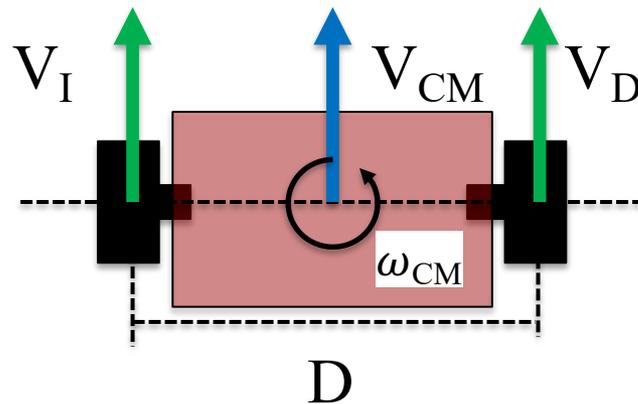




# Cinemática de un robot

## Cinemática directa de un robot con tracción diferencial

- La cinemática directa tiene como objetivo el cálculo de la velocidad lineal y angular del robot a partir de las correspondientes aportaciones de cada una de sus ruedas



- $V_D$  → Velocidad lineal de la rueda derecha
- $V_I$  → Velocidad lineal de la rueda izquierda
- $V_{CM}$  → Velocidad lineal del centro de masas
- $\omega_{CM}$  → Velocidad angular del centro de masas
- $D$  → Distancia entre las ruedas

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$

## Cinemática inversa

- Despejando del sistema de ecuaciones

$$V_D = V_{CM} + \frac{D}{2} \omega_{CM} \quad V_I = V_{CM} - \frac{D}{2} \omega_{CM}$$

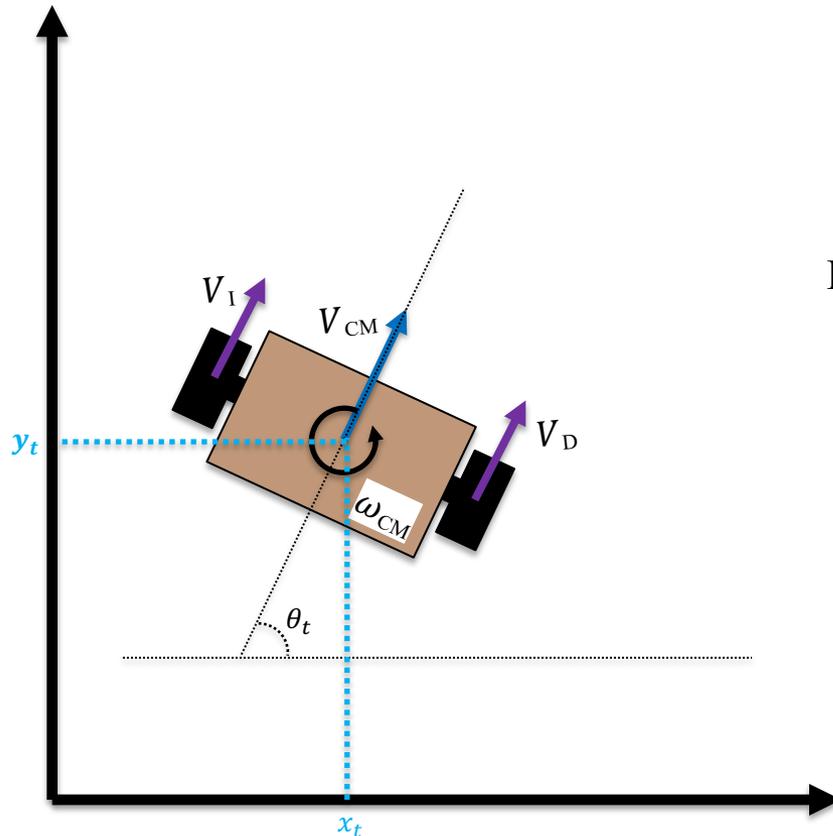
- Permite calcular la velocidad que tiene que tener cada rueda para que el robot tenga una velocidad lineal y angular determinada



# Odometría

## Cálculo de la posición de un robot

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$



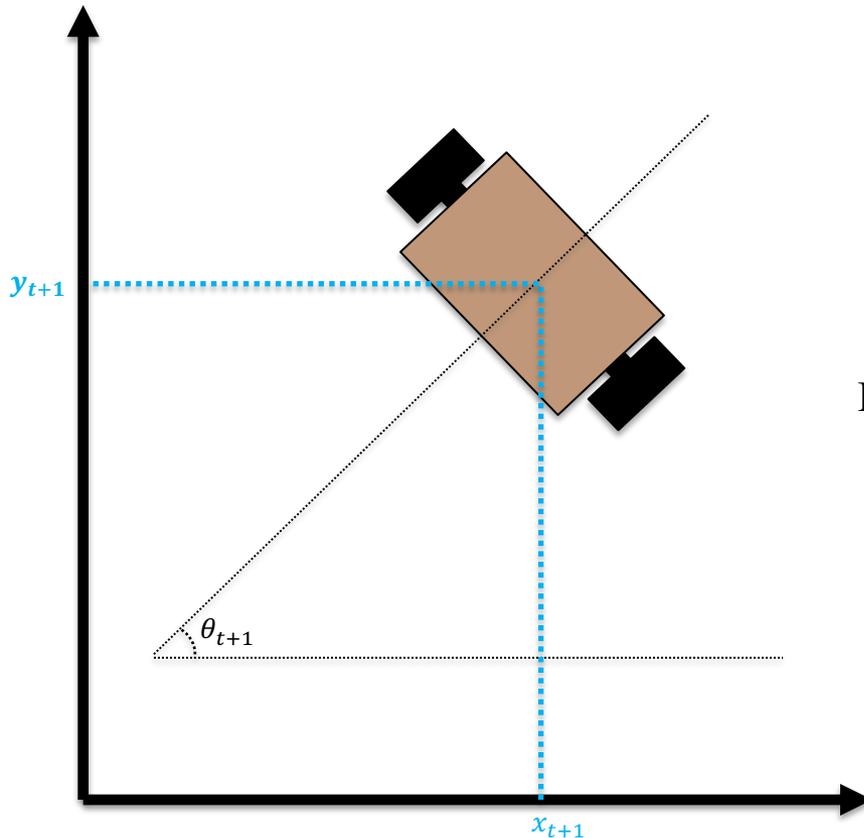
Posición en el instante  $t$   
 $(x_t, y_t, \theta_t)$



# Odometría

## Cálculo de la posición de un robot

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$



Posición en el instante t+1  
( $x_{t+1}$ ,  $y_{t+1}$ ,  $\theta_{t+1}$ )

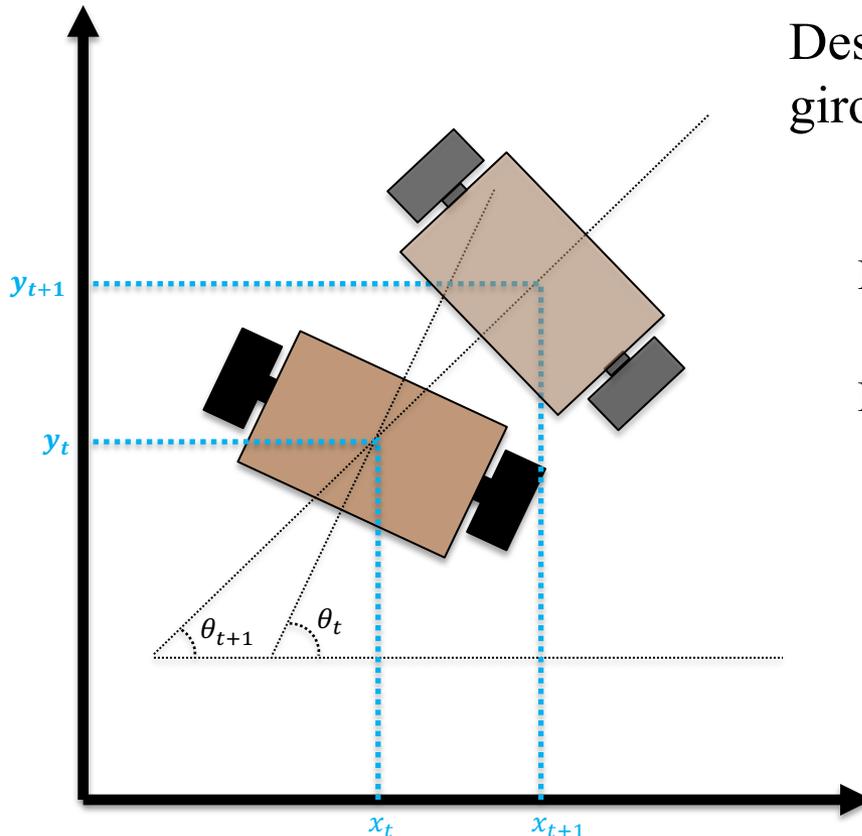


# Odometría

## Cálculo de la posición de un robot

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$

Descomponemos el movimiento en un giro y un avance



Posición en el instante t  
 $(x_t, y_t, \theta_t)$

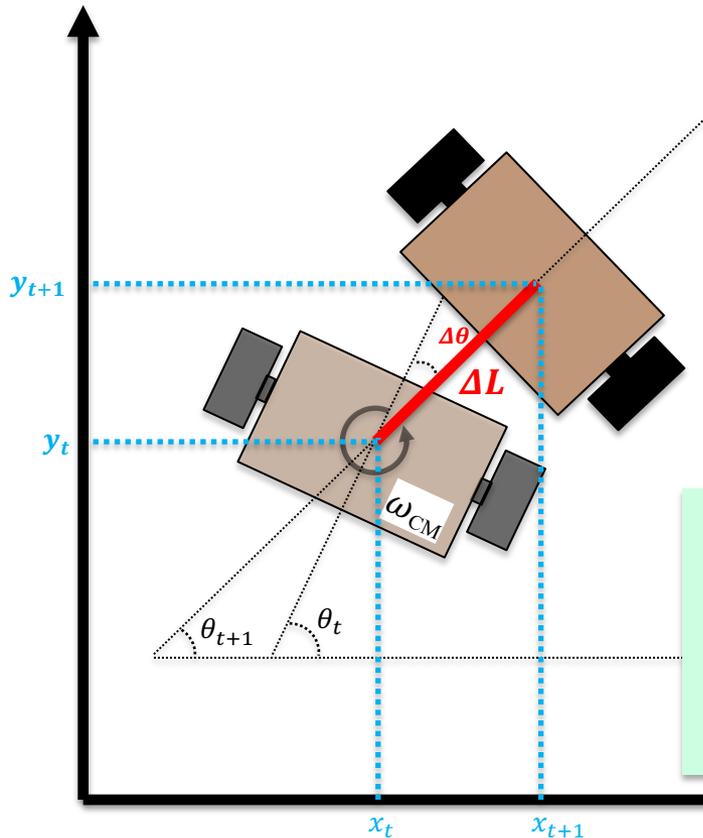
Posición en el instante t+1  
 $(x_{t+1}, y_{t+1}, \theta_{t+1})$



# Odometría

## Cálculo de la posición de un robot

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$



Descomponemos el movimiento en un giro y un avance

Gira  $\Delta \theta$  y avanza  $\Delta L$

Posición en el instante  $t$

$(x_t, y_t, \theta_t)$

Posición en el instante  $t+1$

$(x_{t+1}, y_{t+1}, \theta_{t+1})$

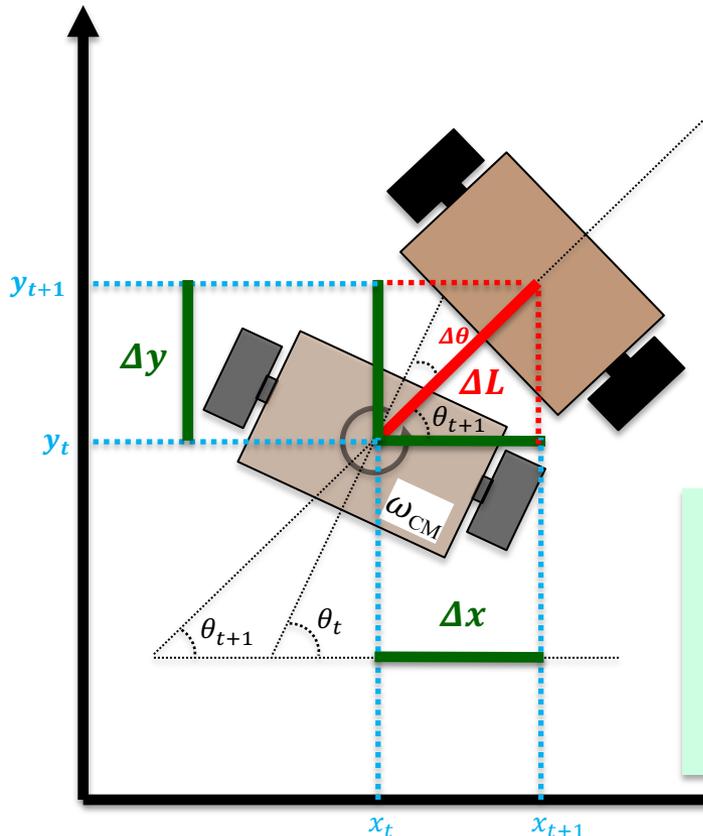
$$\theta_{t+1} = \theta_t + \Delta \theta$$



# Odometría

## Cálculo de la posición de un robot

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$



Descomponemos el movimiento en un giro y un avance

Gira  $\Delta\theta$  y avanza  $\Delta L$

Posición en el instante  $t$

$(x_t, y_t, \theta_t)$

Posición en el instante  $t+1$

$(x_{t+1}, y_{t+1}, \theta_{t+1})$

$$\theta_{t+1} = \theta_t + \Delta\theta$$

$$\begin{cases} \Delta x = \Delta L \cdot \cos\theta_{t+1} \\ \Delta y = \Delta L \cdot \sin\theta_{t+1} \end{cases} \quad \begin{cases} x_{t+1} = x_t + \Delta x \\ y_{t+1} = y_t + \Delta y \end{cases}$$



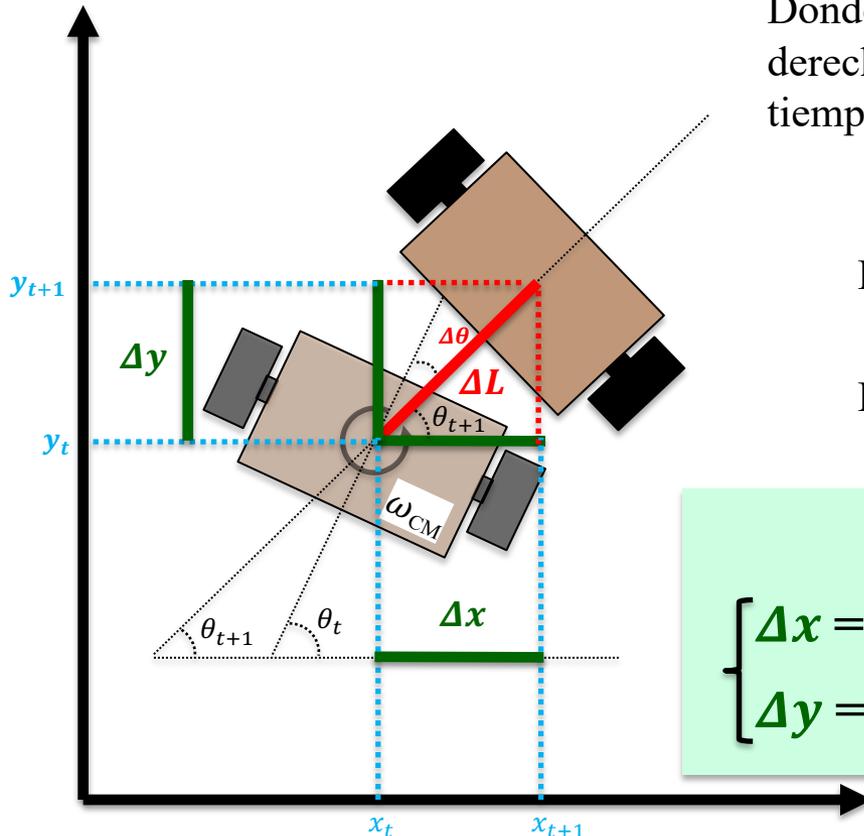
# Odometría

## Cálculo de la posición de un robot

Se puede demostrar que:

$$\Delta L = \frac{\Delta L_D + \Delta L_I}{2} \quad \Delta \theta = \frac{\Delta L_D - \Delta L_I}{D}$$

Donde  $\Delta L_D$  e  $\Delta L_I$  es la distancia avanzada por la rueda derecha e izquierda respectivamente en un intervalo de tiempo en  $\Delta t$



Posición en el instante t

$$(x_t, y_t, \theta_t)$$

Posición en el instante t+1

$$(x_{t+1}, y_{t+1}, \theta_{t+1})$$

$$\theta_{t+1} = \theta_t + \Delta \theta$$

$$\begin{cases} \Delta x = \Delta L \cdot \cos \theta_{t+1} \\ \Delta y = \Delta L \cdot \sin \theta_{t+1} \end{cases} \quad \begin{cases} x_{t+1} = x_t + \Delta x \\ y_{t+1} = y_t + \Delta y \end{cases}$$



# Movimiento de un robot

## Cálculo de la velocidad lineal y angular de un robot

- PPV – Pulsos por vuelta de rueda  $\left(\frac{\text{Pulsos}}{\text{vuelta}}\right)$
- LC – Longitud de la circunferencia de rueda  $\left(\frac{\text{metros}}{\text{vuelta}}\right)$
- $\Delta t$  – Segundos entre interrupciones  $\left(\frac{\text{segundos}}{\text{periodo}}\right)$
- $\Delta P_D$  – Pulsos medidos por periodo en la rueda D  $\left(\frac{\text{pulsos}}{\text{periodo}}\right)$
- $\Delta P_I$  – Pulsos medidos por periodo en la rueda I  $\left(\frac{\text{pulsos}}{\text{periodo}}\right)$

$$V_{CM} = \frac{V_D + V_I}{2} \quad \omega_{CM} = \frac{V_D - V_I}{D}$$

$$V_D \left(\frac{\text{metros}}{\text{segundo}}\right) = \Delta P_D \left(\frac{\text{pulsos}}{\text{periodo}}\right) \cdot \frac{1}{PPV \left(\frac{\text{Pulsos}}{\text{vuelta}}\right)} \cdot LC \left(\frac{\text{metros}}{\text{vuelta}}\right) \cdot \frac{1}{\Delta t \left(\frac{\text{segundos}}{\text{periodo}}\right)} = \frac{LC}{PPV} \cdot \frac{\Delta P_D}{\Delta t}$$

$$V_I \left(\frac{\text{metros}}{\text{segundo}}\right) = \Delta P_I \left(\frac{\text{pulsos}}{\text{periodo}}\right) \cdot \frac{1}{PPV \left(\frac{\text{Pulsos}}{\text{vuelta}}\right)} \cdot LC \left(\frac{\text{metros}}{\text{vuelta}}\right) \cdot \frac{1}{\Delta t \left(\frac{\text{segundos}}{\text{periodo}}\right)} = \frac{LC}{PPV} \cdot \frac{\Delta P_I}{\Delta t}$$



# Movimiento de un robot

## Cálculo de la velocidad lineal y angular de un robot

$$V_{CM} = \frac{V_D + V_I}{2}$$

$$\omega_{CM} = \frac{V_D - V_I}{D}$$

$$V_D = \frac{LC}{PPV} \cdot \frac{\Delta P_D}{\Delta t}$$

$$V_I = \frac{LC}{PPV} \cdot \frac{\Delta P_I}{\Delta t}$$

$$V_{CM} \left( \frac{\text{metros}}{\text{segundo}} \right) = \frac{LC}{PPV} \cdot \frac{1}{\Delta t} \cdot \left( \frac{\Delta P_D + \Delta P_I}{2} \right)$$

$$\omega_{CM} \left( \frac{\text{radianes}}{\text{segundo}} \right) = \frac{LC}{PPV} \cdot \frac{1}{\Delta t} \cdot \left( \frac{\Delta P_D - \Delta P_I}{D} \right)$$



# Odometría

## Cálculo de la velocidad lineal y angular de un robot

$$\omega_{CM} = \frac{d\theta}{dt} = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta\theta}{\Delta t} \right) \xrightarrow{\Delta t \rightarrow 0} \omega_{CM} = \frac{\Delta\theta}{\Delta t} \longrightarrow \Delta\theta = \Delta t \cdot \omega_{CM}$$

$$\Delta\theta = \Delta t \cdot \omega_{CM} = \Delta t \cdot \frac{LC}{PPV} \cdot \frac{1}{\Delta t} \cdot \left( \frac{\Delta P_D - \Delta P_I}{D} \right) = \frac{LC}{PPV} \cdot \left( \frac{\Delta P_D - \Delta P_I}{D} \right)$$

$$V_{CM} = \frac{dL}{dt} = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta L}{\Delta t} \right) \xrightarrow{\Delta t \rightarrow 0} V_{CM} = \frac{\Delta L}{\Delta t} \longrightarrow \Delta L = \Delta t \cdot V_{CM}$$

$$\Delta L = \Delta t \cdot V_{CM} = \Delta t \cdot \frac{LC}{PPV} \cdot \frac{1}{\Delta t} \cdot \left( \frac{\Delta P_D + \Delta P_I}{2} \right) = \frac{LC}{PPV} \cdot \left( \frac{\Delta P_D + \Delta P_I}{2} \right)$$

$$\text{Si } \Delta t \rightarrow 0 \quad \Delta\theta = \frac{LC}{PPV} \cdot \left( \frac{\Delta P_D - \Delta P_I}{D} \right) \quad \Delta L = \frac{LC}{PPV} \cdot \left( \frac{\Delta P_D + \Delta P_I}{2} \right)$$
$$\theta_{t+1} = \theta_t + \Delta\theta. \quad \begin{cases} \Delta x = \Delta L \cdot \cos\theta_{t+1} & x_{t+1} = x_t + \Delta x \\ \Delta y = \Delta L \cdot \text{sen}\theta_{t+1} & y_{t+1} = y_t + \Delta y \end{cases}$$



# Detección de movimiento

## Propuesta de actividades

- **Actividad 1** *Odometría\_1*
  - Envía por el puerto serie la distancia recorrida por el robot y el ángulo de orientación en grados.
  
- **Actividad 2** *Odometría\_2*
  - Modifica el programa anterior para que envíe además la posición en X y en Y del robot.
  
- **Actividad 3** *Odometría\_3*
  - Haz que el robot haga un cuadrado y representa los datos x, y recibidos por el puerto serie en un gráfico de Excel.
  
- **Actividad 4** *Odometría\_4*
  - Conociendo la distancia recorrida por el robot y la orientación, haz un programa que haga que el robot realice un cuadrado perfecto.



# Medida de velocidad

## Lectura de la velocidad de un motor

- La velocidad de un motor es el número de pulsos que avanza el motor en un intervalo de tiempo periódico
  - Este ejemplo es para 100ms

```
long pulsosAnteriorD, pulsosAnteriorI;  
unsigned long incrTiempo;  
long incrPulsosD, incrPulsosI;  
float velD, velI, velLineal, velAngular;<
```

```
millisActual = millis();  
incrTiempo = millisActual - millisAnterior;  
if (incrTiempo > 100) {  
    millisAnterior = millisActual;
```

```
    incrPulsosD = pulsosD - pulsosAnteriorD;  
    incrPulsosI = pulsosI - pulsosAnteriorI;  
    velD = incrPulsosD / incrTiempo;  
    velI = incrPulsosI / incrTiempo;  
    velLineal = (velD + velI)/2;  
    velAngular = (velD - velI)/D;
```

```
}
```



# Interrupción periódica con Arduino

## Interrupción periódica con Arduino

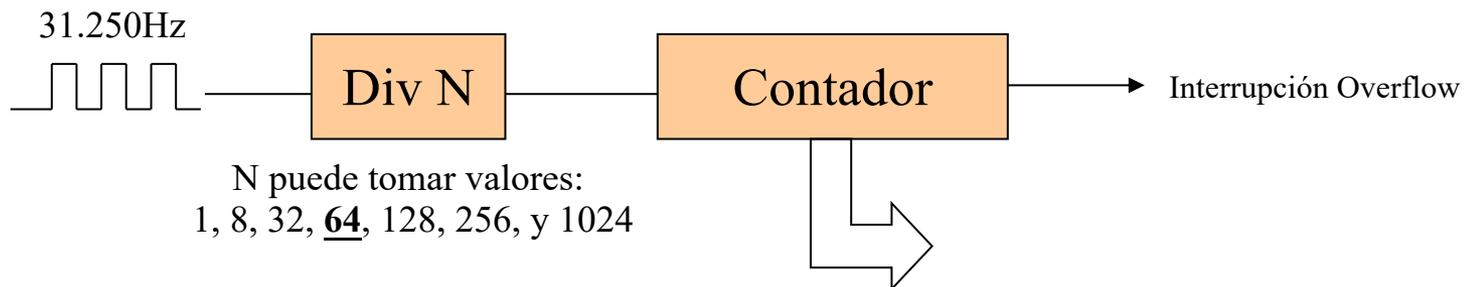
- Para hacer que se ejecute una interrupción periódicamente hay que utilizar los **Timers**.
  - Arduino UNO tiene **tres timers**: Timer0, Timer1 y Timer2
  - Arduino Mega tiene 6 timers: Timer0, ..., Timer5
- Los timers se utilizan para diversas funciones de Arduino por lo que si se utilizan, puede que dejen de funcionar algunas cosas.
- **El Timer2** es un **temporizador de 8bits** que se utiliza para:
  - AnalogWrite de los pines 3 y 11.
  - La función Tone.



# Interrupción periódica con Arduino

## Interrupción periódica con el Timer 2

- El **Timer2** es un **temporizador de 8bits** (cuenta de 0 a 255)
- Un temporizador es un contador digital que incrementa a la velocidad que le indica una señal de reloj.
- La velocidad de reloj es configurable con un divisor de frecuencia o “prescaler”
- Cuando se desborda el temporizador produce un evento que puede generar una interrupción (es necesario habilitarlo).



$$F_{I/O} = F_{OSC}/2 = 8.000 \text{ Hz}$$

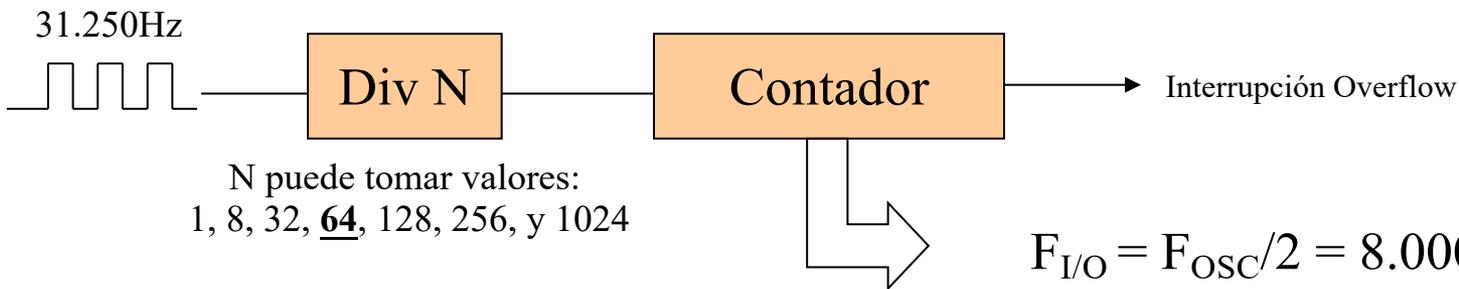
$$T_{overflow} = (2^8 \cdot \text{Clockdiv})/F_{I/O}$$



# Interrupción periódica con Arduino

## Interrupción periódica con el Timer 2

Prescaler	Frec reloj	Periodo Reloj	Frecuencia de Desbordamiento	Desbordamiento	CS2[2:0] <sub>TCCR2B</sub>
1	8.000.000 Hz	0,125 us	31.250 Hz	32 us	1
8	1.000.000 Hz	1 us	3.906,25Hz	256 us	2
32	250.000 Hz	4 us	976,56 Hz	1.024 us	3
64	125.000Hz	8 us	488,28 Hz	2,048 ms	4
128	62.500 Hz	16 us	244,14 Hz	4,096 ms	5
256	31.250 Hz	32 us	122,07 Hz	8,192 ms	6
1024	7.812,5 Hz	128 us	30,515 Hz	32,768 ms	7



$$F_{I/O} = F_{OSC}/2 = 8.000 \text{ Hz}$$

$$T_{overflow} = (2^8 \cdot \text{Clockdiv})/F_{I/O}$$



# Interrupción periódica con Arduino

## Interrupción periódica con el Timer 2

Bit	7	6	5	4	3	2	1	0	
(0xB1)	FOC2A	FOC2B	–	–	WGM22	CS22	CS21	CS20	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
(0x70)	–	–	–	–	–	OCIE2B	OCIE2A	TOIE2	TIMSK2
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 0 – TOIE2: Timer/Counter2 Overflow Interrupt Enable**

When the TOIE2 bit is written to one and the I-bit in the status register is set (one), the Timer/Counter2 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter2 occurs, i.e., when the TOV2 bit is set in the Timer/Counter2 interrupt flag register – TIFR2.



# Interrupción periódica con Arduino

## Interrupción periódica con el Timer de aproximadamente 30 ms

- Es necesario configurar el Prescaler del Timer 2 a 1024 → 32,768 ms

```
TCCR2B = 0b00000111;
```

- Habilitar la interrupción por desbordamiento

```
TIMSK2 = TIMSK2 | 0b00000001;
```

- Escribir la función de interrupción

```
ISR (TIMER2_OVF_vect) {  
  
}
```



# Interrupción periódica con Arduino

## Ejemplo de interrupción periódica

- Interrupción periódica cada 32ms y conmuta pin

```
const byte Pin = 13;

void setup() {
  pinMode(Pin, OUTPUT);
  noInterrupts();
  TIMSK2 = TIMSK2|0b00000001; //Habilita IRQ por desbordamiento
  TCCR2B = 0b00000111; //Preescaler para que desborde cada 32,768ms
  interrupts();
}

ISR(TIMER2_OVF_vect) {
  // Conmuta Pin
  if (digitalRead(Pin))
    digitalWrite(Pin,LOW);
  else
    digitalWrite(Pin,HIGH);
}

void loop() {
}
```

*InterrupcionPeriodica*



# Detección de movimiento

## Propuesta de actividades

### – **Actividad 5**

#### *Odometría\_5*

- Calcula la velocidad del robot (lineal y angular), el ángulo y las posiciones x a y dentro de una interrupción cada 32ms y envía los datos por el puerto serie cada segundo.
- Para comprobar el funcionamiento en simulación puedes poner un pin a nivel alto al entrar y a nivel bajo al salir

### – **Actividad 6**

#### *Odometría\_6*

- Modifica el programa para que la odometría se calcule cada 100ms.



# Referencias

## Referencias

- Qué son y cómo usar interrupciones en Arduino. Luis Llamas
  - <https://www.luisllamas.es/que-son-y-como-usar-interrupciones-en-arduino/>
- Arduino y las interrupciones. Prometec.net
  - <https://www.prometec.net/interrupciones/>
- Arduino y los Timers. Prometec.net
  - <https://www.prometec.net/timers/>
- Arduino Trick: How To Share Interrupt Service Routines(ISR) between Libraries and Application
  - <https://atadiat.com/en/e-arduino-trick-share-interrupt-service-routines-between-libraries-application/>