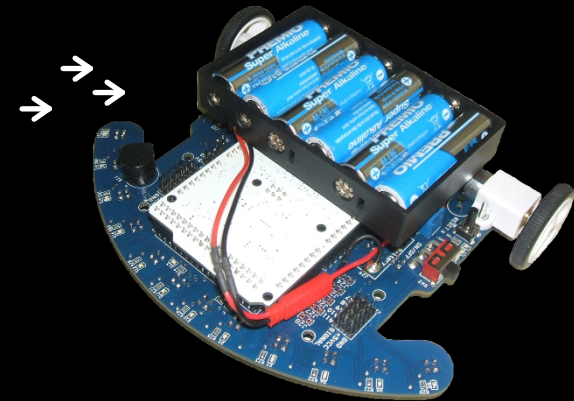
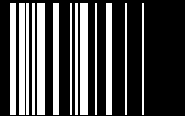


Robótica aplicada con Arduino



# ROBÓTICA APLICADA CON ARDUINO

15 JULIO 2013

Control del GR

Javier Pérez Alepuz

# Índice → →

- Teoría de control
- Controladores P, PD, PI, PID
- Control aplicado al rastreador GR
- Repaso control motores y lectura sensores analógicos



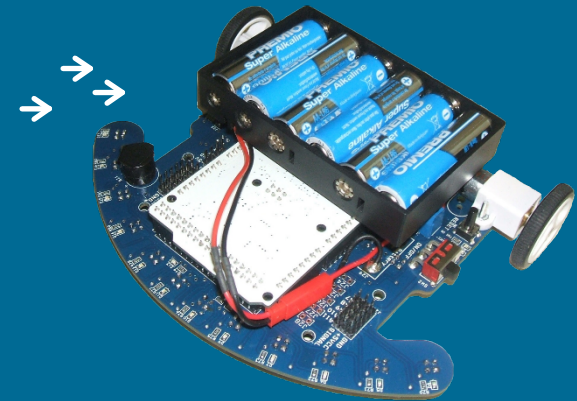
**Robótica aplicada  
con Arduino**



**dfests**  
ua.es



Robótica aplicada con Arduino



# TEORIA DE CONTROL



## Control de procesos

- Surge a partir de los progresos realizados en lo que se conoce la segunda revolución industrial.
- La aparición de las computadoras permiten un avance importante en control.
- Se pueden aplicar ideas de control imposibles hace relativamente pocos años.



Robótica aplicada  
con Arduino





## Teoría de control

- Se hace necesaria la comprensión del principio del control automático.
- Estudia el comportamiento dinámico de un sistema frente a órdenes de mando o perturbaciones.
- A las magnitudes que definen el comportamiento de un sistema se las denomina variables del sistema.
- Si dichas variables son de tipo físico se tendrá un sistema físico.



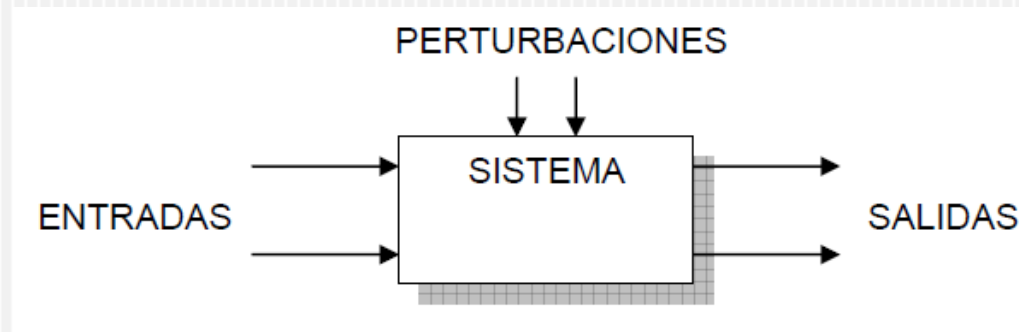
Robótica aplicada  
con Arduino





## Teoría de control (2)

- Se vale de modelos matemáticos abstractos que representan y describen con gran aproximación el comportamiento de un sistema.



Robótica aplicada  
con Arduino





## Bucle abierto, bucle cerrado

- Existen dos tipos de sistemas claramente
- diferenciados, cuya actuación se rige por principios diferentes:
  - sistemas de regulación en **bucle abierto**.
  - sistemas de regulación en **bucle cerrado**.



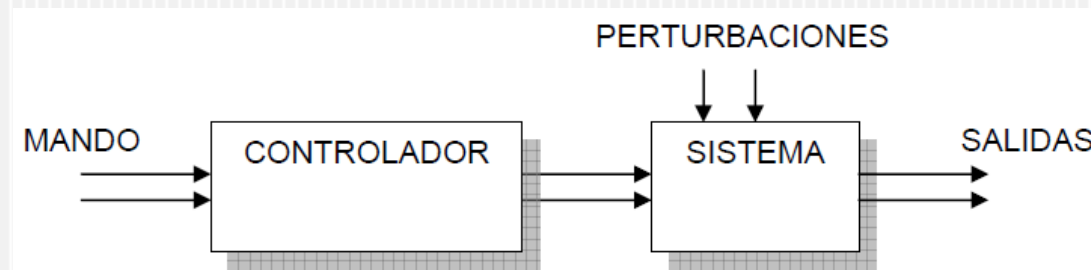
Robótica aplicada  
con Arduino





## Bucle abierto

- Los sistemas de regulación en bucle abierto son aquellos en los cuales las señales de entrada actúan directamente sobre los elementos que controlan el comportamiento del sistema sin ser influida por la salida del mismo.



Robótica aplicada  
con Arduino



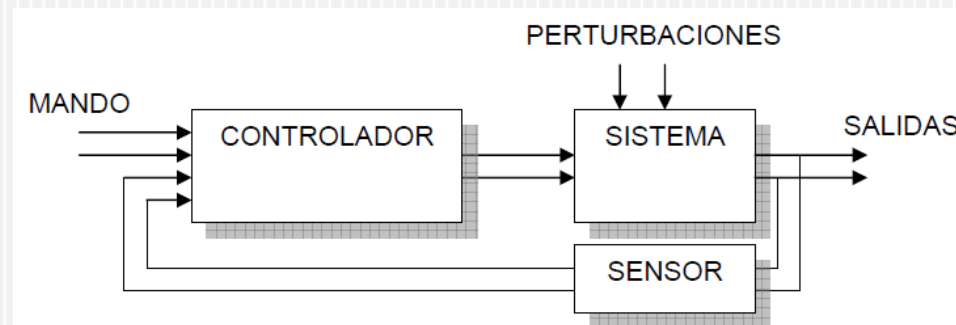
Ej: Lavadora





## Bucle cerrado

- Los sistemas de regulación en bucle cerrado son aquellos en los cuales además de las señales de entrada, el comportamiento del sistema también es influido por la salida del mismo.



Robótica aplicada  
con Arduino

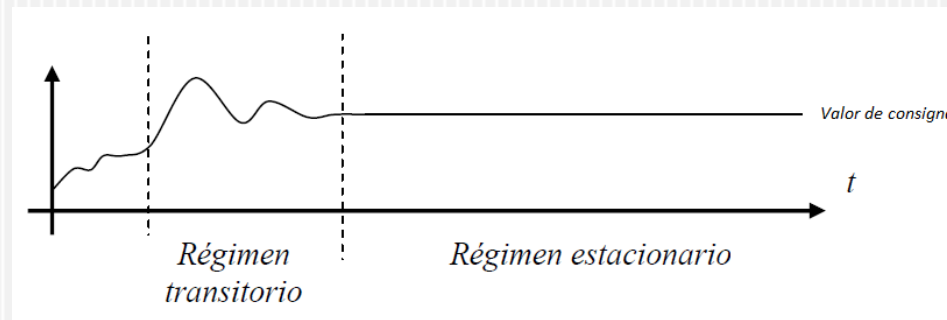


Ej: Control velocidad



## Régimen transitorio-estacionario

- El **régimen transitorio** es el período de tiempo en que la respuesta del sistema va desapareciendo con el tiempo.
- El **régimen estacionario** es el período de tiempo en que la respuesta del sistema se mantiene constante hasta que es afectada por perturbaciones.



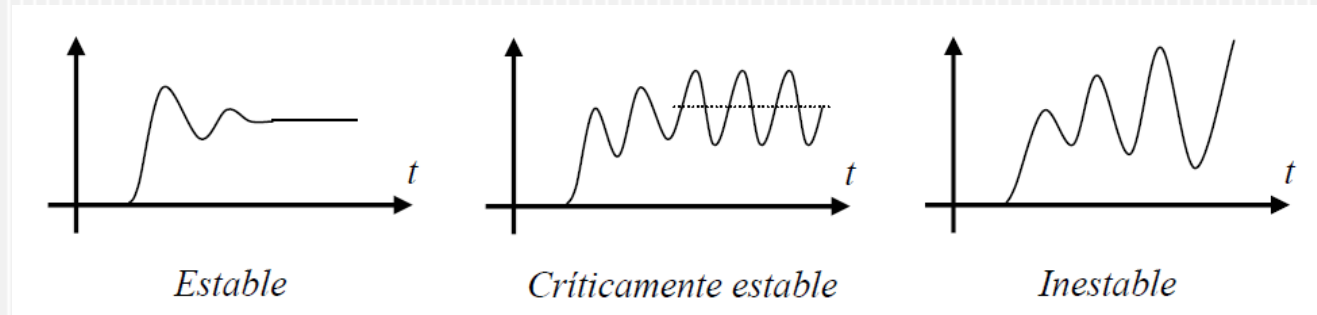
Robótica aplicada  
con Arduino





## Estabilidad - Inestabilidad

- Un sistema es **estable** cuando el comportamiento del mismo converge hacia un estado estacionario.
- Un sistema es **inestable** cuando el comportamiento del mismo diverge de un estado estacionario.



Robótica aplicada  
con Arduino





## Error

- El error de un sistema es la desviación que existe entre el valor de consigna y el valor actual medido/detectado.
- El objetivo del control es conseguir un error 0 en el sistema a controlar.



Robótica aplicada  
con Arduino

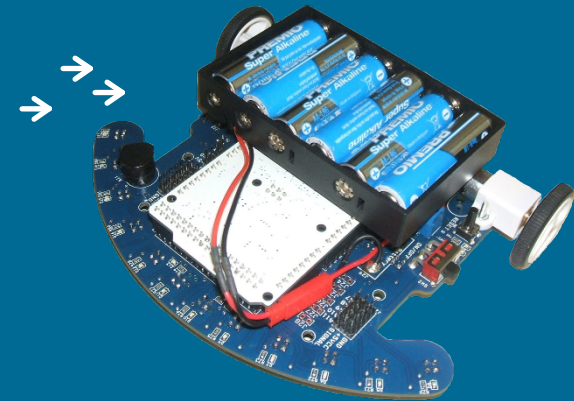


Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es



Robótica aplicada con Arduino



# CONTROLADORES P, PD, PI, PID



## Controlador P

- Un controlador P consta de una constante proporcional, también llamada **ganancia**, que multiplica al error.
- Hace al sistema más rápido en alcanzar el valor de consigna, pero también produce que hayan más sobreoscilaciones.
- Con valores de error pequeños (cuando el sistema está cerca de la consigna), su acción no tiene apenas efecto.



Robótica aplicada  
con Arduino



Ej: Control velocidad



## Controlador PD

- Es un controlador P que además consta una parte **derivativa**.
- Deriva el error y por tanto su acción es proporcional a los cambios que se producen en el error, anticipándose así a la evolución del sistema y suavizando la sobreoscilación.
- Con sistemas lentos es peligrosos pues puede hacerlos inestables.



Robótica aplicada  
con Arduino



Ej: Control velocidad



## Controlador PI

- Es un controlador P que además consta de una parte **integrativa**.
- Acumula el error a lo largo del tiempo y así es capaz de eliminar errores pequeños. Cuando el sistema está cerca de la consigna.
- Debe tener un valor muy pequeño y utilizará algún mecanismo para su control.



Robótica aplicada  
con Arduino



Ej: Control velocidad





## Controlador PID

- Es un controlador que combina los tres elementos descritos anteriormente.
- P, I y D son valores constantes que se pueden calcular matemáticamente si se tiene el modelo del sistema.
- Por norma general, el ajuste de un controlador se hace de manera manual, “prueba y error”, ya que en muchas ocasiones es complicado obtener el modelo de los sistemas.

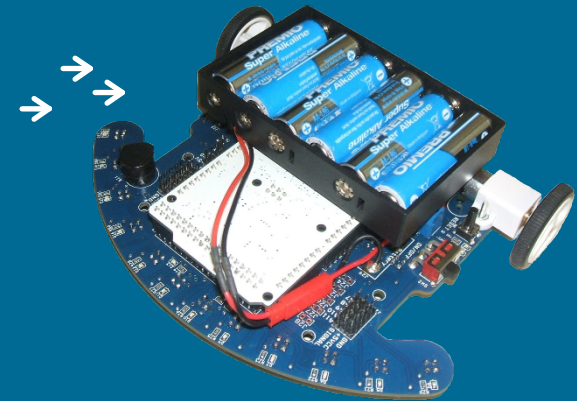


Robótica aplicada  
con Arduino





Robótica aplicada con Arduino



# CONTROL APLICADO AL RASTREADOR GR



## Control del GR

- Para el control del robot rastreador GR es suficiente con usar un control PD, ya que el uso de la parte integral complica la gestión del control y no se obtiene una mejora significativa.
- Sin embargo, es posible implementar un PID completo si el alumno lo desea.

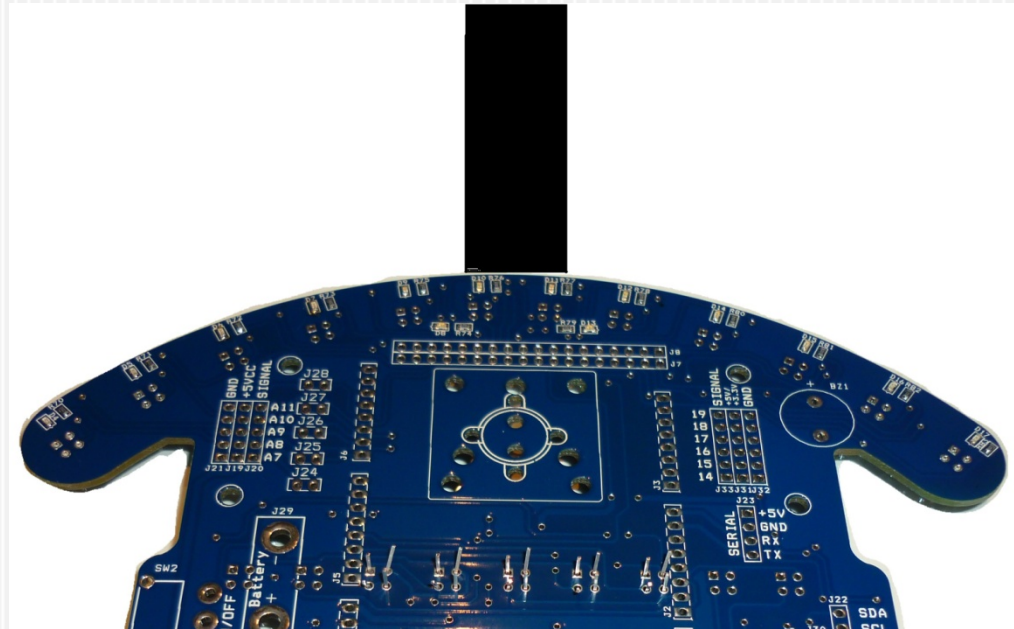


**Robótica aplicada  
con Arduino**





## Objetivo deseado (valor de consigna)



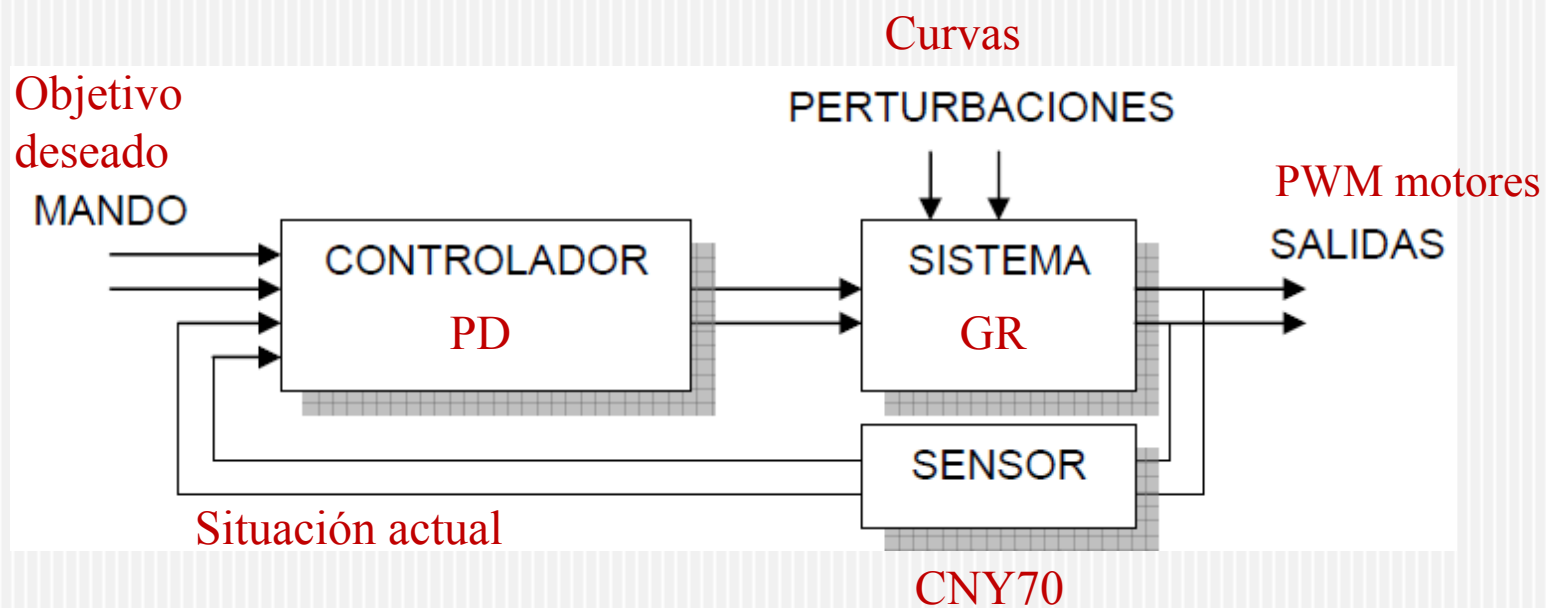
Robótica aplicada  
con Arduino



dfests  
ua.es



## Identificación de elementos del sistema

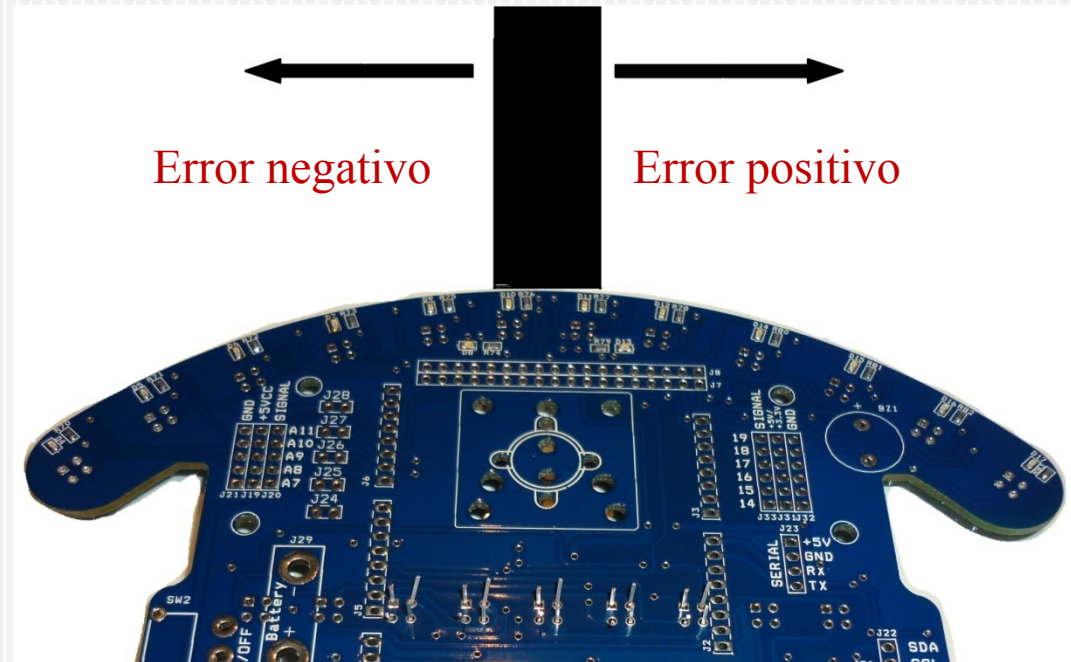


Robótica aplicada  
con Arduino





# Error



Robótica aplicada  
con Arduino



dfests  
ua.es



## Salida del sistema

- Se pretende determinar a qué valores hay que poner los pulsos PWM de los motores.
- Pasos (en código):
  - Implementar el método "get\_Center()" para determinar el centro de la línea leída.
  - Al método "setPid(double kP, double kI, double kD)" pasarle como parámetros 30, 1, 1 respectivamente.



Robótica aplicada  
con Arduino





## Salida del sistema (2)

- El método "getPID()" calcula y devuelve el valor del controlador PID:
  - Leer linea sensores analógicos.
  - Obtener el error con "getCenter()", que ya estará implementado, y almacenarlo en la variable "proportional". La variable "lastProportional" almacena el error del ciclo anterior.
  - Calcular el valor de PID con:

$\text{valor\_a\_devolver} = kP * \text{proportional} + kD * (\text{proportional} - \text{lastProportional})$



Robótica aplicada  
con Arduino







## Salida del sistema (3)

- Se determinará un valor de velocidad máxima para el robot en una variable MAXSPEED. Se comenzará por un valor bajo, por ejemplo 400.
- Se utilizará la función `analogWriteResolution(10)`. El valor 10 establecerá 10 bits como número de bits de resolución para los contadores que generan las señales PWM.



**Robótica aplicada  
con Arduino**





## Salida del sistema (3)

- Si resultado es positivo, aplicar el valor de velocidad **MAXSPEED–resultado\_del\_PID** al motor derecho y **MAXSPEED** al motor izquierdo.
- Si resultado es menor de 0, entonces aplicar el valor **MAXSPEED+resultado\_del\_PID** al motor izquierdo y **MAXSPEED** al motor derecho.
- Es importante no salirse del rango de PWM al establecer la velocidad en los motores. Con 10 bits el rango es 0..1023.



Robótica aplicada  
con Arduino





## Ajuste del PD

- Con la constante derivativa con valor a 1, ir incrementando la constante proporcional.
- Cuando el robot tenga un comportamiento oscilante, ir subiendo la derivativa hasta obtener un comportamiento aceptable.
- Una vez alcanzado este punto, ajustar las dos constantes alrededor de los valores obtenidos hasta obtener un comportamiento satisfactorio.



Robótica aplicada  
con Arduino





## Opciones adicionales

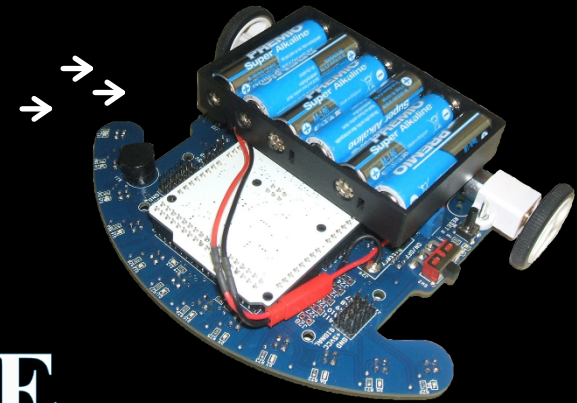
- Calibración de los sensores para que utilizarlos todos con valores mínimos, máximos e intermedios similares.
- Ponderar el error para obtener valores más altos en los extremos permitiendo una corrección mayor.
- Introducir la parte integrativa.



Robótica aplicada  
con Arduino



Robótica aplicada con Arduino

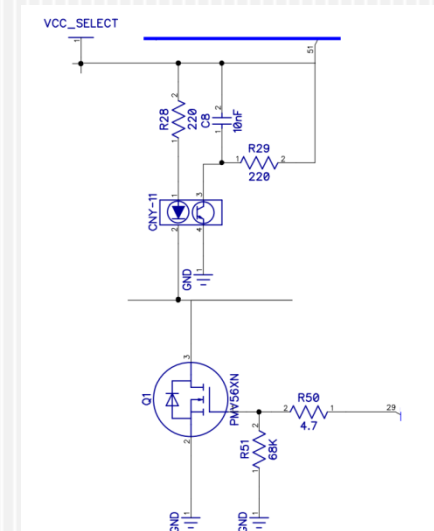


# REPASO CONTROL DE MOTORES Y LECTURA DE SENSORES ANALÓGICOS



# Sensores Analógicos

- Para realizar la lectura hay que seguir los siguientes pasos:
  1. Configurar el pin 51 como salida
  2. Escribir en el pin un estado alto (HIGH) Esperar 15 microsegundos
  3. Configurar pin 51 como entrada
  4. Escribir en el pin un estado bajo (esto desactiva la pull-up interna, véase "Table 3-1. Signal Description List " del datasheet del microcontrolador SAM3X8E )
  5. Encender la línea de sensores poniendo el pin 29 a estado alto.
  6. Esperar 5 microsegundos
  7. Contador = 0
  8. Mientras no el sensor valga estado alto
    1. Leer sensor
    2. Si el estado es bajo
      1. Contador++
    3. Sino
      1. Valor sensor = Contador
    4. Esperar un microsegundo
  9. Apagar la línea de sensores poniendo el pin 29 en estado bajo.



**Robótica aplicada  
con Arduino**





## Calculo del error

- Una vez leídos los sensores se obtendrá un valor de entre 0 y unos 300 us de tiempo de descarga en cada sensor, siendo de  **$S_i$**  el valor obtenido para el sensor  **$i$** .
- Una forma fácil de conocer el central de la linea es utilizar una media ponderada.

$$\text{centro} = \frac{S1*1000 + S2*2000 + S3 * 3000 + \dots + S_n*n*1000}{S1 + S2 + S3 + \dots + S_n}$$



Robótica aplicada  
con Arduino





## Calculo del error

- Obtenido el valor del centro de la linea (según la explicación anterior). El error se puede obtener como 3500-



**Robótica aplicada  
con Arduino**

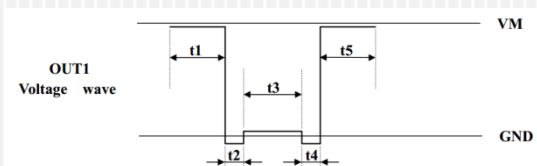
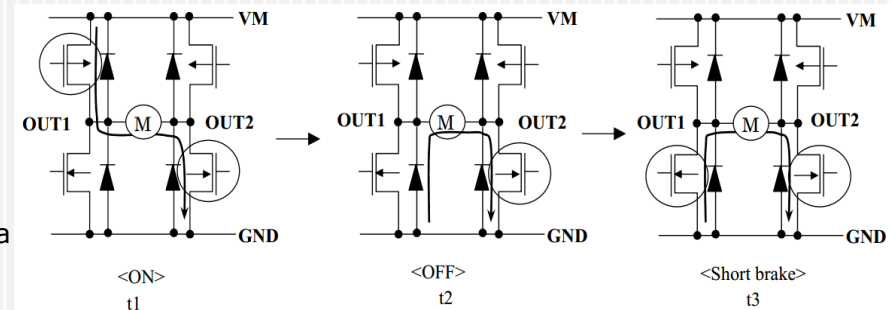






# Actuadores

- **Motor en marcha (Figura ON)**
  - -Se hace circular la corriente a través del motor (según el sentido de la corriente el motor girara en uno u otro sentido).
- **Motor libre (Figura OFF)**
  - El motor gira libremente por su propia inercia hasta que termina frenando lentamente.
- **Freno motor (Fig. Short Brake)**
  - Se pone ambos bornes del motor a la misma tensión de forma que se descarga la bobina del motor y se opone al movimiento del rotor, por lo que se fuerza el frenado.



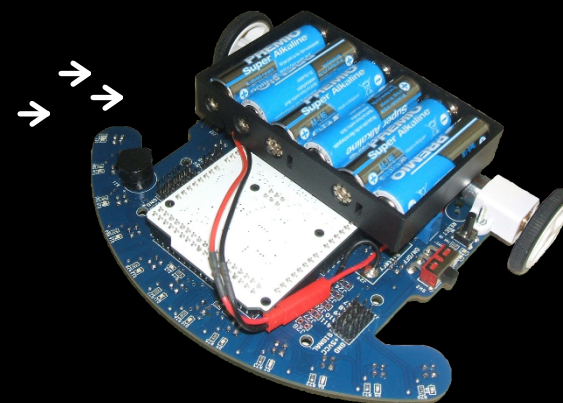
Input				Output		
IN1	IN2	PWM	STBY	OUT1	OUT2	Mode
H	H	H/L	H	L	L	Short brake
L	H	H	H	L	H	CCW
		L	H	L	L	Short brake
H	L	H	H	H	L	CW
		L	H	L	L	Short brake
L	L	H	H	OFF (High impedance)		Stop
H/L	H/L	H/L	L	OFF (High impedance)		Standby



Robótica aplicada  
con Arduino



Robótica aplicada con Arduino



# ROBÓTICA APLICADA CON ARDUINO

15 JULIO 2013

Control del GR

Javier Pérez Alepuz