



Robótica aplicada con Arduino





# ROBÓTICA APLICADA CON ARDUINO

17 JULIO 2013

Sensores digitales y analógicos. Primeras pruebas.

Carlos A. Jara y Santiago T. Puente

[carlos.jara@ua.es](mailto:carlos.jara@ua.es)



## Índice →↗

- Código Arduino
- Primer sigue-líneas
- Uso de la librería del taller
- Sensores digitales
  - Descripción
  - Diseño electrónico
  - Lectura de su estado con la librería
  - Primer sigue-líneas por señal digital
- Sensores analógicos
  - ....
  - ...



Robótica aplicada con Arduino

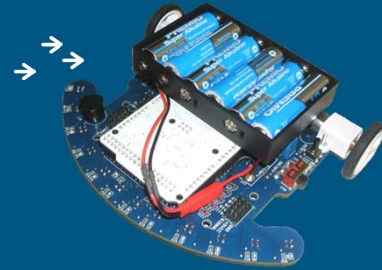


Master Oficial en Automática y Robótica  
Universidad de Alicante



dfests.ua.es

Robótica aplicada con Arduino



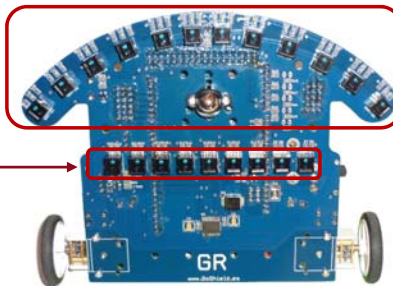
# SENSORES DIGITALES

## Descripción

### Descripción

- La plataforma GR incorpora dos filas de sensores:
  - Una delantera preparada para ser leída en modo analógico.
  - Una central preparada para ser leída en modo digital.

Sensores centrales preparados para lectura digital



Sensores delanteros preparados para lectura analógica



Robótica aplicada  
con Arduino

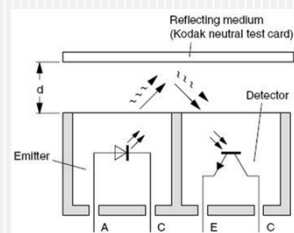
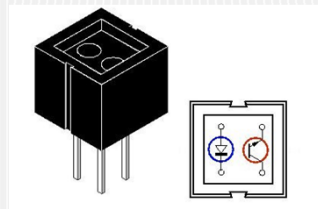


Master Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## Descripción

- Sensores de reflexión CNY70.
  - Se componen de un emisor infrarrojo y un receptor fotosensible.
  - Distinta reflexión del haz infrarrojo en superficies blancas y negras.



Robótica aplicada  
con Arduino

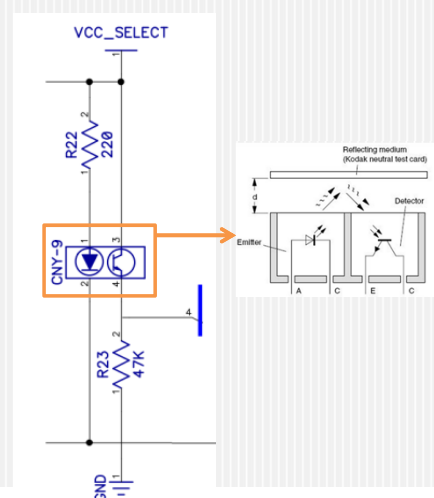


Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## Descripción

- El esquema electrónico diseñado permite leer un sensor a través de las entradas digitales del Arduino DUE.
- Para poder obtener una salida en la lectura (si existe o no línea) es necesario pasar corriente por el *fotodiodo*. R22 ajusta dicha corriente.
- Si existe reflexión de luz, el *fototransistor* es excitado y cierra el circuito (3-4) para pasar corriente y dar una señal Vcc en la entrada Arduino DUE (4) → E. Digital = 1.



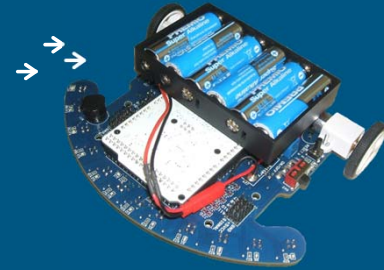
Robótica aplicada  
con Arduino



Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

Robótica aplicada con Arduino



# SENSORES DIGITALES

## Diseño electrónico

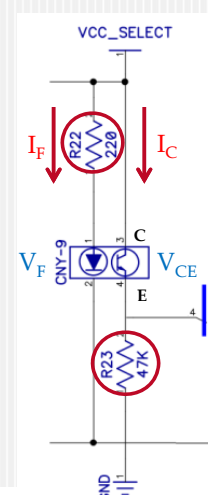
### Diseño del circuito

#### Características del CNY0 (Datasheet).

- ▣ Corriente de excitación del *fotodiodo*:  $I_F$
- ▣ Voltaje de caída en el *fotodiodo*:  $V_F$
- ▣ Corriente de colector:  $I_C$
- ▣ Voltaje de caída del *fototransistor*:  $V_{CE}$
- ▣ Modo *fototransistor* en saturación:  $V_{CEsat}$



Cálculo resistencia R22 y R23



Robótica aplicada  
con Arduino



Maestría Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## ■ Diseño del circuito

### ■ Características del CNY0 (Datasheet).

- Parámetro conocido  $d \approx 5$  mm.
- $I_c \approx 0,1-0,15$  mA.

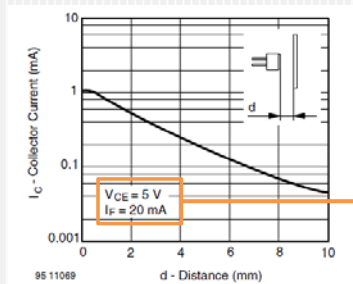
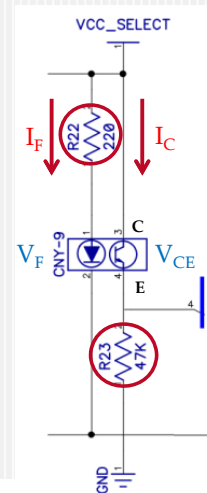


Fig. 9 - Collector Current vs. Distance

$V_{CE} = 5V$   
 $I_F = 20\text{ mA}$



Robótica aplicada  
con Arduino



Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
ua.es

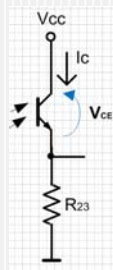
## ■ Diseño del circuito

### ■ Cálculo de la resistencia R23.

#### ▫ Datos.

- $I_c = 0,1$  mA.
- $V_{CEsat} = 0,3$  V.

$I_F = 20\text{ mA}, I_c = 0,1\text{ mA},$   
 $d = 0,3\text{ mm (figure 1)}$       0.3

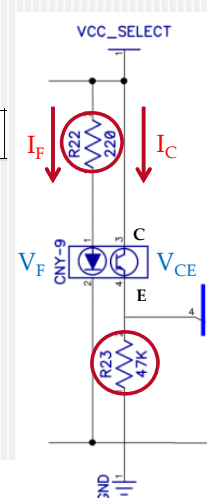


$$V_{CC} = V_{CEsat} + I_c \cdot R_{23}$$

$$R_{23} = \frac{5 - 0,3}{0,1 \cdot 10^{-3}} = 47000 \Omega = 47\text{ k}\Omega$$

Arduino DUE  $\rightarrow V_{CC} = 3,3\text{ V}.$

$$I_c = \frac{V_{CC} - V_{CEsat}}{R_{23}} = \frac{3,3 - 0,3}{47 \cdot 10^3} = 0,063\text{ mA}$$



Robótica aplicada  
con Arduino



Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

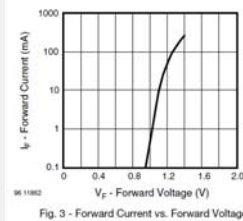
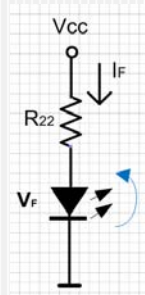
dfests  
ua.es

## ■ Diseño del circuito

### ■ Cálculo de la resistencia R22.

#### □ Datos.

- $I_F = 20 \text{ mA}$ .
- $V_F \approx 1,1 \text{ V}$ .

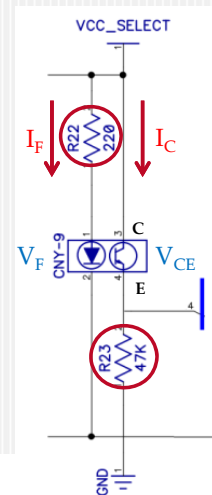


$$V_{CC} = I_F \cdot R_{22} + V_F$$

$$R_{22} = \frac{5 - 1,1}{20 \cdot 10^{-3}} = 195 \Omega \approx 220 \Omega$$

Arduino DUE →  $V_{CC} = 3,3 \text{ V}$ .

$$I_F = \frac{V_{CC} - V_F}{R_{22}} \approx 10 \text{ mA}$$



Robótica aplicada  
con Arduino

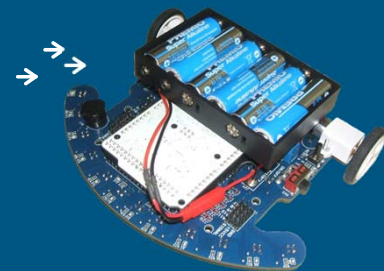


Máster Oficial en  
Automática y Robótica  
Universidad de Murcia

dfests  
ua.es



Robótica aplicada con Arduino



# SENSORES DIGITALES

Lectura del estado de los sensores

## ■ Lectura del estado de los sensores

- Sensores digitales.
  - Programación de Librería C++/Arduino DUE.
  - Lectura sensores digitales.
    - 9 sensores digitales línea central → Estado 0 (l. negra)/ 1 (l. blanca).
    - Lectura inversa: Estado 0 (l. negra) = **true** (existe línea).  
Estado 1 (l. blanca) = **false** (no existe línea).
  - Función de lectura.
    - Inicialización de variables para una nueva lectura.
    - Activación de los diodos infrarrojos (hacer pasar corriente por ellos).
    - Lectura de los 9 sensores ("for (int i=0;i<9;i++){)").
    - Desactivación de los diodos infrarrojos.
    - Devolución del estado.



Robótica aplicada  
con Arduino



## ■ Lectura del estado de los sensores

- Programación Librería C++.
  - Funciones a implementar.
    - Variables declaradas en .h 

```
unsigned char middleSensor [9];
bool middleSensorAux [9];
unsigned int middleSensorValue;
```
    - Inicialización .cpp (init()): 

```
middleSensor = { I_IR1,...,I_IR9 }
```
    - unsigned int ReadMiddleLine(): Lee los sensores centrales y los almacena en el vector *middleSensorAux* y en un entero *middleSensorValue* (valor que devuelve).
      - `bool middleSensorAux; unsigned int middleSensorValue;`
    - unsigned char getMiddleSensor (unsigned char sensor): Obtiene el valor leído de un sensor central.
    - ShowMiddleLine (bool oneLed): Marca en los leds el estado de los sensores infrarrojos.
      - Si se pasa **true** como parámetro se encienden cuando los sensores detectan blanco.
      - Si se pasa **false** como parámetro se encienden cuando los sensores detectan negro.



Robótica aplicada  
con Arduino



## ■ Lectura del estado de los sensores

### ■ Programación Librería C++.

#### ▫ Funciones a implementar.

- `unsigned int ReadMiddleLine()`: Lee los sensores centrales y los almacena en el vector `middleSensorAux` y en un entero `middleSensorValue` (valor que devuelve).
- `bool middleSensorAux; unsigned int middleSensorValue;`

```
unsigned int GoShield_GR_ua::ReadMiddleLine(){
    unsigned int val=0x01;
    middleSensorValue=0;
    //Activar los fotodiodos y un Delay (100 µs)
    ....
    //Bucle de lectura
    for (int i=0;i<9;i++){
        //Actualizar middleSensorAux[i] y middleSensorValue
    }
    //Apagar los fotodiodos
    ....
    return middleSensorValue;
}
```

```
unsigned char middleSensor [9];
bool middleSensorAux [9];
unsigned int middleSensorValue;
```



Robótica aplicada  
con Arduino



Master Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## ■ Lectura del estado de los sensores

### ■ Programación Librería C++.

#### ▫ Funciones a implementar.

- `unsigned char getMiddleSensor(unsigned char sensor)`: Obtiene el valor leído de un sensor central.

```
unsigned char GoShield_GR_ua::getMiddleSensor(unsigned char sensor){
    //Definición de variable de retorno
    unsigned char ret=0;
    //Comprobación estado en middleSensorAux a partir número sensor
    ....
    //Apagar los fotodiodos
    return ret;
}
```

#### Constates Types.h

```
static const unsigned char UNO = 1;
static const unsigned char DOS = 2;
....
```



Robótica aplicada  
con Arduino



Master Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es



## ■ Lectura del estado de los sensores

- Programación Librería C++.
- Funciones a implementar.
  - ShowMiddleLine (bool oneLed): Marca en los leds el estado de los sensores infrarrojos.
    - Si se pasa **true** como parámetro se encienden cuando los sensores detectan blanco.
    - Si se pasa **false** como parámetro se encienden cuando los sensores detectan negro.
    - Leds D4-D17 (12) y 9 sensores digitales. Variable **lineLEDS[]**.
      - Extremos a LOW (D4, D17) → **lineLEDS[0], lineLEDS[11]**.
      - Sensores digitales 1-4 (middleSensorAux[0] - middleSensorAux[3]) → **lineLEDS[1] - lineLEDS[4]**.
      - Sensor digital central (middleSensorAux[4]) → **lineLEDS[5] - lineLEDS[6]**.
      - Sensores digitales 6-9 (middleSensorAux[5] - middleSensorAux[8]) **lineLEDS[7] - lineLEDS[10]**.

### Ayuda

expresión booleana ? valor si cierto : valor si falso  
 (oneLed)? digitalWrite(\*\*\*,LOW) : digitalWrite(\*\*\*,HIGH);

```
void GoShield_GR_ua::ShowMiddleLine(bool oneLed){
  digitalWrite(lineLEDS[0], LOW);
  digitalWrite(lineLEDS[11], LOW);
  //Bucle encendido 4 primeros sensores
  ....
  //Encendido del Sensor central
  ....
  //Bucle encendido 4 últimos sensores
  ....
}
```



Robótica aplicada  
con Arduino



## ■ Lectura del estado de los sensores

- Programación Arduino DUE
  - Programa de llamada a las funciones de lectura de la librería

Librería

Objeto

Variables para el estado  
de los sensores



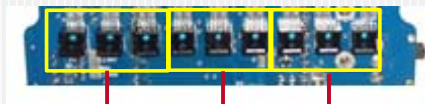
Robótica aplicada  
con Arduino





## ■ Primer sigue-líneas

- Sigue-líneas con los sensores centrales.
  - Implementación función *middle\_Follow()* en la librería.
    - A partir de la lectura de los sensores en la variable *int middleSensorValue*, decidir el giro del coche.



<i>middleSensorValue (bin)</i>	000000111	000111000	111000000
<i>middleSensorValue (hex)</i>	0x07	0x38	0x1C0
Giro coche	Izquierda	Recto	Derecha



Robótica aplicada  
con Arduino



Maestr Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## Primer sigue-líneas

- Sigue-líneas con los sensores centrales.
  - Implementación función *middle\_Follow()* en la librería.

```
void GoShield_GR_ua::middle_Follow(){
  //Centro
  if((middleSensorValue&0x38)>0){
    ....
    delayMicroseconds(50);
  }
  //Izquierda/Derecha
  else{
    ....
    if((middleSensorValue&0x07)>0){
      //Giro Izquierda
    }else{
      //Giro Derecha
    }
  }
}
```



Robótica aplicada  
con Arduino



## Primer sigue-líneas

- Sigue-líneas con los sensores centrales.
  - Implementación programa Arduino DUE.
    - Programa de dos estados: visualizar medidas de los sensores (SENSING) y seguir la línea (FOLLOWING).

```

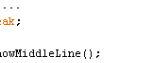
Middle_Follower $
#include <GoShield_GR_ua.h>
GoShield_GR_ua gr;
int estado=SENSING;
boolean boton=false;
int boton;

void setup(){
  boton=0;
  Serial.begin(9600);
  gr.setForwardSpeedRight(0);
  gr.setForwardSpeedLeft(0);
}

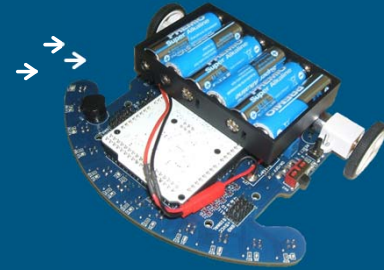
void loop(){
  int ir=gr.ReadMiddleLine();
  switch(estado){
    case SENSING:
      //....
      break;
    case FOLLOWING:
      //....
      break;
  }
  gr.ShowMiddleLine();
}
```



Robótica aplicada  
con Arduino

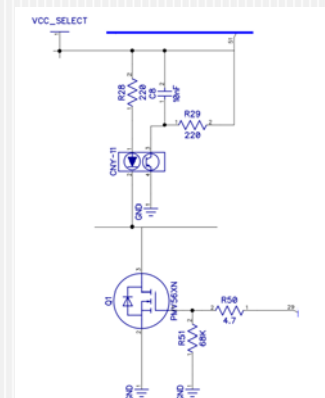
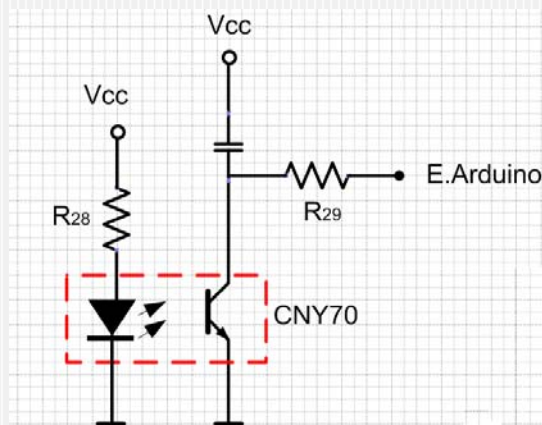


Robótica aplicada con Arduino



# SENSORES ANALÓGICOS

## Sensores Analógicos



Robótica aplicada  
con Arduino

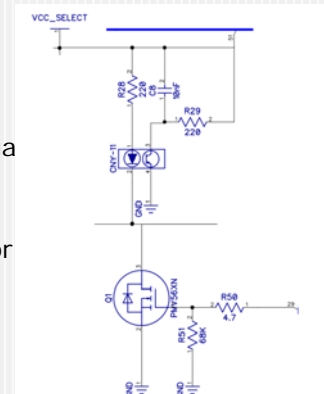


Master Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## Sensores Analógicos

- Los sensores delanteros están montados de forma diferente. Lo que se pretende es medir el tiempo de carga del condensador C8 de forma que según cuanto más excitado se encuentra el sensor más rápido se carga el condensador.
- El tiempo de carga de un condensador depende de la corriente que permite descargar el fotoreceptor (su valor máximo oscila alrededor de un miliamperio).
- Al igual que en el caso anterior al activar la salida 29 se activa la línea de sensores. En caso de estar en alta impedancia, esta fila de sensores permanece inactiva.



Robótica aplicada  
con Arduino

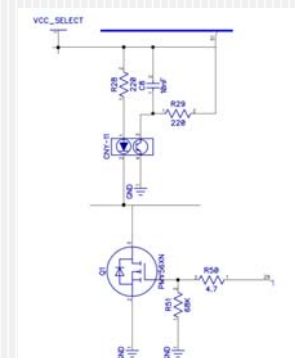


Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

## 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, vease "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 el sensor valga estado alto
    1. Leer sensor
    2. Si el estado es alto
      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



Máster Oficial en  
Automática y Robótica  
Universidad de Alicante

dfests  
.ua.es

