

4. Plataforma giratoria

Tallerine Arduino/LED

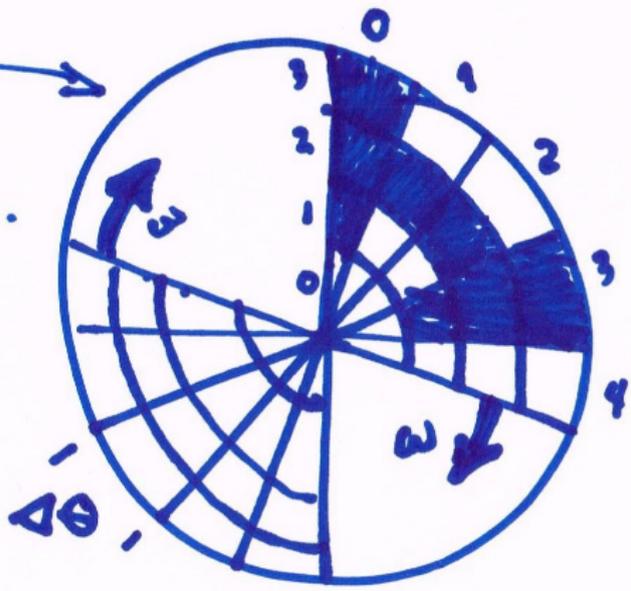
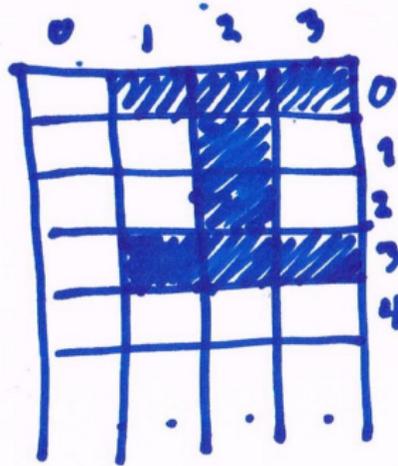
Instituto de Ingeniería Eléctrica

26 de agosto de 2019

Plataforma Giratoria



<https://www.youtube.com/watch?v=W3Gmv9J05eQ>



ω : velocidad angular
(vueltas / seg)

θ : ángulo de pixel

Datos

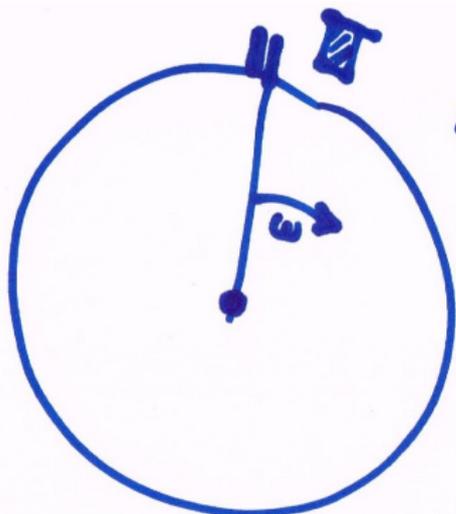
- 1 la *resolución* en pixels por vuelta n_p
- 2 la velocidad angular (en vueltas por segundo) de la rueda ω

Calcular

El *tiempo de pixel* t_p , es decir, cuánto debe durar un LED prendido o apagado para cubrir el área de un pixel al girar la rueda.

Cálculo de velocidad angular

detector de vueltas



$$\omega = \frac{\Delta \theta}{\Delta t}$$

- *Detectar* cada vez que la rueda completa un giro
 - Se requiere algún tipo de *sensor*.
 - Ejemplo: sensor infrarrojo en la rueda, luz led en la horquilla
- *Medir tiempo* Δt entre detecciones
 - Un programa con *memoria*
 - Debe *recordar* momento de detección anterior

Medición de intervalos de tiempo

- Cómo medimos tiempo entre dos eventos?
- Función `millis()` nos da la hora en milisegundos

```
int t_ant; // MEMORIA: tiempo de ultimo evento registrado
void setup() {
  t_ant = millis(); // tiempo en que arranca el Arduino
}
void loop() {
  while (...no pasa nada...) {
    ... esperar un poco
  }
  // evento registrado!
  int t = millis(); // registramos el momento
  // medimos tiempo respecto a evento pasado
  int dt = t - t_ant;
  // actualizamos: tiempo de ultimo evento es ahora t
  t_ant = t;
}
```

Ejercicio

Medir intervalo de tiempo entre dos pulsaciones

- Conectar un pulsador a algún pin de entrada del Arduino (con pull up o down, como gusten)
- medir segundos Δt entre dos pulsaciones
- Mostrar tiempo calculado por el puerto Serie
- prender y apagar LED cada Δt seg

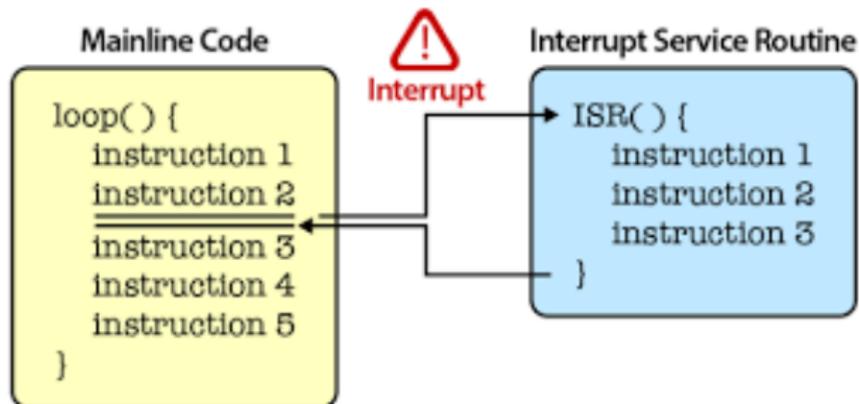
- Cómo mido velocidad de la rueda?
- Cómo mido la posición de la rueda? (sincronismo)
- Fin de carrera
- Sensor Efecto Hall
- ¿Problemas en realizar la medición?

- Cómo mido velocidad de la rueda?
- Cómo mido la posición de la rueda? (sincronismo)
- Fin de carrera
- Sensor Efecto Hall
- ¿Problemas en realizar la medición?

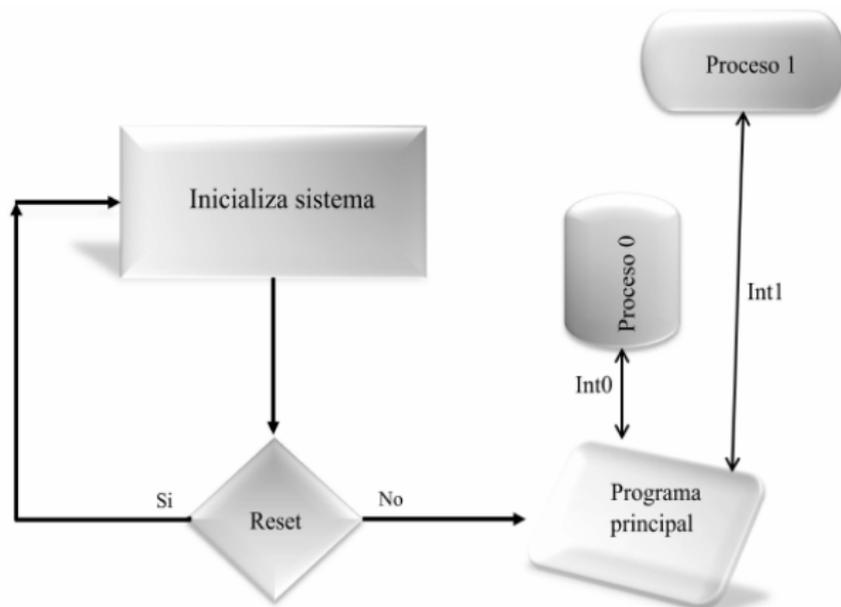
- Cómo mido velocidad de la rueda?
- Cómo mido la posición de la rueda? (sincronismo)
- Fin de carrera
- Sensor Efecto Hall
- ¿Problemas en realizar la medición?

Qué son?

En programación, una interrupción es una señal recibida por el procesador o MCU, para indicarle que debe «interrumpir» el curso de ejecución actual y pasar a ejecutar código específico para tratar esta situación.



Interrupciones



□

Attachinterrupt	Nombre	Pin (puerto digital)
0	INT0	pin 2 (D2)
1	INT1	pin 3 (D3)

Para definir una interrupción necesitamos tres cosas:

- Un pin de Arduino que recibirá la señal de disparo
- Una condición de disparo
- Una función que se ejecutará, cuando se dispara la interrupción



Ejemplo:

```
const int intPin = 2;
volatile int ISRCounter = 0;
int counter = 0;
void setup(){
    pinMode(intPin, INPUT_PULLUP);
    Serial.begin(9600);
    attachInterrupt(0, debounceCount, FALLING);
}
void loop(){
    if (counter != ISRCounter){
        counter = ISRCounter;
        Serial.println(counter);
    }
}

void debounceCount(){
    ISRCounter++;
}
```

A tener en cuenta:

- La función asociada a una interrupción se denomina ISR (Interruption Service Routines) y, por definición, tiene que ser una función que no recibe nada y no devuelve nada.
- El objetivo de una ISR es que tenga el menor tiempo de ejecución posible.
- Frecuentemente la función de la ISR se limitará a activar un flag, incrementar un contador, o modificar una variable.

Eventos que puede detectar una Int:

- RISING, ocurre en el flanco de subida de LOW a HIGH.
- FALLING, ocurre en el flanco de bajada de HIGH a LOW.
- CHANGING, ocurre cuando el pin cambia de estado (rising + falling).
- LOW, se ejecuta continuamente mientras está en estado LOW.



Algunas alertas:

- Para poder modificar una variable externa a la ISR dentro de la misma debemos declararla como «*volatile*».
- Marcar como *volatile* solo las variables que se usan tanto en el bucle principal como dentro de la ISR.
- La función *millis()* no actualiza su valor, por lo que no podemos utilizarla para medir el tiempo dentro de la ISR. (sí podemos usarla para medir el tiempo entre dos ISR distintas)
- La función *delay()* no funciona dentro de una ISR

Ejercicio -2

Medir intervalo de tiempo entre dos pulsaciones utilizando interrupciones

- Conectar un pulsador (o fin de carrera) a algún pin de entrada del Arduino (con pull up o down, como gusten)
- medir segundos Δt entre dos pulsaciones
- Mostrar tiempo calculado por el puerto Serie
- prender y apagar LED cada Δt seg

Traer:

- Solución implementada de los ejercicios
- Armar un circuito de 2 leds en el cual cada uno se prende durante la mitad del periodo. El periodo debe calcularse mediante la distancia temporal entre señales recibidas por el arduino desde el fin de carrera.
- La implementación del código y los leds se probarán en la rueda en la próxima clase.

