



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


Escuela Técnica Superior de Ingeniería del Diseño



TRABAJO FIN DE GRADO:

DISEÑO DE LA APP Y EL CONTROL DE UNA
CÁMARA VIDEO CON ARDUINO.

Alumna: Émili Prieto Antón

Tutor: Roberto Capilla Lladró

Cotutor: José Antonio Gómez Tejedor

Índice

1	Introducción al proyecto.	4
2	Características y explicación de cada componente.	5
2.1	Arduino UNO.	5
2.2.1	Introducción ¿Por qué Arduino?	5
2.2.2	Características Arduino UNO	6
2.3	APP Inventor 2	7
2.4	Bluetooth HC-05/HC-06	7
2.5	Servomotores	8
2.5.1	Características servomotores	8
2.6	Cámara IP	9
3	Montaje del prototipo.	10
3.1	Esquema eléctrico.	12
4	Código.	13
4.1	El código en Arduino.	13
4.2	Código en App Inventor 2.	14
5	Presupuesto.	16
6	Realización del proyecto.	17
7	Alternativas para realizar el proyecto.	21
8	Posibles aplicaciones.	22
9	Conclusión.	22
10	Bibliografía.	23

1 Introducción al proyecto.

Debido al gran avance del desarrollo tecnológico, cada vez es más sencillo poder obtener al alcance de nuestras manos dispositivos y programas que faciliten realizar acciones de la vida cotidiana sin necesidad de esfuerzo (consultas bancarias, tiempo del transporte público, etc.), a su vez, también permite desarrollar aplicaciones más complejas con fines dedicados a proyectos de investigación (monitoreo de datos, control remoto, etc.). Esto hace que cada vez, sea más importante el desarrollo de aplicaciones para dispositivos móviles y por tanto cada vez más, es más importante tener conocimiento de programas y lenguajes de programación dedicados a ello.

De esta manera, este proyecto pretende realizar una aplicación móvil, para sistemas operativos Android, que permita obtener el control de un brazo robot construido con dos servos, el cual se llevará a efecto mediante el sensor acelerómetro del que dispone el propio dispositivo usando para la transmisión de la información la tecnología Bluetooth. A su vez, en la misma aplicación se podrá observar la retransmisión de la imagen en tiempo real de una cámara situada al final del brazo robot. De esta manera se podrán tener dos funciones simultáneas en la aplicación como son; el control de la dirección de la cámara y la visualización de la imagen sin interactuar físicamente.

Para ello se contará con un microcontrolador Arduino UNO, que servirá para poder hacer el control de los servomotores. Dicho control se realizará mediante un Shield Bluetooth HC-05 de Arduino, el cual se encargará de recibir la información del dispositivo móvil. La programación de la aplicación móvil se ha desarrollado a través de la aplicación web que dispone Android, llamada APP Inventor, donde se gestionarán ambas partes del proyecto.

En cuanto a la retransmisión de la imagen, se hará uso de una cámara IP, que permite realizar una conexión mediante WiFi, ya que el Bluetooth no tiene la capacidad suficiente para poder procesar información pesada como son las imágenes en tiempo real. Además, se contará con una Raspberry Pi 3 para el procesado de las imágenes entre la aplicación móvil y la cámara IP.

2 Características y explicación de cada componente.

2.1 Arduino UNO.

2.2.1 Introducción ¿Por qué Arduino?

Arduino es una compañía de hardware libre, se trata de una plataforma de prototipos electrónicos de código abierto (llamado Open-Source) basada en hardware y software flexibles fáciles de usar. Esto se debe a que está pensado para cualquier interesado en crear o desarrollar objetos y entornos, tenga o no experiencia y conocimientos en el tema. Permite interactuar con el entorno mediante la entrada de sensores y realizar acciones sobre el mismo mediante otros dispositivos. También permite tener proyectos autónomos o proyectos con software en ejecución en un ordenador.

Todo ello permite tener placas de circuito impreso (como microcontroladores) a un bajo coste en el mercado además de otros componentes como placas de expansión (Shields) y sensores como el dispositivo Bluetooth que usaremos en este proyecto. En cuanto al software consiste en un entorno de desarrollo (IDE) basado en el entorno de Processing y lenguaje de programación basado en Wiring, todo ello apoyado por una comunidad muy amplia donde encontrar información y código para los posibles proyectos que puede desarrollar este tipo de dispositivos. Permite una gran variedad de librerías y facilidades para de obtener información a la hora de programar incluso para personas que empiezan de cero.

Aunque existen gran variedad de microcontroladores, Arduino ofrece ciertas ventajas sobre el resto:

- Es barato en comparación con el resto de plataformas, su versión más barata puede ser ensamblada a mano e incluso las prefabricadas no superan los 50€.
- Es multiplataforma ya que se puede ejecutar en la mayoría de los sistemas operativos (véase Linux, Macintosh, etc.) mientras que el resto se limitan a Windows.
- El entorno de programación es simple para el uso de principiantes y flexible para usuarios más avanzados, mientras que el resto de plataformas usa un lenguaje de programación complejo y de difícil uso si no se está familiarizado con él.
- El código abierto y su hardware extensible permite a diseñadores experimentados mejorar y extender los módulos pues se encuentran publicados bajo licencia Creative Commons.

Por tanto, debido a su bajo coste en hardware, sus facilidades para programar, los pocos recursos que necesita el IDE, haciendo que lo pueda soportar cualquier sistema operativo y también la compatibilidad con el sistema Android para la programación de

aplicaciones móviles. Se ha optado por escoger este microcontrolador para el desarrollo del proyecto.

2.2.2 Características Arduino UNO

Tipo de placa: Placa computadora, microcontrolador de placa simple.

Microcontrolador: ATmega 328P

EEPROM: 1 KB (ATmega 328P)

Memoria flash: 32 Kb (ATmega 328P) de los cuales 0.5 Kb son usados para el bootloader

Memoria SRAM: 2 Kb (ATmega 328P)

Reloj: 16 MHz

Voltaje medio: 5 V

Voltaje de entrada recomendado: 7-12 V

Voltaje de entrada límite: 6-20 V

E/S Digitales: 14 (6 de ellas permiten salida PWM)

Entradas analógicas: 6

Intensidad de corriente continua por los pines de E/S: 20 mA

Intensidad de corriente continua por el pin de 3.3 V: 50 mA

Tamaño de la placa: 68.6x53.4 mm

Peso: 25 g

Se ha optado por esta placa, debido a que tan solo se necesitan controlar dos servomotores y la conexión al Bluetooth.

La siguiente imagen muestra cómo es una placa Arduino UNO:



Ilustración 1.Arduino UNO

2.3 APP Inventor 2

App Inventor es un entorno de desarrollo de aplicaciones para dispositivos Android. Esta plataforma pertenece a Google Labs y depende directamente del MIT (Massachusetts Institute of Technology) que permite de forma visual y simple realizar aplicaciones para dispositivos móviles desde cualquier dispositivo Android e incluso desde un ordenador mediante su página web. Su simplicidad permite crear aplicaciones sin que el usuario necesite demasiados conocimientos de programación, ya que al ser visual, la programación va en bloques que se van uniendo como un puzzle para realizar el programa final. Además cuenta con una amplia comunidad y soporte para ayudar a realizar mejores aplicaciones. El inconveniente de App Inventor es esa misma simplicidad que lo limita, ya que no permite aplicaciones excesivamente complejas, pero a pesar de ello abarca infinidad de posibilidades.

2.4 Bluetooth HC-05/HC-06

Tanto el bluetooth HC-05 como el 06, son unas placas integradas realizadas por la misma compañía de Arduino con lo que se puede asegurar compatibilidad a la hora de configurarlo y programarlo. Se configura por puerto serie. En el caso del HC-06 el Bluetooth tan solo actúa en modo esclavo con lo que no necesita apenas configuración, en el caso del HC-05 al comportarse tanto en modo esclavo como maestro, hay que realizar normalmente una configuración previa.

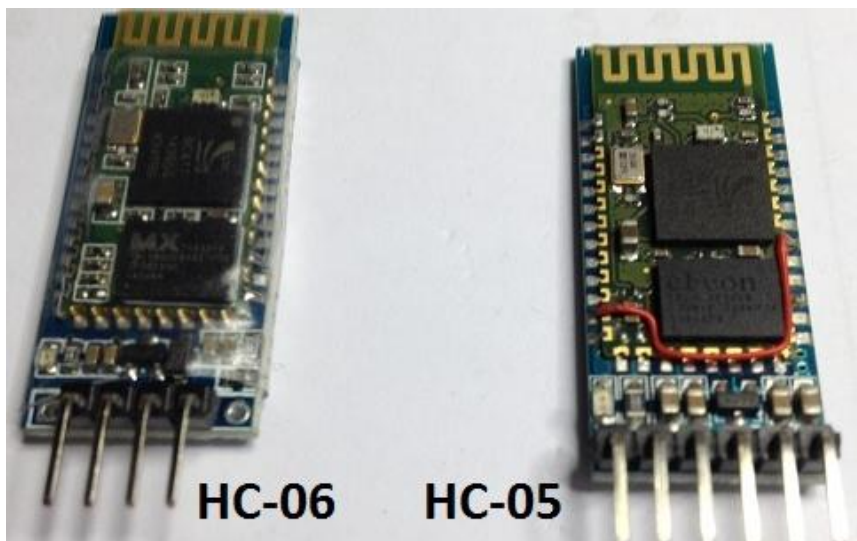


Ilustración 2. Shields Bluetooth

En este proyecto se planteaba la opción de transmitir tanto la imagen de la cámara como la información de los servos a través de Bluetooth. Viendo la incapacidad del Bluetooth para procesar de forma óptima tanta información, se optó por separar la imagen del movimiento. Operando de esta manera hubiera sido más sencillo usar el HC-06 debido a su simplicidad a la hora de trabajar con él, pero como no se disponía de uno, se usó definitivamente el modelo HC-05.

2.5 Servomotores

Un servomotor o más comúnmente llamado servo, es un dispositivo electromecánico parecido a un motor de corriente continua, que es capaz de posicionarse en un punto concreto de su rango de actuación y mantenerse estable en el mismo. Dependiendo de su finalidad, se puede controlar tanto su velocidad como su posición.

La característica principal con respecto a los motores de continua, es que posee más fuerza de par, menor inercia y mayor velocidad. Aunque se pueden modificar para obtener un motor de continua con estas mismas características pero a cambio de sacrificar el control. El control se hace mediante Modulación de ancho de Pulsos (PWM).

En este proyecto, se ha optado por utilizar micro-servos, más pequeños que los servos, normalmente usados para automodelismo. Son más baratos que los servos y tienen menos par, pero el suficiente para este proyecto, ya que la cámara no tiene un peso considerable y por las características de los microservos lo pueden mover aún añadiendo la torreta.

2.5.1 Características servomotores

- Longitud del cable: 15cm
- Velocidad: 0.14sec/60degree al (4.8V) ~0.12sec/60degree al (6V)
- Fuerza de Torque: 1.3kg.cm.at (4.8V) ~1.5kg.cm.at (6V)
- Tensión: 4.8V ~ 6V
- Compatible con Futaba, JR, Sanwa, Hitec, etc
- Dimensiones: 1,46 x 0,94 en en x 0,47 en
- Peso: 0.67 oz = 18.89 g



Ilustración 3. Micro servo

2.6 Cámara IP

Las cámaras IP son cámaras que utilizan el protocolo Ethernet y Wi-Fi para poder enviar la información, lo que permite obtener la información recibida desde cualquier lugar tanto dentro como fuera de la red local. Por ello su uso más común es el de la videovigilancia y la seguridad del hogar, ya que vienen con software incluido para facilitar la instalación y el control de la misma e incluso usando aplicaciones móviles propias para que el usuario pueda controlar la cámara desde cualquier lugar.

En este proyecto se ha utilizado una cámara D-Link DCS-930LB1. Con las siguientes características:

- Peso: 41g
- Dimensiones: 2.5x2.5x2.5cm
- Memoria RAM: 32 MB
- Zoom digital: 4X
- Resolución horizontal: 640 píxeles
- Resolución del sensor óptico: 0.3 Mp
- Resolución: Resolución de captura del video: 640x480
- Potencia eléctrica: 2 W
- Tipo de conector: Fast Ethernet
- Ángulo de visión real: 45.3 grados
- Otras características: Micrófono, WiFi, accesible desde iOS y Android.



Ilustración 4. Cámara IP.

3 Montaje del prototipo.

Pasaremos a realizar una breve explicación de cómo se ha realizado el montaje del proyecto.

Para el brazo robot se ha utilizado un Pan-Tilt, que consiste en 4 piezas de plástico de fácil montaje, que se adaptan a los micro-servos:



Ilustración 5. Pan-Tilt desmontado.

Que una vez montado queda:

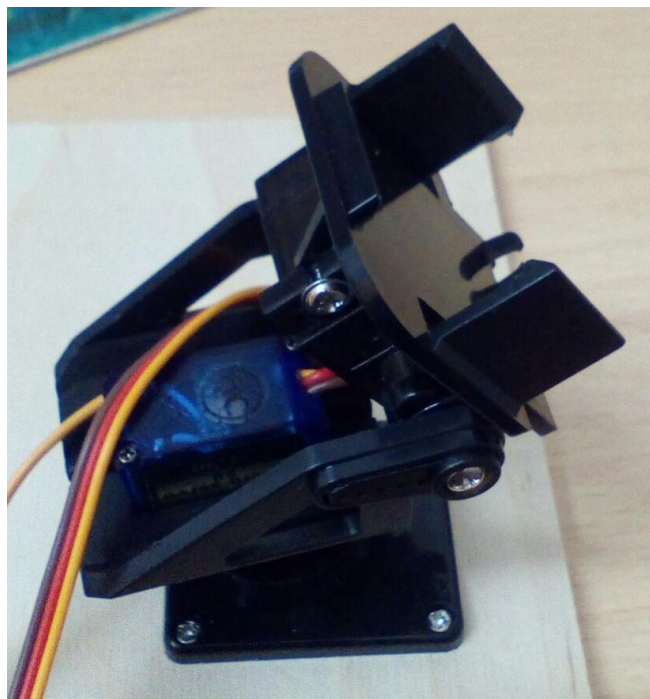


Ilustración 6. Pan-Tilt montado y sujeto a la base

Para sujetar el brazo robot y el resto de componentes, se ha optado por una base de madera simple y barata para poder permitir al prototipo tener algún tipo de sujeción.

Los demás componente irían también encima de la base de madera para proporcionar unidad al montaje.

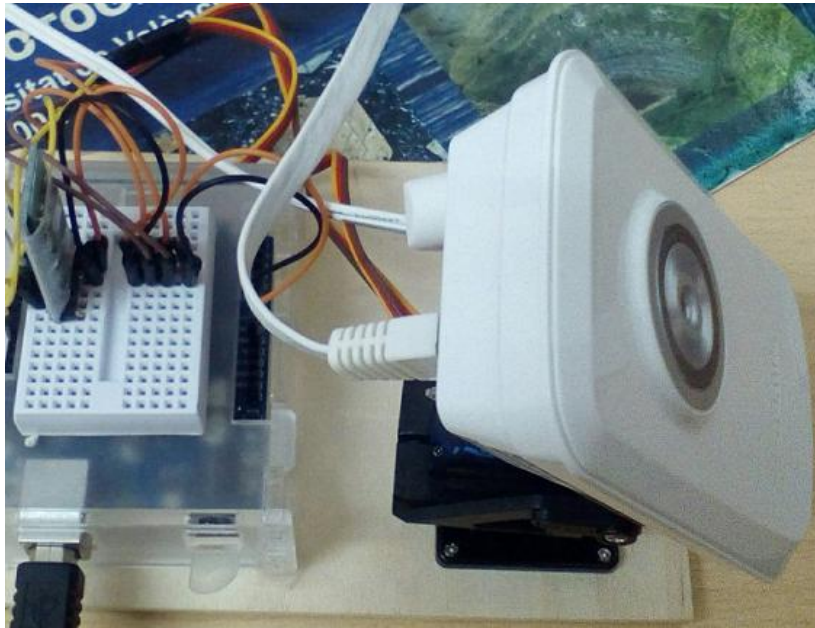


Ilustración 7. Prototipo del proyecto.

Como se puede observar, para la conexión de cables (tipo dupont macho-macho) se ha utilizado una pequeña placa de prototipos adherida a la tapa de la caja del Arduino, de esta manera se pueden realizar las conexiones de manera simple, sin necesidad de utilizar un cableado demasiado largo, quedando todo más compacto y evitando el tener que hacer uso de placas de PCB en las que hay que utilizar soldadura y que para hacer pruebas podría tener inconvenientes.

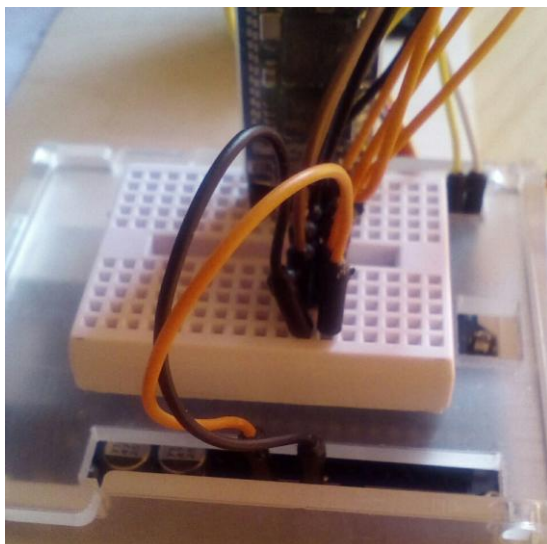


Ilustración 8. Board de conexión.

3.1 Esquema eléctrico.

Aquí se mostrará la conexión realizada entre el Arduino y los servos junto al Bluetooth. Por una parte tenemos el dispositivo Bluetooth, se puede ver que con cuatro conexiones ya es funcional. Este mismo esquema de conexiones se puede aplicar si en vez de un modelo HC-05 dispusiéramos de un modelo HC-06, ya que ambos se conectan de la misma manera para nuestro proyecto:

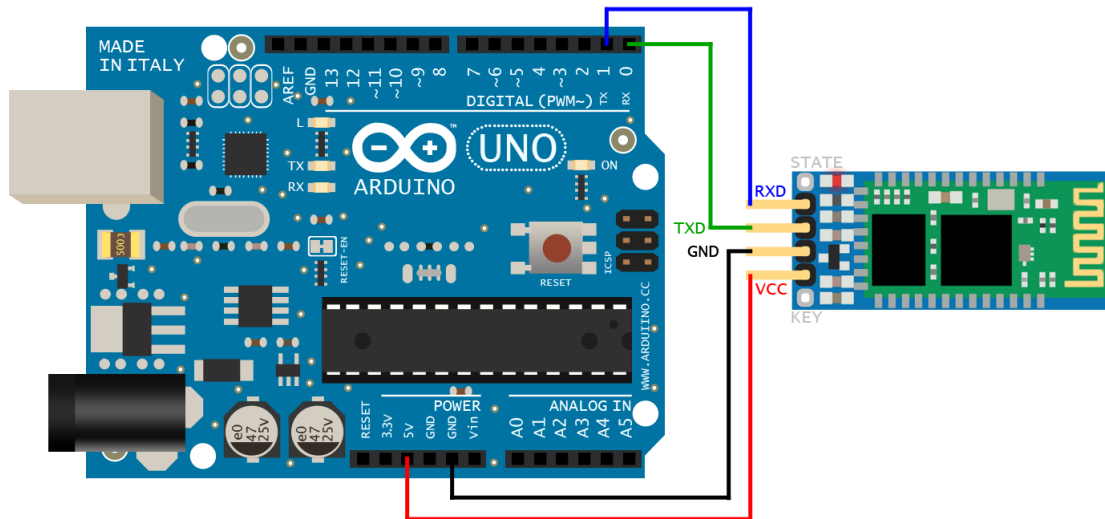


Ilustración 9. Conexión Bluetooth-Arduino.

Por otra parte los micro-servos los conectaríamos de la siguiente manera:

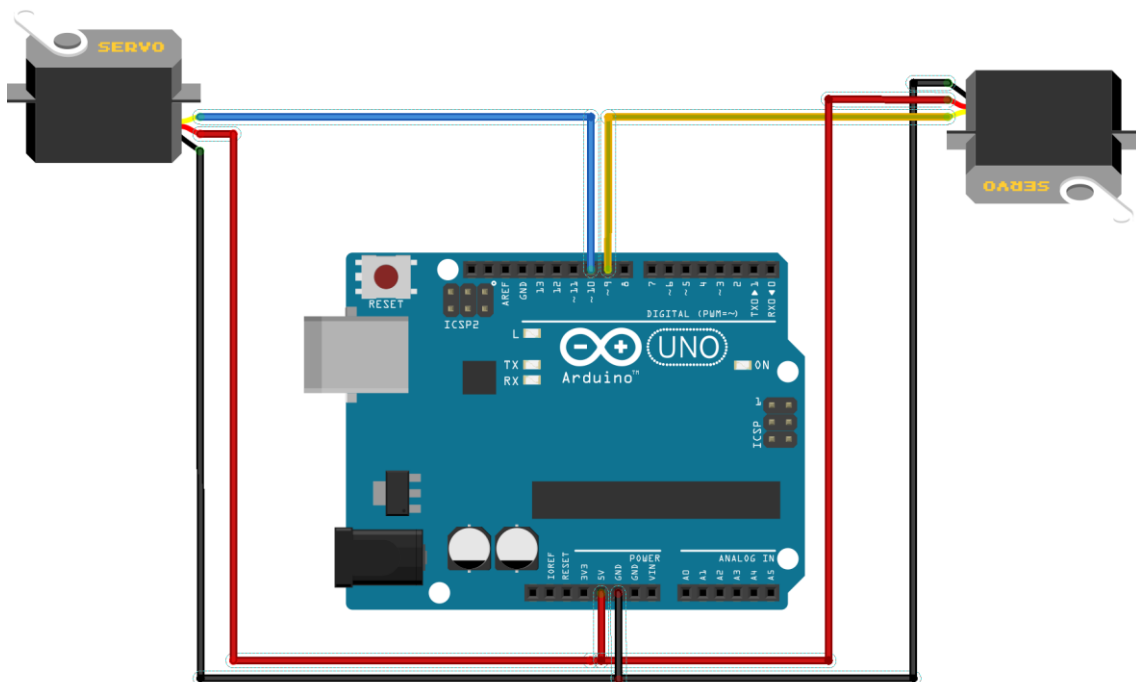


Ilustración 10. Conexión microservos-arduino.

La cámara IP iría a parte conectada a la red de tensión o bien con una batería externa.

4 Código.

En este apartado se explicará de manera breve, cómo se ha programado el Arduino y cómo se ha desarrollado la aplicación móvil con el App Inventor.

4.1 El código en Arduino.

```
#include <Servo.h> //se incluye la libreria para que se puedan utilizar los servos.

Servo servol;
Servo servo2; //se declaran los dos servos
int acelerometer; //se declara una variable para leer el acelerometro

void setup() //se inicializan el puerto serie para el bluetooth y los dos servos segun el pin en el que se conectaran
{
  Serial.begin(9600); //comunicacion (vel)
  Serial.setTimeout(10); //se establece un tiempo de pausa en segundos
  servol.attach(9);
  servo2.attach(10); // 9/10 pines de conexion en el arduino
}

void loop() //programa principal para la lectura del sensor y movimiento de servos
{
  if(Serial.available()>0) //se verifica la conexion al puerto serie
  {
    acelerometer=Serial.parseInt();

    if(accelerometer>100&&accelerometer<270)
    {
      digitalWrite(10, LOW); //con estas dos lineas nos aseguramos de mantener el otro servo parado.
      delay(1);
      acelerometer=map(accelerometer,100,270,0,180); //escala los valores que proporciona el acelerometro y los convierte en un rango de angulos
      servol.write(accelerometer);
    }
    if((accelerometer>300) && (accelerometer<480))
    {
      digitalWrite(9, LOW);
      delay(1);
      acelerometer=map(accelerometer,300,480,0,180);
      servo2.write(accelerometer);
    }
  }
}
```

Ilustración 11. Sketch Arduino.

Como se observa, el código en Arduino consiste en incluir la librería para poder trabajar con los servos y declarar e inicializarlos junto al puerto serie. Una vez hecho se realizan dos condiciones, en las que según el valor que se lea del sensor moveremos un servo o el otro. Para ello se ha procurado dar rangos de valores distintos, para evitar que se solape la información y se muevan los dos servos a la vez.

4.2 Código en App Inventor 2.

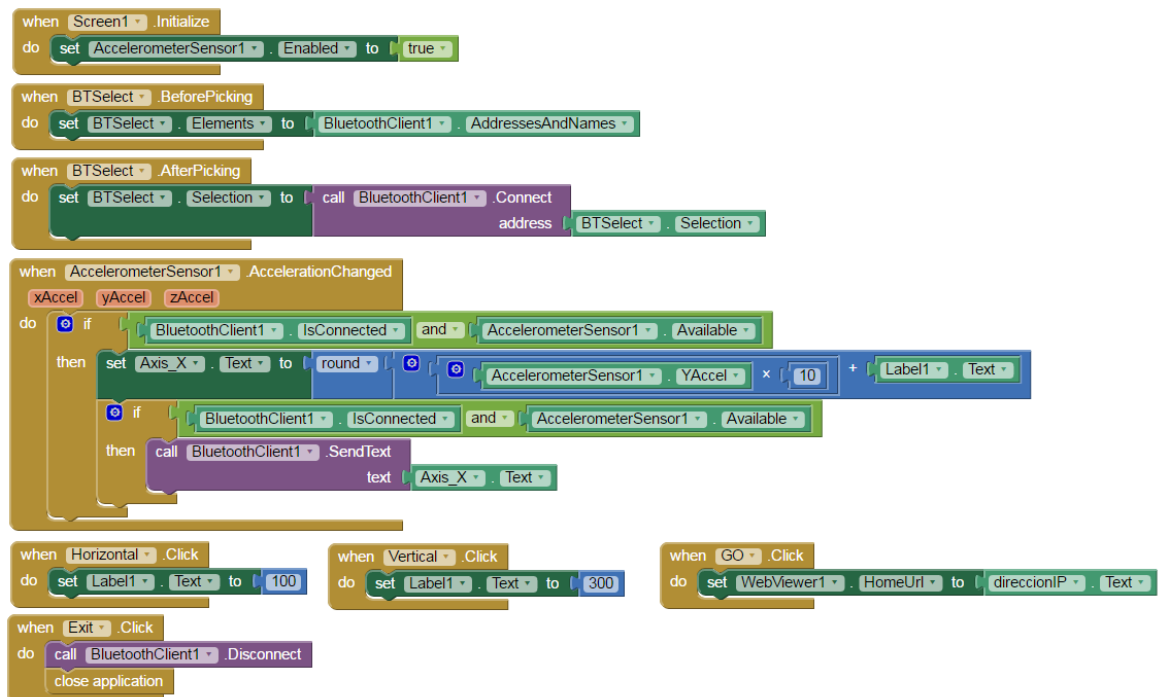


Ilustración 12. Diagrama de bloques en App Inventor 2

El primer bloque corresponde al habilitado del sensor del acelerómetro, mientras que los dos siguientes corresponden a la conexión y selección del Bluetooth. El cuarto bloque es donde se realiza el envío del valor del sensor multiplicado por diez y además sumándole el valor del “Label” correspondiente a los valores de seleccionar un botón y otro para cada servo por separado y además que tengan valores de rango distintos para poder diferenciarlos en el código del Arduino.

El botón GO inicializa la visualización en el “webViewer” (visor Web) de la dirección escrita en el texto correspondiente a la dirección IP.

Para finalizar se ha añadido un botón para poder salir de la aplicación, el último bloque “Exit”.

A continuación se añade la imagen del diseño de la aplicación y cómo se sitúan los diferentes elementos:

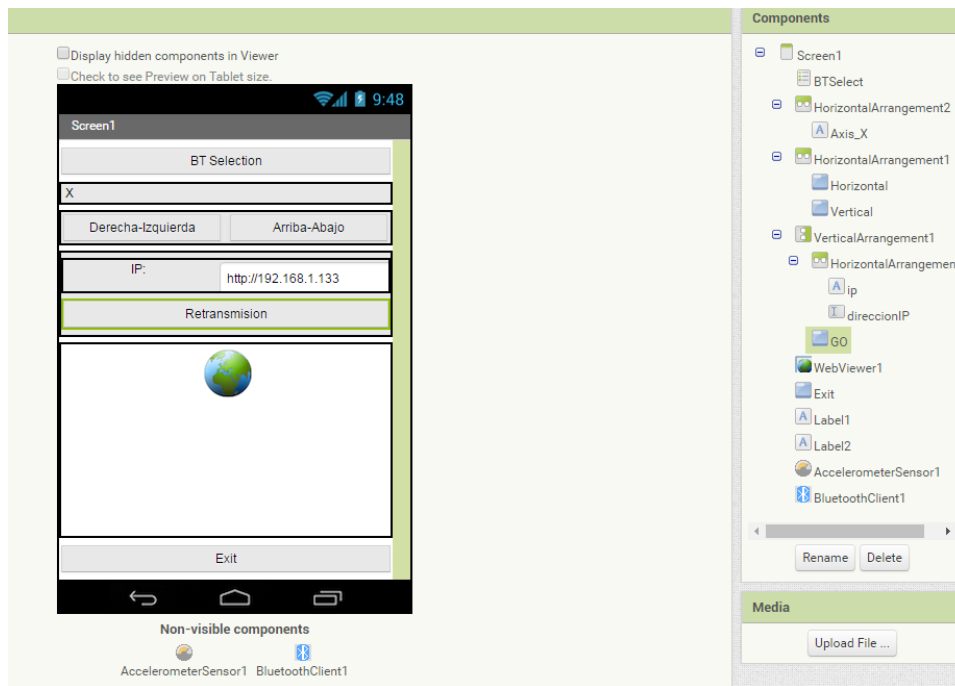


Ilustración 13. Diseño aplicación en App Inventor 2

Por último se muestra una captura de la aplicación realizada desde el móvil con el resultado final:



Ilustración 14. Aplicación móvil, vista final.

5 Presupuesto.

A continuación se detallan tres tablas con los costes que han sido necesarios para la elaboración de este proyecto, así como un presupuesto de los mismos.

COSTES HUMANO			
Concepto	Horas	Precio Hora (€/hora)	Precio final (€)
Mano de Obra (Ingeniero Junior)	450	12	5400.000
Concepto	km	Precio Kilometro (€/km)	Precio Final (€)
Desplazamiento	4.7	0.25	1.175
Total:			5401.175

COSTES VARIOS			
Concepto	Horas	Precio Hora (€/hora)	Precio final(€)
Desarrollo Software	120	12	1440
Total:			1440

COSTES MATERIALES			
Componente	Cantidad	Precio Unidad (€/ud.)	Precio final(€)
Servomotor	2	2.27	4.54
Bluetooth HC-05	1	2.7	2.70
Cámara IP	1	36	36.00
Cables	1	-	0.76
Arduino Uno	1	5.18	5.18
Raspberry Pi 3	1	41	41.00
Torreta Pan-Tilt	1	2.75	2.75
Tabla madera	1	3.60	3.60
Caja Arduino	1	4.30	4.30
Total:			96.53

COSTES TOTALES	
Concepto	Precio final(€)
Costes Humanos	5401.175
Costes Varios	1440.000
Costes Materiales	96.530
Total:	6937.705

Se puede ver que con la cámara IP sube más el presupuesto, pero en el caso de que la cámara no trabajara con complementos Java no haría falta la Raspberry, lo que abarataría el coste total del proyecto. Cabe destacar que la mayoría de los componentes se pueden encontrar más económicos con una buena búsqueda y disponiendo de tiempo para recibirlos. Con una buena búsqueda se puede llegar a ahorrar más de 20€. Además se pueden omitir algunos elementos reflejados en el presupuesto, como la caja acrílica del Arduino Uno, que no son estrictamente necesarios para el proyecto, ya que el diseño del mismo puede variar y estar todo en una misma carcasa, al aire, etc.

6 Realización del proyecto.

Para realizar el proyecto se comenzó por la parte de los micro-servos, que como se ha podido ver en el montaje, tan solo requiere de una pequeña programación y fácil conexión para que funcione. Se comenzó probando primero con un solo servo y una vez se comprobó su correcto funcionamiento se pasó a el control de los dos a la vez. En un principio dio problemas, ya que la comunicación no se realizaba correctamente y se movían los dos sin control, por ello como se ha comentado en el montaje, se optó por poner unos botones en la aplicación para evitar este problema y tener la comunicación separada. Aún así, en ocasiones por la falta de calibrado o las pequeñas tensiones que llegan a los servos, se detecta un leve movimiento del servo que debiera estar parado mientras el otro se está controlando.

Para la realización de la parte de la imagen, se optó por una cámara IP, por su fácil conexión a través de la red Wi-Fi.

Para ello, tan solo hay que configurarla, siendo muy sencillo, ya que todas las cámaras llevan consigo un software de instalación y configuración, con lo cual siguiendo las instrucciones del software de la cámara correspondiente, se puede acceder a ella y tenerla lista.

Una vez, instalada la cámara y configurada con nuestra red local, hay que asegurarse de abrir un puerto en el router para esa dirección IP. Cada modelo de router es distinto, por lo que habría que mirar la marca y seguir las instrucciones de ese modelo. Una vez abierto el puerto ya estaría listo todo para ser usada en la aplicación, pues tan solo hay que poner la dirección IP de nuestra cámara en el apartado de la aplicación que pone IP y darle al botón de Retransmisión para que nos lleve a la página de la visualización de imágenes de la cámara IP, y con eso ya tendríamos la aplicación terminada:



Error: Authorization Required

Authentication was requested

Ilustración 15. Error de autenticación.

Al realizar dicha operación, surgió el error de acceso que se ve en la ilustración. Después de buscar soluciones a dicho error, se averiguó que la cámara usada para el proyecto utilizaba complementos de Java y JavaScript y que el App Inventor 2 no tiene soporte para ellos, lo que hace que no podamos introducir el usuario y la contraseña de la cámara. Con lo cual se buscó una solución alternativa al problema.

Aquí es donde entra la Raspberry Pi comentada al comienzo de la memoria. Al tratarse de un mini-ordenador, puede ser utilizado como servidor web y FTP. Eso es una ventaja para el proyecto ya que la cámara dispone de un servicio para guardar las imágenes en un servidor FTP. Esto nos proporciona la opción de utilizar esas imágenes en bucle, en el servidor Web de la Raspberry utilizando una página en HTML y PHP, con lo cual ya no tendríamos que usar los complementos Java ni tener que introducir usuario ni contraseña, ya que se va directamente a las imágenes procesadas. Además, al ser sistema operativo en Linux, se consiguen los servidores de forma sencilla y libre uso.

Para la configuración de la Raspberry Pi, se ha utilizado el sistema operativo Raspbian, que se obtiene directamente de la misma página de raspberry:

<https://www.raspberrypi.org/downloads/raspbian/>

Una vez instalado el sistema operativo siguiendo las instrucciones de la misma página, tan solo hay que actualizarlo con los comandos en la terminal: `sudo apt-get update` y `sudo apt-get upgrade`.

Una vez actualizado, se instalan los servidores FTP y Web, para ello hay que utilizar los comandos: `sudo apt-get install vsftpd` para el servidor FTP y `sudo apt-get install apache2` para el servidor web. En la bibliografía se encuentran las páginas con las instrucciones a seguir para la correcta configuración de ambas.

Cuando ya se ha configurado la Raspberry, configuramos el envío de imágenes de la cámara a la Raspberry. Cada cámara tendrá su página de configuración, en este caso lo haríamos de la siguiente manera:

DCS-930LB //		VÍDEO EN DIRECTO	CONFIGURACIÓN	MANTENIMIENTO	ESTADO	AYUDA
Asistente	FTP					Sugerencias útiles..
Configuración de red	En esta sección, puede configurar un servidor FTP para recibir las imágenes.					Nombre de host: esta es la dirección IP del servidor FTP al que se va a conectar.
Configuración inalámbrica	<input type="button" value="Guardar parámetros"/> <input type="button" value="No guardar parámetros"/>					Puerto: el puerto predeterminado es el 21.
DNS dinámico	SERVIDOR FTP					Nombre de usuario: el nombre de usuario que ha solicitado acceder al servidor FTP externo.
Configuración de la imagen	Nombre de host	192.168.1.133				contraseña: la contraseña del servidor FTP externo.
Vídeo	Puerto	21 (El valor predeterminado es 21)				Modo pasivo: activando el modo pasivo se permitirá el acceso a un servidor FTP externo si la cámara está detrás de un router protegido por un cortafuegos.
Audio	Nombre de usuario	pi				Activar la carga de imágenes en el servidor FTP: Al marcar esta casilla se permitirá que la cámara cargue imágenes en el servidor FTP mostrado anteriormente. Seleccione Siempre para permitir que la cámara cargue siempre las imágenes en el servidor FTP. Seleccione Programar si desea especificar la fecha y la hora en que la cámara inicia y detiene la carga de las imágenes en el servidor FTP. Seleccione Detección de movimiento/sonido si desea que la cámara cargue imágenes en el servidor FTP únicamente cuando se detecta movimiento o sonido.
Detección de movimiento	contraseña	••••••••				Frecuencia de imagen: el usuario puede elegir y definir las cifras para las imágenes/segundo y los segundos/imagen .
Detección de sonido	Ruta	/var/www/html/				Nombre de archivo básico: el nombre que contiene la información de
Correo	Modo Pasivo	<input checked="" type="radio"/> Sí <input type="radio"/> No				
FTP	PROGRAMA DE TIEMPO					
Fecha y hora	<input checked="" type="checkbox"/> Activar la carga de imágenes en el servidor FTP					
cerrar de sesión	<input checked="" type="radio"/> Siempre <input type="radio"/> Programar Día <input type="checkbox"/> Lun <input type="checkbox"/> Mar <input type="checkbox"/> Mié <input type="checkbox"/> Jue <input type="checkbox"/> Vie <input type="checkbox"/> Sáb <input type="checkbox"/> Dom Periodo de tiempo Iniciar : 00:00:00 (Ejemplo: 06:30:00) Detener : 00:00:00 (Ejemplo: 22:30:00) <input type="radio"/> Detección de movimiento/sonido					
	Frecuencia de imagen	<input checked="" type="radio"/> 1 Imágenes/segundo				
		<input type="radio"/> 1 Segundos/imagen				
	Nombre de archivo básico	pic				
	Archivo	<input checked="" type="radio"/> Sobrescribir <input type="radio"/> Sufijo de fecha/hora <input type="checkbox"/> Crear subcarpeta por 0.5 Hora <input type="radio"/> Sufijo de número de secuencia hasta 1024				
	PROBAR SERVIDOR FTP					
	Se enviará un archivo JPEG al servidor FTP anterior para realizar las pruebas. (Nombre de archivo: test_date_time.jpg) <input type="button" value="Probar"/>					
	<input type="button" value="Guardar parámetros"/> <input type="button" value="No guardar parámetros"/>					

Ilustración 16. Configuración envío imágenes a servidor FTP.

Como se ve, es muy sencillo, tan solo configurar los parámetros de nuestro servidor FTP con los datos del servidor de la Raspberry y la carpeta donde guardar las imágenes, el nombre que se les va a dar y configurar el tiempo de envío.

Una vez se tiene la configuración realizada, creamos un pequeño archivo en la carpeta del servidor web llamado index.php para poder capturar las imágenes de la siguiente manera:

```
2 <?php
3
4
5 $file1=fopen("/home/pi/var/www/html/pic.jpg","r");
6 $file2="/var/www/html/";
7 $dir= chdir($file2);
8 while(true){
9
10     exec("sudo cp /home/pi/var/www/html/pic.jpg /var/www/html/");
11 }
12 ?>
13
14 <head>
15 <script>
16 var i = 0;
17 var images = new Array("/var/www/html/pic.jpg");
18 function cambia(){
19     if (i <= images.length - 1) {
20         document.galeria.src = images[i];
21         i++;
22     }else{
23         i = 0;
24     }
25 }
26 setInterval(cambia,1000);
27 </script>
28 </head>
29 <body>
30 <img src="" name="galeria" />
31 </body>
32 </html>
```

Ilustración 17. Código en el servidor Web.

De esta manera para conectar en la aplicación y ver las imágenes, tan solo habría que poner la dirección IP de la Raspberry, en vez de la de la cámara obteniendo como resultado la imagen de la Ilustración 13 situada en el apartado 4.2 de la memoria del proyecto.

De esta manera, el proyecto estaría finalizado.

Nota: Aprovechando el uso de la Raspberry, también se comprobó de realizar la aplicación con la propia Raspicam de la misma compañía, comprobando que se obtenía una calidad de imagen inferior a la obtenida con la cámara IP, utilizando esta última para el montaje final del proyecto.



Ilustración 18. Prototipo Final

7 Alternativas para realizar el proyecto.

Debido a la investigación llevada a cabo para realizar el proyecto, se han encontrado diversas maneras para poder realizarlo. Aunque para la elaboración de este proyecto se ha optado por usar Raspberry PI 3 y cámara IP. Se hará una breve explicación de las alternativas encontradas que podrían ser posibles a continuación:

Cámara IP:

Es una manera muy sencilla de obtener nuestro objetivo, ya que poniendo en la aplicación la dirección IP de nuestra cámara se consigue obtener la imagen. Pero hay que tener en cuenta que en general las cámaras IP son caras, lo que supondría un aumento en el presupuesto final considerable. También hay cámaras baratas, pero hay que tener en cuenta la compatibilidad con la aplicación, ya que por ejemplo, en este caso se probó con una cámara D-Link DCS-930LB1, la cual lleva complementos Java Script que no son soportados por la aplicación del APP Inventor 2.

WebCam y Raspberry Pi:

Otra manera simple de conseguir el objetivo ya que tanto la Raspberry, como la webcam se pueden conseguir a bajo coste, tan solo hay que instalar un programa en la Raspberry y configurarlo para poder ver el video en nuestra página web de la aplicación. El inconveniente, no todas las webcam tienen soporte en Linux, como por ejemplo la WebCam Trust WB-1400T usada para las pruebas es detectada por la Raspberry, pero en cambio no tiene soporte, para ello existe una lista en internet proporcionada por Raspberry y Linux que indica qué cámaras tienen soporte y funcionan, cuáles tan solo soporte, cuáles no funcionan y cuáles no se han probado, con lo que es conveniente que si no se tiene una webcam y se va a comprar, se revise esa lista previamente para saber cuál es compatible y cumple las necesidades del usuario.

Otros programas:

Para la realización de la aplicación se podrían haber usado otros programas como por ejemplo Android Studio, el cual está diseñado específicamente para hacer aplicaciones móviles en dispositivos Android, o bien otros parecidos como el NeatBeans que son entornos de desarrollo de programación libres que cada vez más van incorporando IDEs para poder realizar aplicaciones móviles ya que es una tendencia cada vez mayor. Las ventajas de utilizar este tipo de entornos es que proporcionan más posibilidades y alternativas a la hora de desarrollar, ya que existen muchas librerías en los propios entornos o bien el propio usuario puede crearlas. El inconveniente es que se necesita de un mayor conocimiento en programación, sobretodo en Java, lo que hace que sean entornos muy útiles y potentes, pero requiere tener un conocimiento mucho mayor previo.

Para el montaje:

Al igual que se podrían haber escogido diversas cámaras y opciones para visualizar la imagen, para hacer el brazo robot incluso la base del prototipo, también se podría haber utilizado la opción de hacer un diseño para imprimirlo con impresora 3D, en este caso, se ha descartado esta opción debido a que requiere del conocimiento y uso de programas para diseñar los planos en 3D y sobretodo tener a disposición una impresora 3D, que aunque es una

tecnología en auge y cada vez más asequible, no todo el mundo dispone de una y aunque hay lugares donde puedes ir a imprimir o incluso hacer un pedido online, no siempre sale rentable y aumentaría el coste del proyecto.

Giróscopo:

Para realizar el movimiento de los servos, también se podría haber elegido utilizar el sensor giróscopo que incorporan los móviles, ya que su uso es más eficiente a la hora de realizar los movimientos del servo. El inconveniente, es que no todos disponen de un sensor de este tipo, lo que hace que la aplicación no funcionara, en cambio el sensor acelerómetro lo llevan todos incorporado.

8 Posibles aplicaciones.

Este proyecto puede tener diversas aplicaciones, entre ellas y las más comunes son la videovigilancia casera por su simplicidad y bajo coste, además de las posibilidades que tiene de programación y diseño, pudiendo personalizar la aplicación al gusto y necesidades del usuario.

Otra aplicación muy común, es la de proporcionar visión a un robot, ya sea dirigido o brazo robótico, puesto que se puede adaptar el diseño y posicionarlo sobre la superficie y de esta manera, a la vez que se mueve el robot, se puede mover la cámara y permitir la visibilidad completa del usuario.

9 Conclusión.

Este proyecto pretende demostrar que es posible realizar una aplicación móvil que, aun sin tener conocimientos específicos de uso y control de cámaras instaladas en brazo robótico, permita acercar esta tecnología a todo el mundo. Se trata de una aplicación que podría tener varios usos prácticos. Gracias a los avances de hoy en día, que abaratan costes de producción y precio final del producto, se pueden realizar proyectos como este por una cantidad mínima de recursos y tiempo. Además de que se han desarrollado diferentes conocimientos en cuanto a configuraciones de redes, sistemas operativos y programación en lenguaje de desarrollo web.

10 Bibliografía.

A continuación se proporcionan algunas de las páginas utilizadas en la búsqueda de información para realizar el proyecto.

Información general:

- ◆ Información sobre Arduino (Manual de usuario):

http://electroship.com/documentos/Arduino_user_manual_es.pdf

- ◆ Página oficial de Arduino:

<https://www.arduino.cc/en/Main/ArduinoBoardUno>

- ◆ Información sobre los servos:

<https://es.wikipedia.org/wiki/Servomotor>

- ◆ Información sobre Bluetooth:

http://hetpro-store.com/TUTORIALES/bluetooth_hc-06_app_arduino/

APP Inventor 2:

- ◆ Información sobre App Inventor:

https://es.wikipedia.org/wiki/App_Inventor

- ◆ Tutoriales App Inventor:

<https://www.youtube.com/watch?v=2wPJd7nrqxo>

<http://www.instructables.com/id/Android-accelero-control-arduino-servo-via-bluetoo/>

<https://www.youtube.com/watch?v=sbnHFV2xdCw>

<https://petestechprojects.wordpress.com/2015/01/07/real-time-dlink-wireless-camera-view-using-appinventor2/>

RaspiCam:

- ◆ Tutoriales RaspiCam:

<http://www.devlopez.com/es/camara-remota-super-simple-en-raspeberry-con-modulo-camera-pi-o-noir/>

<http://www.maquinasvirtuales.eu/raspberry-pi-pi-noir-como-servidor-de-videovigilancia-casero/>

<http://rsppi.blogspot.com.es/2013/08/streaming-de-video-desde-raspberrypi.html>

PHP/HTML:

- ◆ Tutoriales PHP/HTML:

<http://bakia.co/blogtecnologia/45/obtener-imagenes-de-una-carpeta-dinamicamente-por-php/>

<http://www.mediavida.com/foro/dev/imagenes-en-bucle-html-500374>

Cámara IP:

- ◆ Datasheet cámara IP:

http://www.dlink.com/es/es/-/media/consumer_products/dcs/dcs-930l/datasheet/dcs_930l_b1_datasheet_en_eu.pdf

- ◆ Manuales de Usuario:

http://www.dlink.com/es/es/-/media/consumer_products/dcs/dcs-930l/manual/dcs_930l_a1_manual_v1_20_es.pdf