
CARTONBOT

MEMORIA DEL PROYECTO FINAL DEL CURSO DE ARDUINO BÁSICO



Diego de los Reyes Rodríguez
26/04/2013

CONTENIDO

1.	Análisis	2
1)	Enunciado.....	2
2)	Descripción del proyecto	2
3)	Componentes	3
4)	Presupuesto.....	4
2.	Desarrollo.....	5
1)	Diseño de circuitería.....	5
2)	Fabricación de la carcasa	6
3)	Montaje electrónico.....	7
4)	Integración de la electrónica con la carcasa.....	7
5)	Programación.....	8
3.	Manual de Usuario	10
1)	Instrucciones.....	10
4.	Conclusiones	12

1. ANÁLISIS

En este apartado se describirá qué se quiere hacer, por qué y qué necesitamos para hacerlo.

1) ENUNCIADO

Realizar un proyecto nos obliga a aplicar la mayoría de los conocimientos expuestos a lo largo del curso.

Se buscará un proyecto innovador y que utilice varias de las funcionalidades expuestas durante el curso.

Se adjuntará el esquema del circuito, el programa y documentación sobre el funcionamiento y uso.

2) DESCRIPCIÓN DEL PROYECTO

Proyecto

Este proyecto consiste en la fabricación de un prototipo de un robot, al que llamaremos **cartonbot**, que sea capaz de ejecutar diversas habilidades. Estas habilidades se sustentarán sobre una capa que controla las piezas del cartonbot.

Objetivos

Al final de la realización de este proyecto se pretende haber alcanzado los siguientes objetivos:

- **Demostrar conocimientos:** El objetivo inicial es realizar un proyecto que cubra las actividades planteadas en el enunciado, para que, junto con el resto de actividades entregadas a lo largo del curso, pueda demostrar los conocimientos adquiridos.
- **Metodología:** Se pretende buscar una metodología que nos sirva de base para la ejecución de proyectos de una forma clara, sencilla y ordenada.
- **Escalabilidad:** Tanto a nivel hardware como a nivel software, se pretende que el sistema pueda crecer fácilmente, tanto incorporándole nuevos componentes como programas.
- **Modularidad:** Se busca desarrollar pequeños módulos, tanto a nivel hardware como software, cada uno con una funcionalidad completa y definida, con la menor dependencia posible entre ellos.
- **Abstracción entre hardware y software:** Para minimizar la dependencia entre hardware y software, esto es, que los programas ejecutados puedan funcionar sin conocer detalles específicos del hardware, como los pines donde están conectados los componentes.
- **Divertirse y aprender:** El objetivo principal del curso y del proyecto, es pasar un buen rato mientras se aprende.

3) COMPONENTES

Carcasa

Para la fabricación de una carcasa que dé forma física al cartonbot, se ha buscado reutilizar material que se pueda encontrar por casa, abaratando así los costes. Se ha elegido como carcasa una caja de cartón de embalaje de una lámpara de *IKEA*.

A esta caja se le harán dos agujeros para los ojos y una abertura que sirva como boca.

Para los brazos del cartonbot, que servirán a su vez como agujas del reloj, se recortarán dos tiras de cartón.

Como acabado final, se pintará al cartonbot de color marrón, para conservar así su apariencia de caja, se pondrá papel charol para tapar sus ojos, cartulina para tapar su boca y cuatro pequeñas patas para sostenerlo.

Electrónica

Los elementos necesarios para la fabricación del cartonbot son:

- **Ojos:** Dos leds, de color verde, con sus resistencias, conectados ambos al mismo pin como salida PWM para poder ajustar su brillo. En los ojos incluimos un sensor LDR con su resistencia, capaz de percatarse de los cambios de luz, para controlar el apagado de la alarma.
- **Boca:** Un buzzer capaz de reproducir melodías, conectado como salida PWM.
- **Brazos:** Dos servos serán los encargados de mover las dos tiras de cartón que definen los brazos/agujas del cartonbot/reloj.
- **Placa Arduino:** Para orquestar los elementos anteriores y recibir y guardar la configuración del cartonbot, se empleará una placa Arduino UNO, sencilla, barata y con pines más que suficientes para este proyecto.

Esta electrónica se integrará dentro de la carcasa.

4) PRESUPUESTO

Se ha tratado de reutilizar cosas encontradas por casa y se ha fabricado un cartonbot de carcasa para abaratar costes. El cálculo de costes se ha realizado orientativamente consultando la web [BricoGeek](http://BricoGeek.com).

Componente	Tipo	Unidades	Precio unidad	Total
Caja de cartón	Cuerpo	1	-	-
Tiras de cartón	Brazos / Agujas del reloj	2	-	-
Spray pintura	Pintura	1	2,00 €	2,00 €
Tornillería	Tornillería	-	-	-
Tacos	Patas	-	-	-
TOTAL				2,00 €

Electrónica	Tipo	Unidades	Precio unidad	Total
Placa Arduino UNO	Placa Arduino	1	22,00 €	22,00 €
LDR	Entrada analógica	1	1,35 €	1,35 €
Led verde	Salida PWM	2	0,35 €	0,70 €
Zumbador piezo	Salida PWM	1	1,80 €	1,80 €
Servomotor estándar	Salida PWM	2	10,90 €	21,80 €
Resistencia 220Ω	Resistencia	2	0,37 €	0,74 €
Resistencia 2.2 KΩ	Resistencia	1	0,37 €	0,37 €
Cables y clemas	Cableado	-	-	-
TOTAL				48,76 €

CONCEPTO	IMPORTE
Carcasa	2,00 €
Electrónica	49,13 €
TOTAL	50,76 €

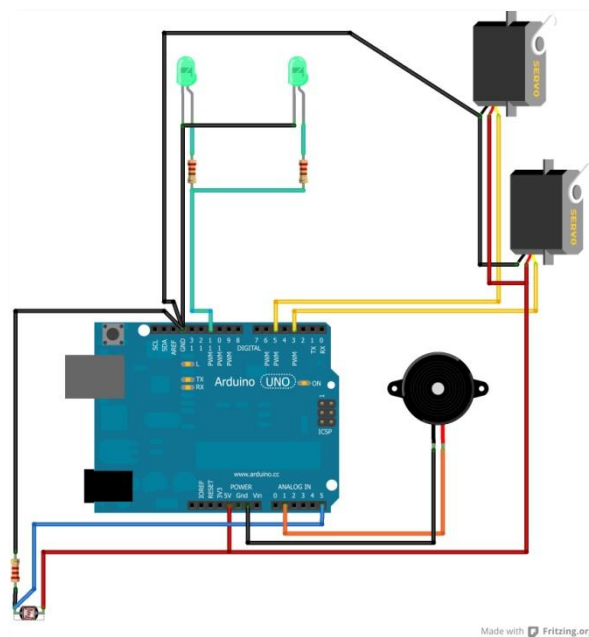
2. DESARROLLO

Explicación detallada de la construcción y desarrollo del cartonbot.

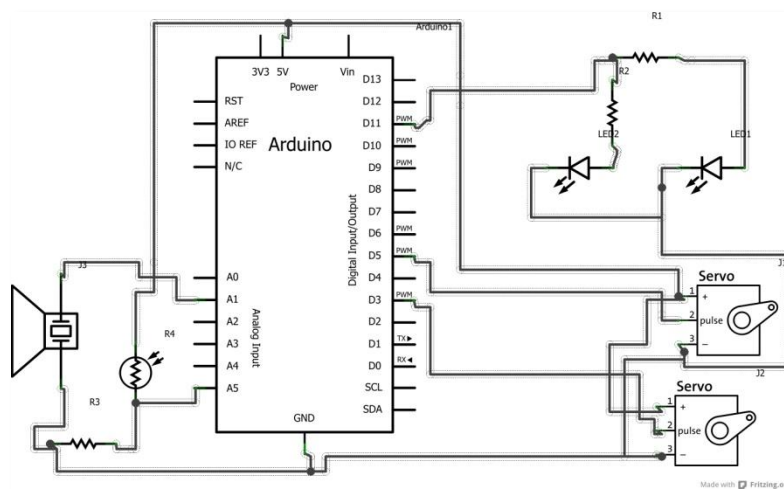
1) DISEÑO DE CIRCUITERÍA

Antes de empezar a montar los componentes electrónicos y la carcasa, debemos diseñar la circuitería, para evitar posteriores errores de ejecución. Los esquemas en alta resolución y el archivo fritzing se encuentran en la carpeta **/Esquemas/**.

Protoboard



Esquema



2) FABRICACIÓN DE LA CARCASA

La carcasa del cartonbot se ha fabricado de forma casera, reutilizando una caja de cartón, uniéndola a dos tiras de cartón y finalmente pintándola.

Elección de la caja

Para la elección de la caja, debemos considerar los esquemas anteriormente mostrados. Una vez hemos comprobado que nuestra circuitería cabe, procedemos a recortarla.

Recortar la caja

Para marcar los ojos y la boca nos ayudamos de una caja de tarjetas de visita. Con un lápiz trazamos los círculos y la boca, ayudándonos de un metro para que queden bien centrados. Finalmente, con ayuda de un cúter, los recortamos.



Colocamos la placa por fuera de la caja para marcar sus conexiones de USB y toma de corriente para que cuando la metamos dentro podamos acceder a ella.

Los huecos de los servos los recortaremos ayudándonos de los mismos servos, marcaremos con el lápiz, recortaremos con la cúter y lo ajustaremos con el servo. Después, con unas tijeras recortamos dos tiras de cartón para los brazos, que serán las

manecillas del reloj.

Ubicar los componentes electrónicos

Unimos los brazos a los servos y a la caja. Metemos también la placa para comprobar que todo encaje bien.

Colocamos un par de clemas en el hueco de los ojos, atornillamos y colocamos los leds. Haremos lo mismo con el LDR en la cabeza del cartonbot.

Una vez comprobado que todo encaja, procedemos a desmontar el cartonbot para poder pintarlo.



Pintar



En un lugar ventilado, rociamos con un spray, por separado, la caja y los brazos del cartonbot y dejamos que se seque.

Patas

Finalmente, colocamos cuatro patas para sostener y elevar al cartonbot.

3) MONTAJE ELECTRÓNICO

Siguiendo el esquema realizado, montaremos todos los componentes en la placa. Lo haremos por partes.

Ojos

Formados por dos leds y un LDR. Unimos los leds y el LDR, junto con sus resistencias, mediante clemas y cables.

Brazos

Formados por dos servos, uno para cada brazo.

Boca

Formada por un buzzer, conectado directamente a la placa.



4) INTEGRACIÓN DE LA ELECTRÓNICA CON LA CARCASA

Una vez tenemos la carcasa lista y se ha secado la pintura y tenemos la circuitería montada y probada, pasamos a integrarla dentro de la caja. Para ello, atornillamos los componentes a la caja según lo habíamos marcado al recortarla.



5) PROGRAMACIÓN

Se utiliza una arquitectura en capas para el diseño del programa que hará funcionar al cartonbot.

- **Capa de Piezas:** Esta capa, formada por varias librerías, se encarga de abstraer el hardware del cartonbot de las aplicaciones que harán uso de él.
- **Capa de Utilidades:** Formada por un conjunto de librerías que encapsulan componentes comunes para facilitar la reutilización de código utilizados por cualquier otra librería.
- **Capa Reactiva:** Instintos del cartonbot que se ejecutan sin pensar.
- **Capa Deliberativa:** Sustentándose en la capa de abstracción del software, estos programas representan las habilidades del cartonbot, como el reloj. Sólo pueden ejecutarse si la capa reactiva no está en ejecución, y sólo puede existir una habilidad deliberativa en ejecución a la vez.
- **Capa Arduino:** Sketch de Arduino, encargado de lanzar las habilidades.

Se adjunta el código dentro de la carpeta **Código**. Dentro de esta carpeta podemos encontrar:

- **examples:** sketches de ejemplo del uso de las librerías desarrolladas, utilizado para el testeo.
- **libraries:** contiene las librerías desarrolladas, necesarias para el funcionamiento del cartonbot. Se describirán en los siguientes apartados.
- **sketchboard:** contiene el sketch principal, **Cartonbot**.

La descripción detallada del código está en los comentarios del programa.

Capa de Piezas

Formado por las siguientes librerías:

- **Boca:** Maneja el buzzer.
- **Ojo:** Maneja los leds que representan un ojo y el ldr que indica la luz que ve el cartonbot.
- **Brazo:** Maneja un servo. En este proyecto maneja los brazos izquierdo y derecho.
- **FabricaPiezas:** Se encarga de la instanciación de las piezas y es la única clase que conoce los pines a los que están conectados los componentes.

Capa de Utilidades

Formada por las siguientes librerías:

- **CodificadorMorse:** Codifica mensajes en morse. Actualmente sólo emite S.O.S.
- **Sonido:** Almacena secuencias de sonido.

Capa Reactiva

Formada por las siguientes librerías:

- **FobiaLuz:** Instinto desarrollado que representa la fobia a la luz. Cuando hay poca luz, el cartonbot siente miedo, pide socorro, levanta los brazos y abre los ojos. Según pasa el tiempo, el cartonbot va dejando de tener miedo y cada vez tarda más en reaccionar ante la ausencia de luz, hasta que llega un punto que se olvida del miedo, la oscuridad le sorprende y vuelve a padecer la fobia. El porcentaje de fobia se almacena en memoria.

Capa Deliberativa

Formada por las siguientes librerías:

- **Bailar:** Habilidad deliberativa que hace que el cartonbot baile al ritmo de una melodía pregrabada.
- **Pasivo:** Habilidad que sólo cambia de pose, utilizada para cuando el cartonbot no sabe qué hacer.

Capa Arduino

- **Cartonbot:** Inicializa las habilidades y las pone en ejecución.

Comunicaciones y almacenamiento

Para poder interactuar con el cartonbot, nos comunicaremos con él mediante el puerto serial. El cartonbot podrá almacenar en memoria persistente datos de configuración.

Comunicación serial

Por el puerto serie se envían órdenes al cartonbot y se obtiene información de su estado. Además, muestra el menú y los mensajes que nos manda el cartonbot sobre lo que está haciendo.

EEPROM

Utilizada por la habilidad reactiva de fobia a la oscuridad. En la dirección 0 de memoria se almacena un valor comprendido entre 0 y 100, siendo 0 miedo a la oscuridad y 100 la ausencia de dicho miedo.

3. MANUAL DE USUARIO

Conectar los componentes a la placa **Arduino UNO** según el esquema de conexión.

Añadir las librerías ubicadas en **Código/librerías** a las librerías de Arduino.

Añadir el programa ubicado en **Código/sketchbook** al sketchbook de Arduino.

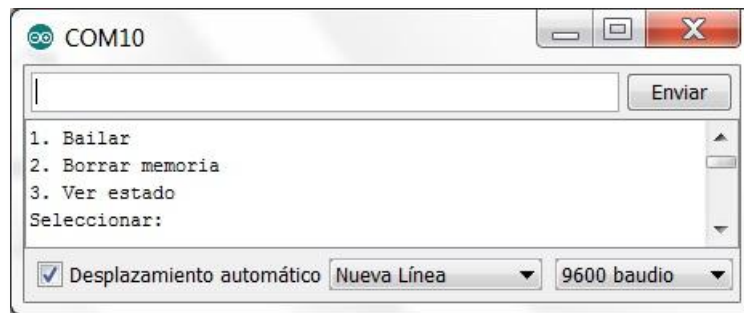
Conectar la placa a por usb con el ordenador para su comunicación por el puerto serial.

Ejecutar y cargar el sketchbook **Cartonbot.ino** en la placa **Arduino UNO**.

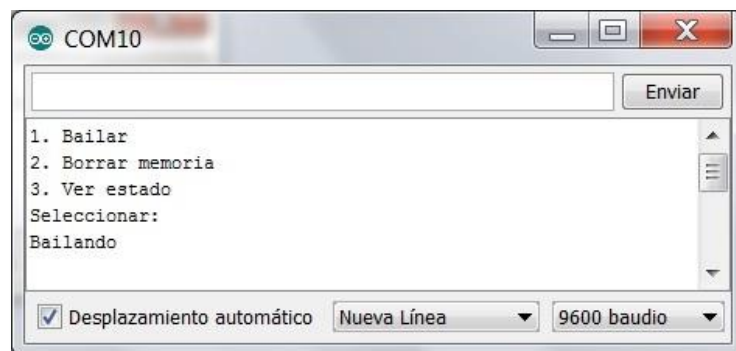
Ejecutar un programa que permita enviar y recibir datos por el puerto serie, como el propio del IDE de Arduino.

1) INSTRUCCIONES

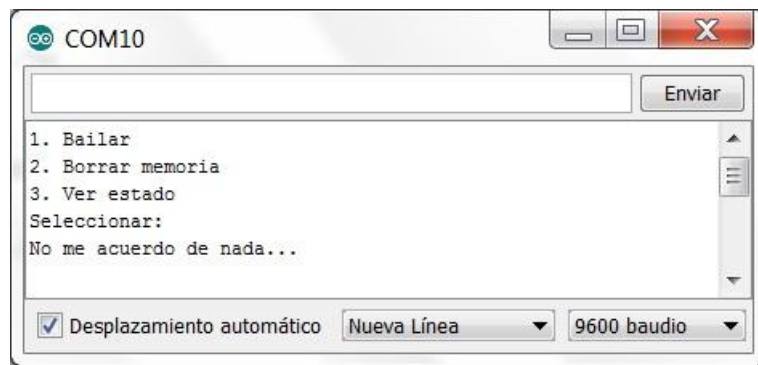
Al ejecutar el programa, el monitor del puerto serie mostrará un menú con las siguientes opciones:



Si se selecciona la opción de Bailar, el cartonbot reproducirá una melodía pregrabada y comenzará a moverse al ritmo.



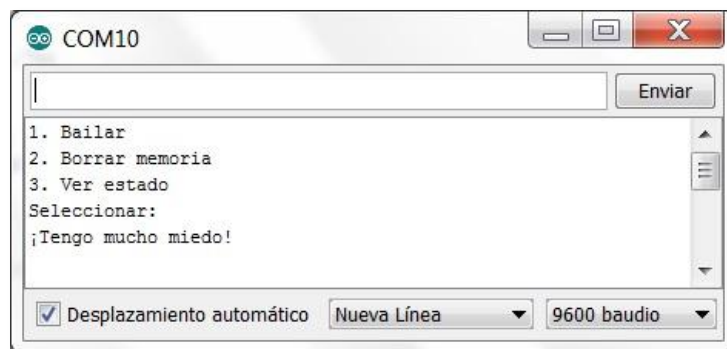
Si se selecciona la opción de Borrar memoria, borraremos toda la configuración.



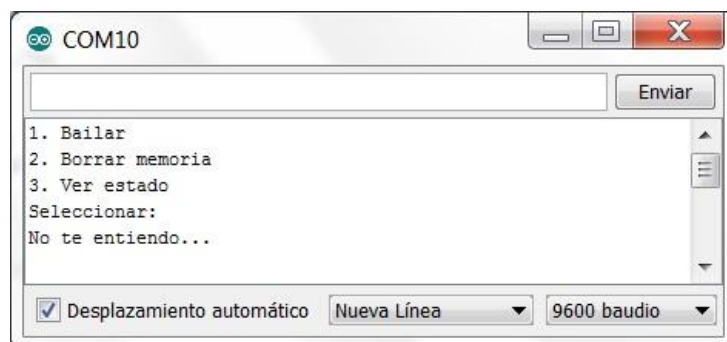
Si se selecciona la opción de ver estado, se mostrará el nivel de fobia.



En cualquier momento, ante la ausencia de luz, el robot emitirá mensajes en función del miedo que tenga (su nivel de fobia a la oscuridad).



Si se selecciona una opción (numérica) desconocida, el robot dirá que no te entiende y ejecutará una acción.



4. CONCLUSIONES

Tras la entrega de mi primer proyecto en Arduino, puedo decir que el resultado ha cubierto mis expectativas. Para verlo con más detalle, haremos un repaso a la lista de objetivos que nos planteamos antes de empezar.

- **Demostrar conocimientos:** En este proyecto se han englobado aspectos de programación, uso de librerías, entradas y salidas, tanto digitales como analógicas y PWM, además del puerto serial y la memoria EEPROM. Aun así, me hubiera gustado poder meter más sensores (distancia, inclinación), que hubieran dado más juego en la programación para el desarrollo de habilidades, además de poder usar otros elementos de comunicación, como bluetooth para poder haber controlado y monitorizado el robot a través de una aplicación Android o un módulo WiFi para hacerlo mediante una web.
- **Metodología:** La metodología seguida para la realización del proyecto me ha parecido adecuada. Esta metodología consta de las tres fases:
 - o **Análisis:** Donde se plantea el proyecto (enunciado), se describe qué se quiere hacer definiendo los objetivos, se describen los componentes necesarios para construir la carcasa y se calcula el presupuesto estimado para estudiar la viabilidad del proyecto.
 - o **Desarrollo:** Antes de montar las piezas, se debe diseñar la circuitería y pensar cómo fabricar una carcasa donde poder ubicar la electrónica. Una vez tengamos todo calculado, procedemos a construir la carcasa y montar la electrónica (estos procesos podrían hacerse en paralelo). Terminadas ambas cosas, procedemos a integrarlas. En paralelo a la construcción de la carcasa, se puede programar el proyecto, por capas, programando en primer lugar las librerías más básicas y escribiendo un sketch de ejemplo y testeo de cada una de esas piezas.
 - o **Entrega:** Terminado el desarrollo y testeo del proyecto, podemos proceder a la redacción del manual de uso y el documento de conclusiones.
- **Escalabilidad:** Para añadir una pieza (por ejemplo, oído), sólo hay que conectarla y:
 - o Crear la librería para su manejo.
 - o Añadir una función a la fábrica de piezas que sepa dónde está conectada la pieza y sea capaz de crearla.
 - o Utilizarla en las habilidades donde corresponda.
- **Modularidad:** El hardware ha sido desarrollado en pequeños módulos (ojo, boca, brazo), tal y como se describió y se mostró en las fotos en esta memoria. Esto a su vez ha sido llevado a la capa de programación, donde las piezas electrónicas se han abstraído en librerías modulares que facilitan la programación. Además, cada habilidad o utilidad es una librería modular.
- **Abstracción entre hardware y software:** A nivel de programación, gracias a la fábrica de piezas, hemos conseguido separar la lógica de funcionamiento de cada pieza del pin en el que está ubicado en la placa, de modo que hemos logrado cierta abstracción entre hardware y software. Sin embargo, en este apartado aún queda bastante por avanzar, ya que actualmente la fábrica de piezas conoce todos los componentes y su pin y cada habilidad debe instanciar cada pieza que necesita. En un futuro, se busca desarrollar una fábrica capaz de leer de un fichero XML ubicado en memoria SD que describa las piezas y el pin al que están conectados. De este modo, la abstracción sería absoluta.

- **Divertirse y aprender:** He podido desarrollar un robot, que era el objetivo inicial por el que me apunté al curso. Ha habido momentos desesperantes, principalmente a la hora de programar, tanto comportamientos “extraños” (por ejemplo, al poner un timer como entero, tras 32 segundos el robot se quedaba inactivo, ya que un int no puede almacenar más, o movimientos extraños de los servos al conectar más de una vez un servo al mismo pin) como limitaciones del lenguaje simplificado de C++, sobretodo en cuanto a la orientación a objetos, algo básico como la herencia no existe, ni se pueden lanzar hilos de ejecución, de modo que he tenido que cambiar la forma de programar para poder ejecutar paso a paso cada habilidad en lugar de lanzarlas en hilos, pero, una vez aprendida la base de cómo hacer las cosas, ver al robot funcionando y haciendo aquello para lo que le había programado ha sido sin duda una gran experiencia.

Adicionalmente, según fui desarrollando las habilidades (en concreto, la fobia a la oscuridad), introduje un factor de inteligencia artificial de aprendizaje, algo que no tenía contemplado en los objetivos iniciales. Es muy básico, pero permite al cartonbot ir aprendiendo a no temer la oscuridad. Este pequeño aprendizaje, en líneas futuras de desarrollo, se podría mejorar para dotar al cartonbot de personalidad y emociones y hacer que pudiera ir aprendiendo de ellas según su experiencia. De este modo, cada cartonbot sería diferente.

Como conclusión final, el cartonbot es un proyecto vivo que puede ir evolucionando, tanto añadiéndole nuevas piezas como nuevas habilidades o aspectos de personalidad y aprendizaje.