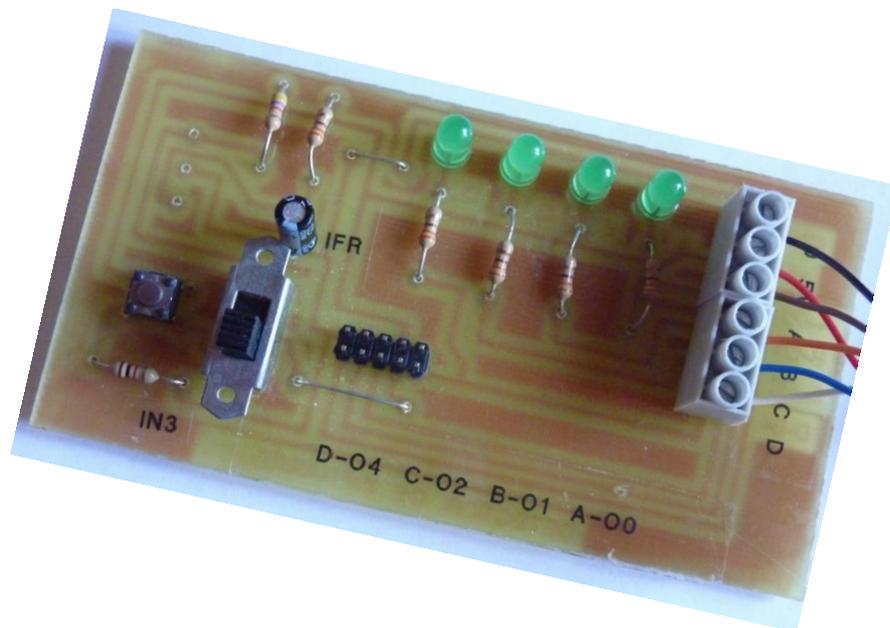




KITS PARA INICIAR EN LA ROBÓTICA Y PROGRAMACIÓN EN SECUNDARIA

PREMIOS ITIC EDUCACIÓN 2014



7 DE NOVIEMBRE DE 2014

IES PANDO - OVIEDO

Autor: Luis A. Díaz Sánchez
albertods61@gmail.com

Contenido

1. Nombre del Centro y responsable de proyecto	4
2. Antecedentes.....	4
3. Objetivos del proyecto	4
4. Descripción del proyecto realizado	5
4.1. Software a utilizar.....	5
4.2. Descripción de las patillas del Picaxe 08M y 08M2	6
4.2.1. Cable de programación USB	7
4.2.2. Alimentación de los KITS	7
4.2.3. Cable plano 10 hilos (esquema de conexiones)	8
4.3. KIT BASE.....	9
4.4. Descripción	9
4.4.1. PCB y Cara vista componentes	10
4.4.2. Presupuesto y lista de materiales	10
4.5. Kit 1 de prácticas.....	11
4.5.1. Cara vista componentes	11
4.5.2. Presupuesto y lista de materiales	12
4.6. Kits de Prácticas 2	12
4.6.1. Presupuesto y Lista de materiales	14
4.7. Kit de Prácticas 3 para semáforo	14
4.7.1. Cara vista componentes	15
4.7.2. Presupuesto y lista de materiales	15
4.8. KIT 4 de prácticas	16
4.8.1. Presupuesto y lista de materiales	16
5. Conceptos de programación en BASIC	17
5.1. Programación de salidas.....	17

5.2.	Programación de entradas digitales	19
5.3.	Tiempos de espera	20
5.4.	Variables	20
5.4.	Bucle for...next.....	21
5.5.	IF condición THEN acción	21
5.6.	IF condición THEN acción1 ELSE acción2 ENDIF	21
5.7.	IF...ELSEIF...ELSE	22
5.8.	LOGICAS AND y OR.....	22
5.9.	DO LOOP WHILE	23
5.10.	Entradas analógicas	23
5.10.1.	Test de entradas analógicas.....	24
5.11.	Select case	25
6.	Ejercicios propuestos a realizar	27
6.1.	EJERCICIOS CON EL KIT 1	27
6.1.1.	Encender 2 LED intermitentes	27
6.1.2.	Encender 2 LED alternativamente	28
6.1.3.	Encender 2 LED un número de veces	28
6.1.4.	Encender 2 LED con una LDR	30
6.2.	EJERCICIOS CON EL KIT 2	32
6.2.1.	Contador binario desde el 0 al 9.....	32
6.2.2.	Contador ascendente con pulsador	34
6.2.3.	Secuencia LED Fórmula 1	35
6.2.4.	Manejo del display con el mando a distancia.....	36
6.3.	KIT 3 semáforo.....	38
6.3.1.	Programa secuencia continua	38
6.3.2.	Semáforo con pulsador	38
6.4.	KIT 4	39

6.4.1. Esquema de conexiones	41
7. Bibliografía	43

1. Nombre del Centro y responsable de proyecto

IES Pando de Oviedo. Luis A. Díaz Sánchez, profesor de Tecnología, Fundamentos de Electrónica y Electrotecnia

2. Antecedentes

Al desarrollar el tema de programación y robótica en la materia de Tecnología de 4º ESO, nos encontramos con que en el temario vienen conceptos de robótica, pero no existen kits para desarrollar las prácticas adecuadamente.

Ante este problema surge la idea de desarrollar unos kits de programación de μ controladores. Se busca una idea que fuese barata, fácil de programar para alumnos de 4º ESO y que se puedan hacer unos kits ampliables para posteriores ideas.

Se decide hacer un KIT base que llevará el μ controlador Picaxe 08M2 y otros kits que llevarán entradas (pulsadores, ldr,...) y salidas(led, motores,...). Ambos se conectarán por un cable plano de 10 hilos.

La casa Picaxe facilita un software de programación -tanto a través de código Basic como diagramas de flujo- gratuitos y que permiten la simulación del programa.

3. Objetivos del proyecto

Entre los principales objetivos del proyecto se encuentran:

- ✓ Diseñar y construir unos kits de robótica para las clases de 4º ESO
- ✓ Iniciar a los alumnos a:
 - Conocer los algoritmos de programación
 - Resolver problemas mediante diagramas de flujo
 - Conocer un lenguaje de Programación (BASIC)
 - Aplicar la programación a resolver problemas de la vida real y aplicarlos a un proyecto electrónico (Robótica)

- ✓ Poner a disposición del profesorado que lo necesite los PCB para realizar los KIT y un manual con ejercicios resueltos para facilitar la labor docente.

4. Descripción del proyecto realizado

El proyecto que se describirá a continuación consiste en realizar unos kits para las prácticas de robótica en las clases de Tecnología de 4º ESO. El proyecto constará de lo siguiente:

- ✓ KIT base que llevará un ucontrolador PIC08M2 y al que se conectarán los KITS de entrenamiento. Serán diferentes KITS con distintos componentes para poder realizar las diferentes prácticas
- ✓ Manual básico de robótica, explicando entradas digitales, analógicas, salidas por LED, amplificadas por transistor y las instrucciones en BASIC para iniciarse a la programación
- ✓ Ejercicios para realizar con cada KIT, resueltos mediante diagramas de flujo y/o código de programación
- ✓ Bibliografía con los enlaces de videos, manuales, fotos, etc

4.1. Software a utilizar

Los programas que vamos a utilizar son libres y facilitados por la casa Picaxe a través de su web www.picaxe.com

Se puede utilizar indistintamente:

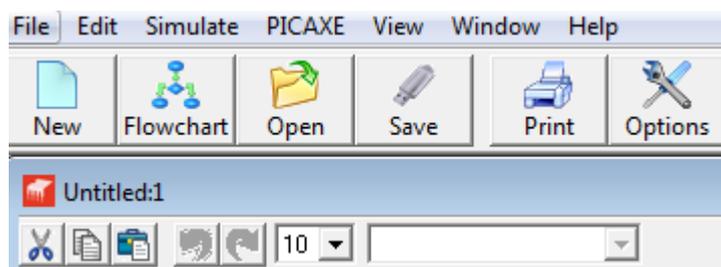
Picaxe Editor 6 o Picaxe Programming Editor

<http://www.picaxe.com/downloads/pe6073/PICAXEEditor6.exe>

<http://www.picaxe.com/downloads/bas805.exe>

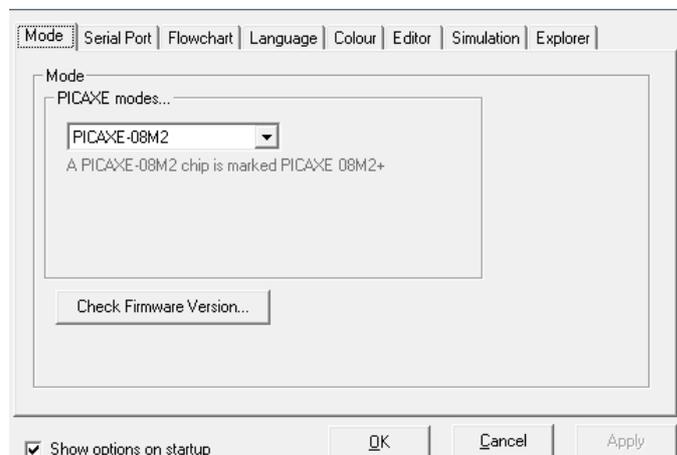
Ambos permiten programar en código y en diagrama de flujo

Desde el código de programación, accedemos al diagrama de flujo mediante el icono Flowchart



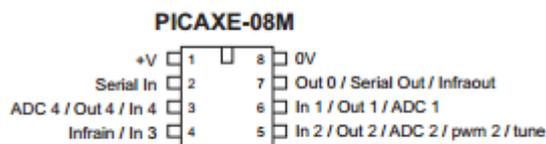
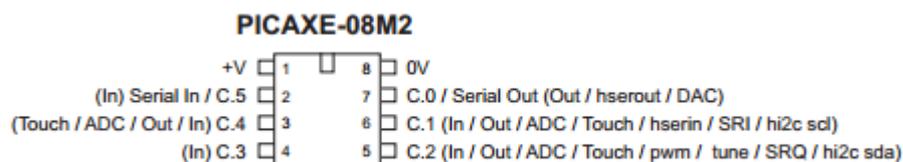
Antes de comenzar a programar, tenemos que elegir en Options->Mode-> El ucontrolador a programar Picaxe 08M o Picaxe 08M2

En Serial Port el puerto al que va conectado el cable de Programación, que suele ser COM3 o COM4



4.2. Descripción de las patillas del Picaxe 08M y 08M2

En un principio los kit estaban pensados para el picaxe 08M. Pero este chip lo dejó de fabricar la casa y lo sustituyo por el 08M2. Se pueden usar ambos indistintamente.



4.2.1. Cable de programación USB

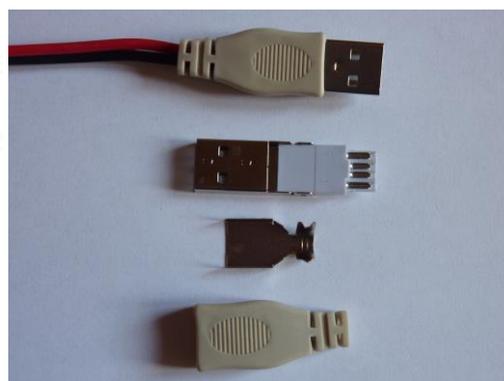
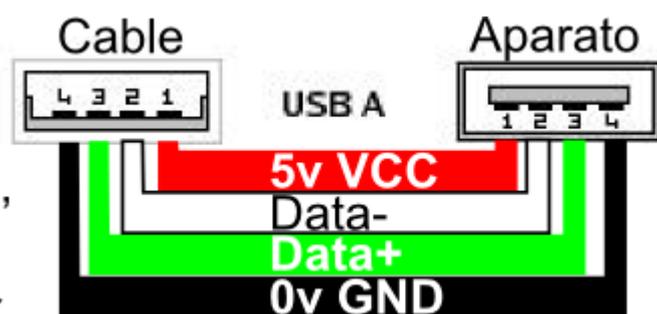
Este cable lo vende la casa picaxe. El precio aproximado es de 12£



http://www.picaxestore.com/index.php/en_gb/picaxe/picaxe-download-cables.html

4.2.2. Alimentación de los KITS

Para programar estos μ controladores se va a trabajar con el software de la casa Picaxe. Los kits tienen que tener una alimentación DC 5V. Dado que sería un inconveniente conectarlos a fuentes de alimentación, lo que hacemos es aprovechar la salida de los puertos usb +5 y 0v y mediante un conector usb realizar un cable para alimentarlos desde el ordenador.



4.2.3. Cable plano 10 hilos (esquema de conexiones)

Para conectar el kit base (dispone de alimentación, puertos de entrada IN y de salida OUT al PICAXE) a los KIT de salida hemos diseñado un cable plano de 10 hilos. En la tabla se muestra el esquema de conexión

PIC 08M KIT BASE	Aplicación KIT 1,2,3,4	Conexión IN / OUT	Conexión cable
1	6	Out0	
2	7	IN1/OUT1/ADC1	
3	8	IN2/OUT2/ADC2	
4	9	IN3/INFRA3	
5	10	IN4/OUT4/ADC4	
6	1	+5V	
7	2	+5V	
8	3	GND	
9	4	GND	
10	5	GND	

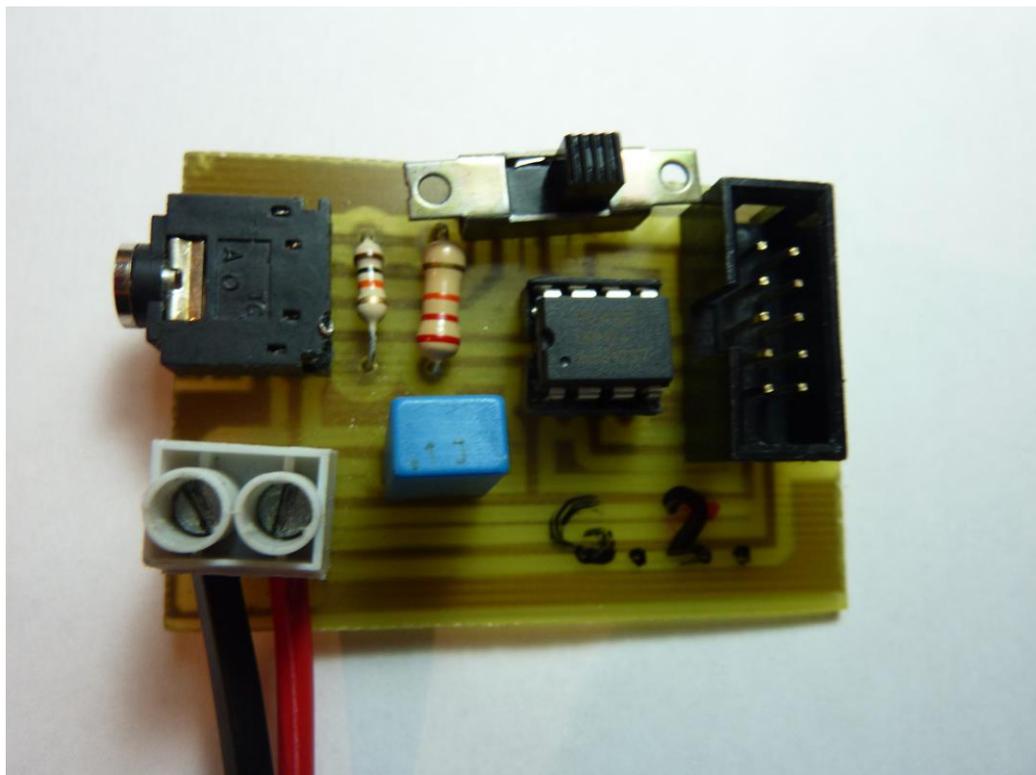
4.3. KIT BASE

Para realizar los kit utilizaremos una placa fotosensible 1 cara 200x300 mm 1,6 mm cuyo coste aproximado es de 20€

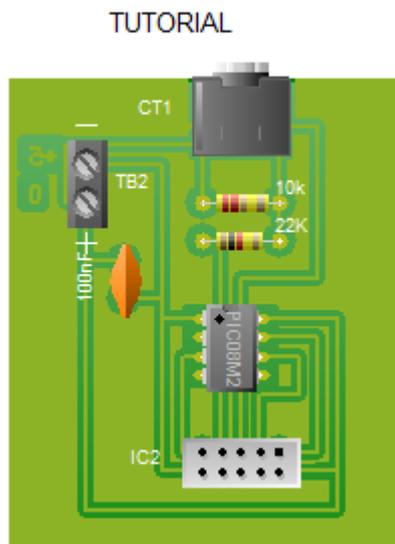
4.4. Descripción

Es el kit donde va a ir el μ controlador. Dispondrá de alimentación 5VDC a través del cable USB. En la parte superior llevará un socket al que se le conecta el cable de programación. Llevar un conmutador. Dado que la patilla 6 del picaxe sirve como Serial OUT o OUT0. Cuando se esté transfiriendo el programa desde el ordenador al pic, se deslizará a la parte superior (conectará a Serial OUT) y cuando se esté ejecutando el programa se deslizará a la parte inferior (OUT 0) para que funcione la salida OUT0

En la parte inferior dispone de un socket para conectar a través del cable plano de 10 hilos de los distintos kits de prácticas que se describirán a continuación.



4.4.1. PCB y Cara vista componentes

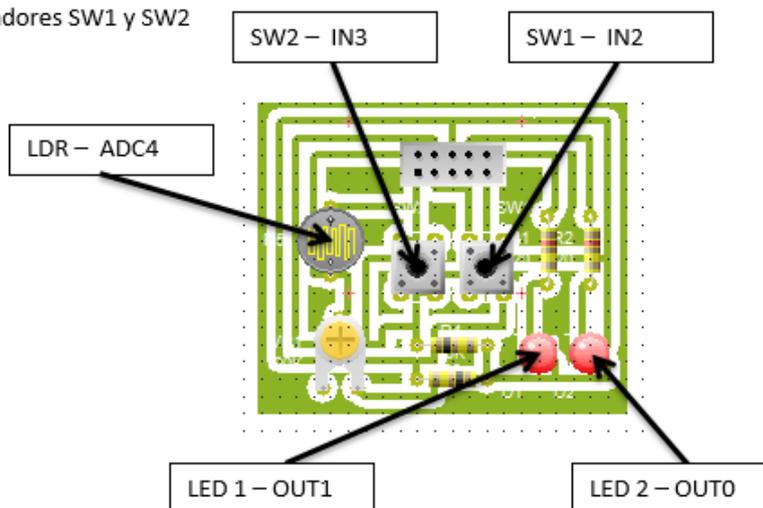


4.4.2. Presupuesto y lista de materiales

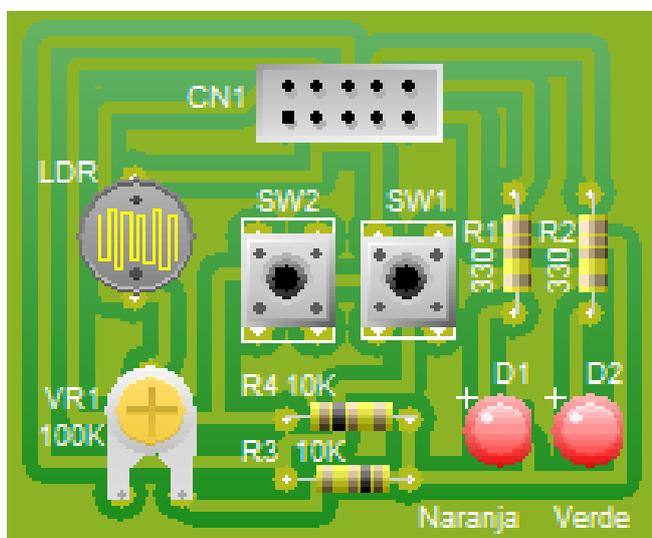
Código	Denominación	Uds	€/Ud.	Total
CT1	Stereo Picaxe Socket	1,000	0,130	0,130
R1	Resistencia 10k -1/4w	1,000	0,022	0,022
R2	Resistencia 22k -1/4w	1,000	0,022	0,022
	Picaxe 08M2	1,000	2,280	2,280
	Zocalo 8 pines	1,000	0,120	0,120
TB2	Regleta 2 pines paso 5mm	1,000	0,290	0,290
C1	Condensador poliester 100nF-100V	1,000	0,200	0,200
IC2	Conector cable plano macho 10 hilos	1,000	0,270	0,270
	Placa CI positivo 60x45mm	0,003	323,000	0,945
			TOTAL	4,279

4.5. Kit 1 de prácticas

- Entradas Digitales: 2 Pulsadores SW1 y SW2
- 1 Entrada Analógica
- Salidas : 2 LED D1 y D2



4.5.1. Cara vista componentes

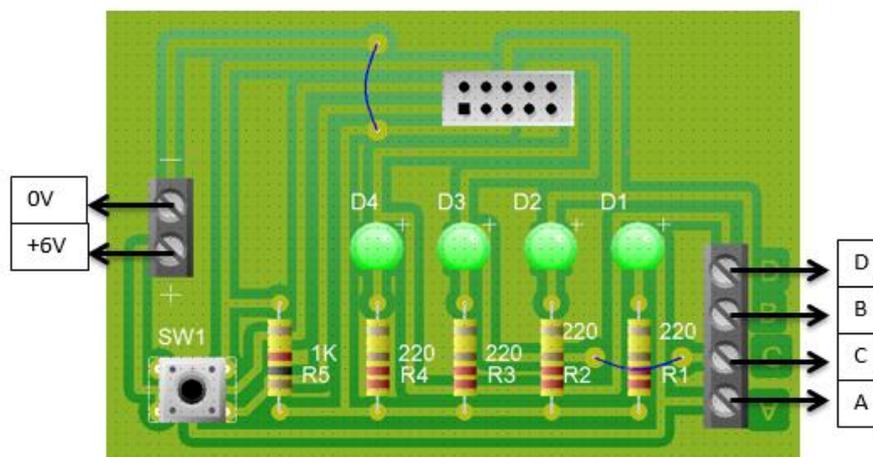


4.5.2. Presupuesto y lista de materiales

Código	Uds	Denominación	€/Ud	TOTAL
CT1*	1	Stereo Picaxe Socket (CON 039)	0,13	0,13
R1	1	Resistencia 10k – 1/4W	0,022	0,022
R2	1	Resistencia 22k-1/4W	0,022	0,022
AXE007M2*	1	Picaxe 08M2	2,28	2,28
ZOC08P	1	Zocalo 8 pines	0,12	0,12
IC2	1	Cont. macho para cable plano 10 hilos	0,27	0,27
C1	1	Condensador poliéster 100nF -100V	0,32	0,32
TB2	1	Regleta paso 5mm 2 pines	0,29	0,29
	1	Placa CI Positiva 80x60	2,56	2,56
TOTAL				6,144

4.6. Kits de Prácticas 2

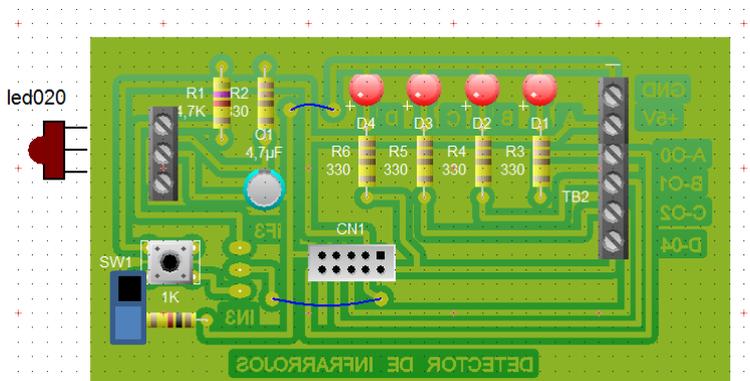
Dispone de 4 LED y un pulsador. Los LED servirán para diferentes prácticas entre ellas un contador de binario de 0 a 9



CODIGOS BCD 8421			
D = 8	C = 4	B = 2	A = 1
OUT0	OUT 1	OUT2	OUT4
LED D4	LED D3	LED D2	LED D1

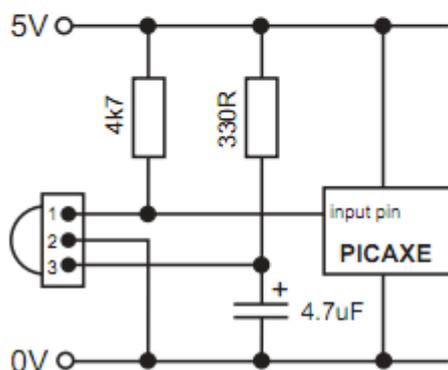
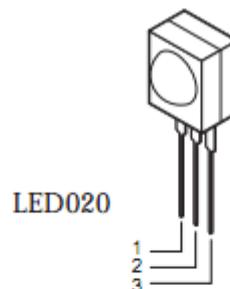
Una variante del kit anterior llevará un Fotodiodo de infrarrojos, para poder programar el kit mediante un mando a distancia

El interruptor en la posición arriba (dibujo)se usará para el sensor de infrarrojos y en la posición de abajo para usar el SW1 como entrada IN3



CONEXIÓN ENTRADA

El detector de infrarrojos LED020 se conecta según figura y se programa a la entrada IN3

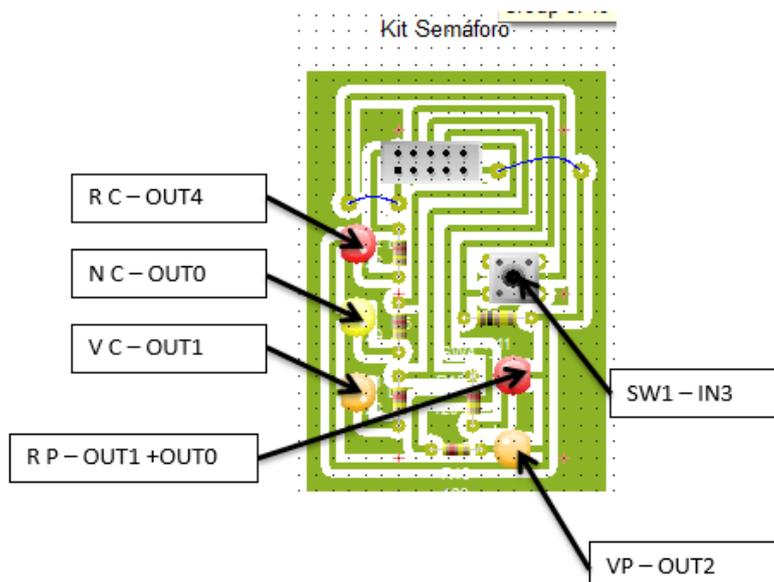


4.6.1. Presupuesto y Lista de materiales

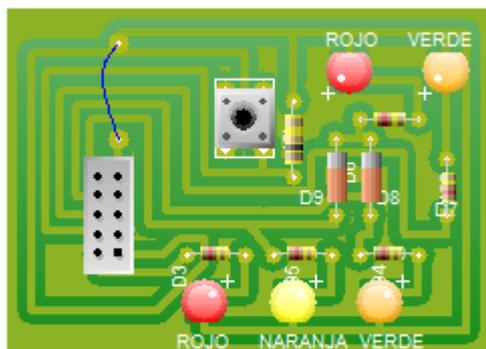
Código	Denominación	Uds	€/Ud.	Total
	Regleta 2 pines paso 5mm	3,000	0,290	1,170
R1,2,3,4	Resistencias R300 1/4W	4,000	0,022	0,044
D1,2,3,4	Led 5mm 2V-10mA	4,000	0,100	0,200
R5	Resistencias 10k - 1/4W	2,000	0,022	0,044
SW1	Pulsador CI 6x6	1,000	0,190	0,380
	Conector cable plano macho 10 hilos	1,000	0,270	0,270
	Placa CI positivo 100x60 mm	0,006	323,000	1,938
TOTAL				2,378

4.7. Kit de Prácticas 3 para semáforo

- Entrada Digital: 1 Pulsador SW1
- Salidas: 5 LED Coches: (Rojo, Naranja, Verde), Peatones: (Rojo, Verde)



4.7.1. Cara vista componentes

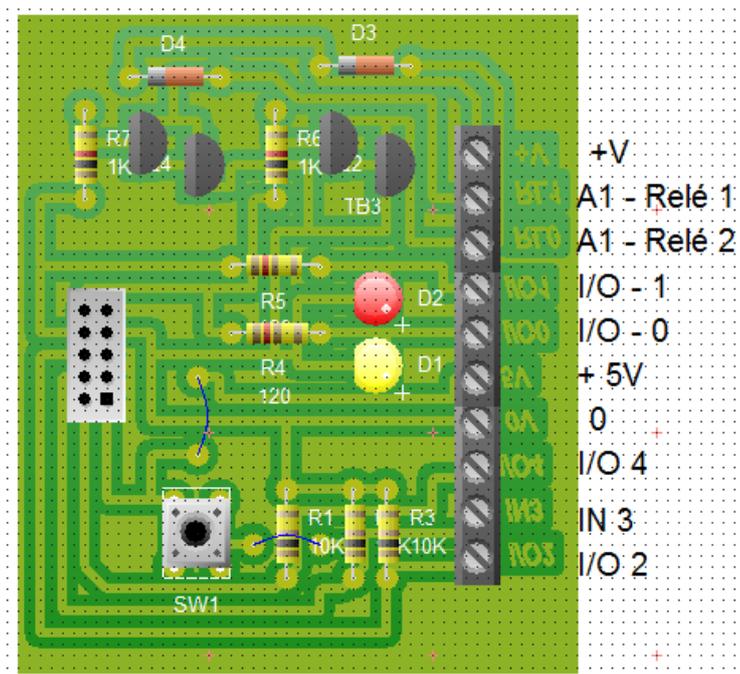


4.7.2. Presupuesto y lista de materiales

Código	Denominación	Uds	€/Ud.	Total
R1, R2	Resistencias R300 1/4W	5,000	0,022	0,110
D1, D2	Led 5mm 2V-10mA	5,000	0,100	0,500
R3,R4	Resistencias 10k - 1/4W	1,000	0,022	0,022
SW1, SW2	Pulsador CI 6x6	1,000	0,190	0,190
	Conector cable plano macho 10 hilos	1,000	0,270	0,270
	Placa CI positivo 65x45 mm	0,003	323,000	0,969
			TOTAL	2,061

4.8. KIT 4 de prácticas

Este kit lleva salidas amplificadas por transistor, por lo que se pueden conectar motores, relés, etc en dichas salidas.



4.8.1. Presupuesto y lista de materiales

Código	Denominación	Uds	€/Ud.	Total
	Regleta 2 pines paso 5mm	5,000	0,290	1,450
D3, D4	Diodo rectificador IN4004	2,000	0,070	0,140
TB1,2,3,4	Transistor BC548 NPN	4,000	0,100	0,400
R1,2	Resistencias 2K2 1/4W	2,000	0,022	0,044
D1, D2	Led 5mm 2V-10mA	2,000	0,100	0,200
R3,4,5	Resistencias R300 - 1/4W	3,000	0,022	0,066
R6	Resistencias 10k - 1/4W	1,000	0,022	0,022
SW1	Pulsador CI 6x6	1,000	0,190	0,190
	Conector cable plano macho 10 hilos	1,000	0,270	0,270
	Placa CI positivo 80x65 mm	0,005	323,000	1,680
			TOTAL	3,0116

5. Conceptos de programación en BASIC

Usaremos el programa siguiente Picaxe que descargaremos gratuitamente desde

<http://www.picaxe.com/Software/PICAXE/PICAXE-Programming-Editor/>

Vamos a ver las instrucciones de programación más básicas. Al final se enlazará a unos manuales en pdf para el que desee profundizar más en la programación en BASIC de los ucontroladores PICAXE

5.1. Programación de salidas

High 1 -> pone la salida 1 a nivel alto => 1 – 5V – H

Low 1 -> pone la salida 1 a nivel bajo => 0- 0V- L

Se pueden activar varias salidas a la vez

High 0,1

Low 0,1

La siguiente orden produce una basculación de la salida. Si vale 0 la pone a 1 y si vale 1 la pone a 0

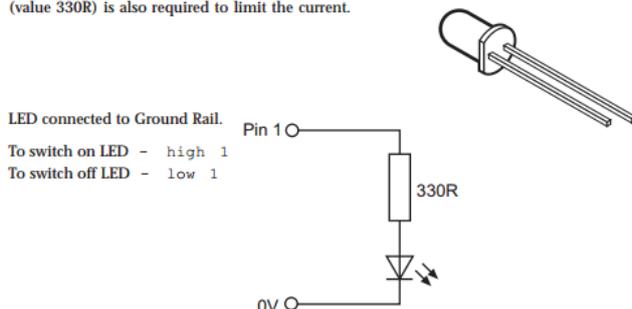
Toggle 0

Vemos como se conectaría un LED a la salida 1

Output Device Interfacing

Output Device 1 - Light Emitting Diode (LED)

The PICAXE microcontroller can sink ("absorb") or source ("give out") a small amount of current, which means that an LED can be connected directly to the output pin. A series resistor (value 330R) is also required to limit the current.



Si lo que deseamos conectar es una lámpara, timbre, motor, rele, etc la salida no proporciona la suficiente intensidad por lo que debemos amplificarla mediante dos transistores en par de Darlington

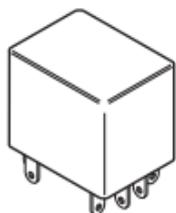
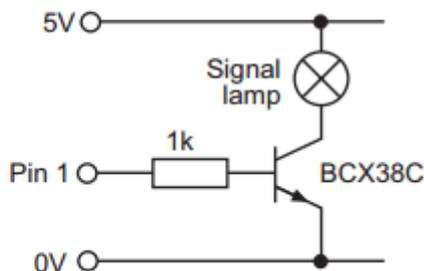
El elemento a conectar debe de ir entre +5V y Colector del transistor

To switch on Lamp - high 1

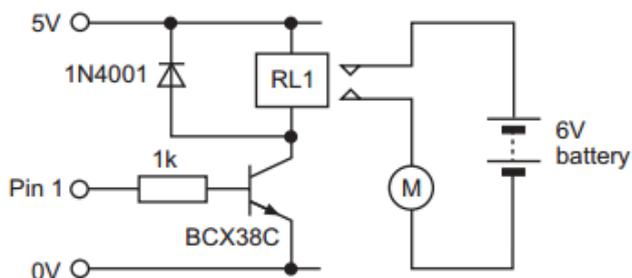
To switch off Lamp - low 1



signal lamp



relay



5.2. Programación de entradas digitales

Entradas digitales puede ser: pulsadores, finales de carrera, interruptores, etc.

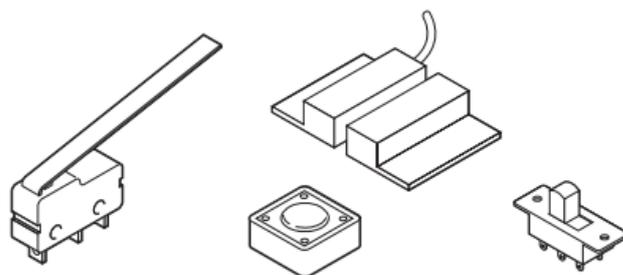
Si el pulsador esta sin activar -> 0V = 0

Si se acciona el pulsador -> 5V = 0

Veamos cómo se realiza la programación:

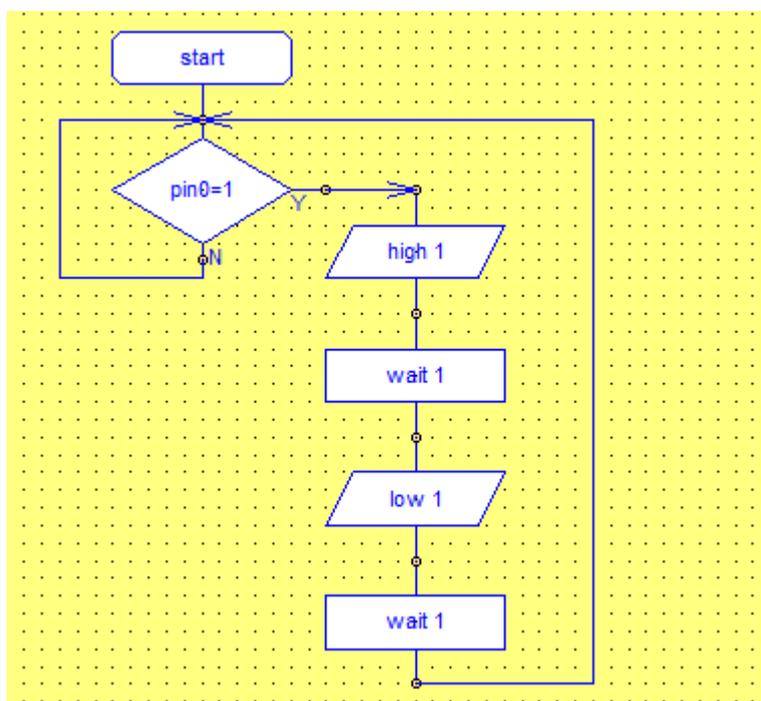
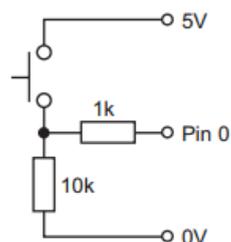
Diagrama de flujo

Si el pulsador no se acciona **N** vuelve a **START** y está esperando a que se pulse. Cuando se acciona **Y** ejecuta la acción que se desee, por ejemplo enciende y apaga un led durante 1 sg.



With this circuit the input pin is **low** when the switch is open and **high** when the switch is closed.

Goto 'jump' when switch is open: if pin0 = 0 then jump
Goto 'jump' when switch is closed: if pin0 = 1 then jump



Código de programación

- Mediante etiquetas

Las etiquetas son puntos de programa (terminados en :) a los que se puede enviar el programa mediante la orden *goto* . El programa está en bucle cerrado `main-> goto main` a la espera de que se accione el pulsador `IN3 =1` y salga del bucle a la etiqueta `secuencia`

```
1 main:
2     if pin3 = 1 then goto secuencia
3     goto main
4
5 secuencia:
6
```

- Mediante el bucle `do....loop`

Este bucle sino se le añade una condición estará en bucle indefinido. El programa saldrá del bucle a la etiqueta `secuencia` cuando se accione el pulsador `IN3 = 1`

```
1 main:
2     do
3         if pin3 = 1 then goto secuencia
4     loop
5
6 secuencia:
7
```

5.3. Tiempos de espera

Wait t -> detiene el programa t (sg)

Pause t -> detiene el programa t (msg)

5.4. Variables

Las variables en basic se denominan por `b0, b1, b2, b3, b4,....`

Se pueden inicializar por ejemplo `b0 = 0; b1 = 1`

Se pueden incrementar **inc b0** => `b0= b0+1`

Se pueden decrementar **dec b0** => $b0=b0-1$

5.4. Bucle for...next

Este bucle repite una acción un número determinado de veces. Por ejemplo si deseamos encender y apagar un led conectado a OUT 0 10 veces durante 1 sg cuando se accione un pulsador conectado a la entrada IN1 el programa sería el siguiente:

```
1  main:
2      if pin1=1 then goto salida
3      goto main
4
5  salida:
6      for b0= 1 to 10
7          toggle 0
8          wait 1
9      next
10     goto main
11
```

la variable b0 valdrá 1,2,..., 10 y la incrementa next

5.5. IF condición THEN acción

Esta instrucción ejecuta una acción si se cumple una condición. En el programa anterior vemos que si $pin1 = 1$ el programa se envía a la etiqueta salida, sino vuelve al bucle main

5.6. IF condición THEN acción1 ELSE acción2 ENDIF

Esta instrucción evalúa una condición, si se cumple se ejecuta la acción 1 y sino se cumple se ejecuta la acción 2

```
1  main:
2      readadc 4, b0
3      if b0 < 125 then
4          goto noche
5      else
6          goto dia
7      endif
8
9      dia:
10
11     noche:
12
```

5.7. IF...ELSEIF...ELSE

Esta instrucción evalúa más de una condición con ELSEIF

```
1  main:
2      readadc 4, b0
3      if b0 <= 50 then
4          goto programa1
5      elseif b0>51 and b0<100 then
6          goto programa2
7      elseif b0>=100 and b0<200 then
8          goto programa3
9      else
10         goto programa4
11     endif
12
```

5.8. LOGICAS AND y OR

Si deseamos que una acción se ejecute cuando se accionen dos pulsadores a la vez conectados a 0 y 1 usamos la función lógica AND

```
1  main:
2      do
3          if pin0=1 and pin3 = 1 then goto secuencia
4      loop
5
6  secuencia:
```

Si deseamos que una acción se ejecute cuando se accionen dos pulsadores uno u otro conectados a 0 y 1 usamos la función lógica OR

```

1  main:
2      do
3          if pin0=1 or pin3 = 1 then goto secuencia
4      loop
5
6  secuencia:
7

```

5.9. DO LOOP WHILE

Este bucle se realizará mientras que se cumpla la condición. Vamos a ver como realizamos un bucle for con esta orden

```

1  main:
2      b0 = 0
3      do
4          toggle 1
5          pause 800
6          inc b0
7          if pin1 = 1 then exit
8      loop while b0 <=9
9
10     goto main
11

```

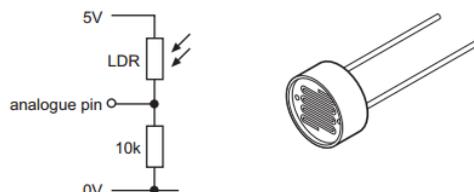
Una variante de este bucle – que ya se ha visto – es quitar la condición while. Con lo que el bucle queda cerrado do...loop a la espera de que se accione un pulsador

5.10. Entradas analógicas

Las entradas analógicas por ejemplo una LDR tienen valores infinitos (en función de la luz) variarán desde 0 a 5 V. Para conectar una LDR por ejemplo a una entrada analógica por ejemplo a la entrada 1 (Se ve si se puede conectar analógico si trae ADC1)

Input Device 3 - Light Dependant Resistor (LDR)

A Light Dependant Resistor (LDR) is a resistor that changes in value according to the light falling on it. A commonly used device, the ORP-12, has a high resistance in the dark, and a low resistance in the light. Connecting the LDR to the microcontroller is very straight forward, but some software 'calibrating' is required.



Programación de entradas analógicas

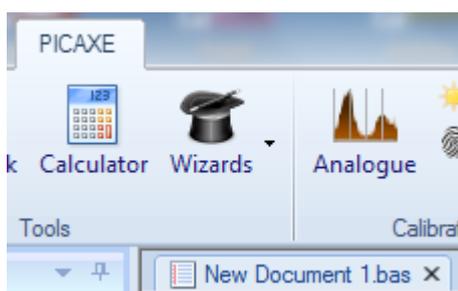
Readadc 4, b0 -> lee el valor de la ldr (0,255) conectado ADC14y se lo asigna a la variable b0

```
1 main:
2     readadc 4, b0
3     if b0 < 125 then
4         goto noche
5     else
6         goto dia
7     endif
8
9     dia:
10
11     noche:
12
```

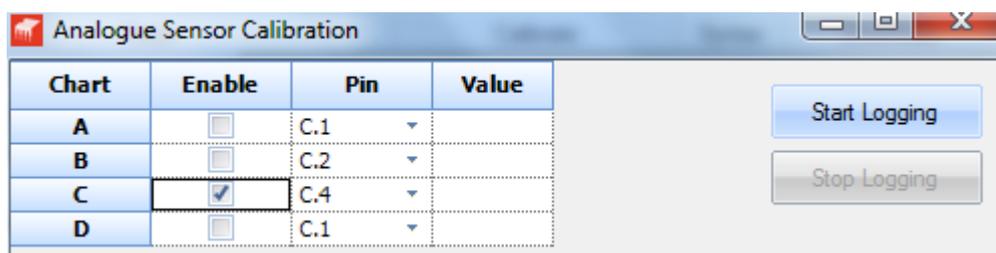
5.10.1. Test de entradas analógicas

Los valores de las entradas analógicas se pueden calibrar desde el picaxe editor

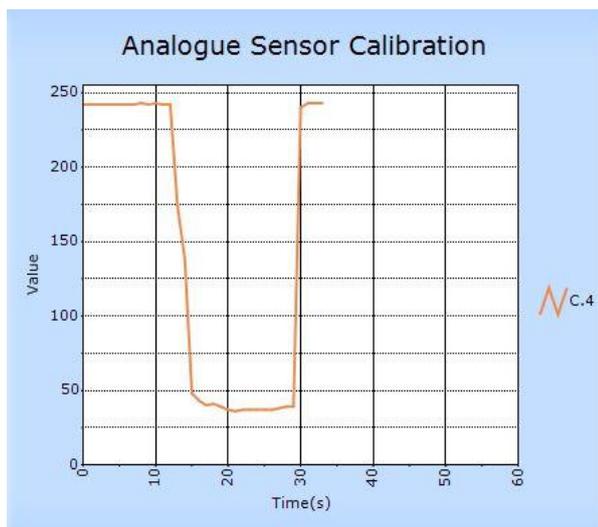
Para ello realizamos lo siguiente. En la pestaña Picaxe - Analogue



Marcamos el puerto al que va conectado el sensor analógico (en este caso C.4) y presionamos Start Logging



A continuación una vez que pasamos el programa al ucontrolador, vemos los valores del sensor analógico (LDR) en la siguiente gráfica



Cuando la LDR recibe luz los valores de la variable $b0 = 240$ y cuando no recibe luz $b0 = 40$

Lo que se hace es coger un valor intermedio 125

Si $b0 > 125$ se ejecuta el programa de día y si $b0 < 125$ se ejecuta el programa de noche. Estos valores los usaremos en el ejercicio 6.1.4

5.11. Select case

Cuando son muchas las condiciones a evaluar, en vez de IF, ELSEIF, ELSE se utiliza esta instrucción

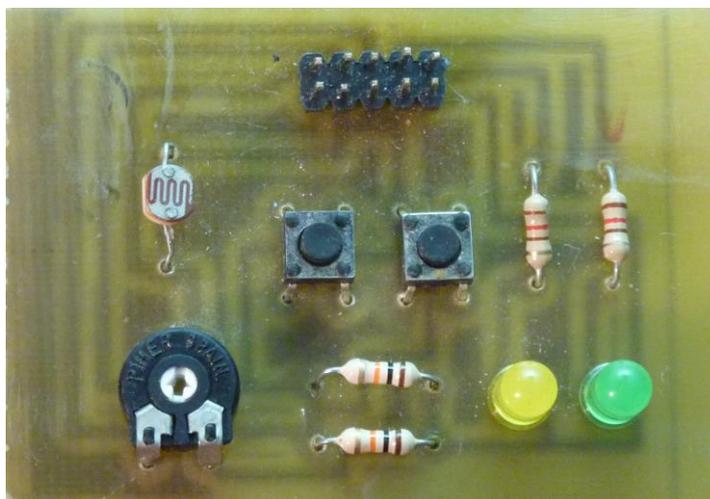
Este programa evalúa la variable $b0$ desde 0 a 9 y en función de ese valor asigna valores a las salidas 0,1,2,4

```
select case b0
  case 0
    low 4,2,1,0
  case 1
    low 4,2,1
    high 0
  case 2
    low 4,2,0
    high 1
  case 3
    low 4,2
    high 0,1
  case 4
    low 4,1,0
    high 2
  case 5
    low 4,1
    high 2,0
  case 6
    low 4,0
    high 2,1
  case 7
    low 4
    high 2,1,0
  case 8
    low 0,1,2
    high 4
  case 9
    low 2,1
    high 4,0
end select
```

6. Ejercicios propuestos a realizar

Vamos a ver una serie de ejercicios para cada KIT y su solución en basic y algunos de ellos en diagrama de flujo.

6.1. EJERCICIOS CON EL KIT 1



[Video con los ejercicios del KIT1](#)

6.1.1. Encender 2 LED intermitentes

Dos LED se encenderán y apagarán a la vez cada 0,8 sg

Solución:

LED 1 -> OUT0 - LED 2 -> OUT1

Utilizaremos el comando *toggle 0,1* y el comando de espera *pause 800*

Estos comandos los situaremos dentro del bucle *do....loop* que es un bucle cerrado, por lo que estarán parpadeando continuamente

```
1 main:
2     do
3         toggle 0,1
4         pause 800
5     loop
```

6.1.2. Encender 2 LED alternativamente

Dos LED parpadearán alternativamente cada 0,8 sg, Mientras uno este encendido el otro estará apagado y al contrario

Solución:

Es una variante del ejercicio anterior. Antes de entrar en bucle *do...loop* lo que hacemos es encender uno de los LED, por ejemplo el LED1 *high 1*. Al entrar en el bucle LED 1 (encendido) y LED2(apagado), por lo tanto cuando pasen por el comando *toggle 0,1* el que estaba encendido (LED1) se apaga y el que estaba apagado(LED2) se enciende

```
1  main:
2      high 0
3      do
4          toggle 0,1
5          pause 800
6      loop
```

6.1.3. Encender 2 LED un número de veces

6.1.3. Al accionar un pulsador los LED1 y LED2, deben de encenderse y apagarse a la vez cada 0,5 sg durante 10sg.

Solución:

LED 1 ->OUT0 - LED 2 ->OUT1

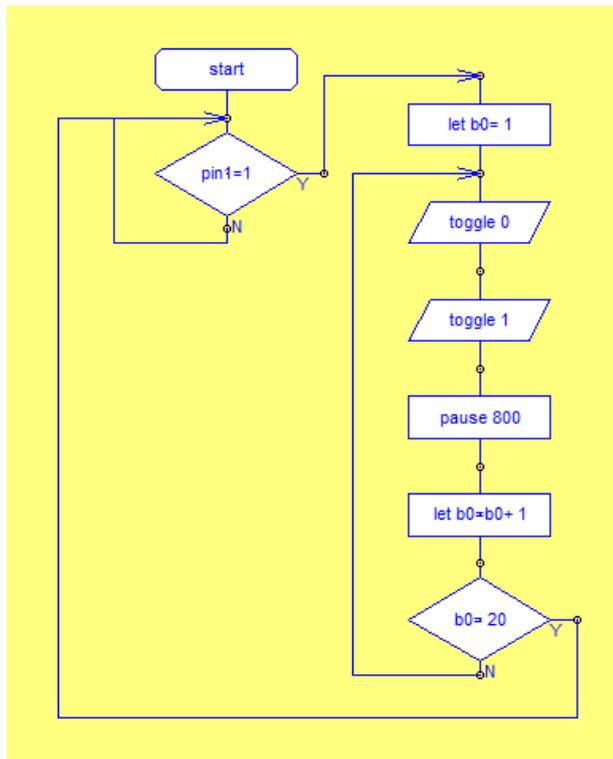
Pulsador ->IN3

El pulsador es una entrada digital, cuando se acciona su valor es 1. Usaremos un bucle infinito *do...loop*. Cuando se accione el pulsador, saldrá del bucle e irá a la etiqueta *secuencia*, para realizar la secuencia. Para ello utilizaremos el bucle *for....next*

El tiempo de encendido de los LED es de 0,5 sg. Como queremos que hagan esa secuencia 10 sg. El bucle se debe de repetir $10/0,5 = 20$ veces

```
1  main:
2      do
3          if pin3=1 then goto secuencia
4      loop
5  secuencia:
6      for b0 = 1 to 20
7          toggle 0,1
8          pause 500
9      next
10     low 0,1
11     goto main
```

Diagrama de flujo



6.1.4. Encender 2 LED con una LDR

6.1.3. Al accionar un pulsador los LED1 y LED2, deben de encenderse y apagarse a la vez cada 0,5 sg durante 10sg.

Solución:

LED 1 ->OUT0

LED 2 ->OUT1

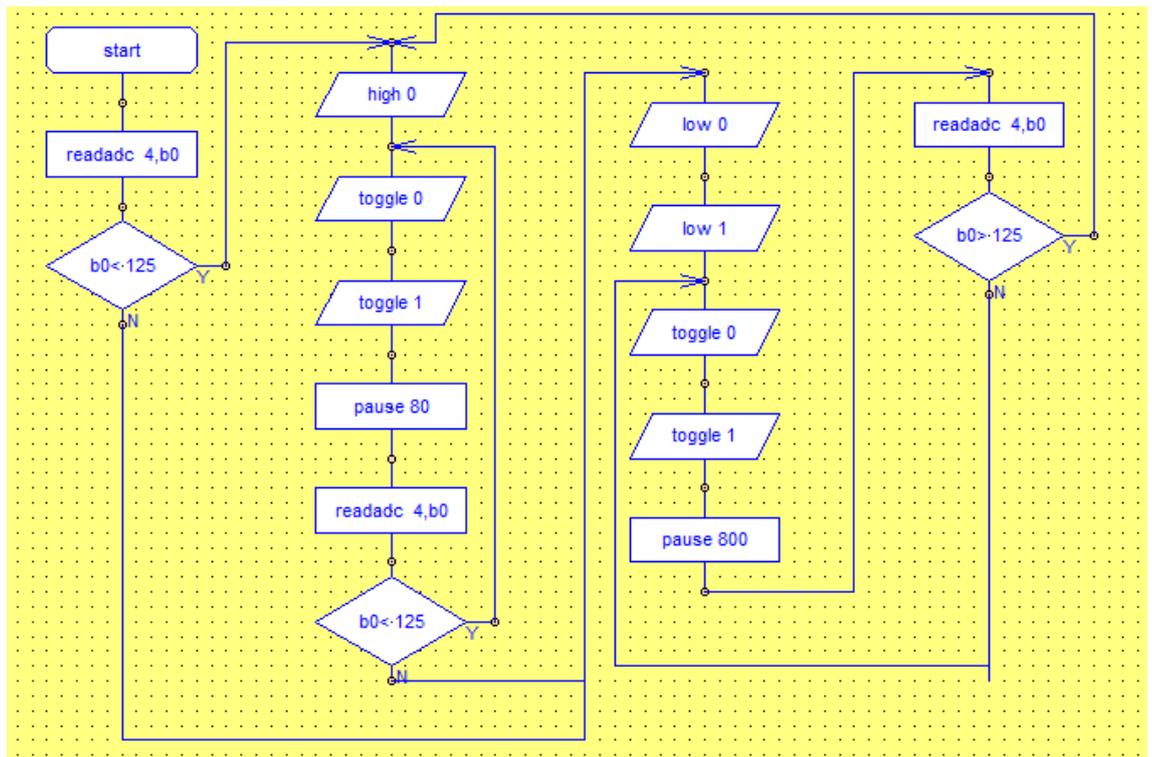
Pulsador ->IN3

El pulsador es una entrada digital, cuando se acciona su valor es 1. Usaremos un bucle infinito *do...loop*. Cuando se accione el pulsador, saldrá del bucle e irá a la etiqueta *secuencia*, para realizar la secuencia. Para ello utilizaremos el bucle *for....next*

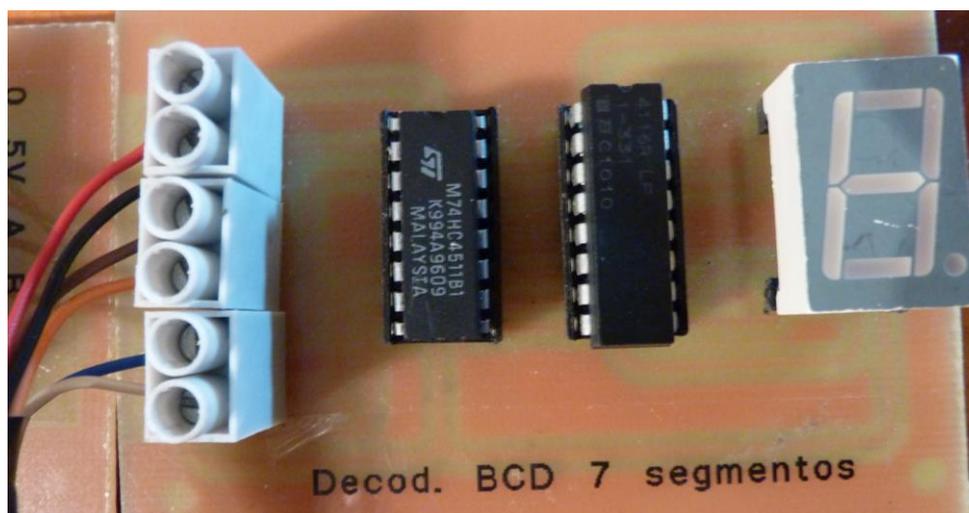
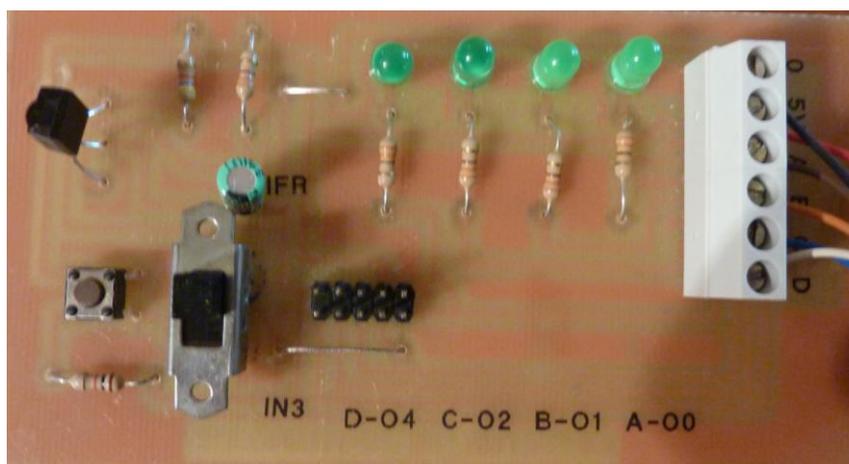
El tiempo de encendido de los LED es de 0,5 sg. Como queremos que hagan esa secuencia 10 sg. El bucle se debe de repetir $10/0,5 = 20$ veces

```
1  main:
2      readadc 4, b0
3      if b0<150 then
4          goto noche
5      else
6          goto dia
7      endif
8
9  dia:
10     do
11         toggle 0,1
12         pause 800
13         readadc 4, b0
14         if b0<150 then goto noche
15     loop
16
17 noche:
18     high 0
19     do
20         toggle 0,1
21         pause 800
22         readadc 4,b0
23         if b0>150 then goto dia
24     loop
```

Diagrama de flujo



6.2. EJERCICIOS CON EL KIT 2



[Video con los ejercicios del KIT2](#)

6.2.1. Contador binario desde el 0 al 9

Aprovecharemos a conectar a las salidas un decodificador BCD – 7 segmentos por lo que aparte veremos también los números en el display

Realizaremos la tabla de la verdad

Nº	D= OUT4	C = OUT2	B = OUT1	A= OUT0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1

4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Al accionar el pulsador irá contando ascendente 0,1,2,...9,0,1,...

```

1   low 0,1,2,4
2   'D(8)=OUT4 C(4)=OUT2 B(2)=OUT1 A(1)= OUT0
3   b0 = 0
4   main:
5       if pin3 = 1 then goto display:
6       goto main
7
8   display:
9       if b0= 9 then
10          b0=0
11       else
12          inc b0
13       endif
14       select case b0
15          case 0
16             low 4,2,1,0
17          case 1
18             low 4,2,1
19             high 0
20          case 2
21             low 4,2,0
22             high 1
23          case 3
24             low 4,2
25             high 2
26          case 4
27             low 4,1
28             high 2,0
29          case 5
30             low 4,0
31             high 2,1
32          case 6
33             low 4
34             high 2,1,0
35          case 7
36             low 0,1,2
37             high 4
38          case 8
39             low 2,1
40             high 4,0
41          case 9
42             low 2,1
43             high 4,0
44
45       end select
46       pause 1000
47       goto main

```

6.2.2. Contador ascendente con pulsador

Al pulsar realiza la secuencia de contar desde 0 a 9

```

1  low 0,1,2,4
2  'D(8)=OUT4 C(4)=OUT2 B(2)=OUT1 A(1)= OUT0
3  b0 = 0
4  main:
5      if pin3 = 1 then goto display:
6      goto main
7
8  display:
9      for b0=0 to 9
10     select case b0
11         case 0
12             low 4,2,1,0
13         case 1
14             low 4,2,1
15             high 0
16         case 2
17             low 4,2,0
18             high 1
19         case 3
20             low 4,2
21             high 0,1
22         case 4
23             low 4,1,0
24             high 2
25         case 5
26             low 4,1
27             high 2,0
28         case 6
29             low 4,0
30             high 2,1
31         case 7
32             low 4
33             high 2,1,0
34         case 8
35             low 0,1,2
36             high 4
37         case 9
38             low 2,1
39             high 4,0
40
41     end select
42     pause 1000
43 next
44 low 0,1,2,4
45 pause 1000
46 goto main

```

6.2.3. Secuencia LED Fórmula 1

Al accionar el pulsador se realizará la secuencia de encendido de LEDs como el de arranque de fórmula 1

- Al accionar el pulsador
 - Se enciende el LED A , espera 1 sg
 - Se enciende el LED B, espera 1 sg
 - Se enciende el LED C, espera 1 sg
 - Se enciende el LED D, espera 1sg
 - Se apagan todos los LED
- Si acciono el pulsador durante mas de 1 sg
 - Se ponen los 4 LED en averia

Si vuelvo a accionar el pulsador , el programa vuelve al inicio

Programa

```
1  main:
2      low 0,1,2,4
3      wait 1
4      do
5          if pin3 = 1 then goto comprobar
6      loop
7
8  comprobar:
9      wait 1
10     if pin3 = 1 then
11         goto averia
12     else
13         goto secuencia
14     end if
15
16  secuencia:
17     low 0,1,2,4
18     high 0
19     wait 1
20     high 1
21     wait 1
22     high 2
23     wait 1
24     high 4
25     wait 1
26     low 0,1,2,4
27  goto main
```

```

28
29 averia:
30     low 0,1,2,4
31     wait 1
32     do
33         toggle 0,1,2,4
34         pause 800
35         if pin3 = 1 then goto main
36     loop

```

6.2.4. Manejo del display con el mando a distancia.

El funcionamiento es el siguiente. Si se pulsa una tecla numérica en el MD saldrá esa cifra en el display. El botón UP irá contando ascendentemente 0,1,2,... y el botón DOWN irá contando descendientemente

Programa

```

1  let b2 = 0
2  low 0,1,2,4
3  main:
4  pause 500
5
6  main2:
7  infrain2
8  if infra = 0 then :let b2 = 1: goto numero: endif 'key 1
9  if infra = 1 then :let b2 = 2: goto numero: endif 'key 2
10 if infra = 2 then :let b2 = 3: goto numero: endif 'key 3
11 if infra = 3 then :let b2 = 4: goto numero: endif 'key 4
12 if infra = 4 then :let b2 = 5: goto numero: endif 'key 5
13 if infra = 5 then :let b2 = 6: goto numero: endif 'key 6
14 if infra = 6 then :let b2 = 7: goto numero: endif 'key 7
15 if infra = 7 then :let b2 = 8: goto numero: endif 'key 8
16 if infra = 8 then :let b2 = 9: goto numero: endif 'key 9
17 if infra = 9 then :let b2 = 0: goto numero: endif 'key 0
18 if infra = 16 then :inc b2: goto numero: endif
19 if infra = 17 then
20     if b2 = 0 then
21         b2 = 9
22     else
23         dec b2
24     endif
25     goto numero
26 endif
27
28 if infra = 21 or infra = 19 then ponerCero
29 goto main2

```

```

30
31     'nuevos
32     'A = 1 => OUT0
33     'B = 2 => OUT1
34     'C = 4 => OUT2
35     'D = 8 => OUT4
36     'OUT4(8) - OUT2(4) - OUT1(2) - OUT0(1)
37
38     swon0: 'numero 0
39     low 0,1,2,4
40     goto main
41
42     swon1: 'numero 1
43     high 0
44     goto main
45
46     swon2: 'numero 2
47     high 1
48     goto main
49
50     swon3: 'numero 3
51     high 0,1
52     goto main
53
54     swon4: 'numero 4
55     high 2
56     goto main
57
58     swon5: 'numero 5
59     high 0,2
60     goto main
61
62     swon6: 'numero 6
63     high 1,2
64     goto main
65
66     swon7: 'numero 7
67     high 0,1,2
68     goto main
69
70     swon8: 'numero 8
71     high 4
72     goto main
73
74     swon9: 'numero 9
75     high 0,4
76     goto main
77
78     numero:
79     low 0,1,2,4
80     select case b2
81     case 0
82     goto swon0
83     case 1
84     goto swon1
85     case 2
86     goto swon2
87     case 3
88     goto swon3
89     case 4
90     goto swon4
91     case 5
92     goto swon5
93     case 6
94     goto swon6
95     case 7
96     goto swon7
97     case 8
98     goto swon8
99     case 9
100    goto swon9
101    else
102    goto swFin
103    endselect
104
105    ponerCero:
106    let b2 = 0
107    goto main
108
109    swFin:
110    if b2 = 10 then
111        let b2 = 0
112    elseif b2 < 0 then
113        let b2 = 9
114    endif
115    goto main

```

6.3. KIT 3 semáforo



[Video con ejercicios del KIT3](#)

6.3.1. Programa secuencia continua

```
1  main:
2      low 0,1,2,4
3      do
4          high 1 'enciende VC y RP
5          wait 10
6
7          high 0 'enciende el NC
8          wait 3
9
10         low 1,0 'apaga VC, RP y NC
11         high 4, 2 'enciende el RC y VP
12         wait 10
13     loop
14     goto main
```

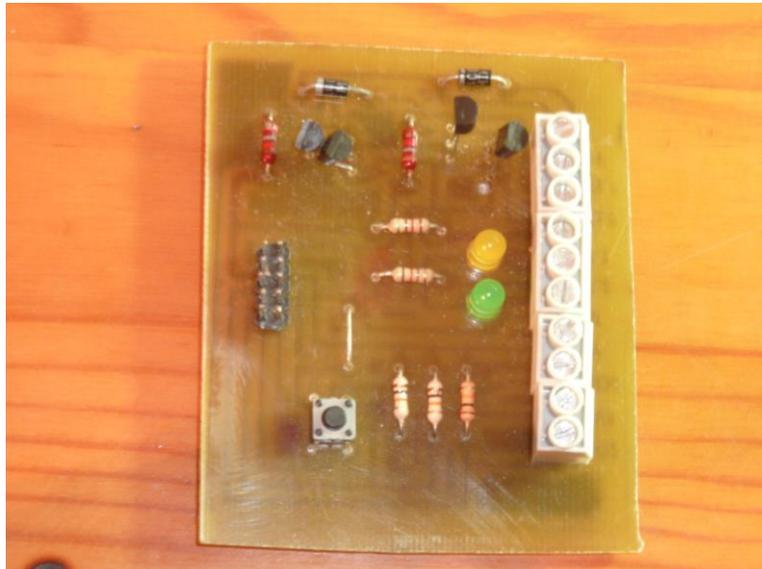
6.3.2. Semáforo con pulsador

El semáforo naranja está parpadeando y cuando se acciona el pulsador después de un tiempo se acciona para el paso de peatones

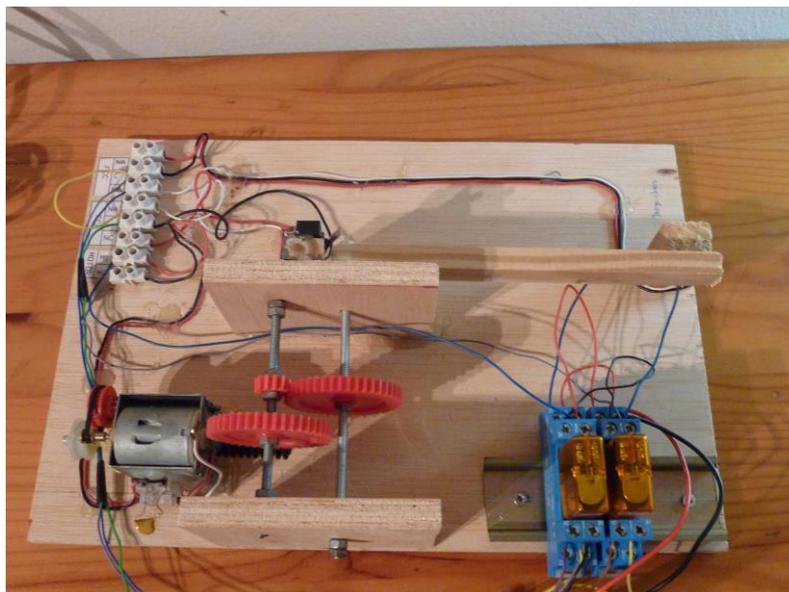
Programa

```
1  main:
2  low 0,1,2,4
3  do
4      toggle 0 'parpadea NC
5      pause 500
6      if pin3 = 1 then goto secuencia
7  loop
8
9  secuencia:
10     for b0 = 1 to 6
11         toggle 0
12         pause 500
13     next
14     high 0
15     wait 3
16     low 0
17     high 4, 2
18     wait 10
19     goto main
```

6.4. KIT 4



[Video con el ejercicio KIT 4](#)



Proyecto final. Conectaremos el KIT a unos relés y a una barrera de garaje con sus finales de carrera FCC y FCA. Cuando accionemos el pulsador se realizará la secuencia completa:

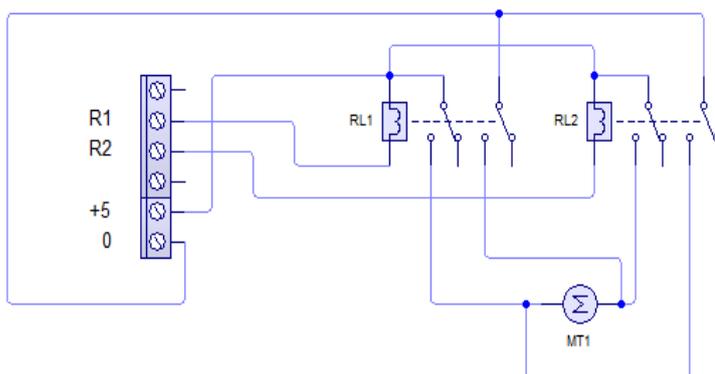
- Empieza a abrir la barrera
- Cuando llegue al final de carrera FCA.
- La barrera se detiene y espera 10 sg
- A continuación empieza a bajar
- Cuando llegue al FCC. La barrera se detiene y vuelve al inicio

FCA = IN2 – FCC = IN4 – Pulsador = IN3

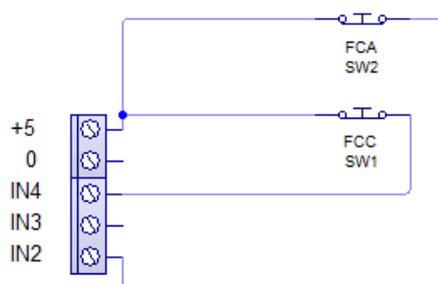
Relé abrir = OUT 0 – Relé cerrar = OUT1

6.4.1. Esquema de conexiones

Salidas: se conectan a 2 relés para invertir el sentido de giro



Entradas: Conexiones de los Finales de carrera

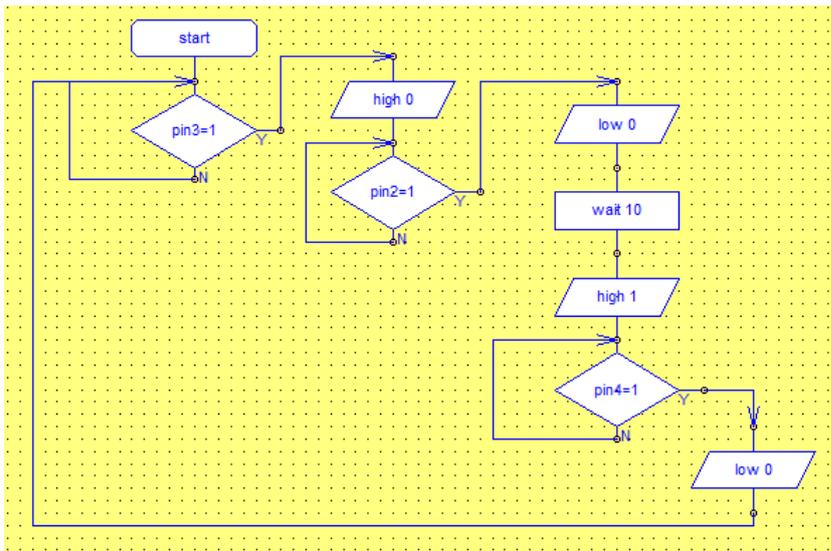


Programa

```

1  main:
2  low 0,1
3  do
4      if pin3 = 1 then goto abrirBarrera
5  loop
6
7  abrirBarrera:
8  high 0
9  do 'esta abriendo la barrera
10     if pin2 = 1 then goto cerrarBarrera 'cuando llega al FCA
11  loop
12
13  cerrarBarrera:
14  low 0 'desactiva el rel? de subir
15  wait 10 'espera 10 sg
16  high 1 'avctiva el rel? de bajar
17
18  do
19     if pin4 = 1 then goto main 'cuando llega al FCC
20  loop
21  |
    
```

Diagrama de flujo



7. Bibliografía

Software y hardware www.picaxe.com

Videos [Ejercicios KIT1](#) [Ejercicios KIT2](#)

[Ejercicios KIT3](#) [Ejercicios KIT4](#)

Fotos www.albertods.com/premiosItic2014/kit_robotica/album/index.html

Power Point <http://www.slideshare.net/ldiaz4/robtica-8198992>

Apuntes www.albertods.com/premiosItic2014/unidad_robotica.pdf

Manual 1 www.albertods.com/premiosItic2014/picaxe_manual1.pdf

Manual 2 www.albertods.com/premiosItic2014/picaxe_manual2.pdf

Manual 3 www.albertods.com/premiosItic2014/picaxe_manual3.pdf

Mando a distancia www.albertods.com/premiosItic2014/mandoDistanciaPicaxe.pdf

PCB circuitos www.albertods.com/premiosItic2014/PCB_KIT_08M2.rar

Fabricar circuitos impresos <https://artechip.wordpress.com/2013/09/09/como-fabricar-placas-de-circuitos-impreso-fotosensibles/>