DISEÑO Y CREACIÓN DE UN ROBOT HUMANOIDE

TRABAJO DE INVESTIGACIÓN

DAVID ASENSIO CAÑAS

Entrega: 18/11/2011
Corrección: 12/1/2012

INTRODUCCIÓN 3

OBJETIVOS 4

CIMIENTOS TEORICOS 5

- **Hardware 5**
- **Software 11**
- **Opciones De Diseño 13**
- **Materiales 13**
- **Métodos De Construcción y Herramientas 14**
- **Conclusión Final 14**

CONSTRUCCIÓN 15

- **Diseño del robot 15**
- **Estructura, Fabricación y montaje 16**
- **Incorporación del hardware electrónico 18**
- **Creación del software 21**
- **Valoración del resultado final 24**

PRESUPUESTO 25

- **Presupuesto del proyecto 25**
- **Presupuesto real del robot acabado 26**

CONCLUSIONES 27

BIBLIOGRAFIA 29

GLOSSARIO 30

SIGLAS 31

AGRADECIMIENTOS 32

ANEXOS 33

INTRODUCCIÓN

¿Que es un robot humanoide?:

Existen muchas clases de robots, seguidores de líneas, cuadrúpedos, hexápodos, autómatas, etc...

En este caso se plantea hacer un robot humanoide, es decir, de apariencia antropomórfica.

Por supuesto, la característica básica de este robot será el estar compuesto por dos piernas que le proporcionaran el movimiento, por dos brazos que le ayudaran a equilibrarse o a sujetarse y por una cabeza que aunque no sea imprescindible, le aportará un aspecto más humano.

Bípedos actuales:

Actualmente, hay bastantes proyectos de robots humanoides, tanto a nivel de aprendizaje como a niveles industriales. El que más destaca, es ASIMO (Advanced Step in Innovative Mobility*)* creado en el año 2000 por la empresa Honda como símbolo de su inversión en I+D. ASIMO es el resultado del compromiso a largo plazo que definió la empresa en los años 80, cuando creó su primer robot bípedo, desde entonces, la empresa ha invertido una gran cantidad de dinero en este tipo de proyectos y realmente ha avanzado mucho, hasta el punto de que ASIMO es considerado el humanoide más avanzado del mundo, ya que es capaz de andar, correr, subir y bajar escaleras y simular muchas más acciones humanas, tal es su complejidad que la última versión presentada, es capaz de identificar objetos, evitar obstáculos e incorpora funciones muy avanzadas de visión artificial.

Después de esta breve introducción, solo cabe destacar, que actualmente, aunque estén muy avanzados, todavía no podemos atribuir ninguna función práctica a los robots humanoides, aunque la intención es que en un futuro sirvan de ayuda en la vida diaria de las personas, como por ejemplo, para proporcionar un soporte a personas discapacitadas, no obstante, este objetivo aún queda muy lejos de la realidad.

OBJETIVOS

En este trabajo se pretende construir un robot humanoide, es decir, de apariencia antropomórfica que sea capaz de llevar a cabo varias funciones, también propias de los humanos, como por ejemplo, andar, saludar, etc... Simultáneamente a esta labor, se intentarán reflejar los problemas que se pueden dar en un proyecto de estas características, las mejores maneras para resolver dichos problemas y las diferentes vías de expansión que puede tener el proyecto en base a los cimientos que se definan al inicio. También se analizarán las diferentes opciones de construcción y se valorará su utilidad en base a la ambición, la complejidad o el propósito del robot.

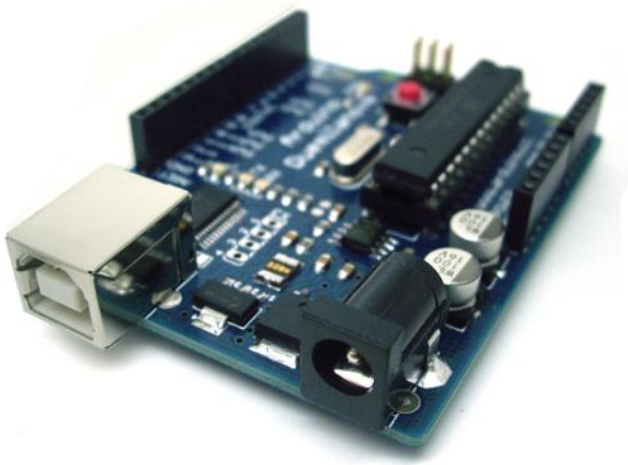
Cabe destacar que este tipo de proyectos aportan un dinamismo inusitado en lo que a posibilidades de expansión se refiere, ya que existen múltiples formas y opciones de añadir nuevas funcionalidades que no habían sido pensadas desde la base del proyecto, sin necesidad de tener que rediseñar todo de nuevo; se conseguiría simplemente con realizar algunos cambios mínimos. Por tanto se aprovechará esta peculiaridad para acentuar la complejidad del proyecto en la medida de lo posible.

CIMIENTOS TEORICOS

Hardware:

+ Placa controladora:

  » Arduino Duemilanove:



ARDUINO DUEMILANOVE

Arduino, es el fruto de un proyecto de código abierto muy ambicioso. Se trata de una placa con entradas y salidas digitales y analógicas, que permite ser programada para multitud de funciones, desde las más simples, como el parpadeo de un LED, hasta otras un poco más complejas como controlar los diferentes motores de un robot bípedo.

A modo de "cerebro electrónico" o microprocesador, la placa Arduino Duemilanove (que es en la que me voy a centrar) tiene un chip ATmega 168 o ATmega 328 con un *bootloader* pre-instalado que nos permite cargar nuestros propios programas directamente desde el puerto USB, evitando así los tediosos programadores convencionales en los cuales tienes que extraer el microprocesador de la placa para programarlo.

Además, Arduino tiene su propio entorno de programación y compilador, que está escrito en Java y basado en Processing, del que ya hablaremos más adelante. El lenguaje que interpreta dicho entorno es muy similar al lenguaje C++, pero con algunas modificaciones, ya que Arduino dispone de sus propias librerías para interactuar mejor con el *hardware.*

Las placas de Arduino han adquirido mucha popularidad en los últimos años, y esto ha sido un gran punto a favor para esta plataforma, ya que diversas empresas han desarrollado extensiones de *hardware* o más comúnmente llamados *shields*, que nos permiten incrementar las funciones de nuestra placa, como por ejemplo leer datos de una tarjeta SD o reproducir archivos en formato MP3. En definitiva, Arduino es una placa versátil, con infinitas opciones de expansión, intuitiva y fácil de programar.

Ventajas de Arduino para este proyecto:

- Muy versátil: Ya he hablado de la cantidad de extensiones de *hardware* que tiene Arduino lo cual posibilita una gran capacidad de adaptación a cualquier proyecto.

- Multiplataforma: Es una de las características más importantes de Arduino , y no solo en cuanto a sistemas operativos, sino que además, fluyen por Internet centenares de proyectos y decenas de librerías que lo hacen compatible con Arndroid, Symbian, iOS, etc... Además también es compatible con Processing, Max y en definitiva, cualquier plataforma capaz de enviar datos por el puerto serie del PC.

- Software y hardware libres: Esta característica permite modificar a nuestro gusto cualquier aspecto de el *hardware* o el *software* que componen Arduino sin tener que preocuparse por los derechos de autor, además, en la página del proyecto nos facilitan los esquemas en Eagle y el código de las placas y del entorno de programación.

- Variedad de hardware: Aunque para este proyecto solo me centre en la placa Duemilanove, hay que tener en cuenta que Arduino dispone de muchas más placas como la placa Mega, la placa Nano, la placa LilyPad, la placa Mini, la placa Pro y un largo etcétera de placas que se adaptan a las distintas necesidades de cada proyecto.

- Tamaño: El reducido tamaño, en este caso de la placa Duemilanove, es perfecto para este tipo de proyectos.

- Coste: El coste de un Arduino Duemilanove montado es muy competitivo, 22€ e incluso más barato si se sabe buscar en Internet, pero se pueden reducir todavía más los costes haciéndotelo tu mismo con los esquemas disponibles en la pagina del proyecto.

Inconvenientes de Arduino para este proyecto:

- No especifica para robótica: Una de sus ventajas en el mundo de la electrónica, se convierte en desventaja al usar un Arduino como parte lógica de un robot humanoide, ya que al no tener un uso concreto, la placa Duemilanove tiene serios problemas para alimentar los 8 servos que inicialmente requiere este humanoide, esto se debe a que los servomotores consumen una intensidad variable en función del esfuerzo al que están sometidos, por esa razón si se quiere usar dicha placa para controlarlos, debemos adaptarla a nuestras necesidades, en este caso, de consumo.

Cabe destacar que si bien una de las ventajas de Arduino es su reducido coste, este aspecto de la adaptación lo incrementa un poco, en todo caso ya se valorará más adelante si el coste es asumible o realmente es mucho mejor utilizar alguna placa más adecuada para robótica.

» Placa Imagina:

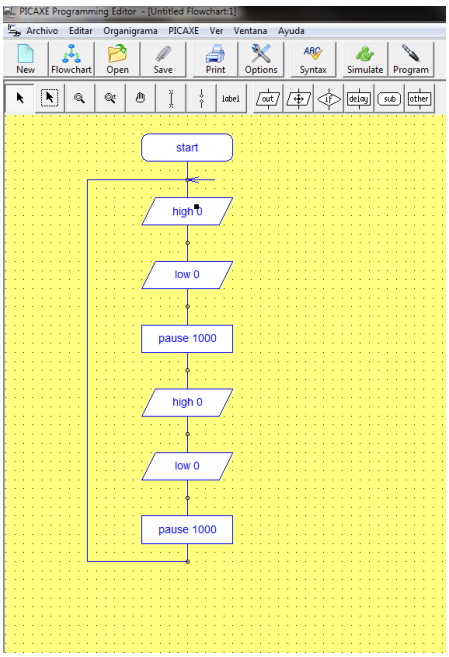
PLACA IMAGINA

Esta placa fue diseñada por el equipo de Robolot, por encargo del "Departament D'educació De Catalunya" con la finalidad de que los alumnos de 4º de la ESO tengan fácil acceso a la robótica y electrónica, a un precio asequible.

A igual modo que la placa Arduino Duemilanove, la placa Imagina dispone de un microprocesador que funciona como "el cerebro" de la placa. Este microprocesador, es un PIC (Peripheral Interface Controller) de microchip, pero que también ha sido pre programado con un *firmware* llamado PICAXE que nos permite programar dicho PIC de una manera

mucho más fácil y rápida, ya que se hace directamente desde el puerto serie del ordenador a la placa.

La plataforma PICAXE también cuenta con su propio entorno y con su propio compilador, este usa el lenguaje BASIC que es sencillo, versátil y potente o también se sirve de diagramas de flujo para generar el código sin que el usuario tenga que tener el mínimo conocimiento de programación.



E AQUÍ UN EJEMPLO DE DIAGRAMA DE FLUJO

PICAXE fue diseñado en un primer momento, con miras de ser una plataforma educativa, pero a raíz del éxito que ha tenido, muchos aficionados lo usan en sus proyectos. Este éxito se debe a que los chips con el *firmware* PICAXE son de bajo costo y fáciles de programar, con una plataforma de software libre y un lenguaje fácil de aprender.

Ventajas de la placa Imagina para este proyecto:

- Multiplataforma: Al igual que Arduino, PICAXE también es multiplataforma, no obstante, lo es de distinto modo, me explico, así como en Arduino el entorno de programación está escrito en Java y por consecuente, se puede ejecutar en cualquier plataforma que soporte una JVM (Java Virtual Machine), en PICAXE existen diferentes compiladores para Windows y Linux, pero el *PICAXE Programming editor* que es el oficial, solo puede ejecutarse en Windows y para Linux o Mac, podría usarse por ejemplo el *AXE Pad.*

- Sencillo e intuitivo: El entorno de programación de PICAXE utiliza BASIC (Beginner's All-purpose Symbolic Instruction Code) como lenguaje para programar los microcontroladores, y como bien indica el nombre, este es un lenguaje pensado para principiantes, y en consecuencia, es bastante fácil de usar y de aprender, además, el *PICAXE Programming Editor* nos permite generar código a partir de diagramas de flujo, con lo cual no hace falta ser un gran programador para hacer funcionar las funciones básicas de un sistema PICAXE. Sin embargo, no debemos olvidar, que en el fondo, los microcontroladores que PICAXE usa

son simples PICs con un *bootloader* que nos facilita la programación de estos, por lo tanto, si disponemos de un programador de PICs como el Pipo o muchos otros, también los podemos programar en C.

- Especifica para robótica: Lo mejor que tiene esta placa, es que ha sido diseñada específicamente para controlar 8 servomotores, lo cual es muy beneficioso para este proyecto, ya que como dije antes, lo mínimo que requerirá serán 8 servos.

Inconvenientes de la placa Imagina para este proyecto:

- Tamaño: Esto no es un problema definitivo, pero aun así es una traba en lo que al diseño se refiere, el tamaño de esta placa es un poco excesivo para un proyecto de este calibre, ya que la escala del robot esta acotada por la potencia de los servos y cuanto más ocupe la placa más tendrá que ocupar el tronco del robot y por consecuente, el peso del robot se verá afectado. Me reitero diciendo que esto no es un gran problema, pero considero que también hay que tenerlo en cuenta, sobretodo en el momento de diseñar el tronco del robot. Este problema, radica en el diseño del PCB, que está hecho a una sola capa, seguramente se reduciría considerablemente el tamaño si se hiciera a doble capa y con componentes de encapsulado SMD, pero esto ya es un cometido del fabricante.

+ Actuadores:

» Servomotores:

Los servomotores son motores direccionales y posicionables que en este caso serán los encargados de dar movilidad a las articulaciones del robot, generalmente, están compuestos de tres partes:

- Pequeño motor DC: En su interior, el servomotor alberga un pequeño motor de corriente continua que está conectado a una reductora, este es el encargado de proporcionar el movimiento.

- Reductora: Una reductora, es un cúmulo de engranajes interconectados entre si y diseñados de tal manera que intercambian potencia de tracción por velocidad, o dicho más coloquialmente, si el motor que conectamos a la reductora gira doscientas veces y dicha reductora tiene ½ de reducción, el eje de salida de la reductora girará solo cien veces pero tendrá el doble de fuerza que el motor.

- Circuito controlador: Interiormente, los servomotores disponen de un circuito para facilitar el posicionamiento del motor DC, debido a que los motores DC solo tienen una conexión de alimentación que los hace girar continuamente, resultaría excesivamente complejo posicionarlos con precisión, por esa razón, los servomotores disponen de un pequeño circuito que interpreta las señales externas y las traduce para controlar el motor, es decir, si el circuito recibe, que el servo debe posicionarse a 30^{o}, el circuito alimenta el motor durante el tiempo necesario y en la polaridad correcta para que esto ocurra, de esta manera, solo debemos preocuparnos de mandar la posición correcta a cada servomotor.

Para este proyecto en concreto, he decidido usar servomotores estándar de la marca Modelcraft, pero cabe destacar que en cuestión de calidad no son estos los más apropiados, no obstante, me he decantado por esta opción por motivos económicos, ya que por sí solos, los servomotores no son excesivamente caros (el precio oscila entre 5 y +40 €) pero teniendo en cuenta que para montar el humanoide se necesitará un mínimo 8, el precio aumenta considerablemente. Este modelo en concreto cuesta en internet 4,95€ más gastos de envío.

+ Sensores:

» Receptor IR:

Un receptor IR es simplemente un sensor que detecta la luz infrarroja (invisible para el ojo humano) y devuelve un valor en base a los milisegundos que ha habido entre ráfaga y ráfaga de luz. Esto es de una gran utilidad, y tiene múltiples aplicaciones, como por ejemplo, controlar aparatos a distancia como una televisión, o en este caso un robot.

» Bluetooth:

El Bluetooth es una tecnología en pleno auge a causa de la actual necesidad de comunicación entre dispositivos móviles. Se basa en el envió de señales similares a las de la radio y se usa mayormente para compartir archivos entre dos dispositivos cercanos. No obstante, tiene muchas más utilidades como la emulación de puertos COM, la cual aprovecharemos para comunicar nuestro robot con un dispositivo móvil y con él ordenador.

En este caso, he elegido el módulo Bluetooth emulador de puerto COM a 9600 baudios de Sure Electronics. He escogido este a causa de su flexibilidad, su reducido costo (apenas 17 € más gastos de envío) y su sencillez de uso.

Software:

+ Placa:

No me voy a demorar mucho ya que anteriormente, en la valoración de las placas, ya he hablado del software que usan, considerando, que lo que se entiende como software de la placa, sería el IDE de Arduino o PICAXE y el *bootloader* que ambas incluyen en su chip controlador.

+ Ordenador:

Tanto para programar como para diseñar, o hacer cualquier otra cosa, se hará sobre el sistema operativo Windows, ya que ofrece compatibilidad para absolutamente todo el software necesario.

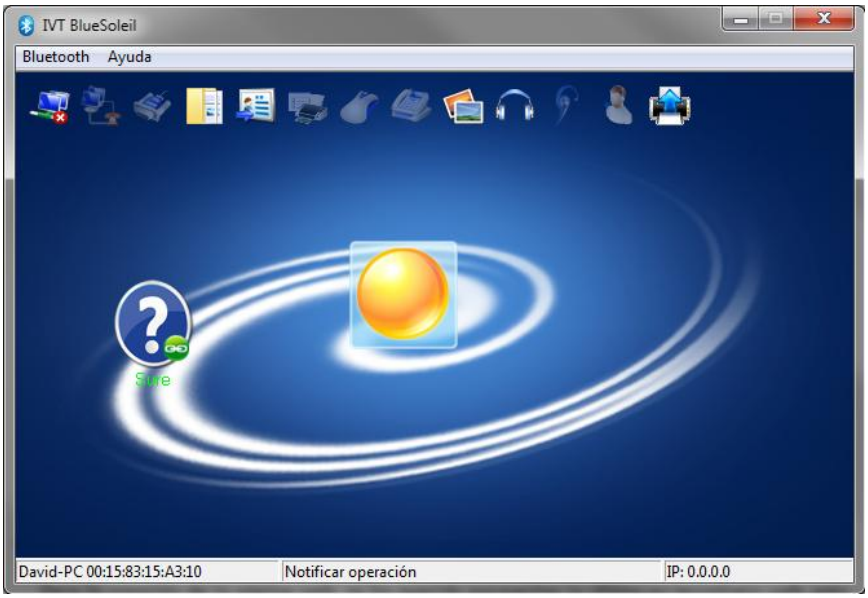
Más adelante, se confeccionará una lista de todos los programas utilizados para llevar a cabo el proyecto.

+ Periféricos móviles:

De los múltiples sistemas operativos para móviles existentes hoy en día (Android, Symbian, PalmOs, IOS, etc..) me he decantado por utilizar Android, y no solo porque es el que tengo más a mano, sino porque además es muy sencillo crear aplicaciones que corran sobre él. Esto es a causa de que la empresa Google (creadora del dicho sistema operativo) ha desarrollado AppInventor, un IDE online para crear aplicaciones sin tener que tocar una sola línea de código. Además, hay un proyecto llamado Amarino, que nos permite pasar datos de puerto serie a cualquier emulador COM Bluetooth sin necesidad de crear ninguna aplicación adicional.

+ Comunicación:

El software que he escogido para realizar la conexión Bluetooth con el ordenador, quizás no es el más apropiado, ya que es de pago. No obstante, existe una versión de prueba que se puede usar durante un tiempo limitado, además, es una simple comodidad, ya que sin necesidad de este programa, se puede configurar puertos COM Bluetooth, no obstante es mucho más engorroso, sobre todo cuando se están realizando pruebas.



IVT Blue Soleil

El software mentado se llama IVT Blue Soleil y es bastante intuitivo y fácil de usar, para empezar realiza un barrido para encontrar cualquier dispositivo Bluetooth cercano al que conectarse, luego, te los muestra en pantalla y te deja la opción de conectarte a ellos. Una vez escogido el dispositivo, el programa analiza los servicios que el dispositivo ofrece (en este caso puerto COM) y te ofrece la opción de conectarte a él usando uno u otro servicio.

Opciones de diseño:

En principio, la idea es partir de un diseño ya existente e irlo modificando en base a las necesidades del proyecto y a la complejidad que se quiera obtener. Como base, he decidido usar un diseño del equipo Robolot el llamado robot HU2, un diseño de Toni Moreno y Joan Pellicer el cual usa 8 servomotores y tiene 2 articulaciones cada hombro y 2 más en cada pierna.

Cuando se haya completado con éxito la construcción de este primero prototipo, se intentará añadir una articulación más en cada pierna, a modo de rodilla y también se modificará el torso para que sea más robusto. Esto encarecerá un poco más el proyecto, ya que se usarán 2 servomotores más, que harán un total de 10.

Materiales:

Para construir el chasis del robot, he decidido usar aluminio de 2 mm en mayor medida. Se ha elegido este material a causa de su robustez, su poco peso, su manejabilidad y su precio. Seguramente, otros materiales, como el titanio o la fibra de carbono hubieran sido muchísimo mejor candidatos para construir un robot de estas características, pero si bien su precio hubiera sido astronómico y su obtención, más dificultosa, tampoco hubieran resultado mucho mejores que el aluminio, además, para dar forma a estos elementos, se necesitan un tipo de maquinaria especial de la cual, en un principio, no dispongo.

También se ha usado, aunque más puntualmente, plástico PVC para los pies y los brazos del robot, ya que es recomendable, que estas partes sean más flexibles, para evitar roturas o dobleces no deseados que alteren el punto de gravedad del robot.

Métodos de construcción y herramientas:

Para construir el primer prototipo, se hará con las herramientas más comunes de un taller, como el taladro, la sierra de calar, limas, destornilladores, etc...

Sin embargo, para construir el modelo final, se considerará la opción de crear los planos en un programa CAD (Diseño asistido por computadora) e introducirlos en una máquina de control numérico para que corte las piezas, haciendo así que el resultado final sea mucho más ameno para programarlo, ya que será más preciso, y mucho más vistoso.

Conclusión final:

En base a las ponencias anteriores, en este apartado se valorará y se escogerá con que materiales, software, hardware, etc... Se va a empezar a confeccionar el robot.

Respecto a la placa, finalmente, he decidido utilizar la placa Arduino Duemilanove, ya que esta ofrece múltiples opciones de expansión y personalmente, me siento más cómodo programando en C que en BASIC, si bien es cierto que quizás el hardware se tiene que adaptar, esto puede ser beneficioso más adelante para explicar un poco más a fondo la parte electrónica del proyecto. No obstante, creo necesario decir que la placa IMAGINA ostenta múltiples cualidades que también la hacen muy capaz para este proyecto.

Respecto a los actuadores y los sensores, no hay nada que escoger, pues solo cabe una posibilidad en los tres casos y ya he hecho mención de ella antes.

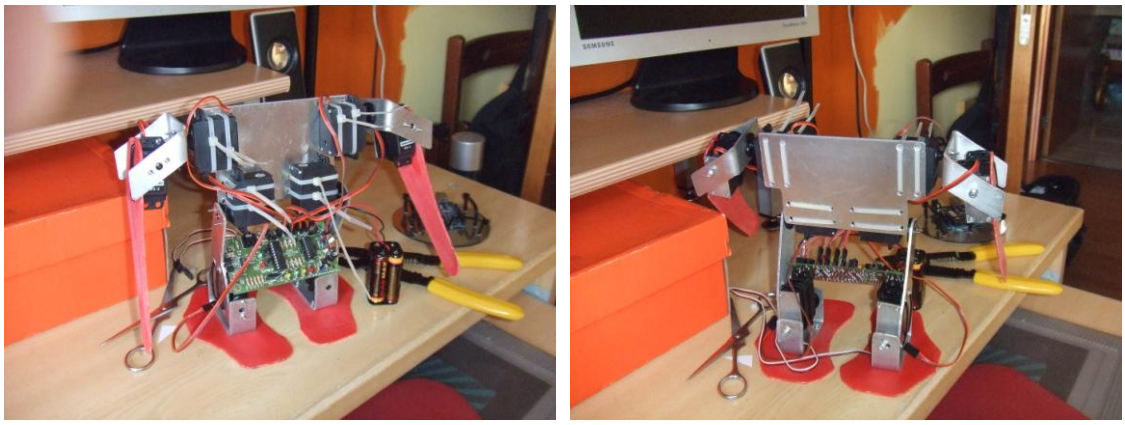
El software que se utilizará, está condicionado por la placa, por tanto, solo hay una opción, el IDE de Arduino para programar la placa, Windows como sistema operativo y a causa de la disponibilidad, usaré Android para los dispositivos móviles, ya que lo tengo más a mano.

CONSTRUCCIÓN

Diseño del robot:

Sin duda alguna, para empezar a construir un robot, primero se tienen que diseñar sus partes y pensar dónde irá cada pieza que lo componga. En el caso de este proyecto, a priori se montará un primer prototipo, que como bien he dicho antes, se hará a partir de unos planos desarrollados por el equipo Robolot y a posteriori, se diseñaran nuevas piezas para aumentar la funcionalidad del robot.

Después de montar el primer prototipo, el tronco del cual estaba compuesto por CD's, se ha observado que tiene poca consistencia, y que los CD's son demasiado frágiles para componer el cuerpo del robot. Para solventar este problema, se ha diseñado un nuevo cuerpo, que será realizado en aluminio para añadir robustez al robot, cabe destacar que esto es una modificación estrictamente necesaria, ya que los CD's son realmente frágiles, pues sin ni siquiera poner en funcionamiento el robot, se agrietaron por los puntos de sujeción de la placa controladora, dejando el conjunto inservible.



PRIMER PROTOTIPO

Otra pega de los CD's es que son realmente estrechos y a causa de esto las piernas quedaban estrictamente juntas, hasta el punto de ser una traba para el correcto funcionamiento del robot. Por estos motivos, y a falta de reiterarme, he diseñado un tronco más largo, más robusto y a mi modo de ver, más vistoso.

Otra de las modificaciones que cabe destacar en el diseño del robot, son las extremidades superiores de las piernas, estas se añadieron, a diferencia del tronco, no por necesidad sino para

añadir nuevas funcionalidades al robot, como por ejemplo, agacharse o subir un escalón. Las extremidades superiores de las piernas, que proveen de rodillas al robot, estarán echas, al igual que todo el conjunto, de aluminio, y estas no se incluirán hasta el diseño final.

En cuanto al diseño integro del robot, lo he desarrollado completamente con AutoCAD, a causa de sus múltiples opciones y su sencillez y su interface intuitiva. También probé un software libre llamado qCAD pero al final me decanté por AutoCAD porque personalmente, ya tengo cierta soltura al usarlo, aunque eso sí, es de pago, no como qCAD.

La razón por la cual decidí realizar los planos en un programa CAD radica en que quería dejar abierta la posibilidad de introducir los planos en una máquina de control numérico para que creara las piezas, posibilidad que, como ya veremos más adelante, al final se hizo tangible.

Estructura, fabricación y montaje:

El primer prototipo, se construyó en el taller, con herramientas comunes, se usó la sierra de calar para dar forma a las piezas de aluminio y unas tijeras para dar forma a las de PVC, una vez cortadas las piezas, se hicieron los agujeros convenientes, para fijar los servomotores y demás piezas. Respecto al funcionamiento del primer prototipo, se consiguió que caminara, aunque con ciertas dificultades, ya que el eje del "tobillo" derecho estaba desviado a causa de que los agujeros no coincidían convenientemente. Después de esta traba, causada por la poca precisión que se obtiene cortando aluminio a mano, decidí acabar los planos con AutoCAD, incluyendo también las piezas de la parte superior de la pierna y acotarlos convenientemente para su posterior realización por control numérico.

Las cotas de los planos fueron realizadas en base a las indicaciones dadas por el operario de la máquina CNC trabajador de la empresa SHINE de Vidreres, el cual indicó que era conveniente acotar las piezas desde un punto 0 que sería el punto de referencia que tendría la máquina respecto al resto de la pieza, también indicó que las piezas se tenían que acotar por separado. Una vez terminadas todas las cotas, se procedió a la conversión de los planos a lenguaje Heidenhain, que es el lenguaje que interpreta la máquina de control numérico, las piezas más sencillas se pasaron a lenguaje manualmente, pero las piezas con un poco más de complejidad, se tradujeron con un programa llamado CAMworks y un plugin de la empresa creadora del lenguaje Heidenhain. **(Para una explicación más detallada, mirar anexo nº3)**

Para cortar las piezas, se hizo de una en una y indicando mediante el lenguaje Heidenhen el cambio de herramienta, que en este caso, se necesitaron brocas de varios tamaños, una fresa de 0,5 cm de diámetro, una broca de avellanado para repasar los agujeros con tornillos y una rosca chapa de métrico 3 para hacer las muescas de los tornillos.

Durante el proceso de corte, surgieron varios problemas, para empezar, el primer inconveniente fue la holgura de la chapa de aluminio a causa de la cual, los avellanados de los tornillos no eran uniformes, esto se solventó rápidamente colocando las bridas de sujeción cada 5 cm más o menos y cerca de donde realizaba el corte la máquina, después, hubo un problema con el liquido refrigerador de la máquina, coloquialmente llamado “taladrina”, este dejó de salir espontáneamente y por consecuencia, el borde de una de las piezas se derritió un poco a causa del calor originado por la velocidad de fresado, pero esto se solventó enseguida accionando el segundo circuito de refrigerado de la máquina, no obstante, la pieza se pudo aprovechar, aunque se distingue el borde más rugoso que el de las demás.

En tercer y último lugar, tuvimos un problema con las piezas más pequeñas, las tapas de sujeción de los servomotores, que a causa de su reducido tamaño cedían a la potencia de la máquina y se rompían quedando inservibles, pero esto se solventó bajando la velocidad de fresado, tanto de rotación como de avance y sujetando cuidadosamente la pieza con una varilla de punta engomada.

Una vez cortadas todas las piezas, se pulieron los bordes de estas con una cuchilla para cortar aluminio y se allanaron los ángulos interiores que no se habían cortado a 90° a causa de que la herramienta fresadora de la máquina tiene forma redonda, también se repasaron los avellanados y los agujeros que no habían quedado del todo definidos y se pulieron las piezas con una especie de estropajo especial para pulir metales con poca dureza, también se tuvo que hacer una pequeña modificación en las piezas que constituyen la parte superior de la pierna, ya que en los planos, no se había dejado suficiente margen para la colocación de los servos, pero esto fue una minucia que se solventó lijando a mano cada pieza.

En realizar todo el proceso de corte, teniendo ya de antemano los planos convertidos a lenguaje Heidenhain, se invirtieron un total de 4 horas.

Con las piezas ya terminadas y los servos en mano, se procedió a montar el robot, en este paso, me topé con un inconveniente, los tornillos de sujeción de los servomotores son un poco extraños, y bastante costosos de encontrar, pero finalmente los encontré en una tienda de maquetas en Calella. Para el montaje, se usó, una cortadora, un cúter, un destornillador, una mini-cortadora radial, bridas, *Superglue* y unos alicates de electrónica y el único inconveniente que surgió, fue que los servos que constituyen la pelvis de robot, tenían cierta holgura, pero lo solucioné introduciendo una pieza de metal entre el servo y la brida y fijándola con *Superglue*, también usé una vara con muescas de tornillo para agregar robustez al conjunto. Finalmente, se atornilló el Arduino y se insertó el *Shield*, del que más adelante hablaré.

Incorporación del hardware electrónico:

Con hardware electrónico, me refiero a la placa Duemilanove y al *Shield* creado para dar potencia a los servos. En un principio, a causa de mis escasos conocimientos de electrónica, cometí el fallo garrafal de conectar los servos a la placa Arduino con un simple circuito que no requería alimentación externa, la obtenía toda de la placa, pero enseguida me di cuenta, en cuanto conecté más de 3 servos, que esta alimentación no era suficiente y tuve que investigar por internet para encontrar alguna solución, después de mirar varios esquemas, descubrí que existen unos reguladores de tensión llamados 7805 que regulan cualquier corriente de entrada a una corriente de salida que proporción 5 voltios y un máximo de 1 amperio por cada regulador, compre dichos reguladores y empecé a crear el primer *Shield* de alimentación para la placa Arduino Duemilanove.

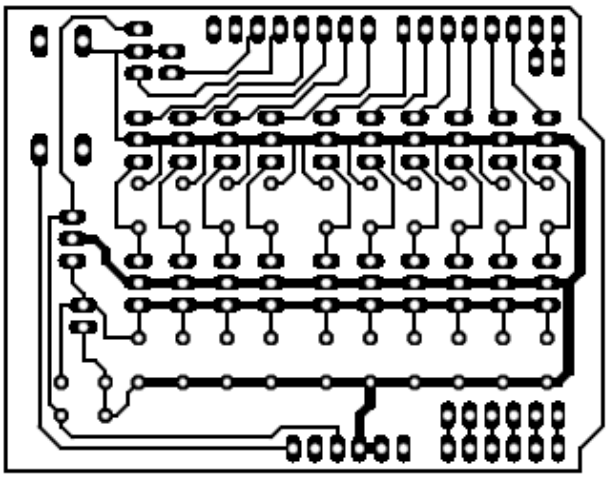
+ Modificaciones u adaptaciones:

  La primera adaptación llevada a cabo, fue la incorporación del *Shield* de alimentación, el cual se creó en un primer momento con un programa llamado Fritzing, pero dicho programa, está pensado para la creación de PCB's muy simples y con componentes electrónicos muy, muy básicos y por tanto, tuve problemas para incorporar los 7805, por esta razón, busqué otro software llamado Eagle, que está muy extendido en el mundo de la electrónica, tiene múltiples funciones y dispone de miles de librerías de componentes como por ejemplo la de Arduino, que permite crear los ya mencionados *Shields* o extensiones de *hardware*, por supuesto, los reguladores 7805 también disponen de una librería propia, que facilita mucho la incorporación de estos en cualquier circuito que hagamos.

Otra cualidad a tener en cuenta de este *software*, es la posibilidad de exportar los archivos que hemos creado a un formato aceptado por los fabricantes industriales de PCB's, dando opción así a enviar nuestros esquemas a una central especializada que nos haga el circuito de una manera más profesional, aunque por el momento, los PCB's que realice serán "caseros", no obstante creo que es un punto a tener en cuenta en miras de obtener un acabado más profesional, ya que si se mandan los circuitos a una empresa especializada, se puede reducir en mucho su tamaño, y esto es un punto a favor.

En la elaboración del primer circuito, cometí el segundo fallo garrafal al incorporar el botón de reseteado de la placa, ya que en vez de puentear el pin de reset con el pin de masa, por un descuido lo puenteé con el de 5,5 voltios, y además, no fui consciente, de que la masa de la placa de alimentación y la de la placa Duemilanove, tenían que ser comunes para que los servos funcionasen. No obstante, con una pequeña modificación, este circuito me permitió comprobar que la alimentación con reguladores de tensión funcionaba y me dispuse así a hacer la segunda versión del circuito, en la que además de corregir el error del reset y de la masa, también incorpore capacitores a la entrada y la salida de los reguladores, ya que esto lo recomienda el fabricante para reducir el ruido del circuito, cosa que en este caso es bastante beneficiosa, porque los servomotores generan gran cantidad de ruido en el circuito. **(Para una explicación más detallada, mirar anexo nº2)**

La segunda versión del circuito, ya fue la definitiva, ya que era totalmente operativa y no tuve que gastar ni un céntimo, mas que en los capacitores, ya que los reguladores, los desoldé del circuito anterior. Además, a esta nueva versión de la placa, le añadí una entrada para conectar el sensor IR.



CIRCUITO FINAL

Con el circuito de alimentación terminado, ya se podía empezar a programar el robot. No obstante, más adelante explicaré algunas de las pegas que tiene este circuito, ya que, otra vez por mi falta de conocimientos en electrónica, no tuve en cuenta que los reguladores tenían que soportar una caída de tensión muy acentuada, a causa de que la batería utilizada para el proyecto da un voltaje de 11 voltios, y al tener que convertirlos a 5 voltios, el regulador se calienta notablemente, aunque esto es fácilmente solucionable con una batería que proporcione un voltaje más bajo.

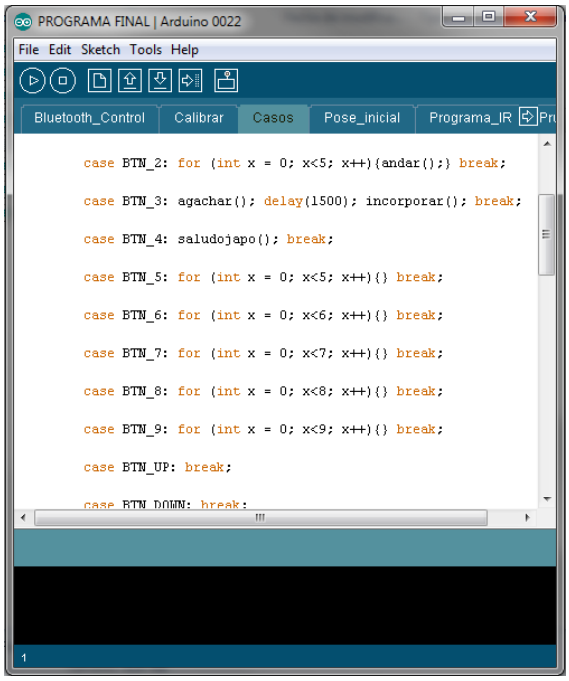
Otra modificación llevada a cabo en el hardware fue la adaptación de los servomotores, aunque sea una adaptación mínima, me di cuenta de que era bastante practica a la hora de crear el circuito, esta adaptación consiste simplemente en el cambio de posición de los pines del servomotor, la cual está marcada por un estándar que define el orden de los pines en pin de control, pin de alimentación de 5 voltios y pin de masa, y yo, para ahorrarme dificultades en la creación del PCB, alterné estos dos últimos, de manera que convergían mejor con la ordenación de pines del 7805 y por tanto, me simplificaba bastante la conexión entre estos y los servos.

A parte de estas modificaciones, todo lo demás son minucias, aunque creo que es bueno destacar también la creación necesaria de un circuito independiente para el modulo Bluetooth, ya que por necesidad, este va conectado a unas resistencias indicadas por el fabricante y además de ahorrarme espacio en el circuito principal, permite también la posibilidad de usar el modulo de manera independiente, posibilitando así la opción de usarlo en futuros proyectos sin tener que comprar otro.

Creación del software:

+ Firmware placa:

El firmware de la placa se compone de unas cuantas partes, la primera parte, sería el *bootloader* del chip ATmega de la placa Arduino, el cual ha sido desarrollado por el equipo Arduino, y con el cual yo no tengo nada que ver, solo hago uso de él y de las comodidades que ofrece. La segunda parte es el código escrito por mí, y es el que hace que el robot funcione.



IDE ARDUINO

Este código, lo he dividido principalmente en cuatro bloques para que sea más cómoda su modificación y la incorporación de nuevas funciones, sin necesidad de modificar la estructura del programa, estos bloques son:

- Programa base: El programa base está constituido básicamente por tres partes, la primera parte, es la llamada a las librerías de las que se hará uso, las cuales simplifican el programa y la declaración de las variables convenientes para hacer más ameno el trabajo. La segunda parte está compuesta íntegramente por una función llamada *SETUP*, esta función es la función de configuración del programa, es decir, solo se ejecuta una vez y generalmente se usa para definir los pines de salida y entrada de la placa, los cuales es mejor que sean estáticos, no por necesidad sino por pragmatismo, aunque

dicha función también se puede usar para ejecutar un fragmento de código sólo al inicio del programa.

Finalmente, la tercera parte de este bloque, está compuesta por la función *LOOP*, el usó de la cual es deducible a partir del nombre, dentro de esta función se incluye el código que va a ser ejecutado indefinidamente por el programa, es decir, todo lo que va a estar ejecutándose de una manera continua, ya sea a la espera del cambio de valor de algún sensor o simplemente, un programa que se ejecute continuamente, en este caso, esta función contiene llamadas a los demás bloques del programa.

- Funciones motrices: Las funciones motrices, son en si las que originan el movimiento de los servomotores del robot, de tal manera que estos describan un movimiento entendible para el ojo humano, me explico, estas funciones, resumen el posicionamiento de cada uno de los servomotores del robot en movimientos tales como agacharse, levantarse, sostenerse en pie, etcétera. De esta manera, cada vez que se desee que el robot haga una de estas funciones, simplemente se llamara a cada función, ahorrándome así el espacio que ocuparía volver a escribir todos los movimientos de cada servomotor.

  Dichas funciones están compuestas por dos partes, en primer lugar, una declaración de variables locales (útiles solo dentro de cada función) las cuales contienen los valores generales del posicionamiento de los servomotores, de esta manera, en la función andar, por ejemplo, nos encontramos con que se compone de las variables, amplitud de paso, velocidad, etc... De esta manera si se quiere modificar la manera de andar, solo es necesario modificar estos valores, esto es muy útil en el momento de programar nuevas funciones y ver si funcionan aplicando un método puro de inducción. En segundo lugar, tenemos lo que podríamos llamar el cuerpo de cada función, en este, se hace uso de los valores de las variables locales y se define la posición de cada servomotor en cada instante de tiempo a partir de estos valores. Esta ultima parte, no tiene que ser modificada necesariamente para cambiar el comportamiento de cada función.

- Control de mando IR: Este bloque contiene lo que en programación se denomina como un *SWITCH CASE* esto no es más que un fragmento de código, que analiza el valor de una variable (en este caso, una variable modificada por el sensor IR) y reacciona en base

a este valor, esto permite la asignación de cada botón en el mando, a una o varias funciones motrices, de tal manera que el robot llevará a cabo una u otra acción dependiendo de qué botón sea pulsado en cada momento.

- Control de mando Bluetooth: El control por Bluetooth, es un bloque sumamente parecido al anterior, aunque en este caso, se analizan los datos obtenidos por el puerto serie de la placa, (al que está conectado el modulo Bluetooth), y otra diferencia, es que aquí no se usa un *SWITCH CASE* sino que se usan una serie de condiciones *IF/ELSE*. No porque necesariamente tenga que ser así, sino que a mí personalmente, me es más sencillo programarlo así, ya que de la otra manera no logré el resultado esperado.

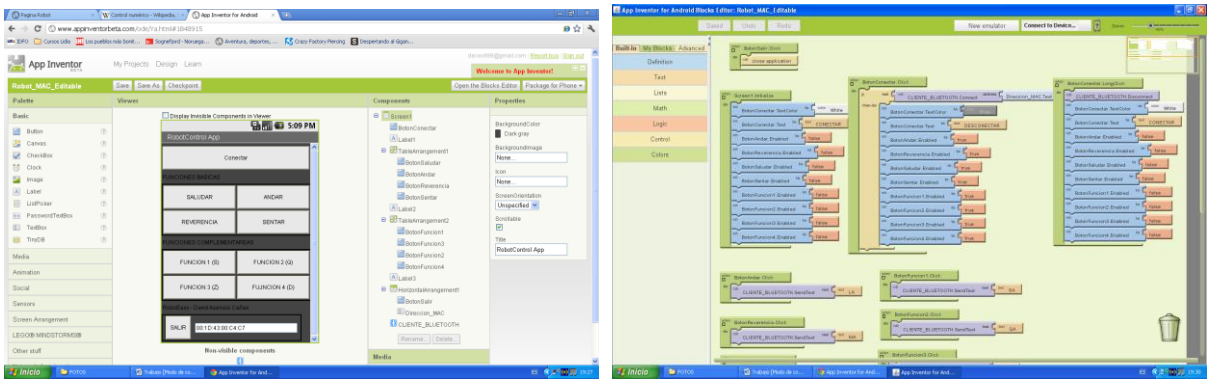
+ Software PC:

En un principio, estuve barajando la posibilidad de crear un software con Processing para controlar el robot desde un PC, pero a causa de la poca estabilidad que tiene el puerto COM Bluetooth, Processing se bloquea cada vez que intenta enviar un dato por dicho puerto, no digo que esto sea un problema de Processing, sino que seguramente, no he logrado dar con la manera de enviar datos correctamente por Bluetooth, de hecho, considero que no tengo mucha experiencia con el lenguaje de programación de Processing, que aún ser extremadamente parecido al de Arduino, incorpora funciones bastante complejas, que el tiempo material de desarrollo del proyecto no me ha permitido estudiar con detenimiento, no obstante, si conectamos el robot por cable en vez de por Bluetooth, si que he logrado enviar y recibir datos satisfactoriamente, de hecho, para calibrar los servomotores del robot, he escrito un pequeño programa en Processing, que permite posicionar cada servo y devuelve el ángulo de este, de esta manera, me he facilitado de todas, todas y en mucho, la programación del robot, evitándome así el calcular ángulos a ojo, lo cual es poco preciso y complejo. A pesar de esto, sin escribir ningún tipo de programa adicional, he logrado enviar datos por Bluetooth, con la hiperterminal que incluye el IDE de Arduino y creo que esto ya se puede considerar como comunicación entre el PC y el robot, ya que se puede manejar el robot, aunque de una manera muy ruda y tediosa, aprovechando el código escrito para la comunicación con dispositivos móviles.

+ Software de periféricos móviles:

Con este título, que puede suponer confuso, me refiero a la creación de la aplicación para el sistema operativo de dispositivos móviles Android. La aplicación está creada con AppInvetor, el compilador de Google al cual ya había hecho alusión en un apartado anterior del trabajo.

La forma de programar aplicaciones que AppInventor nos ofrece, es bastante peculiar, ya que en vez de escribir un código fuente que después un programa interpreta y compila, se distribuyen las acciones en bloques, que se van montando como si de un puzle se tratara, de tal manera, que en vez de programar desde cero, se programa en base a unas funciones ya elaboradas que facilitan en mucho el trabajo.



APP INVENTOR 1

Valoración del resultado final:

Como resultado final, se ha obtenido un robot humanoide que realiza diversas funciones y puede ser controlado desde varios dispositivos, tales como un mando, un teléfono móvil o un ordenador. A demás, también se ha conseguido orientar el proyecto hacia un ámbito más educativo, con la realización de una aplicación en PHP, que permite generar el código del robot simplemente asignando una función (andar, saludar, etc…) a cada botón del mando, permitiendo así, que alguien que no sepa de programación, pueda programar el robot de una manera muy sencilla.

PRESUPUESTO

En este apartado se valorarán dos tipos de presupuestos, en primer lugar, el presupuesto del proyecto, que corresponde al coste en sí de la realización del robot, es decir, el coste de todas las placas utilizadas y todos los materiales, aunque no se acaben utilizando en el resultado final. Por otra parte, está el coste del robot acabado, que seria el precio de todas las piezas que componen el resultado final del robot.

Presupuesto del proyecto:

PRODUCTO	CANTIDAD	PRECIO	IMPORTE
PLACA FOTOSENSIBLE FIBRA DE VIDRIO	3	4,95	14,85
RESISTENCIAS	4	0,1	0,4
POLSADOR	3	0,5	1,5
SENSOR IR	1	4,8	4,8
CINTA DESOLDADORA	1	3,2	3,2
CONECTOR HEMBRA RECTO	4	1,15	4,6
CONECTOR MACHO RECTO	4	0,9	3,6
CONDENSADOR POLIESTER	15	0,08	1,2
REGULADOR TENSION 7805	15	0,7	10,5
ESTAÑO	1	3,9	3,9
ARDUINO	2	22	44
KIT PLACA IMAGINA	1	46	46
BLUETOOTH SERIAL 9600 BAUDIOS	1	17	17
SERVOMOTOR	10	5,3	53
BATERIA LI-PO	1	9	9
ALUMINIO 1M2	1	10	10
10 TORNILLOS SERVOS	3	2,5	7,5
10 TUERCAS TORNILLOS	3	2,5	7,5
CARGADOR LI-PO	1	15	15
Atmega 328	1	8	8
GASTOS DE ENVIO	1	9	9
TOTAL			274,55
TOTAL + IVA 16%			**318,478**

Presupuesto real del robot acabado:

PRODUCTO	CANTIDAD	PRECIO	IMPORTE
PLACA FOTOSENSIBLE FIBRA DE VIDRIO	1	4,95	4,95
RESISTENCIAS	2	0,1	0,2
POLSADOR	1	0,5	0,5
SENSOR IR	1	4,8	4,8
CONECTOR HEMBRA RECTO	1	1,15	1,15
CONECTOR MACHO RECTO	1	0,9	0,9
CONDENSADOR POLIESTER	10	0,08	0,8
REGULADOR TENSION 7805	10	0,7	7
ARDUINO	1	22	22
BLUETOOTH SERIAL 9600 BAUDIOS	1	17	17
MANDO A DISTANCIA	1	6	6
CABLE USB	1	1,95	1,95
SERVOMOTOR	10	5,3	53
BATERIA LI-PO	1	9	9
ALUMINIO 1M2	1	10	10
10 TORNILLOS SERVOS	3	2,5	7,5
10 TUERCAS TORNILLOS	3	2,5	7,5
CARGADOR LI-PO	1	15	15
GASTOS DE ENVIO	1	9	9
TOTAL			178,25
IVA 16%			28,52
TOTAL + IVA 16%			**206,77**

Como podemos observar, el precio del robot acabado es mucho menor que el del proyecto en sí, esto es lógico, ya que en un proyecto se realizan múltiples pruebas que al final no se incluyen en el resultado final.

CONCLUSIONES

Si analizamos los objetivos planteados al inicio del trabajo, podemos ver que se han cumplido de manera bastante exitosa, y si analizamos la función práctica del robot, podemos decir que su utilidad seria la de robot educativo, para que los iniciados en la programación, entiendan mejor como funciona un robot. No obstante, inicialmente estaba más orientado a que el robot tuviera unas cuantas funciones y a lo largo del trabajo, este objetico ha ido perdiendo fuerza y la han ganado otros que en un principio eran secundarios, como por ejemplo, la comunicación con un dispositivo móvil o la creación de una aplicación encargada de generar un código con solo definir unas funciones y pulsando un botón, sin tener que programar. De todas maneras, el resultado final, ha sido en mucho el deseado.

Como al principio del trabajo se define, este es un proyecto ampliamente expandible, ya que ofrece la posibilidad mejorar o incrementar la funcionalidad del robot, sin tener que desmontar lo que ya se ha hecho en un principio, a continuación, listaré una serie de opciones de mejora que se podrían haber interiorizado en el robot.

Posibilidades de expansión:

Creación de una interface para PC: Anteriormente ya se ha hablado de esta opción, no obstante, solo se ha hablado de el lenguaje Processing, aunque una aplicación de estas características, se podría haber realizado con cualquier lenguaje de programación, como por ejemplo, Visual Basic.

Creación de un circuito más compacto: Partiendo de la idea de que el circuito final es un extensión de Arduino, sobresale mucho del cuerpo del robot, esto sería evitable si se creara una placa independiente de Arduino, solo aprovechando el microcontrolador, que llevaría la misma programación. También se podría reducir el tamaño de la placa si se sustituyeran todos los reguladores de tensión por un regulador de tensión variable correctamente calibrado a 5V ya que este tipo de reguladores, ofrecen hasta 5A de intensidad, lo cual seria suficiente para alimentar los servos y reduciría en gran medida el amaño del hardware. Además de esta opción, también existe la alternativa de enviar la placa a una empresa especializada para que la realicen con instrumental adecuado y con componentes SMD, aunque esto encarecería el proyecto.

Más funciones motrices: A causa lo dicho antes sobre la orientación del proyecto, no ha dado tiempo a realizar demasiadas funciones motrices pero también hubiera sido bueno, que el robot

pudiera hacer más cosas, como por ejemplo, levantarse del suelo, pero como he dicho, a causa de ausencia de tiempo, no ha podido ser.

Control del robot por internet: Una idea que me surgió mientras realizaba la pagina web del proyecto, fue la de controlar el robot mediante internet, usando la pagina web como mando, no obstante, esto ya es posible, aunque no desde la web, una opción sería usando un programa de escritorio remoto.

Funciones fonéticas: Investigando por internet, se encuentran diversas librerías de Arduino que permiten componer palabras a base de fonemas, si hubiera profundizado más en este tema, se podría haber realizado un robot que hablara, no obstante, esta tecnología es dificultosa de utilizar, y podría ser un callejón sin salida a la hora de ponerla en practica.

Reconocimiento de voz: Para esta funcionalidad, que en principio parece muy dificultosa, no se necesita ningún tipo de hardware adicional, ya que desde el mismo teléfono móvil se podría realizar una aplicación que reconociese la voz y mandase datos al robot en base a lo que se haya pronunciado, permitiendo así, controlar el robot con la voz. AppInvetor, incluye una opción que haría posible esto sin demasiado esfuerza adicional.

Finalmente, cabe decir, que para este proyecto se han utilizado las ultimas tecnologías que el mercado ofrece, ya que Arduino, Android, HTML5, CSS3 y alguna otra tecnología usada en este trabajo, son lo mas nuevo (o casi) en cuanto a tecnologías se refiere. Además, a nivel personal, esto me ha permitido aprender mucho mas de lo que ya sabia y me ha iniciado en el apasionante mundo de la robótica, lo cual me ha permitido entender mucho mejor el funcionamiento de los aparatos electrónicos y de los lenguajes de programación.

Una última observación que veo oportuna; Al principio del proyecto se decide usar una placa Arduino Duemilanove, que es de la que se disponía en un principio, pero a lo largo de la realización del proyecto, dicha placa se estropeó y al comprar una nueva, se adquirió en vez de esta, una palca Arduino UNO, que es la misma placa en esencia y en forma, solo que con alguna modificación en el Bootloader, simplemente es una versión actualizada de la Duemilanove.

BIBLIOGRAFIA

USUARIO OSCAR. *Android + Processing + Bluetooth* [En línea]. Versión desconocida. Año de publicación 2011 <http://webdelcire.com/wordpress/archives/1045>[Fecha de la consulta: 10/11/2011]

MARIO SACCO. *Bluetooth + Android + PIC + LED = "Hola Mundo"* [En línea]. Versión desconocida. Año de publicación 2011 <http://www.neoteo.com/bluetooth-android-pic-led-hola-mundo>[Fecha de la consulta: 10/11/2011]

KEN SHIRRIFF'S. *A Multi-Protocol Infrared Remote Library for the Arduino* [En línea]. Versión desconocida. Año de publicación 2009 <http://www.arcfn.com/2009/08/multi-protocol-infrared-remote-library.html >[Fecha de la consulta: 1/7/2011]

AUTOR DESCONOCIDO. *Realización de PCB por insolado* [En línea]. Versión desconocida. Año de publicación 2005 <http://www.micropic.es/index2.php?option=com_content&do_pdf=1&id=1>[Facha de la consulta: 1/7/2011]

AUTOR DESCONOCIDO. Make remote controls and listeners [En línea]. Versión desconocida. Año de publicación 2011 <http://www.ladyada.net/learn/sensors/ir.html>[Fecha de la consulta: 1/7/2011]

VARIOS AUTORES. *ARDUINO ESPAÑA* [En línea]. Edición Española Año de publicación 2011 <http://arduino.cc/es/>[Fecha de la consulta: 5/10/2011]

VARIOS AUTORES. *ARDUINO* [En línea]. Edición internacional. Año de publicación 2011 <http://arduino.cc/en/>[Fecha de la consulta: 5/10/2011]

EQUIPO GOOGLE. *Google* [En línea]. Edición Española. <http://www.google.com>[Fecha de la consulta: 17/11/2011]

EQUIPO GOOGLE. *AppInventor* [En línea]. Versión Beta. Lloc de publicació: Año de publicación 2011 <http://www.appinventorbeta.com>[Fecha de la consulta: 17/11/2011]

EMPRESA CADSOFT. *Eagle* [En línea]. <http://www.cadsoftusa.com/>[Fecha de la consulta: 12/10/2011]

VARIOS AUTORES. *Wikipedia* [En línea]. Varios años de publicación <http://es.wikipedia.org>[Fecha de la consulta: 17/11/2011]

GLOSSARIO

Hardware: El *hardware* está compuesto por los mecanismos, el cableado y en general la parte física de un ordenador o cualquier útil eléctrico o mecánico.

Software: Se podría denominar como la parte adversa al *hardware* es decir, es la parte no tangible de un ordenador, comúnmente es denominado como programas de un ordenador.

Librería (o biblioteca): En términos informáticos, es un conjunto de subprogramas utilizados para desarrollar *software*. Normalmente sirven para simplificar notablemente la cantidad de código escrito por un desarrollador y su complejidad. (Según Wikipedia)

Entorno de programación (IDE): También llamado entorno de desarrollo integrado, es un programa informático compuesto por un conjunto de herramientas de programación (compilador, depurador, corrector de código, GUI (Graphical User Interface) etc…) que permiten, o al menos facilitan la creación de otros programas o aplicaciones.

Compilador: A grandes rasgos, un compilador es un programa, que "traduce" a código binario un programa escrito en algún lenguaje de programación entendible por los humanos, se usa para amenizar la escritura de código a los desarrolladores, y evitarles así tener que andar haciendo complicadísimos cálculos en código binario.

Código binario: También conocido como código máquina, es el "lenguaje", por decirlo de alguna manera, que entienden las máquinas, dicho lenguaje, se compone exclusivamente de unos y ceros, y no es entendible por un humano, al menos de manera sencilla e intuitiva.

Firmware: Un *firmware* es un bloque de instrucciones grabadas en una memoria de tipo no volátil (ROM, EEPROM, Flash, etc…) para llevar a cabo funciones especificas como por ejemplo, encender un simple LED o mover un motor. (Según Wikipedia)

Bootloader: Un *bootloader* es un gestor de arranque, es decir, un programa que llama a otro programa, generalmente es un programa sumamente sencillo que prepara todo lo necesario para que otro programa, normalmente más complejo, pueda funcionar.

Hiperterminal: Una hiperterminal es un programa que se encarga de monitorizar e interactuar con el puerto serie de un ordenador, de tal manera que muestra los datos que se están recibiendo por uno u otro puerto y además, permite mandar datos a este puerto.

COM: También llamado puerto serie, es una interface de intercambio de datos digitales por el puerto paralelo del ordenador, donde la información es enviada bit a bit, enviando solo n bit cada vez.

SIGLAS

GUI: Graphical User Interface o traducido, Interfaz Grafica de Usuario, es un programa que actúa como interface de cara al usuario, muestra imágenes y objetos gráficos que camuflan las ordenes de un ordenador, es decir, sin la GUI una acción tan simple como copiar un archivo, se convertiría en una instrucción en la que se definiría la ruta de origen, la acción a realizar (en este caso copiar) y la ruta de destino. Por su puesto, esto es aplicable a todos los dispositivos con interface grafica (ordenadores, móviles, etc…)

PIC: Peripheral Interface Controller o traducido controlador de interfaz periférico, es un tipo de controlador electrónico de la casa Microchip, seria similar al ATmega de Arduino.

JVM: Java Virtual Machine o Máquina Virtual de Java, a grandes rasgos, una JVM es un programa escrito específicamente para cada plataforma que lo ejecuta y que traduce las acciones de un programa escrito en lenguaje Java. De esta manera, el programador puede despreocuparse y escribir su aplicación íntegramente en Java y sin tener que hacer modificación alguna, esta aplicación funcionará en cualquier sistema operativo que soporte una JVM.

BASIC: Beginner's All-purpose Symbolic Instruction Code que traducido viene a ser algo como Código de Instrucciones Simbólicas Orientado a Principiantes, es un lenguaje de programación de alto nivel que fue diseñado en 1964 con la finalidad de facilitar la programación a estudiantes que no fueran de ciencias. Actualmente se sigue usando, sobretodo con fines educativos, pero también para proyectos complejos.

Encapsulado SMD: Surface Mount Device o Dispositivo de Montaje Superficial, deriva de SMT (Surface Mount Technology), un componente SMD es simplemente el equivalente a cualquier

componente electrónico, solo que muchísimo más pequeño, generalmente, se usan en productos fabricados en serie, ya que soldarlos a mano es muy dificultoso.

Máquina CNC: O máquina de Control Numérico por Computador, es un sistema de automatización de máquinas herramienta que son operadas mediante comandos programados en un medio de almacenamiento, en vez de ser manejadas manualmente. Las máquinas CNC son maquinaria industrial y tienen un precio muy costoso, pero en este caso, he tenido la suerte de que me han dejado usar una para realizar el trabajo.

AGRADECIMIENTOS

Agradezco la colaboración de mi familia, por apoyarme tanto económicamente como moralmente, sin los cuales, este proyecto no hubiera sido el mismo, también agradezco la tutoría tanto de María Reyner como de Jaume Romero, por orientarme en este trabajo y por sacar horas de su tiempo libre para acompañarme al concurso de robótica Robolot, dónde presenté mi proyecto y esto me permitió ganar el segundo premio del trabajo tecnológico, premio que se invirtió en parte de la financiación del robot. Finalmente, agradezco profundamente al foro Arduino la aceptación y la ayuda que me prestaron para resolver ciertos problemas y seria un craso error, excluir de los agradecimientos a la empresa SHINE de Vidreres, que no solo me dejó utilizar toda la maquinaria de control numérico, sino que también me proporcionó los materiales necesarios para el robot sin tener que pagar un céntimo.

Aprovecho también para agradecer a Robolot, no solo el premio, sino también el hecho de organizar una convención, a mi modo de ver, muy beneficiosa para el aprendizaje de la robótica, ya que allí conocí a varias personas, que me dieron su opinión sobre el proyecto y eso me ayudó a darle nuevos enfoques.

Gracias a todos, ellos y a todo el que haya contribuido de alguna manera a la realización del proyecto.

ANEXO Nº 1

+ Pagina web:

Para la creación de la página web, se ha querido aprovechar lo último en tecnología web, esto es HTML 5 como lenguaje de maquetación, CSS3 como lenguaje de estilos y PHP para interaccionar con el servidor, la página web se compone de diversas partes, el contenido y la maquetación web, los estilos y la programación.

Contenido y maquetación: La maquetación es la parte desarrollada en HTML5, esta parte es la encargada de decirle al navegador web, las partes que va a tener la pagina y como van a estar distribuidas, el contenido de la pagina web son los textos, imágenes y demás, de los que esta compuesta la web, y se podría decir que también forman parte del maquetado.

Estilos: Los estilos, son los encargados de dar un aspecto más visual e intuitivo a la pagina web, mediante estos, se indica al navegador, los colores y la situación de cada elemento definido en la maquetación, sirven sobre todo para hacer la pagina mas entendible, ya que sin ellos, seria todo simple texto. En este caso, se han creado con CSS3.

Programación: La programación, en este caso, ha servido para realizar el generador de código, se trata de un simple programa en PHP que en base a los datos obtenidos por un formulario que rellenará el usuario, generará un código para programar el robot, esto tiene la finalidad de que el usuario que no sepa programar, pueda incorporar las funciones del robot en el orden que quiera.

Para realizar la pagina web, primero se procede a maquetar y decidir que partes va a tener, normalmente, estas partes son, la cabecera, el logo de la pagina web, un espacio para el contenido y un pie de página para las especificaciones, el copyright, el contacto, etc…

HTLM es un lenguaje que se compone de etiquetas, y cada etiqueta corresponde a un “objeto” de la pagina web, así por ejemplo, la etiqueta <ul> corresponde a una lista no ordenada, o la etiqueta <li> que corresponde a un elemento de esta lista. Casi siempre, una etiqueta designa el inicio y el final de un objeto, para verlo mejor, voy a poner un ejemplo:

```
<ul>
        <li>Elemento 1</li>
        <li> Elemento 1</li>
</ul>
```

Esto es una lista no ordenada, iniciada por la etiqueta <ul> y finalizada por la etiqueta, la cual contiene 2 elementos cada uno de ellos iniciados por <li> y finalizados por .

Esta pequeña muestra de HTML es una minucia, pero ya expone bastante bien lo que generalmente es el lenguaje, que es bastante jerárquico y ordenado, generalmente siempre tiene una estructura base y de ella deriva todo lo demás.

Una vez maquetada la pagina web, es necesario darle un aspecto visual un poco vistoso, esto se consigue mediante CSS3. Es lógico que esta parte se haga después de maquetar, porque lo que hace CSS3 es dar unas propiedades a cada objeto, con lo cual, los objetos tienen que existir o estaríamos haciendo una página web a ciegas.

CSS3 se limita a dar propiedades de la siguiente forma:

```
li{
text-color: white;
}
```

Como se puede apreciar, aquí se está definiendo, que todos los elementos que se encuentren entre etiquetas <li> tendrán el color de texto blanco. Por lo general, CSS3 es bastante ordenado y entendible.

Una vez maquetada y estilizada a tu gusto la pagina web, se procede a añadir el contenido, y en este punto es donde se descubre si se han realizado bien los trabajos previos, ya que si está bien maquetada y estilizada, una página web se debería adaptar sola a cualquier contenido para el que ha sido preparada, y no solo eso, también a cualquier resolución de pantalla, que suele ser esto lo que más dolores de cabeza da.

A veces, no basta con una simple web de contenido, y sobre todo en estos momentos, que estamos en pleno apogeo de la web 2.0 o web de contenido dinámico, el encargado de dar este dinamismo, es PHP en mi caso, aunque también existen otros lenguajes como AJAX, java script, etc…

PHP es un lenguaje de servidor, esto quiere decir, que el programa se ejecuta en un servidor y se envía el resultado al PC del usuario, esto tiene múltiples ventajas y desventajas que no me voy a poner a analizar, lo que si diré es que una de las desventajas es que necesitas tener disponible un servidor que ejecute PHP para probar el código. Generalmente configurar un servidor puede llevar varias semanas, pero existen paquetes preconfigurados que pueden servirnos almenos para desarrollar el proyecto en nuestro propio ordenador sin tener que recurrir un servidor externo. Para esta página web, yo he usado XAMPP, que personalmente, considero que es el mejor que he probado. XAMPP viene configurado con Apache, el servidor web, PHP, el encargado de ejecutar los scripts de contenido dinámico y MySQL las bases de datos, en nuestro caso, solo usaremos Apache y PHP y yo solo explicaré cómo funciona PHP ya que Apache solo es relevante si sirven servicios a los ordenadores de los usuarios que visiten la web, pero en mi caso, como lo uso para desarrollar el proyecto que después subiré a un servidor externo, no me interesa demasiado.

El código PHP converge perfectamente con HTML, de tal manera que este mismo, se inicia y se termina con una etiqueta <?php…..?>. De esta manera, podemos elegir cuando se muestra o no una parte del contenido. En este caso no voy a poner ningún ejemplo, ya que el código escrito para esta aplicación, es muy extenso, no obstante su funcionamiento es sencillo.

El código a generar del robot, solo tiene una parte que cambia, es decir, el código siempre es igual, excepto en una parte en la que se tiene que elegir entre poner diversas funciones (saludar();,andar();, etc…) de esta manera, lo que hace PHP es leer los datos que se piden en el formulario e inserir los fragmentos de código en base a esta lectura, en el orden adecuado.

Una vez acabada la pagina web, se procede a colgarla en internet, esto lo he decidido hacer en un servidor gratuito que usé en su día para otra página, y aunque muy a pesar mío, este tenga publicidad, va bastante bien.

La dirección de la página será la siguiente: http://treballsintesis.webcindario.com (es recomendable abrirla con Google Chrome, ya que al ser HTML5 un lenguaje tan nuevo, no todos los navegadores lo aceptan)

ANEXO Nº 2

+ Creación de PCB's:

La creación de PCB's no es muy compleja, por lo general se hacen a una sola capa y el procedimiento es siempre igual.

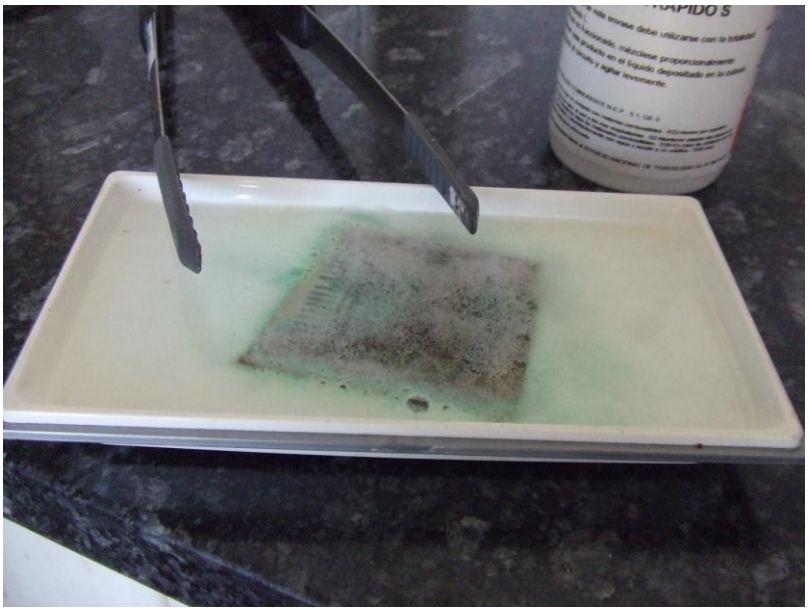
En este caso, he decidido que la mejor opción es hacerlo por insolado, esto consiste en unos pasos que hay que seguir con bastante minucia para que la placa salga bien.

Crear el negativo del circuito: Para crear el negativo del circuito, como ya he dicho en algún momento, se usará un programa llamado Eagle, este generará una plantilla del circuito, la cual se imprimirá en un papel transparente, de tal manera que la tinta negra solo tapará las partes que vayan recubiertas de cobre del circuito.

Insolar el circuito: Una vez seca la tinta del papel transparencias, se procede al insolado, se coge un trozo de placa fotosensible a la que no le haya dado la luz, esta placa no es más que una placa de fibra de vidrio recubierta con una fina capa de cobre, que a su vez está recubierta de una capa sensible al sol, es decir, reacciona a la luz ultravioleta. En mi caso he insolado el circuito con una inmoladora, esto no es más que una caja que dispone de un cristal para sujetar el circuito y que debajo de este, emite luz ultravioleta. Para insolar, basta con pegar el negativo imprimido al trozo que hayamos cortado de la placa fotosensible, i una vez pegados, situarlos encima del cristal con la parte fotosensible mirando hacia la luz, de esta manera, la luz pasará por la parte del papel transparencias que no tenga tinta, revelando así solo la parte que no lleva cobre del circuito.

Revelar el circuito: El proceso de insolado dura unos 8 minutos, después de este, aunque miremos muy atentamente la superficie de la placa, no veremos ningún resultado, esto es a causa de que falta todavía el revelado, que consiste en extraer mediante un proceso químico, el material fotosensible que ha sido insolado, esto se tiene que hacer a oscuras, usando agua y sosa caustica, en una medida ajustada. Una vez introducida la placa en la mezcla, empieza a aparecer el dibujo de nuestro circuito y cuando lo veamos bien definido, estará listo para salir de la mezcla.

Atacar el circuito: Con el proceso anterior, hemos logrado dejar al descubierto la parte no deseada de cobre y con el proceso de atacado, lograremos quitar esta parte y dejar intacta la parte cubierta todavía por material fotosensible. Esto se consigue con un acido, obtenido de mezclar agua oxigenada de 110 volúmenes con “salfumant” y agua, las proporciones son, 3 partes de agua, una parte de agua oxigenada y una parte de “salfumant”. Esta mezcla corresponde a la del atacador rápido, que es el que a mí personalmente me gusta más. Una vez hecha la mezcla, se introduce en ella el circuito y se espera hasta que desaparezca completamente todo el cobre no deseado, una vez desaparece, se extrae la placa y se lava con agua.



PROCESO DE ATACADO

Limpiar las pistas: Después del proceso de atacado, la parte de cobre que nos interesa, queda cubierta todavía por el material fotosensible, mientras la parte que no nos interesaba, ha desaparecido por completo. Después de esto, se tienen que limpiar las pistas de la placa con alcohol, para después poder realizar una correcta soldadura.

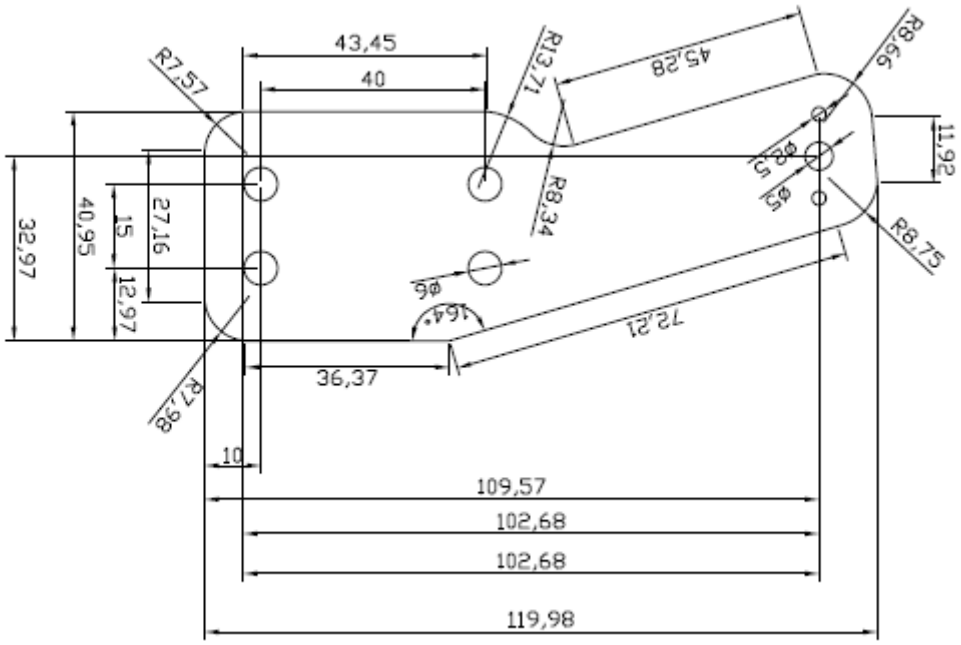
Hacer los agujeros: Una vez tenemos el circuito acabado, se hacen los agujeros para los componentes, estos generalmente se hacen con una broca de 1 mm, que deja un espacio para soldar bastante aceptable.

Soldar los componentes: Cuando tenemos el circuito con los agujeros hechos, se introducen los componentes electrónicos de manera correcta y si lo hemos hecho todo bien, ya tenemos nuestro circuito acabado.

ANEXO Nº 3

+ Corte de piezas en una máquina CNC:

En un apartado anterior, ya he explicado este proceso, pero me parece interesante añadirlo como anexo, ya que así se podrán añadir fotos y poner ejemplos de código Heidenhain, que desde mi punto de vista, es lo más interesante, a demás de explicar mejor el funcionamiento de una máquina de control numérico.



En primer lugar, como ya he dicho antes, se realizaran todas las cotas desde un puto 0 y si es posible, que este punto marque un eje de simetría en la pieza acotada.

Después de realizar esto, se pasaran las cotas al operario de la máquina, para que este traduzca los planos al lenguaje entendible por la máquina. Este lenguaje define muchos aspectos de cada pieza, de tal manera, que no se tiene en cuenta solo la forma de la misma, sino que también se tienen en cuenta otros parámetros como la velocidad de avance de la máquina, el radio de la fresa que se va a utilizar, la velocidad de giro, estas últimas cosas se definen a partir del tipo de material a moldear. Para hacer más cómoda y rápida la realización de las piezas, la máquina CNC tiene un “cargador” de herramientas, que van cambiando en base a si se tiene que

agujerear, fresar, o avellanar. El cambio de estas herramientas, también está indicado en el lenguaje Heidenhain.

No voy a mostrar ningún código entero, ya que son muy extensos y ocuparían mucho espacio, no obstante, se pueden descargar de la pagina web.

Esto es una pequeña muestra de uno de los códigos.

```
0  BEGIN PGM tobillo MM

1  BLK FORM 0.1 Z X-70 Y+0 Z-10

2  BLK FORM 0.2 X+70 Y+60 Z+0

3  TOOL DEF 1

4  TOOL CALL 1 Z S1200 F15000

5  TOOL DEF 8
```

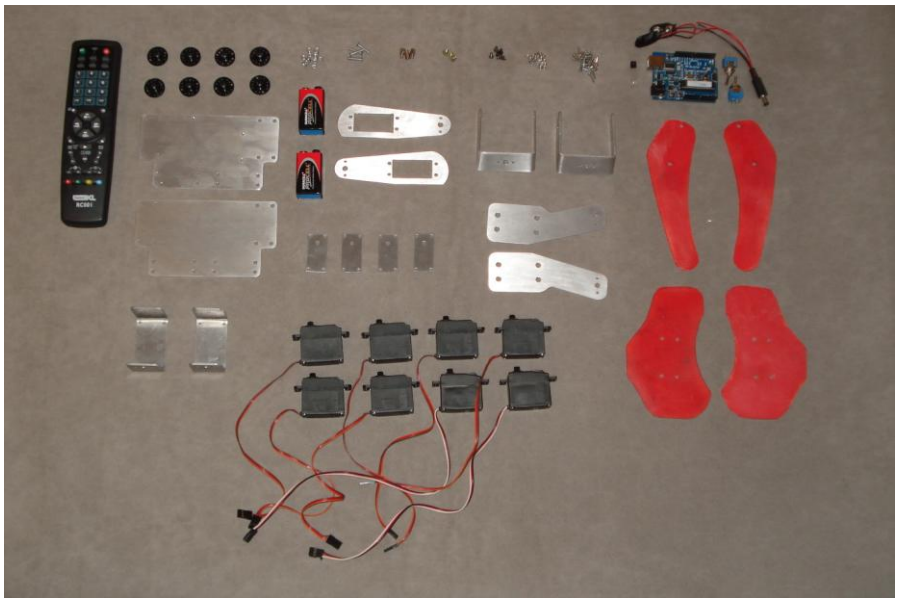
Aquí se hacen las llamadas a las herramientas que van a ser utilizadas y se definen los márgenes de la pieza, de todas maneras, yo no tengo ninguna noción de este lenguaje, por lo tanto tampoco sabría comentar convenientemente todo el código, como bien he dicho antes, de esto se encarga un operario especializado.

Después de generar todos los códigos, estos se introducen en la máquina, y con pulsar un solo botón, esta realiza la pieza a la perfección, aunque en alguna pieza, tuvimos que modificar el código por que había algún fallo, pero esto no tuvo demasiada importancia.

Otro aspecto a destacar, tiene que ver con lo que al principio comentaba de las cotas desde un punto 0 que marca un eje de simetría en la pieza, ya que para piezas exactamente simétricas, se genera solamente el código de la mitad de la pieza y después se le indica a la máquina que haga una simetría, de esta manera, las piezas salen perfectas y se ahorra mucha parte en código.

PANEL DE LA MÁQUINA CNC

Una vez realizadas todas las piezas, tuvimos que doblar algunas de ellas, ya que el diseño del robot lo requería, normalmente, este proceso se hace con una dobladora de piezas industrial, pero por recomendaciones del operario, lo hicimos manualmente, ya que estas dobladoras están pensadas para piezas extremadamente grandes y seria una traba doblar piezas tan pequeñas, ya que seguramente, no tendrían una sujeción estable Y el doblez quedaría desviado. Para doblarlas, usamos simplemente una maza de plástico y un soporte acolchado para no rayar la pieza y l final, el resultado fue del todo satisfactorio.



PIEZAS DEL ROBOT

Por último, cabe destacar, que este tipo de máquinas están muy bien para hacer un solo ejemplar del robot, pero en vista de hacer una producción en serie, son mucho mas rápidas y precisas las máquinas que funcionan con laser, ya que hacen todas las piezas de una vez y son más adecuadas para piezas pequeñas, ya que no ejercen una fuerza de tracción al cortarlas. A demás, si se usa una laser para cortar las piezas, no sería necesario hacer retoques como el lijado de ángulos interiores o el pulir las rebabas, ya que el corte es mucho más limpio.