

```

/**Programació de les maniobres d'aparcament**//

//Declaració de les variablesdel comptador de pulsos

volatile unsigned long timerCounts;
volatile boolean counterReady;
float pulsos;
unsigned int timerTicks, timerPeriod;

//Variables generals
float velocitat, pwm_direccio, pwm_ant=0, error, error_ant, graus;
int motor, Recepcio, sentit, lecturaPoten = A0, valorPoten = 0, direccio, Aut

//Variables d'odometria
float teta, x, y, L, x_ant, y_ant, teta_ant, omega;
int posicio=-1, estat;

//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'interrupció


// 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 q
// Timer 2 - interromp 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

// inici del setup

```

```

void setup ()
{
    Serial.begin(9600);

}
// fi del setup

//inici del Loop
void loop ()
{

while (Serial.available() > 0) //Quan hi ha recepció de dades les llegeix
{
    int Mode =Serial.parseInt(); //Llegim el primer valor enter (0-1) i e

    int y_pant =Serial.parseInt(); //Llegim el primer valor enter (0-100)

    int x_pant =Serial.parseInt(); //Llegim el primer valor enter (0-100)

    if (Serial.read() == '\n') //Si rebem \n significa que la primera sèr
    {
        Auto = Mode;
        pwm = y_pant;
        pwm2 = x_pant;
    }

    } //Fi de la recepció de valors

//Crida funció comptatge
    startCounting (100); // durant 100 ms crida la funció de comptatge de pulsos
while (!counterReady)
    { } // crea un bucle fins que el temporitzador no hagi finalitzat

//Conversió dels pulsos de l'encoder a velocitat
    pulsos = timerCounts;

//Determinem la velocitat del vehicle
    if(motor < 175){sentit = -1;} //Determina el sentit de gir
    else if (motor > 175){sentit = +1;}
    /* Tenim els pulsos cada 0.1s per tant, multiplicant per 10 tenim
       els pulsos/segon per 60 segons tenim els pulsos per minut i si 90 pulsos
       són una volta tenim rpm; després fem les relacions de transmissió i ho
    */
    velocitat = ((((((pulsos*10*60)/90)*1.5)/4.4)/2.75)*2*PI)/60)*0.031*100*s

//Lectura analògica dels graus de la direcció
    valorPoten =analogRead(lecturaPoten);

```

```

    graus=map(valorPoten,0,1023,-120,120); //Escalem el valor analògic a graus

//CONTROLADOR DIRECCIÓ
    error = direccio - graus; //L'error és la consigna menys l'angle de gir actual
    pwm_direccio = 180; //inicialitzem
    pwm_direccio=(error)*1.35-(error_ant)*0.78+pwm_ant; //Regulador dels graus

    //Limitació dels límits superior i inferior del controlador
    if(pwm_direccio>255)
        pwm_direccio=255;
    if(pwm_direccio<50)
        pwm_direccio=50;

    //Actualitzem els valors anteriors d'error i de pwm
    error_ant=error;
    pwm_ant=pwm_direccio;

    //Odometria:
    L=25.5; //distància entre l'eix de la direcció i el de la tracció
    omega= (graus*3.15)/180; //Conversió de graus a radians
    teta = (0.1*velocitat/L)*tan(omega)+teta_ant;
    x = 0.1*velocitat *cos(teta)+ x_ant; //multiplica per 0.1 perquè mostreja centímetres
    y = 0.1*velocitat *sin(teta)+ y_ant;
    //Actualitzem els valors anteriors de x, y i teta
    x_ant = x;
    y_ant = y;
    teta_ant = teta;

    if (Auto == 0) //si Auto és 0 significa que estem en mode manual
    {
        //Assignem a les variables del motor i direcció els valors enviats pel sensor
        motor=map(pwm,100,0,130,225); //Com que els valors rebuts van de 0-100
        direccio=map(pwm2,0,100,-30,30); //Com que els valors rebuts van de 0-100
        x=0; //inicialitzem les posicions d'aparcament a (0,0)
        y=0;

    }

    //Màquina d'estats per realitzar la maniobra d'aparcament
    else if (Auto == 1) //Si Auto és 1 inicia l'aparcament lateral
    {
        motor=100;
        direccio=0;
        switch(posicio)
        {
            //Estat inicial de repòs.
            case -1:
                direccio = -30;

```

```

    posicio = 0;
break;

case 0:
    motor = 225;//Endavant
    direccio = -30;

    if( x < 15 )
    {
        posicio = 0;
    }
    else if (x >= 15)
    {
        posicio=1;
    }
break;

//Posició 1
case 1:
    motor = 100;
    direccio = 30;
    if( x > -5 )
    {
        posicio = 1;
    }
    else if (x <= -5)
    {
        posicio = 2;
    }

break;

//Posicio 2
case 2:
    motor = 100;
    direccio = 0;

    if( x > -15 )
    {
        posicio = 2;
    }
    else if (x <= -15)
    {
        posicio = 3;
    }

break;

```

```
//Posició 3
case 3:

motor = 100;
direccio = -30;

    if( x > -20 )
    {
        posicio = 3;
    }
    else if (x <= -20)
    {
        posicio = 4;
    }

break;

//Posició 4
case 4:

motor = 225;
direccio = 30;

    if( x < -10 )
    {
        posicio = 4;
    }
    else if (x >= -10)
    {
        posicio = 5;
    }

break;

//Posició 5
case 5:

motor = 100;
direccio = -30;

    if( x > -18 )
    {
        posicio = 5;
    }
    else if (x <= -18)
    {
        posicio = 6;
    }
}
```

```

break;

    //Posició 6
case 6:
motor = 225;
direccio = 30;

    if( x < -10 )
    {
        posicio = 6;
    }
    else if (x >= -10)
    {
        posicio = 7;
    }

break;

    //Posició 7 - Estat de repòs -> Cotxe aparcant
case 7:
motor = 175;
direccio = 0;

break;

} // Fi maniobra d'aparcament

} // Fi Auto

else if (Auto == 2) //Si Auto és 2 inicia l'aparcament en bateria
{
motor=175;
direccio=0;
switch(posicio)
{
    //Estat inicial de repòs.
case 0:
    direccio = -30;
    posicio = 1;
break;

case 1:
    motor = 225; //Endavant
    direccio = -30;

    if( x < 20 )
    {
        posicio = 1;

```

```

    }
    else if (x >= 20)
    {
        posicio=2;
    }
    break;

//Posició 1
case 2:
    motor = 100;
    direccio = 30;
    if( y < -10 )
    {
        posicio = 2;
    }
    else if (y >= -10)
    {
        posicio = 3;
    }

//Posició 2
break;

case 3:
    motor = 225;
    direccio = -30;
    if( y > -25 )
    {
        posicio = 2;
    }
    else if (y <= -25)
    {
        posicio = 3;
    }

    break;

//Posició 3
case 4:

    motor = 100;
    direccio = 30;
    if( y < -5 )
    {
        posicio = 4;
    }
    else if (y >= -5)
    {
        posicio = 5;
    }

```



```

    }

    break;

//Posició 4
case 5:

    motor = 100;
    direccio = 0;
    if( y < 4 )
    {
        posicio = 5;
    }
    else if (y >= 4)
    {
        posicio = 6;
    }

    break;

//Posició 5 - Estat de repòs -> Cotxe aparcad
case 6:
    motor = 175;
    direccio = 0;

    break;

}

}

//Escrivim les sortides
    analogWrite(3,pwm_direccio);
    analogWrite(11,motor);

} // Fi del loop

```