Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Diseño e implementación del control de motor paso a paso mediante dispositivos embebidos RIO

(ANEXOS)

Javier Gómez Pindado

Grado en Ingeniería Eléctrica

Tutor: Moisés San Martín Ojeda

Código:01003

Septiembre 2018

Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Anexo 1- Manual de uso

Anexo 1. Manual de uso

1. Introducción al manual

Mediante este circuito impreso en el que se encuentran el Arduino Nano, los pulsadores, el controlador ULN2803A y los diodos led de señalización se realizará el control del motor, para ello será necesario establecer las conexiones adecuadas y activar los pulsadores correctos para cada función.

Además, se crea un ejecutable de LabVIEW desde el que se controlará el motor mediante pulsadores y controles numéricos, utilizando para ello el puerto serie USB.

2. Conexiones:

Para el control físico únicamente será necesario conectar la fuente de alimentación y el motor a la placa. Sin embargo, para el control mediante LabVIEW habrá que realizar las tres conexiones que se detallan a continuación:

- Alimentación: la conexión se realizará como viene detallado en el esquema.

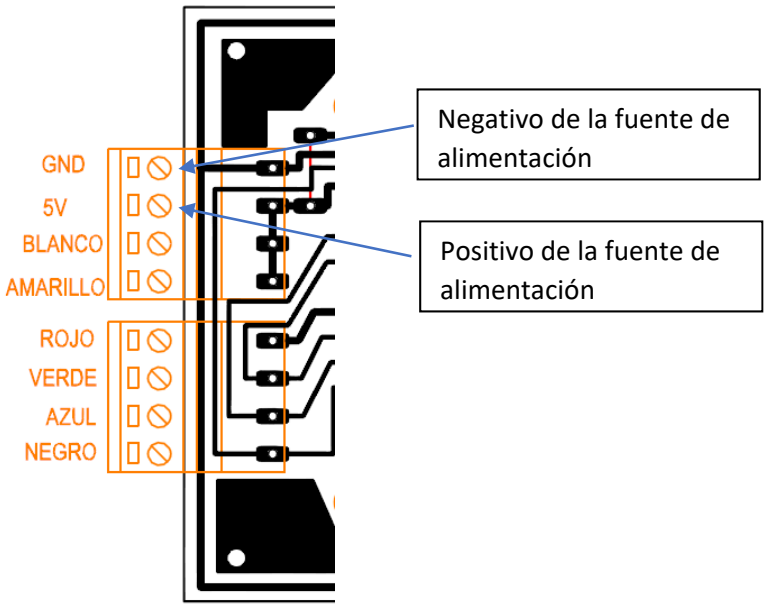


Figura 1 Conexión alimentación. Fuente: elaboración propia.

- Motor: se conectarán los cables del motor de tal forma que el color del cable corresponda con el color que aparece en el esquema.
- Ordenador personal vía USB: para realizar la conexión con el ordenador en primer lugar se tendrán que haber instalado los *drivers* necesarios para que el dispositivo del puerto serie sea reconocido.

Como en este caso estamos utilizando un Arduino Nano la conexión se realizará con un cable USB Mini-B.

Figura 2 USB Mini B. (2018). Recuperado de: https://tiendas.mediamarkt.es/p/nanocable-cable-usb-2-0-tipo-a-m-mini-us-1395966?gclid=EAIaIQobChMIzNrt0bKz3QIVhKwYCh1ztALMEAQYAiABEgKYIvD_BwE&gclsrc=aw.ds&dclid=CN2AoYazs90CFW2jUQodNbgGgg

3. Dispositivo físico de control:

El dispositivo físico de control consiste en una botonera compuesta por 7 pulsadores, tres de ellos se utilizan para realizar el giro hacia ambos sentidos y parar el motor. Los cuatro pulsadores restantes se utilizarán para alimentar individualmente cada uno de los pares de bobinas del motor. De esta forma hay dos formas de uso diferenciadas: el motor girando de manera autónoma controlado por la programación embebida en el Arduino y otra en la que se activan las secuencias de manera totalmente manual. Para pasar de una a la otra es preciso pasar por el botón de paro.

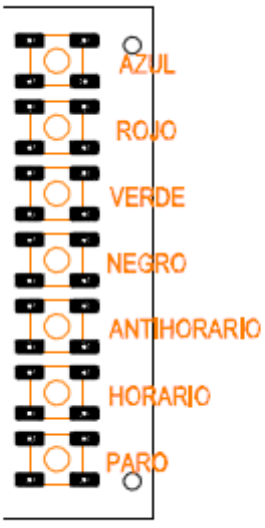


Figura 3 Botonera placa de control. Fuente: elaboración propia

Para girar en sentido horario bastará con pulsar el botón correspondiente. En cualquier momento se pueden pulsar tanto el botón de paro como el de giro

antihorario y el motor responderá a la señal. Al pulsar el botón de paro el motor parará y la alimentación se interrumpirá.

En el caso de querer activar la secuencia de manera manual las dos secuencias serán:

- Sentido antihorario: Negro-Rojo-Verde-Azul- Negro-Rojo-Verde-Azul…
- Sentido horario: Negro-Azul-Verde-Rojo- Negro-Azul-Verde-Rojo…

En caso de que se quisiera realizar el avance con medio paso:

- Sentido antihorario: Negro Rojo-Rojo-Rojo Verde- Verde - Verde Azul- Azul- Azul Negro…
- Sentido horario: Negro Azul- Azul- Azul Verde- Verde - Verde Rojo-Rojo-Rojo Negro…

En todo momento se encenderá el diodo led correspondiente al par de bobinas alimentado.

4. Interfaz Labview:

4.1 Ajustes iniciales:

En primer lugar, se deberá conectar el dispositivo al ordenador vía USB, una vez conectado habrá que seguir los siguientes pasos:

1. Buscar en que puerto COM está nuestro dispositivo, para ello habrá que acceder en nuestro ordenador a Panel de control→ Dispositivos→Otros dispositivos y ahí aparecerá nuestro dispositivo:

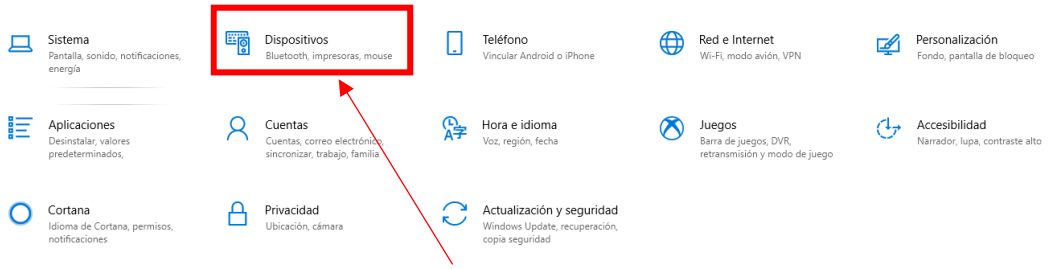


Figura 4 Panel de control Windows 10. Fuente: elaboración propia

Figura 5 Panel de control, dispositivos. Windows 10. Fuente: elaboración propia

2. A continuación, en la interfaz de LabView seleccionamos como *VISA resource name* nuestro COMxx:

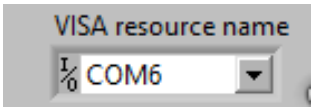


Figura 6 VISA resource name. Fuente: elaboración propia

3. Se fija el valor de los pasos por vuelta del motor

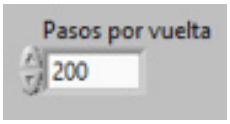


Figura 7 Pasos por vuelta. Fuente: elaboración propia

4. Por último, activamos el botón *Highlight Execution* (bombilla) de la barra de herramientas del diagrama de bloques y ejecutamos el programa para asegurar que la comunicación entre el programa y el dispositivo es correcta. Tendremos que obtener la siguiente respuesta:

4.2 Prestaciones en pantalla:

Una vez realizados los ajustes previos se podrá ejecutar de manera normal el programa LabVIEW.

En la interfaz de LabVIEW contaremos con 4 botones (paro, sentido horario, sentido antihorario y el último será para variar la velocidad), un control numérico que servirá para fijar el nuevo valor de la velocidad, dos indicadores, en uno obtendremos la lectura del buffer de nuestro dispositivo y el otro simulará el motor y girará en función de los pasos que el motor haya dado.

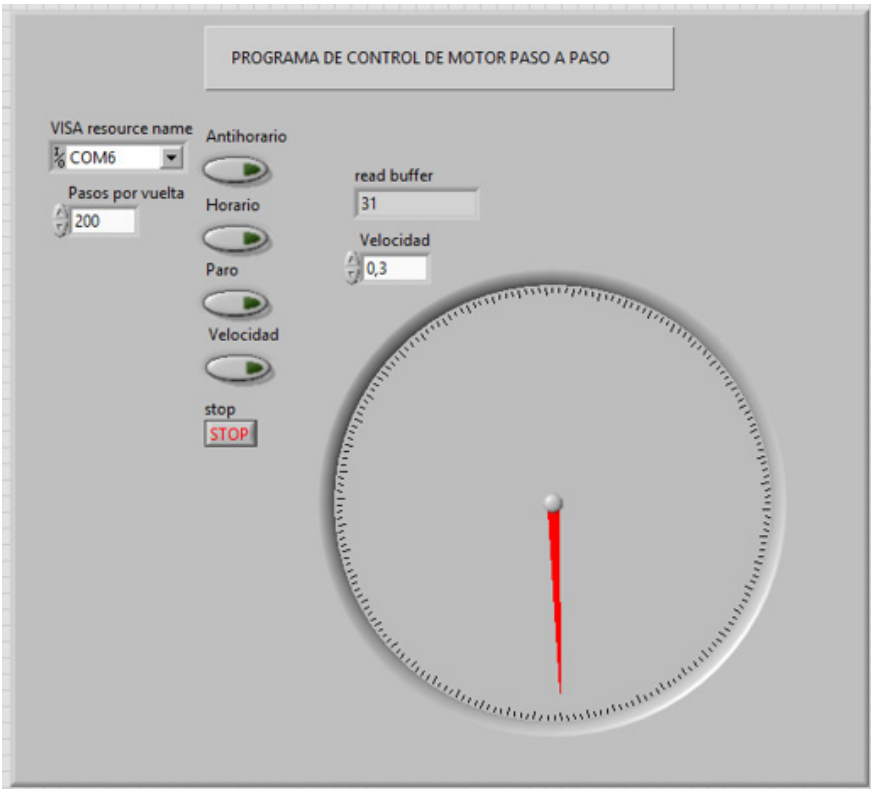


Figura 8 Panel frontal. Fuente: elaboración propia

Los botones de paro, giro horario y giro antihorario tienen la misma función que sus homónimos físicos. Por lo que se utilizaran de la misma manera.

Para variar la velocidad en primer lugar habrá que escribir el valor en el control numérico y a continuación pulsar el botón para enviar el dato al dispositivo.

Tanto el control físico como el control vía LabVIEW pueden actuar de forma simultánea, no obstante, habrá que tomar las mismas precauciones que en el control físico si se quiere activar la secuencia de forma manual.

5. Resolución de problemas

Al abrir el puerto serie del dispositivo, debido a una característica intrínseca de las placas Arduino, este se reseteará por lo que volverá a valores de inicio (motor parado y contador a 0). Una vez establecida conexión se podrá volver a controlar de manera habitual.

En el caso de que sin estar conectado el dispositivo al puerto serie no responda habrá que retirar la tapa de la envolvente y pulsar el botón de RESET que se encuentra en el centro del Arduino Nano.

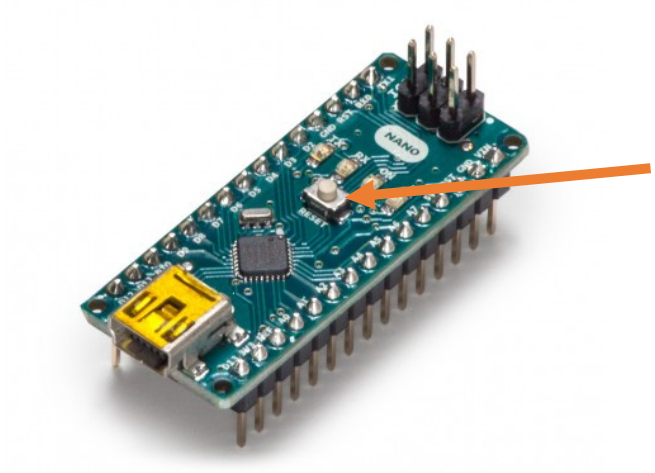


Figura 9 Arduino Nano(2018) Recuperado de Recuperado de https://store.arduino.cc/arduino-nano

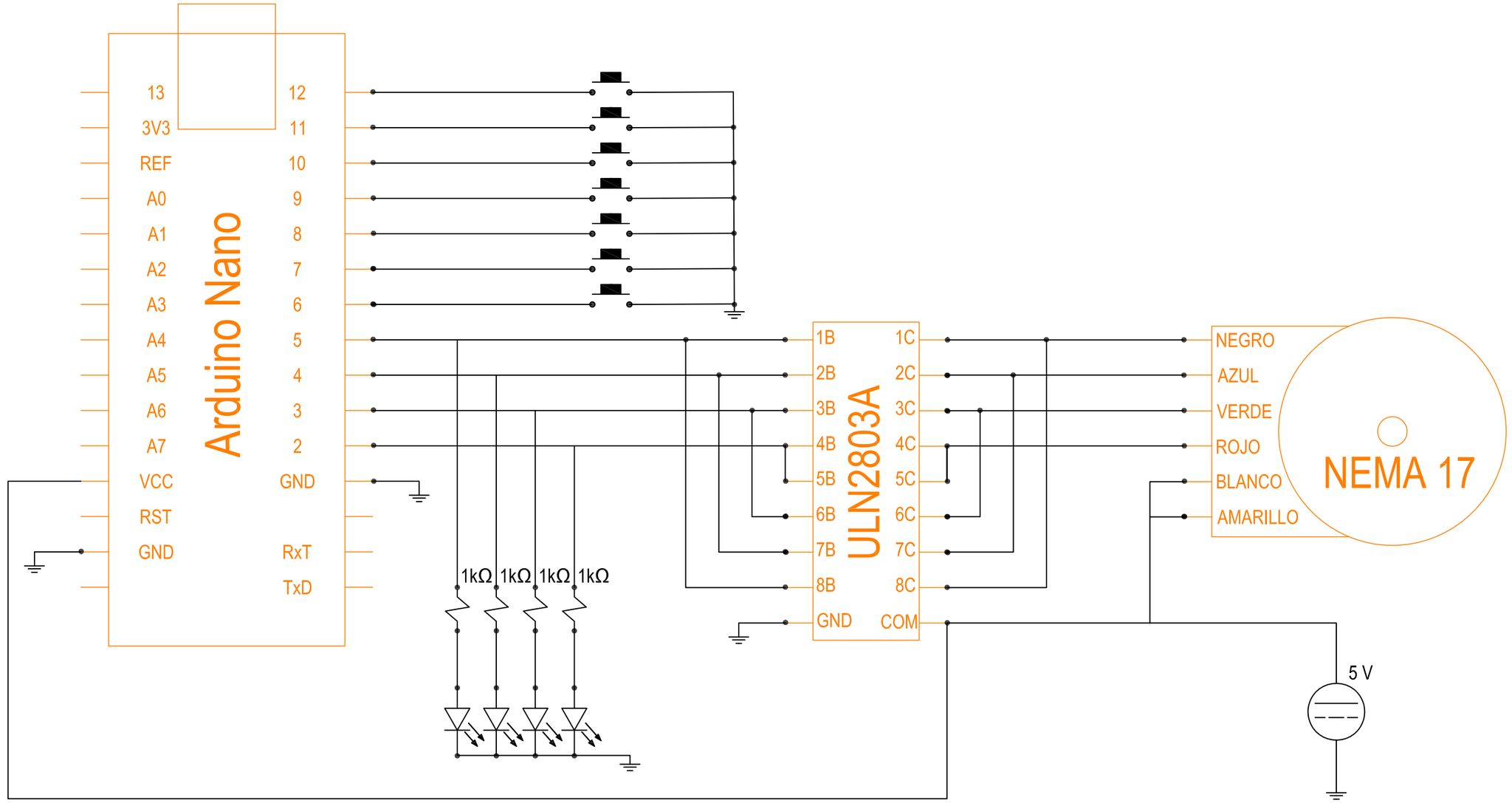
Si hay problemas con la comunicación serie en primer lugar hay que cerrar el programa de LabVIEW y pulsar el botón de RESET. Si el error sigue persistiendo, habrá que desconectar y conectar el USB.

Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Anexo 2 Planos

Índice



Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Trabajo Fin de Grado

Diseño e implementación del control de motor paso a paso mediante dispositivos embebidos RIO

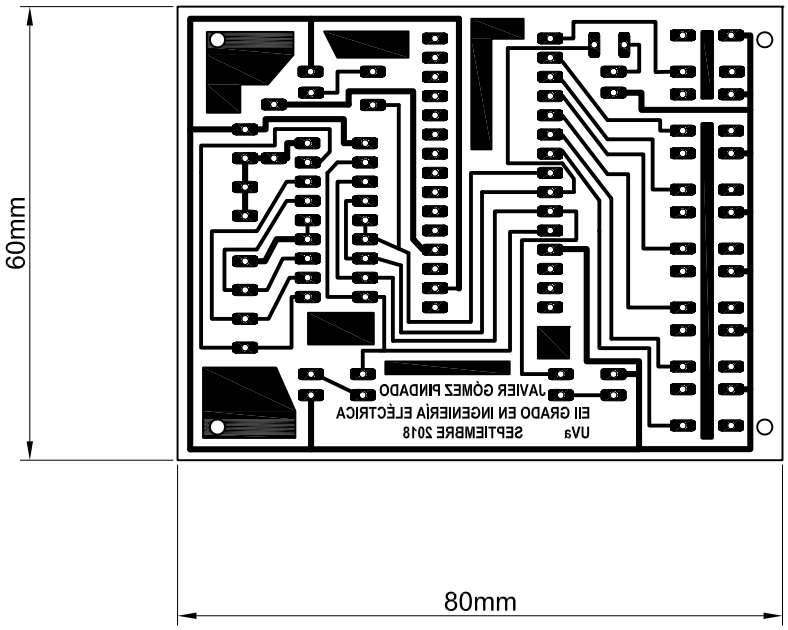
Autor: Javier Gómez Pindado
Tutor: Moisés Luis San Martín Ojeda

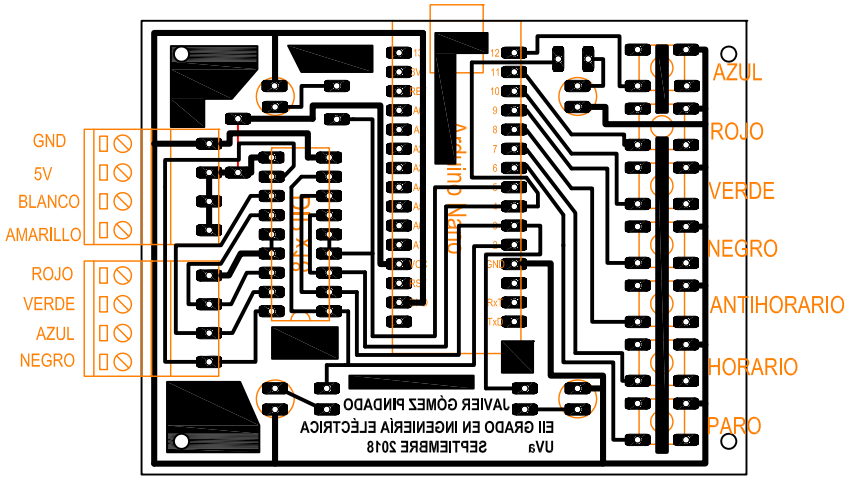
Título del plano: Esquema eléctrico

Dibujado por	Escala
JGP	S/E
Formato	**Nº de plano**
DIN A4	1

FORMATO ORIGINAL A3 (420 x 297)

0 150 MM





Título del plano	Dibujado por	Escala	Formato	Nº de plano
Placa del circuito impreso	JGP	1:1	DIN A4	2

Universidad de Valladolid

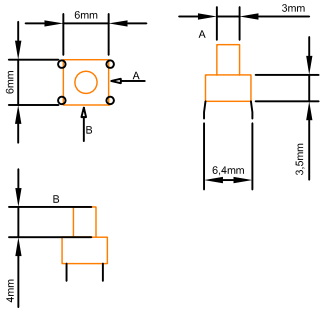
ESCUELA DE INGENIERÍAS INDUSTRIALES

Trabajo Fin de Grado

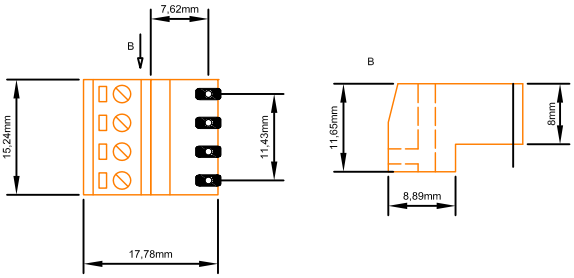
Diseño e implementación del control de motor paso a paso mediante dispositivos embebidos RIO

Autor: Javier Gómez Pindado
Tutor: Moisés Luis San Martín Ojeda

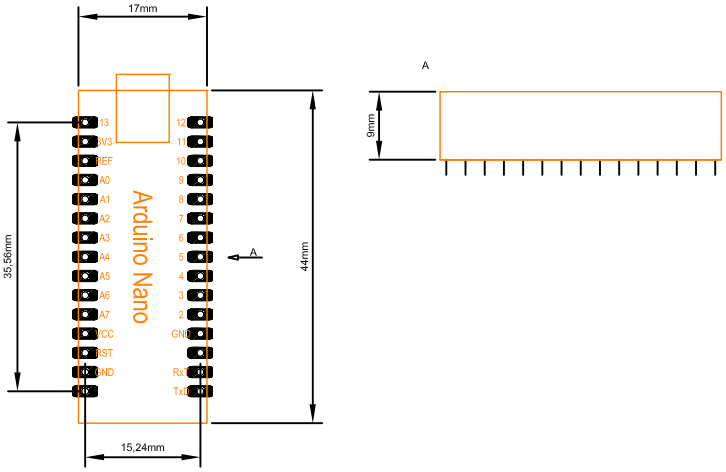
Pulasdor



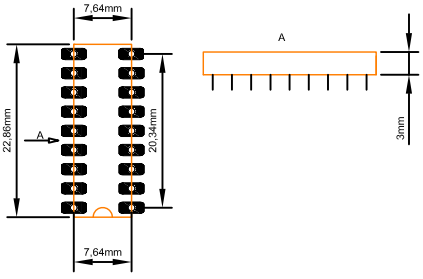
Conector PCB



Arduino Nano+ Zócalo



Zócalo DIP18



Título del plano	Dibujado por	Escala	Formato	Nº de plano
Piezas de la placa de circuito impreso	JGP	1:1	DIN A4	3

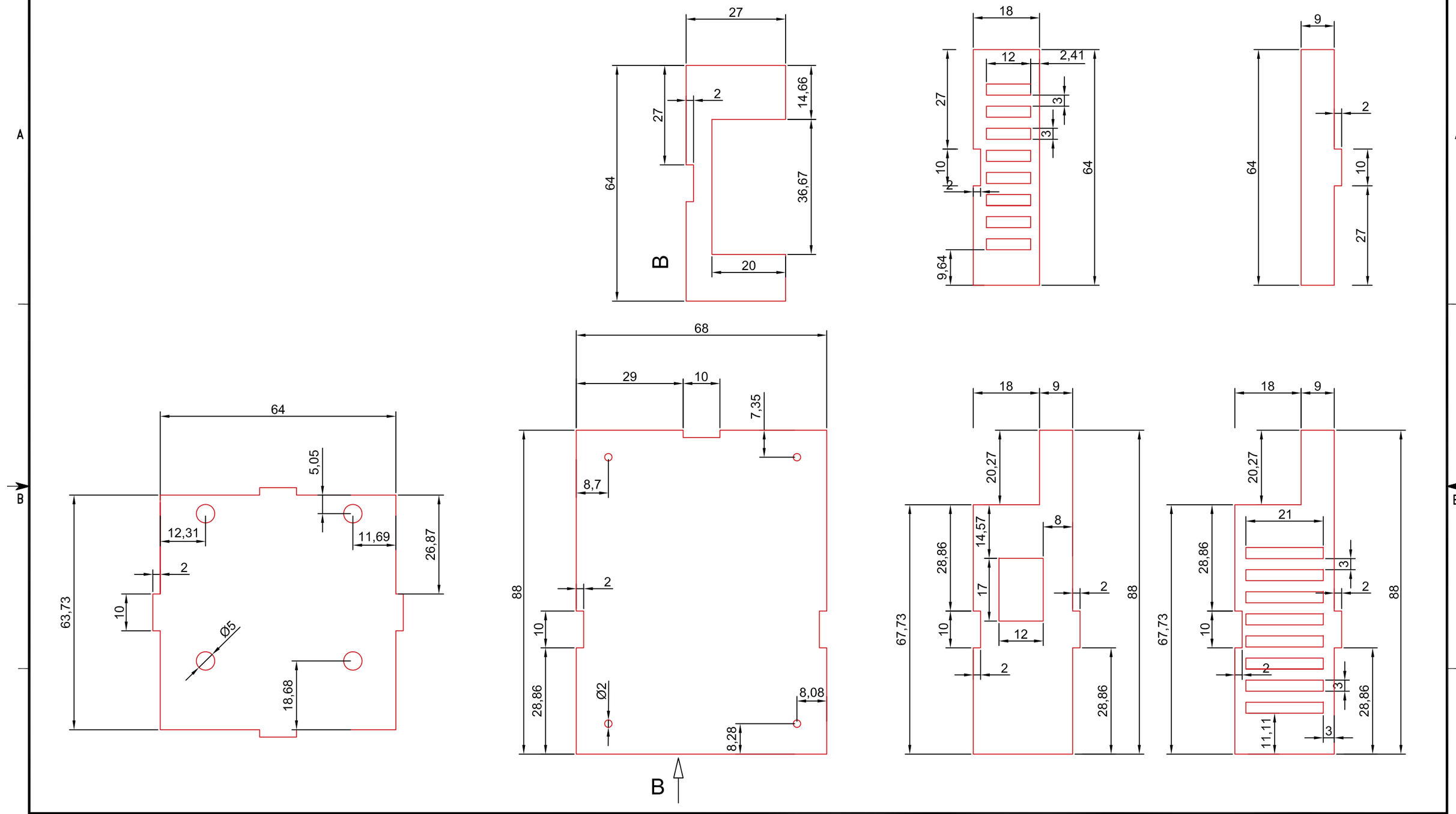
Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Trabajo Fin de Grado

Diseño e implementación del control de motor paso a paso mediante dispositivos embebidos RIO

Autor: Javier Gómez Pindado
Tutor: Moisés Luis San Martín Ojeda



Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Trabajo Fin de Grado

Diseño e implementación del control de motor paso a paso mediante dispositivos embebidos RIO

Autor: Javier Gómez Pindado

Tutor: Moisés Luis San Martín Ojeda

Título del plano: Caja envolvente

Dibujado por	Escala
JGP	1:1
Formato	**Nº de plano**
DIN A4	4

DATA

FORMATO ORIGINAL A3 (420 x 297)

0 150 MM

Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Anexo 3- Hojas de datos

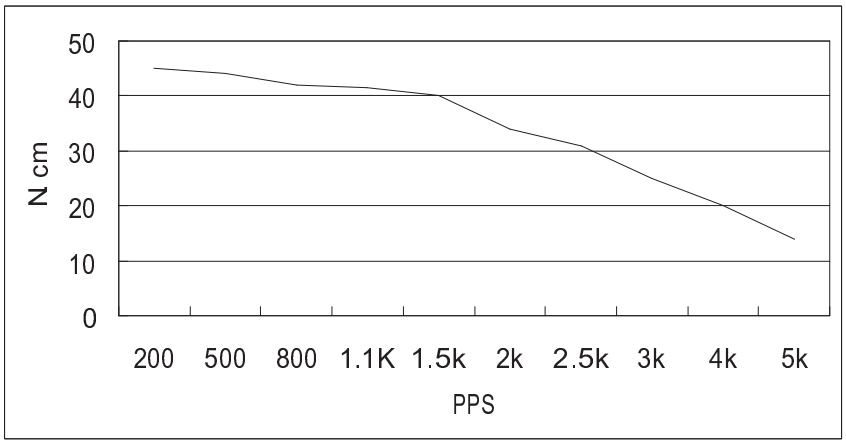
Índice

HIGH TORQUE HYBRID STEPPING MOTOR SPECIFICATIONS

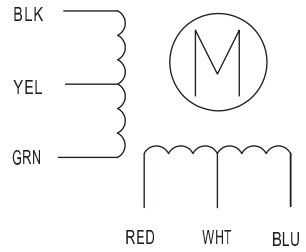
General specifications		Electrical specifications	
Step Angle (°)	1.8	Rated Voltage (V)	4
Temperature Rise (°C)	80 Max (rated current.2 phase on)	Rated Current (A)	1.2
Ambient temperature (°C)	-20～+50	Resistance Per Phase (±10%Ω)	3.3 (25°C)
Number of Phase	2	Inductance Per Phase (±20%mH)	2.8
Insulation Resistance	100MΩ, Min (500VDC)	Holding Torque (Kg.cm)	3.17
Insulation Class	Class B	Detent Torque (g.cm)	200
Max.radial force (N)	28 (20mm from the flange)	Rotor Inertia (g. cm^2)	68
Max.axial force (N)	10	Weight (Kg)	0.365

● Pull out torque curve:

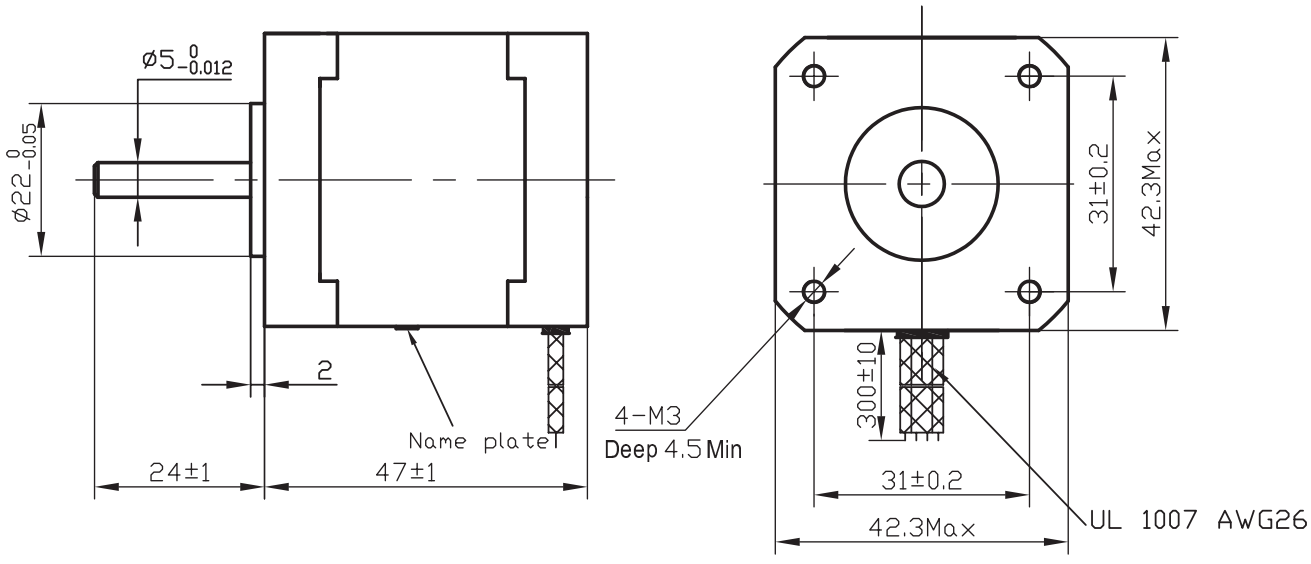
VOLTAGE: 24VDC, CONSTANT CURRENT: 1.2A, HALF STEP



● Wiring Diagram:



● Dimensions:
(unit=mm)



REV	REVISIONS	DESCRIPTION	BY	DATE
2				
1				
DRAW	任飞飞 2010.06.29			
CHECK				
APPROVE				

SY42STH47-1206A

TECHNICAL CONDITIONS

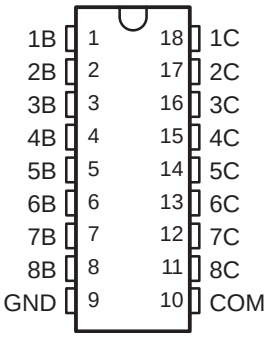
CHANGZHOU SONGYANG MACHINERY & ELECTRONICS NEW TECHNIC INSTITUTE

060047000

SLRS049C − FEBRUARY1997 − REVISED AUGUST 2004

- **500-mA Rated Collector Current (Single Output)**
- **High-Voltage Outputs . . . 50 V**
- **Output Clamp Diodes**
- **Inputs Compatible With Various Types of Logic**
- **Relay Driver Applications**
- **Compatible with ULN2800A Series**

DW OR N PACKAGE
(TOP VIEW)



description/ordering information

The ULN2803A is a high-voltage, high-current Darlington transistor array. The device consists of eight npn Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of each Darlington pair is 500 mA. The Darlington pairs may be connected in parallel for higher current capability.

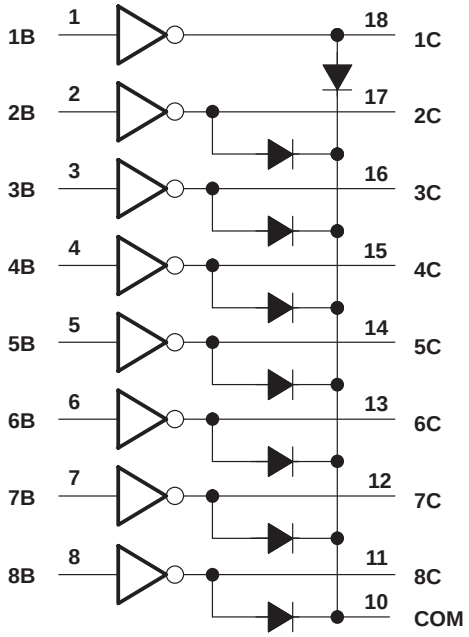
Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. The ULN2803A has a 2.7-kΩ series base resistor for each Darlington pair for operation directly with TTL or 5-V CMOS devices.

ORDERING INFORMATION

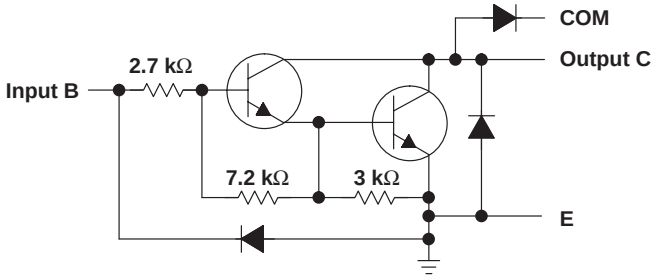
T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
−40°C to 85°C	PDIP (N)	Tube of 20	ULN2803AN	ULN2803AN
	SOIC (DW)	Tube of 40	ULN2803ADW	ULN2803A
		Reel of 2000	ULN2003ADWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

logic diagram



schematic (each Darlington pair)



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Collector-emitter voltage . . . . . 50 V
Input voltage (see Note 1) . . . . . 30 V
Continuous collector current . . . . . 500 mA
Output clamp diode current . . . . . 500 mA
Total substrate-terminal current . . . . . −2.5 A
Package thermal impedance, θ_{JA} (see Notes 2 and 3): DW package . . . . . TBD°C/W
N package . . . . . TBD°C/W
Operating virtual junction temperature, T_J . . . . . 150°C
Storage temperature range, T_{stg} . . . . . −65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltage values, unless otherwise noted, are with respect to the emitter/substrate terminal GND.
2. Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
I_{CEX}	Collector cutoff current	V_{CE} = 50 V, See Figure 1	I_I = 0,			50	µA
$I_{I(off)}$	Off-state input current	V_{CE} = 50 V, T_A = 70°C,	I_C = 500 µA, See Figure 2	50	65		µA
$I_{I(on)}$	Input current	V_I = 3.85 V,	See Figure 3		0.93	1.35	mA
$V_{I(on)}$	On-state input voltage	V_{CE} = 2 V, See Figure 4	I_C = 200 mA			2.4	V
			I_C = 250 mA			2.7	
			I_C = 300 mA			3	
$V_{CE(sat)}$	Collector-emitter saturation voltage	I_I = 250 µA, See Figure 5	I_C = 100 mA,		0.9	1.1	V
		I_I = 350 µA, See Figure 5	I_C = 200 mA,		1	1.3	
		I_I = 500 µA, See Figure 5	I_C = 350 mA,		1.3	1.6	
I_R	Clamp diode reverse current	V_R = 50 V,	See Figure 6			50	µA
V_F	Clamp diode forward voltage	I_F = 350 mA,	See Figure 7		1.7	2	V
C_i	Input capacitance	V_I = 0 V,	f = 1 MHz		15	25	pF

switching characteristics at 25°C free-air temperature

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low- to high-level output	V_S = 50 V, C_L = 15 pF,	R_L = 163 Ω, See Figure 8		130		ns
t_{PHL}	Propagation delay time, high- to low-level output				20		
V_{OH}	High-level output voltage after switching	V_S = 50 V, See Figure 9	$I_O \approx$ 300 mA,	V_S − 20			mV

PARAMETER MEASUREMENT INFORMATION

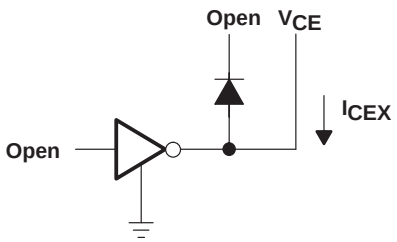


Figure 1. I_{CEX} Test Circuit

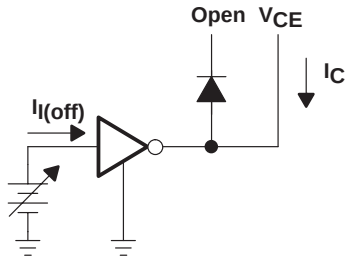


Figure 2. $I_{I(off)}$ Test Circuit

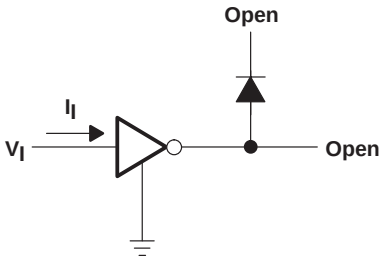


Figure 3. $I_{I(on)}$ Test Circuit

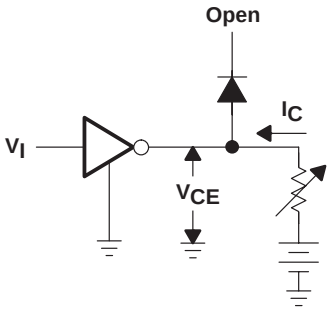


Figure 4. $V_{I(on)}$ Test Circuit

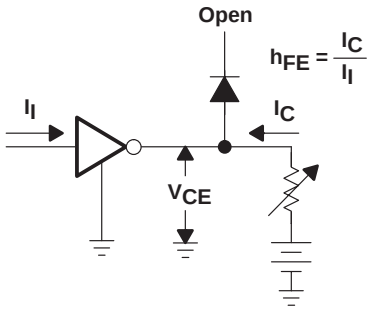


Figure 5. h_{FE}, $V_{CE(sat)}$ Test Circuit

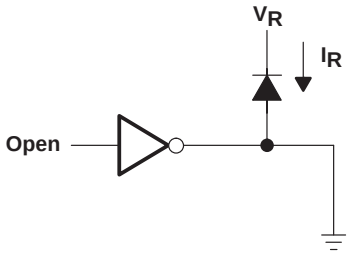


Figure 6. I_R Test Circuit

PARAMETER MEASUREMENT INFORMATION

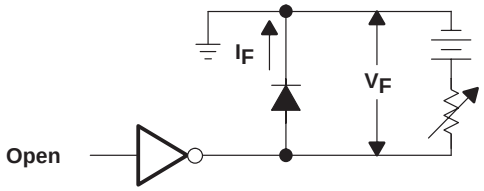
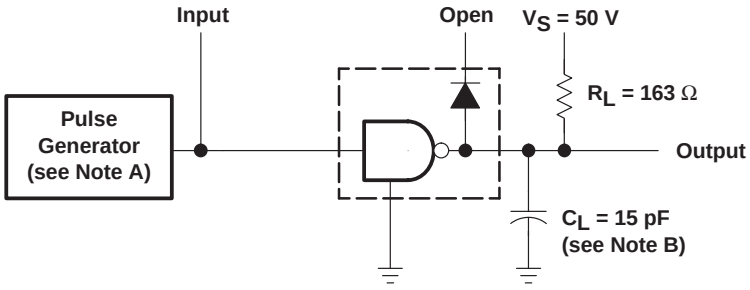
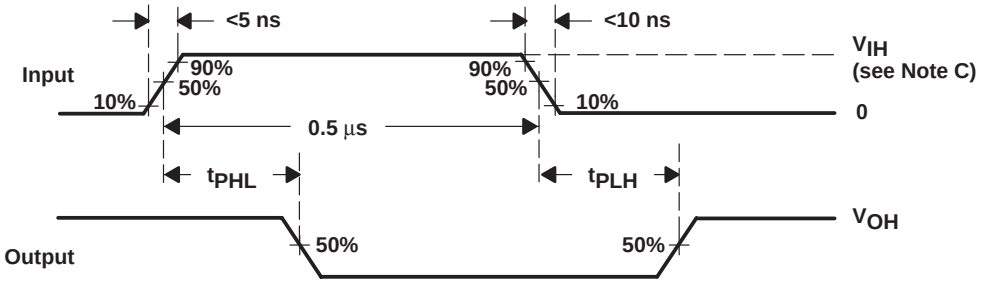


Figure 7. V_F Test Circuit



Test Circuit

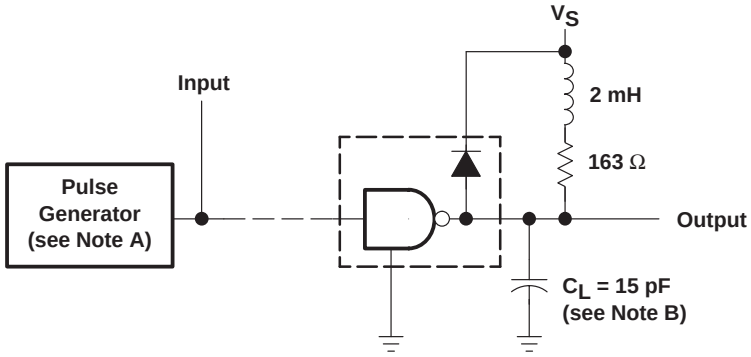


Voltage Waveforms

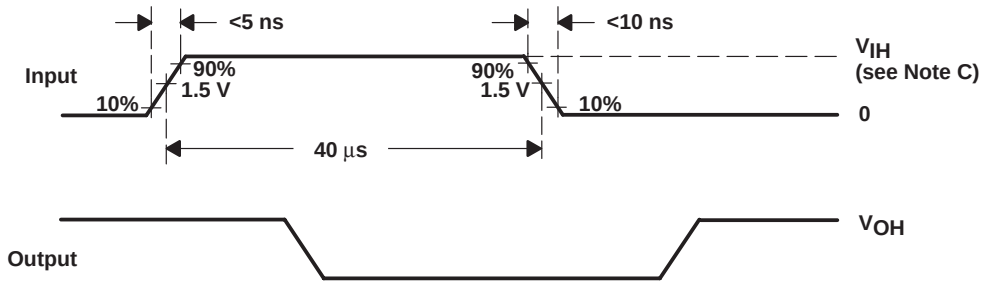
NOTES: A. The pulse generator has the following characteristics: PRR = 1 MHz, Z_O = 50 Ω.
B. C_L includes probe and jig capacitance.
C. V_{IH} = 3 V

Figure 8. Propagation Delay Times

PARAMETER MEASUREMENT INFORMATION



Test Circuit

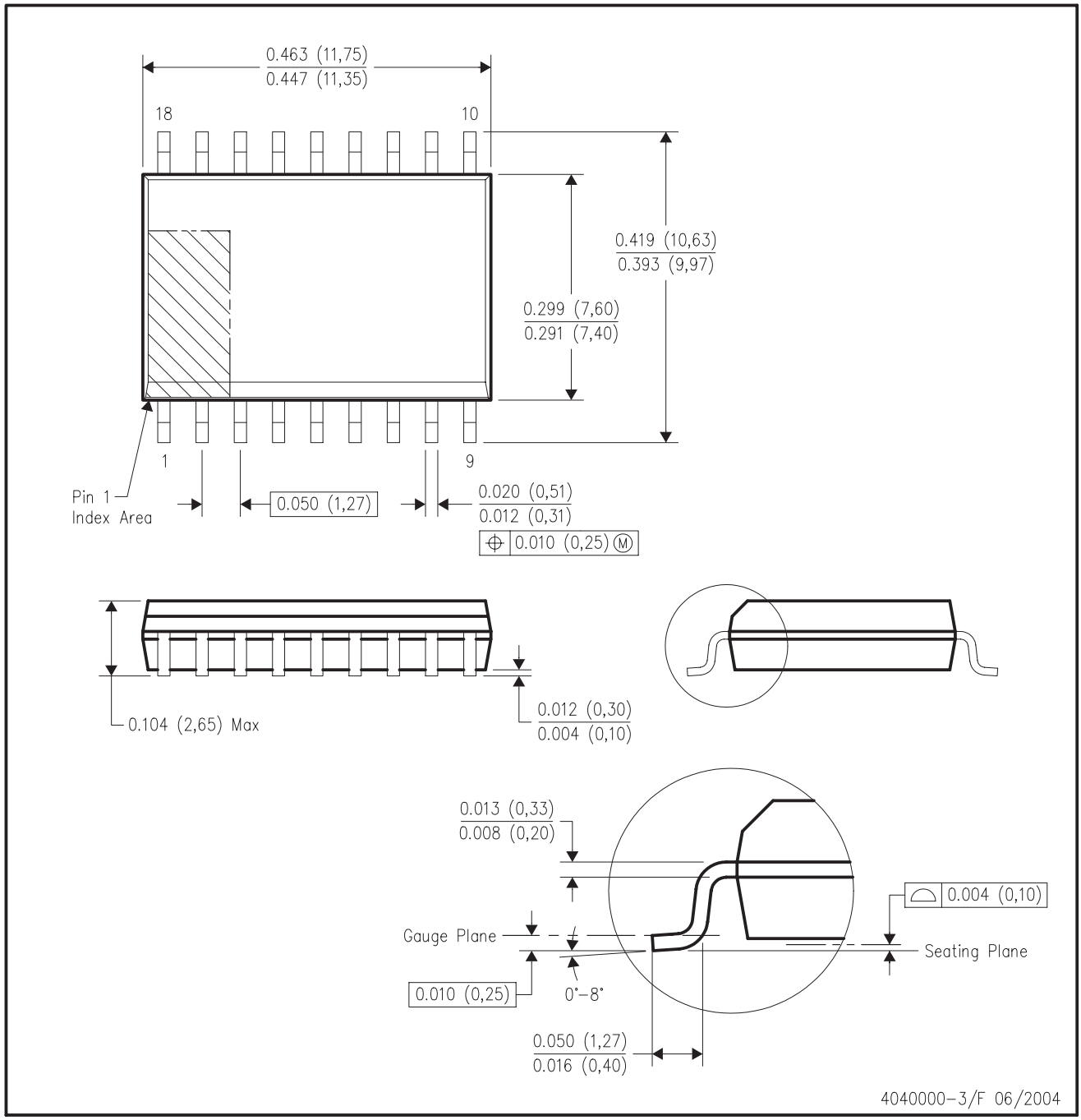


Voltage Waveforms

NOTES: A. The pulse generator has the following characteristics: PRR = 12.5 KHz, Z_O = 50 Ω.
B. C_L includes probe and jig capacitance.
C. V_{IH} = 3 V

Figure 9. Latch-Up Test

DW (R–PDSO–G18) PLASTIC SMALL–OUTLINE PACKAGE

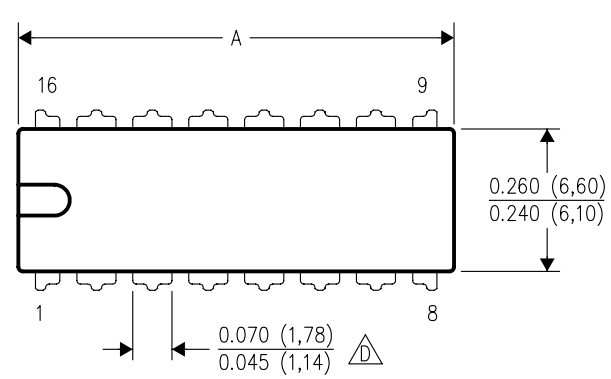


NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS–013 variation AB.

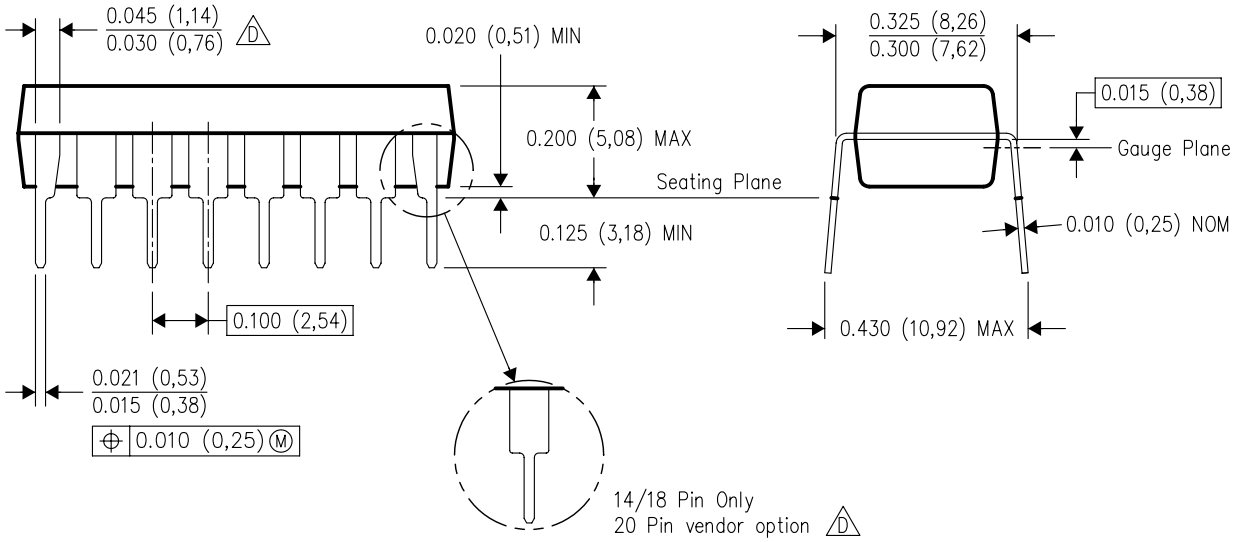
N (R–PDIP–T**)

PLASTIC DUAL–IN–LINE PACKAGE

16 PINS SHOWN



DIM \ PINS **	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
MS–001 VARIATION (C)	AA	BB	AC	AD



4040049/E 12/2002

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS–001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

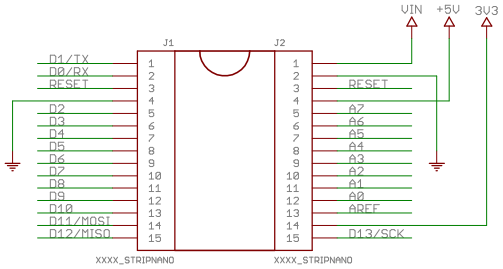
Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

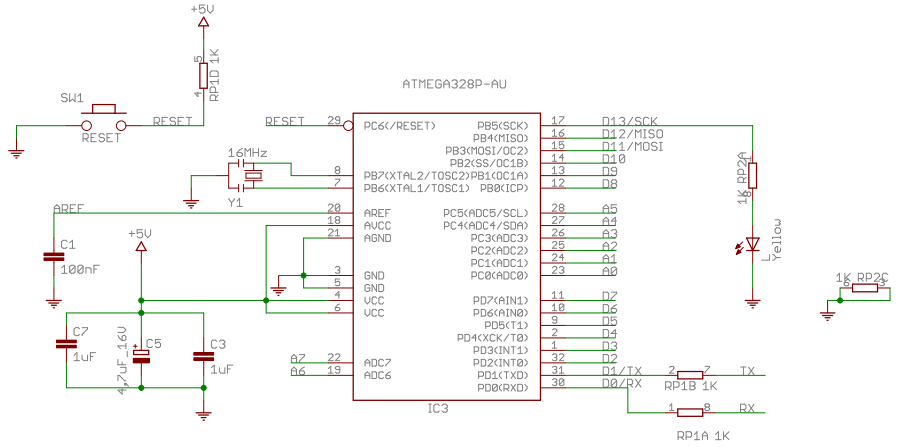
Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

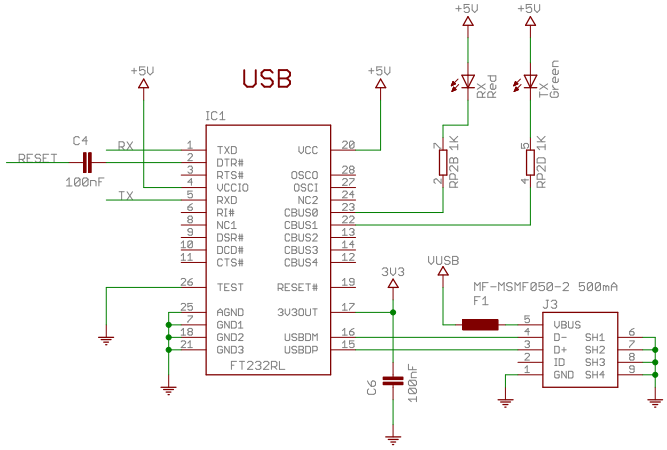
Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

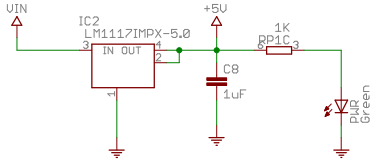
Based on design by Gravitech (gravitec.us)

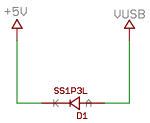
Arduino Nano



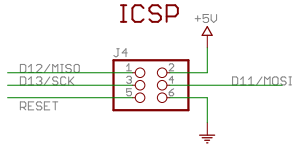


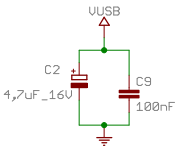






+5V AUTO SELECTOR





Author: E.Vita	
TITLE: Arduino Nano-Rev3.2	
Document Number:	REV: 3.2
Date: 30.12.2014 15:30:26	Sheet: 1/1

Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

Anexo 4- Código

```
/* Programa para controlar un motor paso a paso
 * mediante pulsadores.
 *
 * Además se podrá controlar mediante puerto serie
 * utilizando los comandos adecuados.
 *
 * Javier Gómez Pidnado
 * EII Uva Grado en Ingeniería Eléctrica
 * Septiembre 2018
 */

#define black 2     //Pin que alimenta la fase NEGRA
#define green 4     //Pin que alimenta la fase VERDE
#define red 3       //Pin que alimenta la fase ROJA
#define blue 5      //Pin que alimenta la fase AZUL
#define paro 6      //Pin que lee el pulsador de paro
#define horario 7   //Pin que lee el pulsador de giro horario
#define antihorario 8 //Pin que lee el pulsador de giro antihorario
#define negro 9     //Pin que lee el pulsador para la fase NEGRA
#define verde 10    //Pin que lee el pulsador para la fase VERDE
#define rojo 11     //Pin que lee el pulsador para la fase ROJA
#define azul 12     //Pin que lee el pulsador para la fase AZUL

int dir=0, cont;
int vel=7812;
char comandoTexto[10];
volatile boolean chivato=false;
int bobina=0;
boolean led = false;
```

```
void setup(){
  Serial.begin(9600);
  pinMode(black, OUTPUT);
  pinMode(green,OUTPUT);
  pinMode(red,OUTPUT);
  pinMode(blue,OUTPUT);
  pinMode(horario,INPUT_PULLUP);
  pinMode(antihorario,INPUT_PULLUP);
  pinMode(paro,INPUT_PULLUP);
  pinMode(negro,INPUT_PULLUP);
  pinMode(verde,INPUT_PULLUP);
  pinMode(rojo,INPUT_PULLUP);
  pinMode(azul,INPUT_PULLUP);

  // Paramos todas las interrupciones antes de contigurar un timer.
  noInterrupts();
  // El registro de control A queda todo en 0.
  TCCR1A = 0;
  // Activamos el modo CTC en Timer1.
  TCCR1B = 0;
  TCCR1B |= (1 << WGM12);
  //Prescaler 1024: 16.000.000/1024=15625 tics/s.
  //Se genera un tic cada 1024 ciclos.
  TCCR1B |= (1 << CS12) | (1 << CS10);
  // Inicializamos el contador en 0
  TCNT1 = 0;
  // Registro comparador 7812 equivale a 0,5s.
  OCR1A = 7812;
  // Inicializamos el comparador para el registro A.
  TIMSK1 |= (1 << OCIE1A);
```

```
  // Activamos interrupciones nuevamente.
  interrupts();
}



ISR(TIMER1_COMPA_vect) {
chivato=true;
 }

void interpreta(char *comando){
 switch(comando[0]){
   case 'a':      // Giro horario.
     dir=2;
     if(bobina==0) bobina=2;
     break;
   case 'b':      // Giro antihorario.
     dir=1;
     if(bobina==0) bobina=2;
     break;
   case 'c':     // Paro.
     dir=0;
     break;
   case 'd':
     dir=0;
     cont=0;
     break;
   case 'v':      // Cambiar velocidad (valores entre 391 y 9999)
     if (comando[4]>0){
     vel   =   2*((comando[4]-'0')*1000+(comando[3]-'0')*100+   (comando[2]-
'0')*10+(comando[1]-'0'));
     }
```

```
      else{
        vel = 2*((comando[3]-'0')*100+ (comando[2]-'0')*10+(comando[1]-'0'));
      }
      break;
 }
}
void loop() {
if(digitalRead(7)==0){  //Giro horario
  dir=2;
  if(bobina==0) bobina=2;
}
if(digitalRead(8)==0){ //Giro antihorario
  dir=1;
  if(bobina==0) bobina=2;
}
if(digitalRead(6)==0) dir=0; //Paro
if(dir==0){
  digitalWrite(bobina,LOW);
  bobina=0;
}

//Activación de secuencia mediante pulsadores (previo paso por paro).
if(digitalRead(negro)==0 && dir==0) digitalWrite(black,HIGH);
if(digitalRead(negro)==1 && dir==0) digitalWrite(black,LOW);

if(digitalRead(verde)==0 && dir==0) digitalWrite(green,HIGH);
if(digitalRead(verde)==1 && dir==0) digitalWrite(green,LOW);

if(digitalRead(rojo)==0 && dir==0) digitalWrite(red,HIGH);
if(digitalRead(rojo)==1 && dir==0) digitalWrite(red,LOW);
```

```
if(digitalRead(azul)==0 && dir==0) digitalWrite(blue,HIGH);
if(digitalRead(azul)==1 && dir==0) digitalWrite(blue,LOW);

if (chivato==true){ //Cuando la interrupción activa el chivato:

 Serial.println(cont); //Escribe el número de
                       //pasos dados por el puerto serie.
 digitalWrite(bobina,LOW);//Deja de alimentar la fase.
 if (dir==1){   //Secuencia automática giro antihoario.
  cont--;
    switch (bobina){
    case black:
    bobina=red;break;
    case green:
    bobina=blue;break;
    case red:
    bobina=green;break;
    case blue:
    bobina=black;break;}
  }
   if (dir==2){  //Secuencia automática giro horario.
    cont++;
    switch (bobina){
    case black:
    bobina=blue;break;
    case green:
    bobina=red;break;
    case red:
    bobina=black;break;
    case blue:
    bobina=green;break;}
```

```
    }
    digitalWrite(bobina,HIGH); //Alimenta la nueva fase.
    chivato=false; //Desactiva el chivato para que este código
                //solo se reproduzca una vez.
}

 while(Serial.available()){
    char c = Serial.read();
    if(c=='<')                //  Inicio de nuevo comando.
      sprintf(comandoTexto,"");
    else if(c=='>')           // Fin de nuevo comando.
      interpreta(comandoTexto);
    else if(strlen(comandoTexto)<10)  // Si comandoTexto tiene espacio,
                              //se lee el carácter.
      sprintf(comandoTexto,"%s%c",comandoTexto,c);
                          // Si no, se ignora y se busca  '<' o '>'.
 }


   OCR1A=vel; //El valor de la velocidad cambia el comparador para variar
           //el ritmo de las interrupciones.
}
```