την κατάσταση του λαμπακίου ως φωτεινό
                                led_state = LED_FLASHING;
                        }
                        else
                        {
                                tc->Channel_0.CCR = 0x02; //ΔΙΑΚΟΠΗ ΜΕΤΡΗΤΗ
                                // Θέσε την κατάσταση του λαμπακίου ως σβηστό
                                led_state = LED_IDLE;
                        }
                }
        }
        else
        {
                // Αν το κουμπί είναι πατημένο
                if(button_state == BUT_PRESSED)
                        // Θέσε την κατάσταση του κουμπιού ως μη πατημένο
                        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
                // Ενεροποίησε μόνο το δεύτερο bit στον καταχωρητή SODR
                pioa->SODR = data_out & 0x02;
                // Ενεροποίησε μόνο το δεύτερο bit στον καταχωρητή CODR
                pioa->CODR = data_out & 0x02;
                //Έναρξη μετρητή
                tc->Channel_0.CCR = 0x05;
        }
}
```

Εκτέλεση Πειράματος

Αρχικά, συνδέσαμε το λαμπάκι, το κουμπί και την αντίσταση με το ολοκληρωμένο κύκλωμα AT91 και την γείωση, βασιζόμενοι στην συνδεσμολογία που παρουσιάζεται στην Άσκηση 1.

Έπειτα, εκτελέσαμε τις παρακάτω εντολές αποσφαλμάτωσης με σκοπό να ελέγξουμε την συνδεσμολογία και την αναμενόμενη λειτουργία του κυκλώματος.

mw 0 0xFFFFF400 0x02
mw 0 0xFFFFF410 0x02
mw 0 0xFFFFF430 0x02
mw 0 0xFFFFF434 0x02

mw 0 0xFFFFF400 0x01
mw 0 0xFFFFF414 0x01
mw 0 0xFFFFF464 0x01
md 0 0xFFFFF43C 1

Μετά την επιτυχή αποσφαλμάτωση, διορθώσαμε τα compile error που υπήρχαν στον κώδικα. Πιο συγκεκριμένα, αλλάξαμε τους χαρακτήρες ‘e’ σε 'e' και ορίσαμε την μεταβλητή tmp, ωστόσο θα μπορούσαμε να την αφαιρέσουμε πλήρως, χωρίς να εμφανιστεί κάποιο πρόβλημα κατά την εκτέλεση του κώδικα. Με την ολοκλήρωση των αλλαγών αυτών, κάναμε compile το αρχείο Askisi1.c και εκτελέσαμε τον εκτελέσιμο αρχείο.

Κατά την εκτέλεση του κώδικα, πατήσαμε το κουμπί και κατά όταν σταματήσαμε να ασκούμε πίεση, παρατηρήσαμε το λαμπάκι να ανάβει.

Προκειμένου, το λαμπάκι να ανάβει με το πάτημα του κουμπιού, ήταν αναγκαία η παρακάτω αλλαγή:

if(data_in & 0x01) σε if(!data_in & 0x01)

Κατά την εκτέλεση του αλλαγμένου κώδικα, το πάτημα του κουμπιού ανάβει το λαμπάκι.

Η τελευταία αλλαγή που εφαρμόσαμε στον κώδικα ήταν η παρακάτω:

if(data_in & 0x01) σε if(data_in | 0x01)

Με την παρακάτω αλλαγή, το δεύτερο bit είναι πάντα 1 οπότε με ένα πάτημα του κουμπιού, το λαμπάκι παραμένει αναμμένο.