

# MICROBOT CLÓNICO

## 1 INTRODUCCIÓN

El microbot Clónico es una evolución del microbot básico Tritt. Se ha cambiado la estructura física y se han añadido más sensores, mientras que los motores y la electrónica de control se han mantenido igual. La estructura física ahora es de metal, y en lugar de ruedas incorpora orugas. Los nuevos sensores son uno de luz y dos de contacto, conocidos vulgarmente como 'LDR' y 'bumpers' respectivamente. Gracias a esto se pueden programar algoritmos más complejos y por lo tanto realizar mayor cantidad de aplicaciones.

Al mantener la electrónica y los motores iguales que los que lleva el modelo Tritt, el manual de usuario de éste se mantiene para el microbot Clónico, tan solo habrá cambios en la parte del montaje de la estructura y en la localización de los sensores. Todo lo demás (preparación y utilización de los motores, programación de la CT6811 y programación de la CT293+) se mantiene igual.

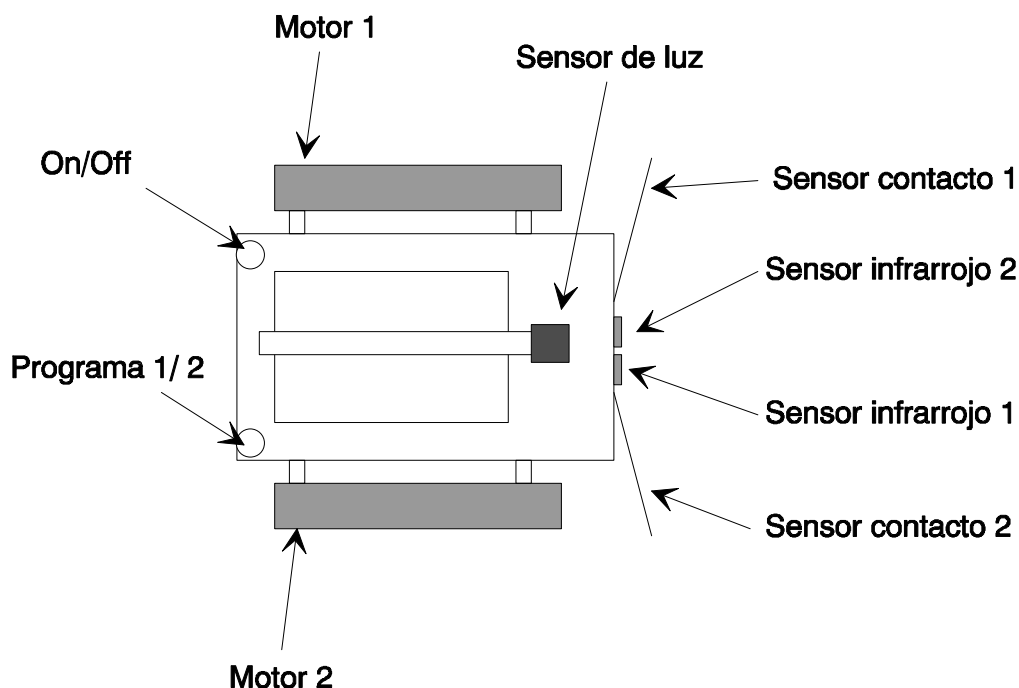
Este apéndice completará la información del manual del microbot, describiendo la localización de los nuevos sensores, la programación de éstos y finalmente se explica el algoritmo que trae el microbot por defecto.

## 2 PROGRAMACIÓN DEL MICROBOT

El microbot Clónico es un robot más avanzado que sus hermanos anteriores. La gran diferencia es que incorpora más sensores y tiene una estructura robusta y fácilmente ampliable. Todo esto permite programar algoritmos más variados, no sólo basados en la diferenciación de fondo negro/blanco como en los anteriores, sino también en la intensidad de luz y en la detección de objetos.

Antes de empezar la programación hay que tener en cuenta la situación física de los actuadores (motores y sensores), y también su situación lógica, es decir la localización de los bits de control en el mapa de memoria. Con estos bits se activarán los motores, se leerán el estado de los bumpers, infrarrojos, y de los demás sensores.

A continuación se muestra un plano de los actuadores del microbot, y en la tabla siguiente la localización lógica de los mismos.



Actuador	Localización	Dirección memoria	Descripción
Motor_1 On/Off	Puerto A: bit PA3	\$1000 : \$08	Activa motor 1: mueve oruga 1
Motor_1 Dirección	Puerto A: bit PA5	\$1000 : \$20	Cambia el sentido de giro motor_1
Motor_2 On/Off	Puerto A: bit PA4	\$1000 : \$10	Activa motor 2: mueve oruga 2
Motor_2 Dirección	Puerto A: bit PA6	\$1000 : \$40	Cambia el sentido de giro motor_2
Sensor Infrarrojo_1	Puerto A: bit PA7	\$1000 : \$80	Distingue blanco/negro
Sensor Infrarrojo_2	Puerto A: bit PA1	\$1000 : \$02	Distingue blanco/negro
Sensor Contacto_1	Puerto E: bit PE0	\$100A : \$01	Detecta contacto con obstáculo
Sensor Contacto_2	Puerto E: bit PE1	\$100A : \$02	Detecta contacto con obstáculo
Sensor de Luz	Puerto E: bit PE3	\$100A : \$08	Detecta intensidad de luz
Switch Programa ½	Puerto E: bit PE2	\$100A : \$04	Permite conmutar programa

A continuación se van a realizar una serie de programas de ejemplo, cada uno mostrará de forma sencilla como manejar cada actuador del microbot. Los programas se van a realizar para la RAM interna, por eso comenzarán con la directiva `ORG $0`. Cuando se quiera dejarlos grabados en la EEPROM interna (microbot autónomo), habrá que cambiar la directiva por `ORG $B600`.

Para volcar los programas, una vez compilados, al microbot habrá que conectarlo al PC y utilizar el 'DOWNMCU', si el programa es para la EEPROM habrá que utilizar el 'CTDIALOG'. La forma de usar estos programas se detalla en el manual de usuario del microbot, en el capítulo 'Tarjeta de desarrollo CT6811: manual de usuario'.

### Ejemplo 1: Poniendo en marcha el microbot

Este programa hace que el microbot avance indefinidamente. Para ello habrá que activar los motores y poner el giro adecuado. Los bits PA3 y PA4 del PUERTO A son los encargados de la activación, por lo tanto tendrán que ponerse a nivel alto '1'. Los bits PA5 y PA6 son los encargados de los sentidos de giro, en este caso deberán estar a nivel bajo '0'. Con estos valores se consigue que el motor 1 gira a izquierdas y que el 2 gire a derechas, el resultado es que el microbot avanza.

El valor en hexadecimal que hay que escribir en el puerto A es \$18, el cual se ha obtenido pasando de binario a hexadecimal el byte de control del puerto A.

Puerto A	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
\$1000	---	0	0	1	1	---	---	---

```
; Ejemplo 1: Poniendo en marcha el microbot
```

```
ORG $0 ; programa para la RAM interna
```

```
LDAA #$18 ; Cargo el valor $18 en el acumulador A
```

```
STAA $1000 ; Escribo el contenido del acumulador A en el Puerto A
```

```
Bucle BRA Bucle ; Dejo el programa en un bucle infinito
```

## Ejemplo 2: Leyendo los sensores de contacto

Este programa hace avanzar al microbot, en cuanto se pulsa el sensor de contacto 1 el microbot se para, para volver a ponerlo en marcha hay que pulsar el sensor de contacto 2.

Lo importante es saber que cuando un sensor de contacto esta activado, es decir se esta apretando, el bit que refleja su estado estará a nivel alto '1'. Mientras que si no esta activado estará a nivel bajo '0'. El switch 'programa 1/2' funciona como un sensor de contacto.

```
; ejemplo 2: El microbot se para cuando pulsemos el sensor de contacto 2, y
; se pone en marcha cuando
;           pulsemos el otro.
; el sensor de contacto_1 se lee en el PUERTO E, bit PE0
; el sensor de contacto_2 se lee en el PUERTO E, bit PE1

PORTA EQU $0
PORTE EQU $A

        ORG $0           ; programa para la RAM interna

        LDX #$1000      ; cargamos en el Registro X el valor $100 0

avanza LDAA #$18
        STAA PORTA,X    ; PORTA + X = $0 + $1000 = $1000 = PUERTO A
        LDAA PORTE,X    ; guardo en A el contenido del PUERTO E
        ANDA #$01       ; Me quedo con el valor del bit PE0,sensor contacto 1
        CMPA #$1        ; ¿Esta a nivel alto? = ¿ Esta pulsado el sensor?
        BNE avanza     ; la comparación no es cierta -> no se para

para   CLRA             ; la comparación es cierta, se ha pulsado el sensor
        STAA PORTA,X    ; paramos el microbot desactivando los bits del PUERTO A
        LDAA PORTE,X    ; guardo en A el contenido del PUERTO E
        ANDA #$02       ; Me quedo con el valor del bit PE1,sensor_contacto2
        CMPA #$2        ; ¿Esta a nivel alto? = ¿ Esta pulsado el sensor?
        BNE para       ; la comparación no es cierta ->no se activa microbot
        BRA avanza     ; la comparación es cierta, se activa el microbot

        END
```

## Ejemplo 3: Leyendo los sensores de infrarrojos

En este ejemplo el microbot va a ser capaz de seguir una línea negra. Para detectar la línea negra se utilizarán los sensores de infrarrojos situados en la parte delantera del microbot. Este ejemplo viene en el manual de usuario del microbot, en la sección de la tarjeta CT293+. Conviene leerlo para saber cómo se obtienen los datos siguientes, pero a la hora de programar tener en cuenta que en este microbot el sensor\_1 corresponde al bit PA7.

Sensor 1 (PA7)	Sensor 2 (PA1)	Acción Microbot	Estado Motor 1 (PA3)	Dirección Motor 1 (PA5)	Estado Motor 2 (PA4)	Dirección Motor 2 (PA6)	Byte Control
1	1	Avanzar	1	0	1	0	\$18
0	1	Izquierda	1	1	1	0	\$38
1	0	Derecha	1	0	1	1	\$58



```

espera BRCLR ADCTL,X $80 espera ; espera a que termine la conversión
LDAA ADR4,X ; leer el resultado de la conversión
CMPA #$E0 ; comparar un el umbral
BLO avanza ; si rebaso el umbral avanza
CLRA
STAA PORTA,X ; no rebaso umbral: parado
BRA espera ; vuelve a realizar conversión

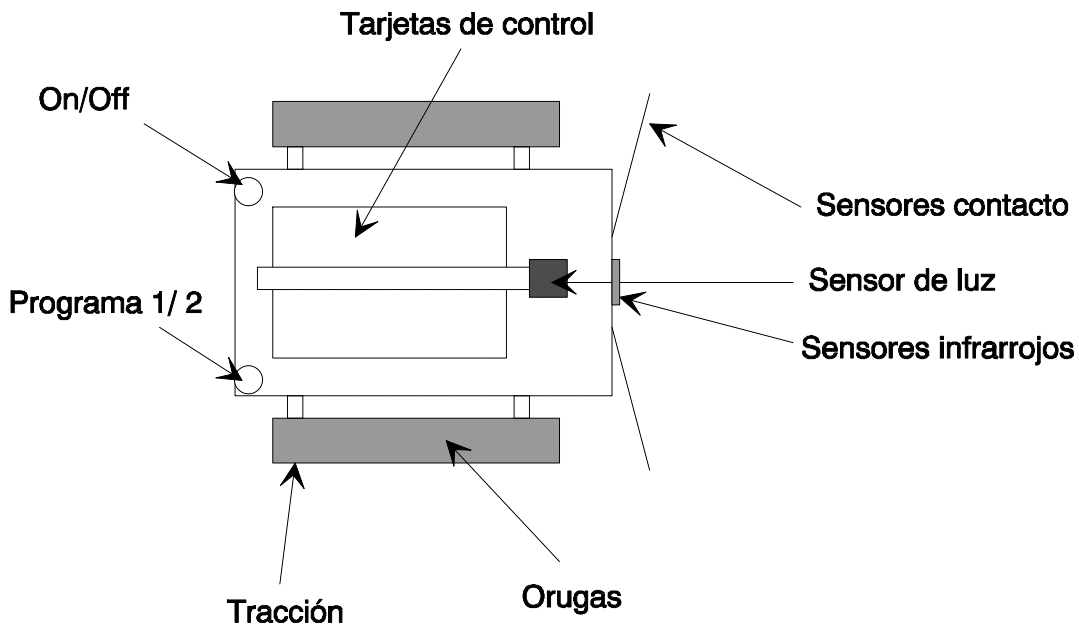
avanza LDAA #$18 ; supero umbral: avanzamos
STAA PORTA,X
BRA espera ; vuelve a realizar conversión

END

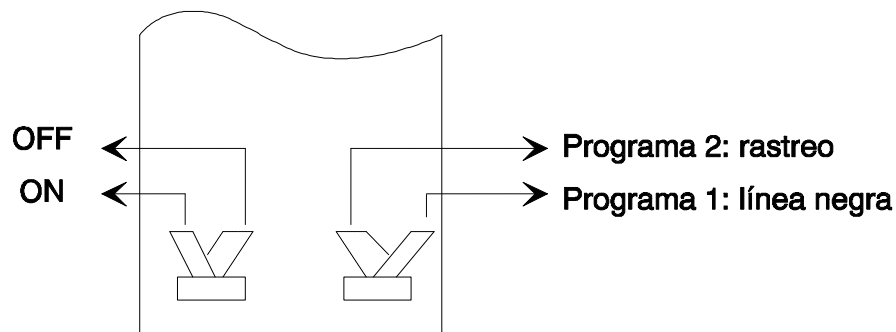
```

## 2 PROGRAMA DEL MICROBOT CLÓNICO

El microbot clónico que usted ha adquirido ha sido programado con dos algoritmos. El primero hace que el microbot siga una línea negra (cinta aislante) por el borde de la misma. El segundo hace que el microbot sea capaz de empujar objetos, evitar bordes, y ser controlado por luz, utilizando para ello todos sus sensores.



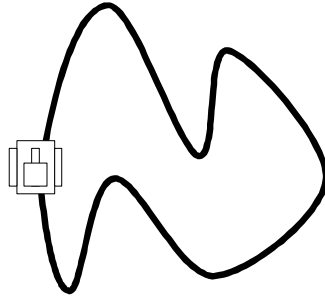
En la estructura mecánica se han incorporado dos switches. El que esta situado a la izquierda se encarga de encender y apagar el microbot. El situado a la derecha selecciona el programa que se desea.



### Programa 1: Seguir una línea negra.

Para probar este programa se recomienda trazar un recorrido sinuoso sobre un fondo claro, a ser posible blanco. Para trazar el recorrido se utilizará una cinta aislante negra, también se puede utilizar un rotulador negro grueso, aunque con esto no se garantiza el resultado. Depende mucho de las características del mismo, por eso es mejor utilizar directamente la cinta aislante negra.

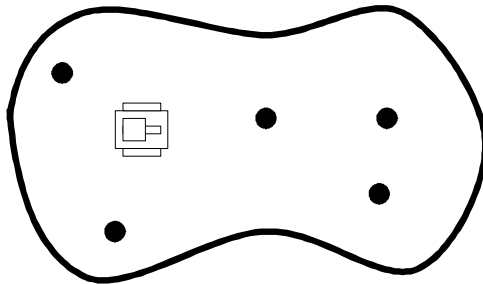
Seleccionaremos el programa 1, es decir el switch derecho estará situado a la derecha. Se situará al microbot sobre la línea negra y se encenderá. (Switch izquierdo en su posición izquierda).



### Programa 2: Rastreo

Para probar este programa se recomienda situar al microbot en el interior de un recinto, con forma de rectángulo, círculo, elipse, que tenga su borde delimitado con cinta aislante negra.

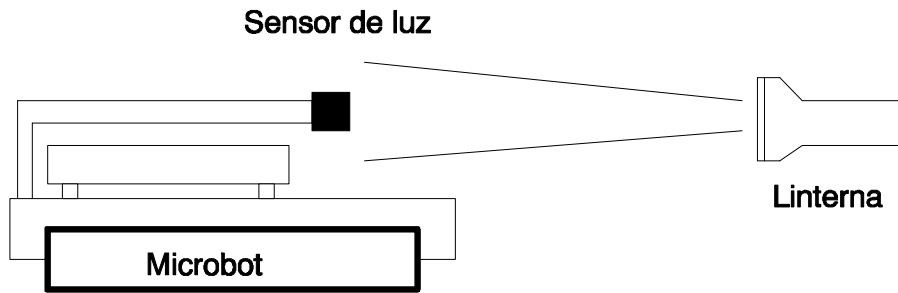
En el interior del recinto se pondrán una serie de objetos, lo suficientemente grandes para ser detectados por el microbot y lo suficientemente ligeros para poder ser empujados por él.



Seleccionaremos el programa situando a la izquierda el switch derecho. Encenderemos el microbot y este empezará a avanzar. A partir de ese momento podrán ocurrir las siguientes cosas:

1. El microbot llega al borde del recinto. Sus sensores al detectar la línea negra hacen que el microbot retroceda primero y gire después para seguir otra ruta. En el momento de ir hacia atrás deja de hacer caso a su sensor de luz, y a sus sensores de contacto.
2. El microbot por medio de sus sensores de contacto detecta un objeto situado en el interior del recinto. En ese momento empieza a orientarse para poder empujar al objeto. Si nada lo detiene lo empujará hasta llegar al borde del recinto. En ese momento sus sensores detectarán la línea negra y se aplicará lo dicho en el punto anterior.
3. Si el microbot se choca con un objeto grande o con una pared, de tal forma que sus dos sensores de contacto se activen a la vez, empezará a retroceder para buscar otra ruta alternativa. En el momento de retroceder no hará caso del sensor de luz ni de los de contacto.

4. Si en cualquier momento, excepto cuando el microbot se esta orientando, se le ilumina con una linterna o cualquier otra fuente de luz, el microbot empezará a girar hasta que deje de recibir la luz. El sensor de luz se ha introducido en una cápsula para evitar interferencias con otras fuentes del entorno, por ejemplo la luz solar. Por eso el microbot detectará que le están iluminando cuando reciba el rayo de luz directamente por el orificio de la antena. Dicho rayo deberá tener la suficiente energía para superar el umbral de detección. Podemos utilizar la fuente de luz para orientar al microbot y llevarle hasta otros objetos.



Este programa aunque es autónomo requiere de la participación de un ayudante, sobretodo para indicarle al microbot dónde están los obstáculos del recinto. Ampliando la electrónica se puede hacer que esta última función sea autónoma también. Para ello habrá que conectar unos ultrasonidos y añadir al programa la capacidad de detección y orientación del microbot al localizar algún objeto.

Por último se menciona que con este kit se proporcionan las herramientas necesarias para poder programar el microbot al gusto de cada uno. Realmente tenemos entre manos una plataforma de trabajo e investigación en sistemas autónomos. Con él, podremos aprender de una forma práctica y entretenida, diferentes técnicas de programación, electrónica y control. Para finalmente poder crear nuestros propios diseños autónomos.