



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΟΛΟΓΙΑΣ & ΑΡΧΙΤΕΚΤΟΝΙΚΗΣ

ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΩΝ

5^Η ΑΣΚΗΣΗ

04/06/2021

Αλέξανδρος Νατσολάρι ΑΜ: 1057769

Παναγιώτης Μπαρμπούνης ΑΜ: 1054382

Παραδοχές :

- Στο LED0 βλέπουμε τις λειτουργίες του θερμοσίφωνα όπως ορίζονται από το διάγραμμα ροής.
- Στο LED1 βλέπουμε τις λειτουργίες του ανεμιστήρα όπως ορίζονται από το διάγραμμα ροής.
- Στο LED2 βλέπουμε τις λειτουργίες του water leakage όπως ορίζονται από το διάγραμμα ροής.
- Στο LED3 θα φαίνονταν οι λειτουργίες του κλειδώματος σπιτιού αλλά λόγω περιορισμένου χρόνου δεν προλάβουμε να υλοποιήσουμε το διάγραμμα ροής του κλειδώματος σπιτιού. Ωστόσο το έχουμε προβλέψει στον κώδικα ώστε στο διάγραμμα ροής επιλογής συσκευής να εμφανίζεται σαν επιλογή.
- Το ped το βάλαμε 120 γιατί σύμφωνα με τον τύπο στις διαφάνειες και μετρήσεις που κάναμε υπολογίσαμε ότι το timer λειτουργεί για περίπου 15 δευτερόλεπτα.

Πείραμα

Ο κώδικας της άσκησης είναι ο ακόλουθος:

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

#define ped 120

int SW5=0;
int SW6=0;
//An InitialStateFan=0 anemhsthras kleistos alliws an InitialStateFan=1 anemhsthras ανοικτος
int InitialStateFan=0;
int LED1=0;
//An InitialStateWLeakage=0 to water leakage apenergopoihmeno alliws an InitialStateWLeakage=1 to water leakage einai energopoihmeno
int InitialStateWLeakage=0;
//An InitialStateBoiler=0 o thermoshfonas einai kleistos alliws an InitialStateBoiler=1 o thermoshfonas einai ανοικτος
int InitialStateBoiler=0;
int interr=0;
int main() {
    PORTD.DIR |= PIN2_bm; //LED2 gia water leakage

    //initialize the ADC for Free-Running mode
    ADC0.CTRLA |= ADC_RESSEL_10BIT_gc; //10-bit resolution
    ADC0.CTRLA |= ADC_FREERUN_bm; //Free-Running mode enabled
    ADC0.CTRLA |= ADC_ENABLE_bm; //Enable ADC
    ADC0.MUXPOS |= ADC_MUXPOS_AIN7_gc; //The bit
    //Enable Debug Mode
    ADC0.DBGCTRL |= ADC_DBGRUN_bm;
    //Window Comparator Mode
    ADC0.WINLT |= 10; //Set threshold
    ADC0.INTCTRL |= ADC_WCMP_bm; //Enable Interrupts for WCM
    ADC0.CTRLE |= ADC_WINCM0_bm; //Interrupt when RESULT < WINLT
```

```

sei();
ADC0.COMMAND |= ADC_STCONV_bm; //Start Conversion

//pullup enable and Interrupt enabled with sense on both edges
PORTF.PIN5CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc;
//pullup enable and Interrupt enabled with sense on both edges
PORTC.PIN6CTRL |= PORT_PULLUPEN_bm | PORT_ISC_BOTHEDGES_gc;
//LED palmnwn
PORTD.DIR |= PIN1_bm; //LED1 gia anemhsthra

PORTD.DIR |= PIN0_bm; //LED0 gia thermoshfonas

sei(); //begin accepting interrupt signals

sei(); //enable interrupts
while(1){
    while (SW5==0) { //boiler
        if (SW6==1)
        {
            if (InitialStateBoiler==0){
                //Afou InitialStateBoiler=0 o thermoshfonas einai kleistos ara:
                InitialStateBoiler=1;
                PORTD.OUT |= PIN0_bm; //Anoikse to LED0
                TCA0.SINGLE.CNT = 0; //clear counter
                TCA0.SINGLE.CTRLB = 0; //Normal Mode (TCA_SINGLE_WGMODE_NORMAL_gc
                TCA0.SINGLE.CMP0 = ped; //When reaches this value -> interrupt CLOCK

                TCA0.SINGLE.CTRLA = TCA_SINGLE_CLKSEL_DIV1024_gc; //(= 0x7<<1 )
                TCA0.SINGLE.CTRLA |= 1; //Enable
                TCA0.SINGLE.INTCTRL = TCA_SINGLE_CMP0_bm; //Interrupt Enable (=0x10)

            }
            else if(InitialStateBoiler==1){
                //Afou InitialStateBoiler=1 o thermoshfonas einai anoiktos ara:
                PORTD.OUTCLR = PIN0_bm; //Kleise to LED0
                InitialStateBoiler=0;
                TCA0.SINGLE.CTRLC = 0; //Disable
            }
            SW6=0;
            SW5++;
        }
    }
    sei();
    while (SW5==1) { //fan

        if (SW6==1)
        {

            if (InitialStateFan==0){
                //Afou InitialStateFan=0 o anemhsthra einai kleistos ara:
                InitialStateFan=1;

```

FREQUENCY/1024

```

        //prescaler=1024
        TCA0.SINGLE.CTRLA=TCA_SINGLE_CLKSEL_DIV1024_gc;
        TCA0.SINGLE.PER = 254; //select the resolution
        TCA0.SINGLE.CMP1 = 90; //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;

        TCA0.SINGLE.INTCTRL |= TCA_SINGLE_CMP1_bm;

        TCA0.SINGLE.CTRLA |= TCA_SINGLE_ENABLE_bm; //Enable

    }
    else if(InitialStateFan==1){
        //Afou InitialStateFan=1 o anemhsthra einai anoiktos ara:
        PORTD.OUTCLR = PIN1_bm; //Kleise to LED1
        LED1=0;
        InitialStateFan=0;
        TCA0.SINGLE.CTRLC = 0x00; //Disable
    }

    SW6=0;
    SW5++;
}

}
sei();
while (SW5==2) { //water leakage

    if (SW6==1)
    {
        if (InitialStateWLeakage==0)
        {
            //synexizei na einai sbhsto to LED2
        }
        else if(InitialStateWLeakage==1)
        {
            PORTD.OUTCLR = PIN2_bm; //Kleise to LED2
            InitialStateWLeakage=0;
        }
        SW6=0;
        SW5++;
    }

}

}
sei();
while (SW5==3) { //house lock

    if (SW6==1)
    {

        SW6=0;
        SW5=0;
    }

}

```

```

    }
}

ISR(PORTF_PORT_vect){//interrupt gia to SWITCH5

    //clear the interrupt flag
    int intflags = PORTF.INTFLAGS;
    PORTF.INTFLAGS=intflags;
    if (interr==1)
    {
        PORTD.OUT |= PIN0_bm;//Anoikse to LED0
        interr=0;
    }

    if (SW5==3)
    {
        SW5=0;
    }
    else{
        SW5++;
    }
}

ISR(PORTC_PORT_vect){//interrupt gia to SWITCH6
    //clear the interrupt flag
    int intflags = PORTC.INTFLAGS;
    PORTC.INTFLAGS=intflags;
    if (interr==1)
    {
        TCA0.SINGLE.CTRLA = 0; //Disable
        interr=0;
    }
    SW6=1;
}

ISR(TCA0_OVF_vect){//interrupt gia ton palmo tou anemhsthra
    //clear the interrupt flag
    int intflags = TCA0.SINGLE.INTFLAGS;
    TCA0.SINGLE.INTFLAGS = intflags;
    if (LED1==0)
    {
        PORTD.OUT |= PIN1_bm;//Anoikse to LED1
        LED1=1;
    }
    else if (LED1==1)
    {
        PORTD.OUTCLR = PIN1_bm;//Kleise to LED1
        LED1=0;
    }
}

ISR(ADC0_WCOMP_vect){//interrupt gia to water leakage
    int intflags = ADC0.INTFLAGS;
    ADC0.INTFLAGS = intflags;

    PORTD.OUT |= PIN2_bm;//Anoikse to LED2

```

```

    InitialStateWLeakage=1;
}

ISR(TCA0_CMP0_vect){//interrupt gia ton timer tou boiler
    //clear flag
    int intflags = TCA0.SINGLE.INTFLAGS;
    TCA0.SINGLE.INTFLAGS=intflags;
    PORTD.OUTCLR = PIN0_bm;//Kleise to LED0
    interr=1;
    sei(); //enable interrupts
}

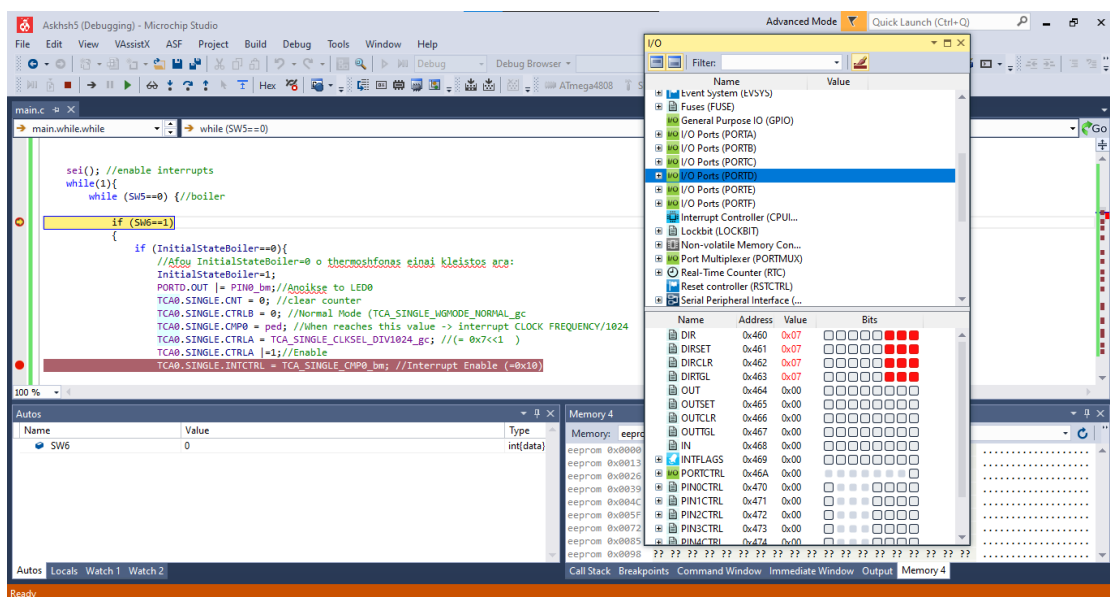
```

Αποτελέσματα Εκτέλεσης

Ξεκινώντας βάζουμε breakpoint στα εξής σημεία για να ελέγχουμε τα ενδιάμεσα αποτελέσματα(να σημειώσουμε εδώ ότι δεν στέλνουμε το zip με τα breakpoint ήδη μέσα γιατί κατά το testing παρατηρήσαμε ένα bug, καθώς αφού έστειλε ο ένας το zip στον άλλο τα breakpoint δεν δούλευαν κατά την εκτέλεση του κώδικα και υπήρχε πρόβλημα): γραμμή 56, 67, 75, 86, 104, 114, 127,136, 147, 166, 169, 184, 186, 197, 202, 214, 221.

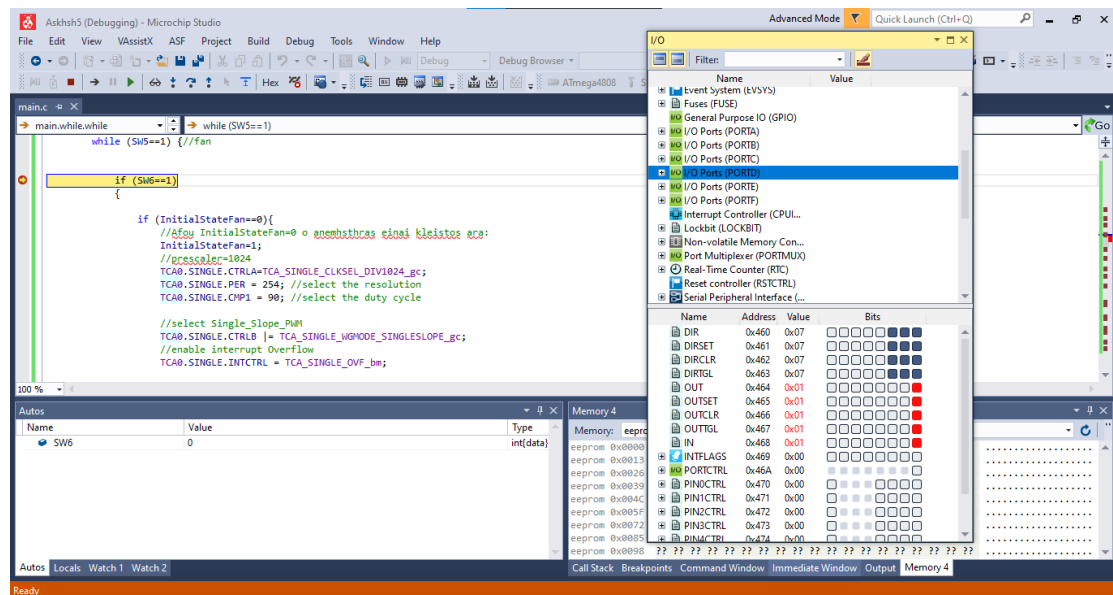
Αρχικά θα πρέπει να θέσουμε την τιμή του RES στον ADC μεγαλύτερη από 10 ώστε να μην ενεργοποιηθεί το interrupt για το water leakage.

Αφού η μεταβλητή SW5 είναι 0 είμαστε στην επιλογή Θερμοσίφωνα.



Αν πατηθεί το PIN6 του PORTC πηγαίνουμε στην συνάρτηση Θερμοσίφωνα όπου η λειτουργία της θα εξηγηθεί παρακάτω στο διάγραμμα ροής Θερμοσίφωνα.

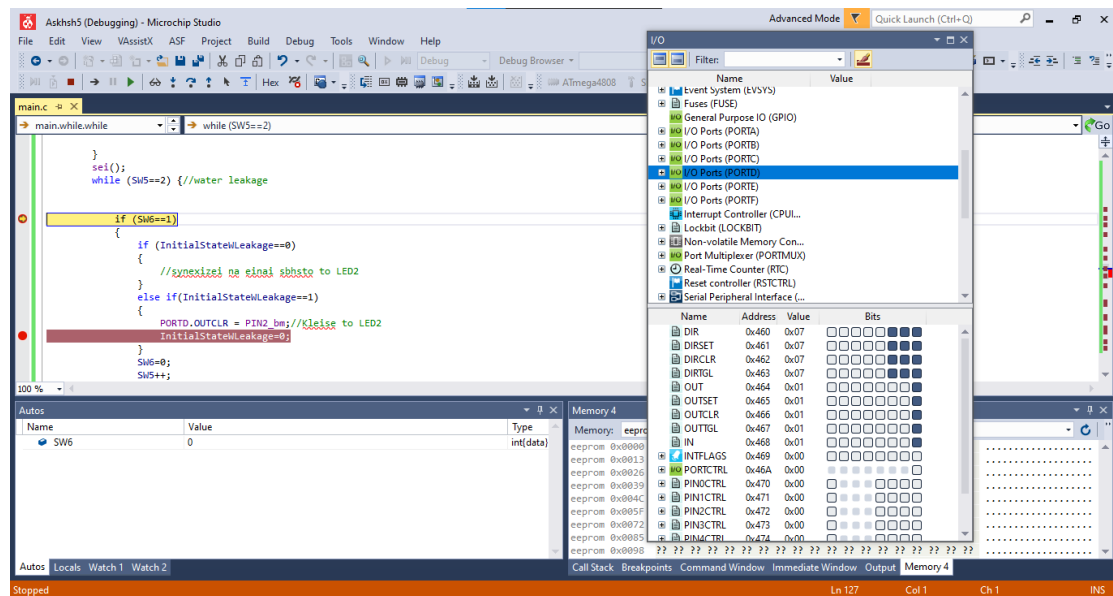
Ενώ αν πατηθεί PIN5 του PORTF τότε θα μεταβούμε στην επιλογή Ανεμιστήρας αφού η μεταβλητή SW5 θα γίνει ίσον με 1.



Εδώ ανοίγει και το LED0 όπως ορίζει η λειτουργία του Θερμοσίφωνα που θα εξηγηθεί παρακάτω.

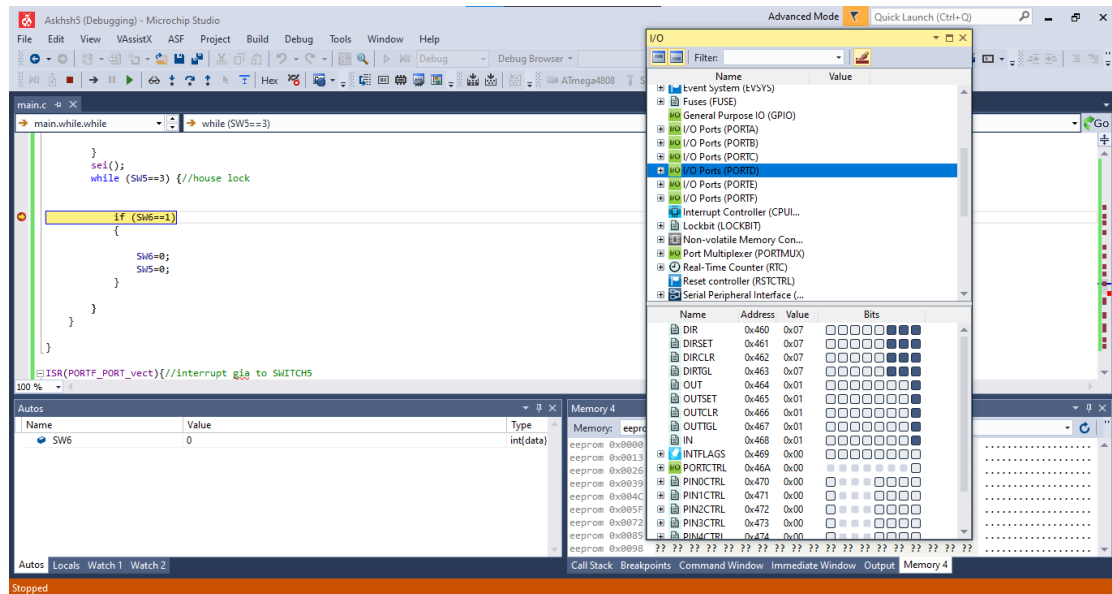
Αν είμαστε στην επιλογή Ανεμιστήρα και πατηθεί το PIN6 του PORTC πηγαίνουμε στην συνάρτηση Ανεμιστήρα όπου η λειτουργία της θα εξηγηθεί παρακάτω στο διάγραμμα ροής Ανεμιστήρα.

Αν πατηθεί PIN5 του PORTF τότε θα μεταβούμε στην επιλογή water leakage αφού η μεταβλητή SW5 θα γίνει ίσον με 2.



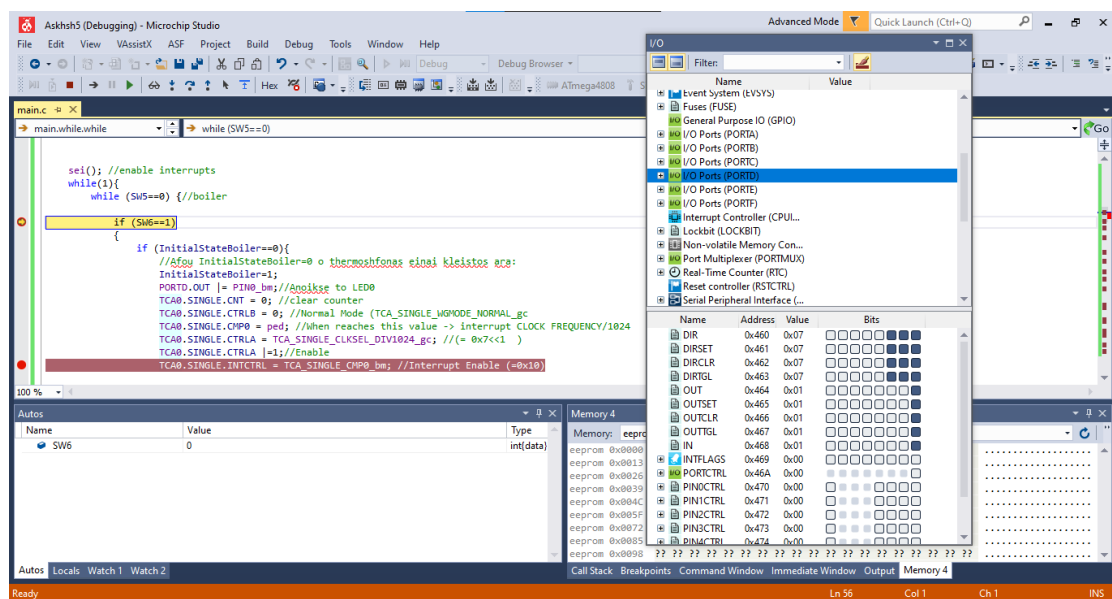
Αν πατηθεί το PIN6 του PORTC πηγαίνουμε στην συνάρτηση water leakage όπου η λειτουργία της θα εξηγηθεί παρακάτω στο διάγραμμα ροής water leakage.

Αν πατηθεί PIN5 του PORTF τότε θα μεταβούμε στην επιλογή κλειδώματος σπιτιού αφού η μεταβλητή SW5 θα γίνει ίσον με 3.



Αν πατηθεί το PIN6 του PORTC πηγαίνουμε στην συνάρτηση κλειδώματος σπιτιού την οποία δεν προλάβαμε να υλοποιήσουμε.

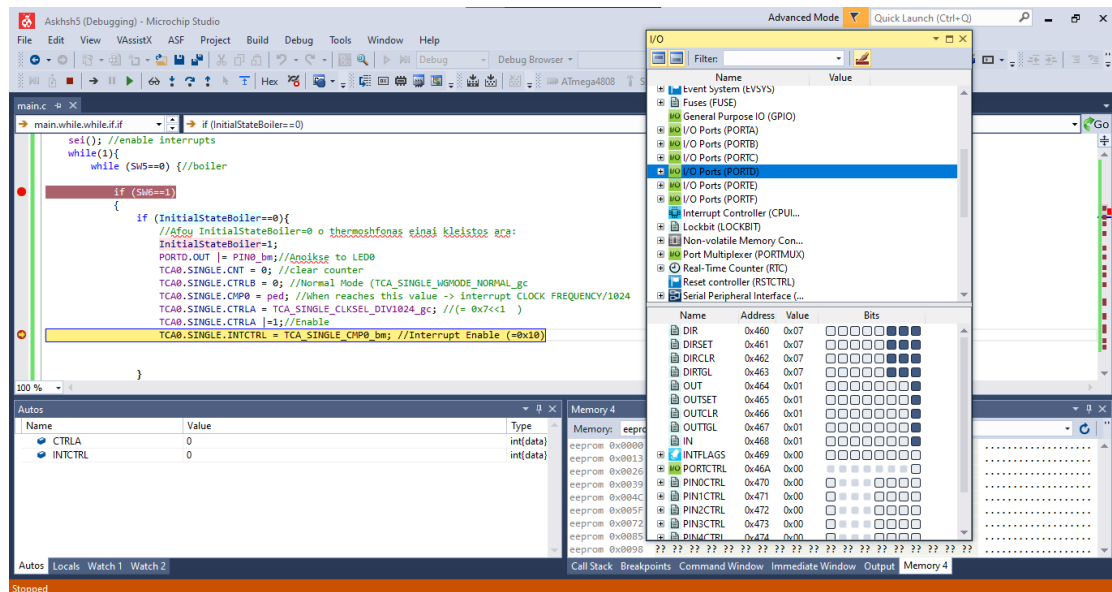
Ενώ αν πατηθεί το PIN5 του PORTF πηγαίνουμε πάλι στην επιλογή Θερμοσίφωνας έχοντας κάνει ένα πλήρη κύκλο.



ΣΥΝΑΡΤΗΣΗ ΘΕΡΜΟΣΙΦΩΝΑ

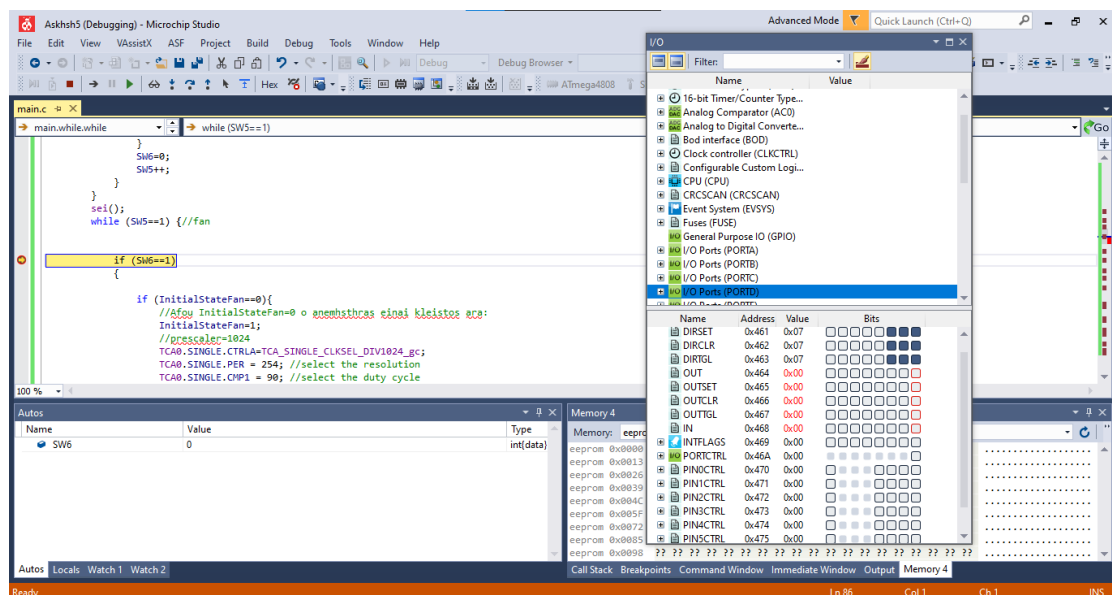
Αφού είμαστε στην συνάρτηση Θερμοσίφωνα έχουμε ορίσει μια μεταβλητή την `InitialStateBoiler` η οποία δείχνει αν είναι ενεργοποιημένος ή απενεργοποιημένος ο Θερμοσίφωνας όταν εισερχόμαστε στην συνάρτηση Θερμοσίφωνα.

Αφού πατηθεί το PIN6 του PORTC και ο Θερμοσίφωνας είναι απενεργοποιημένος (δηλαδή InitialStateBoiler=0) τότε ανάβει το LED0 και ενεργοποιείτε ο timer για ένα μεγάλο χρονικό διάστημα.

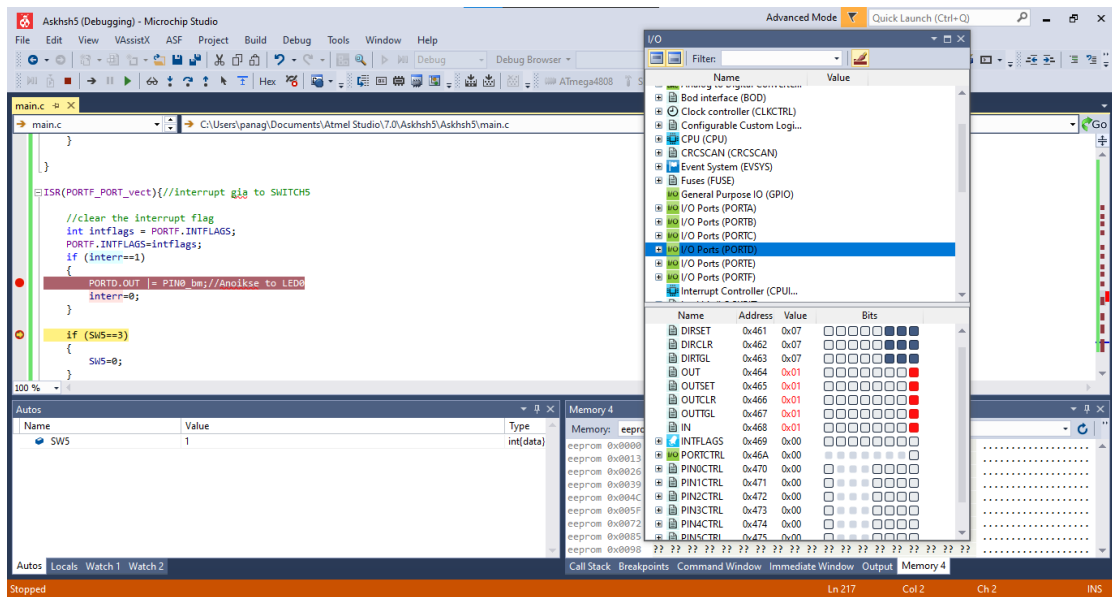


Στο σημείο αυτό αφαιρούμε το breakpoint από την γραμμή 86 ώστε να μην επηρεαστεί ο timer.

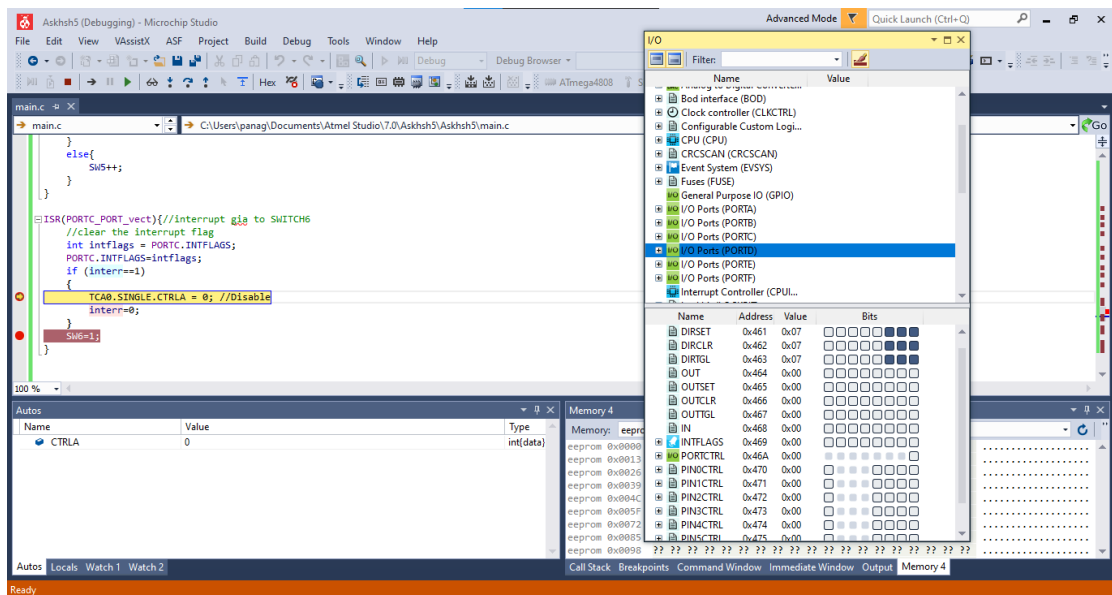
Έπειτα πηγαίνει στο ISR (`TCA0_CMP0_vect`) όπου σβήνει το LED0. Επίσης σε αυτό το σημείο επαναφέρουμε το breakpoint στην γραμμή 86 για να συνεχιστεί η ομαλή λειτουργία του κώδικα.



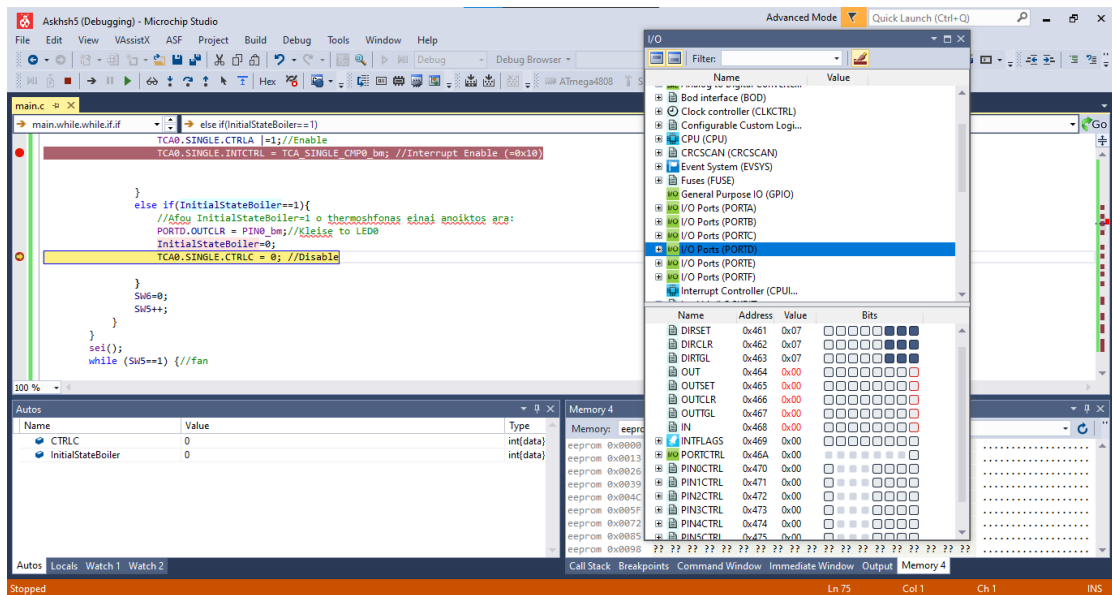
Αν πατηθεί PIN5 του PORTF τότε ανοίγει πάλι το LED0 και συνεχίζει ο timer.



Αλλιώς αν πατηθεί το PIN6 του PORTC θα παραμείνει κλειστό το LED0 και θα απενεργοποιηθεί ο timer.



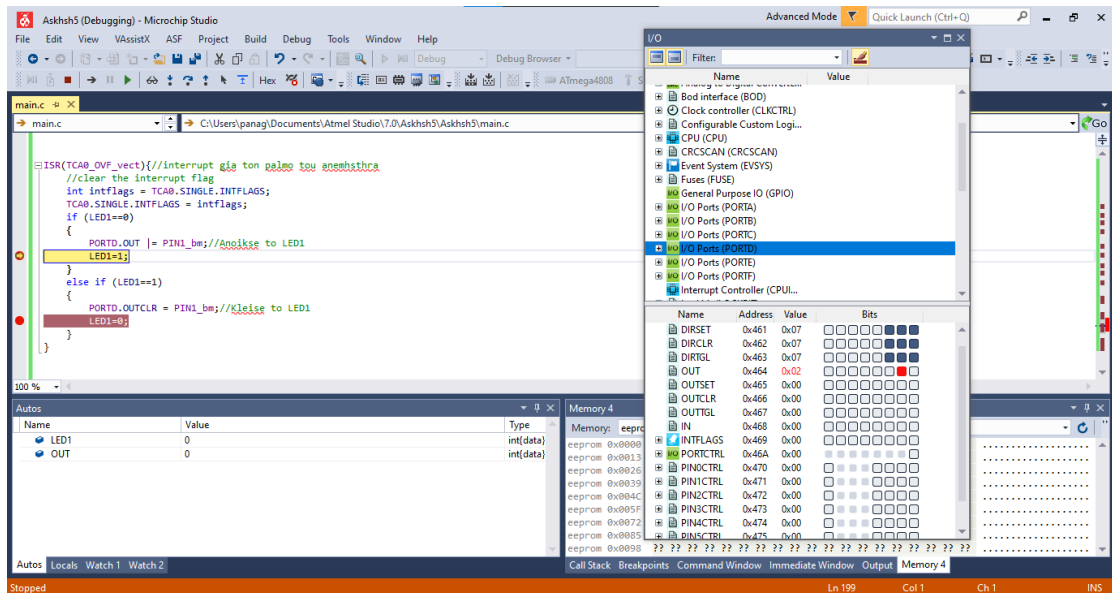
Εάν μπούμε στην συνάρτηση Θερμοσίφωνα και είναι ενεργοποιημένος (δηλαδή InitialStateBoiler=1) αν πατηθεί PIN5 του PORTF βγαίνουμε από την συνάρτηση Θερμοσίφωνα. Αλλιώς αν πατηθεί PIN6 του PORTC σβήνει το LED0 , απενεργοποιείτε ο timer και βγαίνουμε από την συνάρτηση Θερμοσίφωνα.



ΣΥΝΑΡΤΗΣΗ ΑΝΕΜΙΣΤΗΡΑΣ

Αφού είμαστε στην συνάρτηση Ανεμιστήρα έχουμε ορίσει μια μεταβλητή την InitialStateFan η οποία δείχνει αν είναι ενεργοποιημένος ή απενεργοποιημένος ο Ανεμιστήρας όταν εισερχόμαστε στην συνάρτηση Ανεμιστήρα.

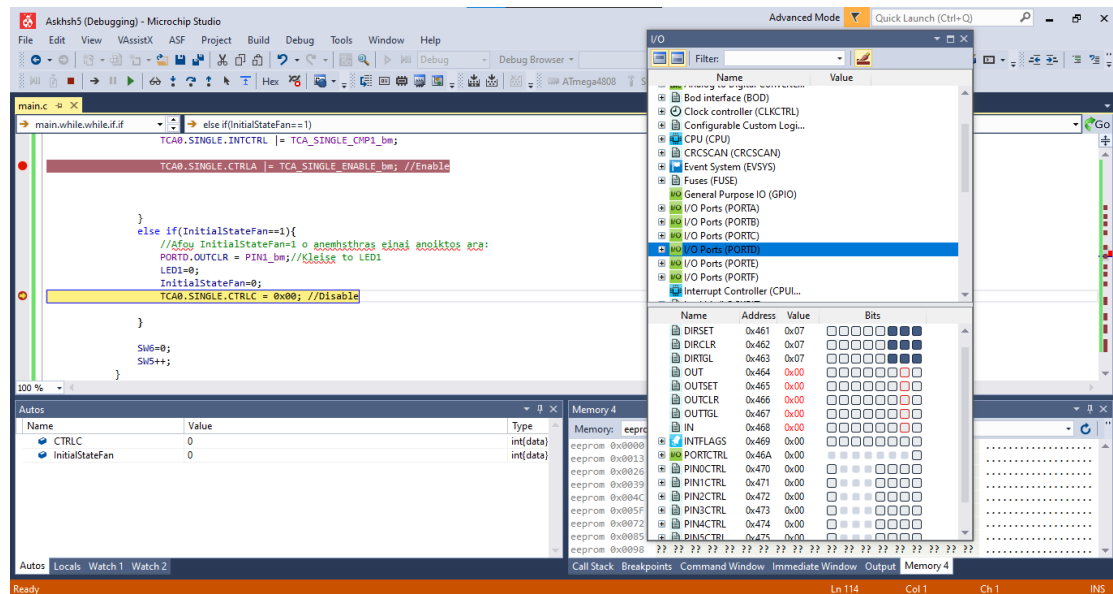
Αφού πατηθεί το PIN6 του PORTC και ο Ανεμιστήρας είναι απενεργοποιημένος (δηλαδή InitialStateFan=0) τότε ενεργοποιείτε ο PWM και ανάβει το LED1 σε rising edge και σβήνει το LED1 πάλι σε rising edge.



Έπειτα έχουμε έξοδο από την συνάρτηση Ανεμιστήρα.

Αν είμαστε στην συνάρτηση Ανεμιστήρα και είναι απενεργοποιημένος (δηλαδή InitialStateFan=0) και πατηθεί PIN5 του PORTF τότε έχουμε έξοδο από την συνάρτηση Ανεμιστήρα.

Αφού πατηθεί το PIN6 του PORTC και ο Ανεμιστήρας είναι ενεργοποιημένος (δηλαδή InitialStateFan=1) τότε σβήνει το LED1, απενεργοποιείτε ο PWM και έχουμε έξοδο από την συνάρτηση Ανεμιστήρα.

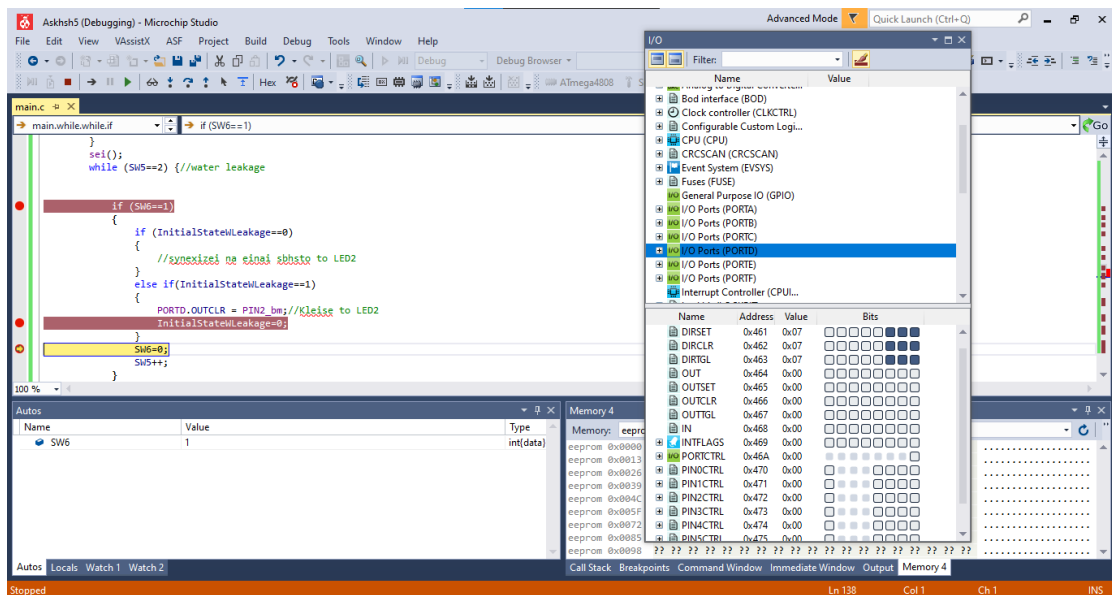


Αν είμαστε στην συνάρτηση Ανεμιστήρα και είναι ενεργοποιημένος (δηλαδή InitialStateFan=1) και πατηθεί PIN5 του PORTF τότε έχουμε έξοδο από την συνάρτηση Ανεμιστήρα.

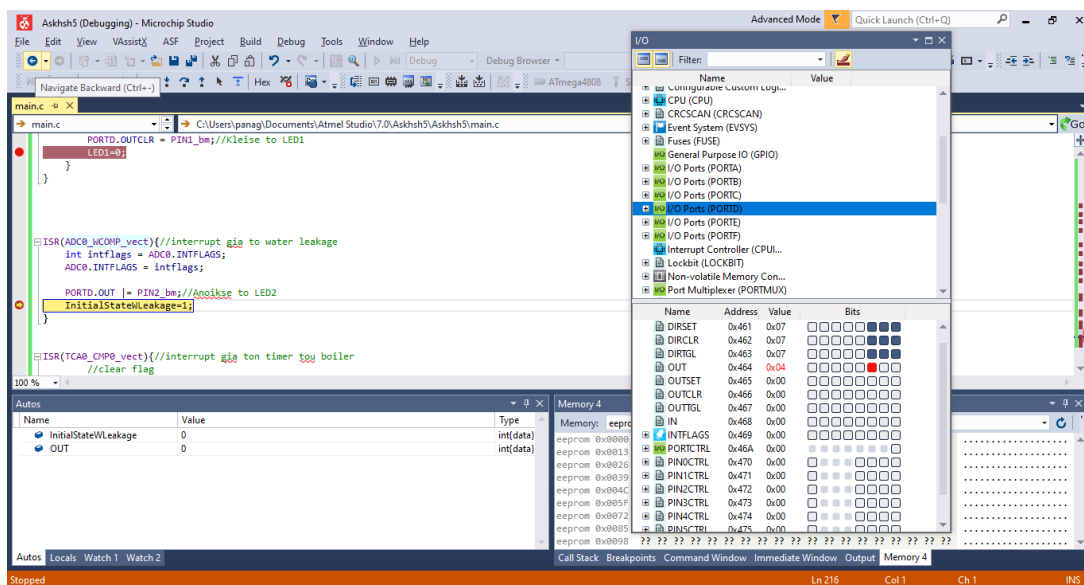
ΣΥΝΑΡΤΗΣΗ WATER LEAKAGE

Αφού είμαστε στην συνάρτηση water leakage έχουμε ορίσει μια μεταβλητή την InitialStateWLeakage η οποία δείχνει αν είναι ενεργοποιημένος ή απενεργοποιημένος ο αισθητήρας water leakage όταν εισερχόμαστε στην συνάρτηση water leakage.

Αφού πατηθεί το PIN6 του PORTC και ο αισθητήρας water leakage είναι απενεργοποιημένος (δηλαδή InitialStateWLeakage=0) τότε συνεχίζει να είναι σβηστό το LED2 και έχουμε έξοδο από την συνάρτηση water leakage.



Έστω ότι ο αισθητήρας water leakage (δηλαδή ο ADC<10) εντοπίζει διαρροή νερού ενεργοποιείτε το `ISR(ADC0_WCOMP_vect)` όπου ανάβει το LED2.



Αφού ανάψει το LED2 επιστρέφει στην κανονική ροή του συστήματος.

Αφού πατηθεί το PIN6 του PORTC και ο αισθητήρας water leakage είναι ενεργοποιημένος (δηλαδή InitialStateWLeakage=1) τότε σβήνει το LED2 και έχουμε έξοδο από την συνάρτηση water leakage.

