

```

// Input: Pin D5
//Declaració de variables:
boolean flanc;
float comptador, valor, valor_ant, velocitat;
int motor;

//Declaració de variables del comptador de pulsos:
volatile unsigned long timerCounts;
volatile boolean counterReady;
float count;
long pulsos;
unsigned int timerTicks;
unsigned int timerPeriod;

//Funció de comptatge de pulsos de l'encoder del motor
void startCounting (unsigned int ms)
{

    counterReady = false;          // temps inferior a 100 ms
    timerPeriod   = ms; // quants ms ha de comptar
    timerTicks    = 0; // reinicia el comptador d'interruptió

    // reset Timer 1 i Timer 2
    TCCR1A = 0;
    TCCR1B = 0;
    TCCR0A = 0;
    TCCR0B = 0;

    // Timer 1 - compte els flancs del pin D5
    // Timer 2 - ens dóna un interval de 1ms
    // El rellotge és de 16 MHz (62.5 nS per tic) - prescaled de 64
    // El comptador incrementa cada 8 µS.
    // per tant nosaltres comptem 250 vegades per tenir 1000 µS (1 mS)
    TCCR0A = bit(WGM01) ; // CTC mode, és un mode de comparació
    OCR0A  = 249; // compte fins a 250 (zero relatiu) és el valor que
    // Timer 2 - interrompt cada 1 ms
    TIMSK0 = bit(OCIE0A); // habilita la interrupció del Timer 2

    TCNT1 = 0; // Inicialitza els dos comptador a 0
    TCNT0 = 0;

    // Reinicia els prescalers
    GTCCR = bit(PSRASY);
    // Inicia el Timer 2
    TCCR0B = bit(CS01) | bit(CS00) ; // Prescaled de 64
    // Comença el Timer 1
    // Configura el pin (D5) com una entrada que incrementa a cada flanc de pujada
    TCCR1B = bit(CS10) | bit(CS11) | bit(CS12);

```

```

    } // fi de startCounting

//*****
// La interrupció del Timer2 és cridada cad 1ms
// 16Mhz / 64 / 250 = 1000

ISR (TIMER0_COMPA_vect) // (Rutina d'interrupció) cada 1ms
{
    // Asigna el valor del comptador
    unsigned int timer0CounterValue;
    timer0CounterValue = TCNT1; // li dóna el valor del comptador

    // Mira si s'ha arribat al temps de 100ms
    if (++timerTicks < timerPeriod)
        return; // si encara no hi ha arribat retorna

    // Quan ja hi ha arribat es paren els Timers per veure els pulsos

    TCCR1A = 0; // stop timer 1
    TCCR1B = 0;

    TCCR0A = 0; // stop timer 2
    TCCR0B = 0;

    TIMSK1 = 0; // Desactiva la interrupció del Timer1
    TIMSK0 = 0; // Desactiva la interrupció del Timer2

    // Calcula el valor

    timerCounts = timer0CounterValue; // valor actual del comptador

    counterReady = true; // set global flag for end count period
} // end of TIMER2_COMPA_vect

void setup ()
{
    Serial.begin(9600);
    // Serial.println("Frequency Counter");

    // end of setup

void loop ()
{

```

```

//Crida la funció de comptar els impulsos
startCounting (100); // cada 100 ms crida la funció

while (!counterReady)
{ } // Crea un bucle mentre no ha acabat de llegir els pulsos que dura

//S'encarrega de canviar la consigna cada "x" temps
comptador = random(0,3000);
valor=millis();
if (valor-valor_ant > comptador) //Condició per fer el canvi de consigna segon
{

    if (flanc == 0)
    {
        motor = 140; //Consigna 1
        flanc = !flanc; //Alterna el valor de flanc que per fer el canvi de consigna
    } else
    {
        motor = 130; //Consigna 2
        flanc = 0;
    }
    valor_ant=valor; //Es passa el valor de temps actual a la variable del valor anterior
}

//Converteix el valor del comptador de pulsos a velocitat
pulsos = timerCounts;
velocitat = ((((((pulsos*10*60)/90)*1.5)/4.4)/2.75)*2*PI)/60)*0.032*100;
//Tenim els pulsos cada 0.1s per tant, multiplicant per 10 tenim els pulsos per segon
//segons tenim els pulsos per minut i si 90 pulsos son una volta tenim rpm
//les relacions de transmissió i ho passem a cm/s

//Escribim en el port sèrie els valors que ens interessen
analogWrite(11,motor);
Serial.print(";"); // Limita els nombres per poder-los diferenciar amb els de la velocitat
Serial.print(motor);
Serial.print(";"); // Limita els nombres per poder-los diferenciar amb els de la velocitat
Serial.println(velocitat);

// let serial stuff finish
//delay(100);

} // Fi del loop

```