

ΕΡΓΑΣΤΗΡΙΟ 5

ΑΣΚΗΣΗ 1: Να δημιουργήσετε ένα πρόγραμμα το οποίο θα χρησιμοποιεί το PIN 9 (PB1) και το NORMAL mode του 16bit Timer Counter 1, channel A ώστε να καλεί το Overflow Interrupt του συγκεκριμένου Timer. Να μετρηθεί η συχνότητα κλήσης του Timer1_OVF_vector.

Υπόδειγμα 1:

```
volatile long timer; //Timer to keep count of the ISR occurred
static unsigned long period; // zero-unsigned-long
void setup(){
    cli(); //Disable global interrupts
    TCCR1A = 0; //Register set to 0
    TCCR1B = 0; //Register set to 0
    TCNT1 = 0;
    OCR1A = 15999;
    //TCCR1B |= (1 << WGM12); //CTC mode//Normal Mode
    TCCR1B |= (1 << CS10)|(1<<CS12); //Prescaler=1024
    TIMSK1 |= (1 << TOIE1); //Overflow interrupt mode

    sei(); //Enable global interrupts

    Serial.begin(9600);
}

ISR(TIMER1_OVF_vect){
    timer++; //Increment timer each ISR
}

void loop(){
    //. That is exactly once a second Polling .
    period+=micros();
    if (timer==1) {
        timer=0;
        unsigned long pp=period;
        period=micros()-pp;
        Serial.println(period/(1000.0*1000));
        period=0;
    }
}
```

ΑΣΚΗΣΗ 2: Να δημιουργήσετε πρόγραμμα που θα χρησιμοποιεί το PIN 9 (PB1) και το CTC (Counter Compare mode) του 16bit Timer Counter 1, ώστε να καλεί το Compare Match Interrupt για τον καταχωρητή OCR1A με συχνότητα κλήσης 1KHz. Να τροποποιήσετε τη συχνότητα στα 10KHz.

Υπόδειγμα 2:

```
volatile long timer; //Timer to keep count of the ISR occurred
void setup(){
    cli(); //Disable global interrupts
    TCCR1A = 0; //Register set to 0
    TCCR1B = 0; //Register set to 0
    TCNT1 = 0;
    OCR1A = 15999;
    TCCR1B |= (1 << WGM12); //CTC mode
    TCCR1B |= (1 << CS10); //No prescaler
    TIMSK1 |= (1 << OCIE1A); //Compare interrupt mode

    sei(); //Enable global interrupts

    Serial.begin(9600);
}

ISR(TIMER1_COMPA_vect){
    timer++; //Increment timer each ISR
}

void loop(){
    // That is exactly once a second Polling .
    static unsigned long old_millis = 0UL; // zero-unsigned-long

    if( millis() - old_millis >= 1000UL)
    {
        Serial.println(timer); //Print timer value

        // advance to next event, regardless the time needed by Serial.println()
        old_millis += 1000UL;
    }
}
```

ΑΣΚΗΣΗ 3: Δίνεται servo-motor 5V DC, RG90 (RG92R) με pinouts όπως αναφέρονται στη θεωρία 6 Σελ. 64. Να γραφεί πρόγραμμα χωρίς τη χρήση της Servo βιβλιοθήκης που θα μετακινεί τον servo-motor στις θέσεις 0 -90 -180.

Το mode που θα χρησιμοποιήσετε είναι το 14 για το PIN 9.

```
void setup()
{
    sei();
    // set up 2 PWM channels on PB1 and PB2 using Timer1
    TCCR1A = 0; // disable all PWM on Timer1 whilst we set it up
    ICR1 = 4999; // frequency is every 20ms
    // Configure timer 1 for Fast PWM mode via ICR1, with no prescaling
    TCCR1A = (1 << WGM11);
    TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS10) | (1 << CS11);
    // Set PB1 and PB2 as outputs
    DDRB |= _BV(1) | _BV(2);
    TCCR1A |= 2 << 6; // enable PWM on port B1 in non-inverted compare mode 2
    TCCR1A |= 2 << 4; // enable PWM on port B2 in non-inverted compare mode 2
    OCR1B = ICR1 * 2 / 20; // 2ms pulse to right motor on PB2
}

void loop()
{
    OCR1A = ICR1 * 0.5 / 20; // 2ms pulse to left motor on PB1
    delay(250);
    OCR1A = ICR1 * 1 / 20; // 2ms pulse to left motor on PB1
    delay(250);
    OCR1A = ICR1 * 1.5 / 20; // 2ms pulse to left motor on PB1
    delay(250);
    OCR1A = ICR1 * 2.0 / 20; // 2ms pulse to left motor on PB1
    delay(250);
    OCR1A = ICR1 * 2.5 / 20; // 2ms pulse to left motor on PB1
    delay(250);
}
```



ΠΑΝ ΔΥΤ. ΜΑΚΕΔΟΝΙΑΣ
ΧΕΙΜ. ΕΞΑΜ. 2014-2015

ΤΜΗΜΑ ΜΗΧ. ΠΛΗΡΟΦΟΡΙΚΗΣ κ ΤΛΠ.
ΜΙΚΡΟΕΠΕΞΕΡΓΑΣΤΕΣ
Διδάσκων: Κοντογιάννης Σωτήριος