

# Robot AIO

## MANUAL DE INSTALACIÓN DEL ROBOT AIO

*Diciembre del 2016*

**David Stiven Ávila González**  
**Miguel Ángel Bermeo Ayerbe**  
**Fabián Andrés Merchán Jiménez**

---

Correo electrónico [aio.javeriana@gmail.com](mailto:aio.javeriana@gmail.com)

Repositorio

<https://github.com/AIO-Javeriana>

Página oficial

<http://pegasus.javeriana.edu.co/~CIS1630IN03/>

---

## Tabla de contenido

Módulo AudioVisual	1
Visio general del Modulo	1
Dependencias	1
Librerías	1
Ejecutar Modulo	2
Servicios Ofrecidos	3
Módulo de Movilidad	6
Visio general del Modulo	6
Dependencias	6
Librerías	6
Ejecutar Modulo	6
Servicios Ofrecidos	8
Módulo de Manipulación	9
Visio general del Modulo	9
Dependencias	9
Librerías	9
Ejecutar Modulo	9
Servicios Ofrecidos	11

---

## Módulo AudioVisual

---

### Visio general del Modulo

El módulo audiovisual de AIO controla todos los recursos de visuales y auditivos del robot. Está desarrollado en JavaScript, y los recursos visuales se controlan a través de CSS y HTML. El código fuente del proyecto se encuentra publicado en [GitHub](#).

### Dependencias

- **Software:**
  - [NodeJS](#)
  - [Chromium browser](#)<sup>1</sup>
- **Hardware:**
  - Parlante
  - Micrófono

### Librerías

Las dependencias externas incluidas en el repositorio del código, se listan a continuación:

- **AudioOutput Module**
    - Reproducir Sonidos: [Web Audio API](#)
    - Text to speech: [Web Speech API](#)
  - **AudioInput Module**
    - Speech to text: [Web Speech API](#)
  - **Video Module**
    - Reproducir videos: [YouTube Player API](#)
  - **Visual Module**
    - Renderizar SVG: [SVG Morpheus](#)
- 

<sup>1</sup> Google Chrome Aunque el proyecto está diseñado y probado para ser utilizado en Chromium browser (navegador soportado por dispositivos con procesador ARM), todas sus opciones son compatibles con el navegador Google Chrome

- Cuadros de diálogo: [Bootstrap alerts](#)
- **Canal de comunicación:**
  - WebSockets: [Socket.io](#)
- **Otros**
  - Mensajes en pantalla: [Bootstrap](#)
  - Botones del menú: [material-floating-button](#)
  - Manejo del DOM: [jQuery](#)
  - Iconos: [Ionicons](#)
  - Comandos de voz: [Annyang](#)

## Ejecutar Modulo

Primero se debe ejecutar el servidor web que expone la lógica del módulo en el navegador estando en la terminal ubicados en la carpeta fuente se ejecuta el siguiente comando:

```
$ node server.js
```

Luego abrir el chromium en la siguiente dirección web:

```
http://localhost:8080/
```

Cabe aclarar que el módulo AudioVisual funciona únicamente cuando está conectado al actor, quien le dice que acción debe realizar y debe estar activo antes de cargar la página en el navegador.

## Servicios Ofrecidos

El módulo ofrece los siguientes servicios:

<b>Nombre:</b>	DECIR
<b>Descripción:</b>	Emite un texto como sonido.
<b>Parámetros</b>	
<b>Nombre:</b>	TEXTO
<b>Descripción:</b>	Texto que se va a emitir como sonido.
<b>Nombre:</b>	TONO
<b>Descripción:</b>	Tono de la voz
<b>Ejemplo en el Script</b>	
<b>DECIR:</b> <b>TEXTO:</b> oh no! voy tarde al colegio <b>TONO:</b> HABLAR	

<b>Nombre:</b>	REPRODUCIR-SONIDO
<b>Descripción:</b>	<p>Reproduce un sonido.</p> <p>Estos sonidos están almacenados en la carpeta:</p> <p style="text-align: center;"><i>/public/assets/sounds/</i></p> <p>y se declaran en el archivo utils.js dentro en la ruta:</p> <p style="text-align: center;"><i>/public/static/js/api/utils.js</i></p> <p>de la siguiente forma:</p> <pre><b>var</b> availableSounds = {   <b>path:</b> './assets/sounds/',   <b>availableSoundFiles:</b> [     { <b>name:</b> 'name', //Id of the sound file       <b>file:</b> 'file.mp3' //Name of the file     }   ] };</pre> <p>Este servicio, usa el Web Audio API, para mayor información acerca de archivos de audio soportados por favor vea la <a href="#">documentación de la Web Audio API</a></p>
<b>Parámetros</b>	
<b>Nombre:</b>	SONIDO
<b>Descripción:</b>	Nombre del sonido que se va a reproducir.
<b>Ejemplo en el Script</b>	
<b>REPRODUCIR-SONIDO:</b> <b>SONIDO:</b> colegio	

<b>Nombre:</b>	RESPONDER
<b>Descripción:</b>	Activa el micrófono para que el usuario pueda realizar una pregunta al usuario.
<b>Parámetros</b>	
<b>Ejemplo en el Script</b>	
<b>RESPONDER :</b>	

---

## Módulo de Movilidad

---

### Visio general del Modulo

El módulo de movilidad de AIO, se encarga de trasladarse en el escenario, por medio de un sistema mecánico físico. La lógica de control se desarrolló en C++11 usando el framework de AIO para este lenguaje. El código fuente del proyecto se encuentra publicado en [GitHub](#).

### Dependencias

- **Software:**
  - [Scoket.io-client-cpp](#)
  - [LibBoost](#)
  - [WiringPi](#)
- **Hardware:**
  - Odroid-C2
  - Módulo de movilidad AIO
  - 4 Pilas AA

### Librerías

Las dependencias externas incluidas en el repositorio del código, se listan a continuación:

- **Gestión de periféricos**
  - Perifericos: [WiingPi](#)
- **Canal de comunicación:**
  - WebSockets: [Scoket.io-client-cpp](#)

### Ejecutar Modulo

Para ejecutar el modulo tener instalada la librería de [Wiring](#) especial para la ODROID-C2, y la herramienta CMAKE para compilar el proyecto. Teniendo esto en cuenta, primero debemos instalar las dependencias con el siguiente comando:

```
$ sudo apt-get install cmake libblkid-dev e2fslibs-dev libboost-all-dev libaudit-dev
```

Luego se ejecutar los siguientes comandos:

```
$ cmake
```



```
$ make  
$ ./mobility_module
```

El modulo está configurado para conectarse con el módulo de procesamiento como local host y el puerto 9090. Si desea cambiarlo al ejecutarlo agregando la propiedad -h seguido la dirección del módulo procesamiento y -p seguido del puerto. Por ejemplo:

```
$ ./mobility_module -h 192.168.0.7 -p 8182
```

## Servicios Ofrecidos

El módulo ofrece los siguientes servicios:

<b>Nombre:</b>	MOVE
<b>Descripción:</b>	El modulo se mueve siguiente la línea de acuerdo a las propiedades.
<b>Parámetros</b>	
<b>Nombre:</b>	DIRECTION
<b>Descripción:</b>	Este parámetro indica, si está en una intersección debe moverse a la izquierda(TURN-LEFT) o a la derecha(TURN-RIGHT) o seguir derecho (NONE).
<b>Ejemplo en el Script</b>	
<b>MOVE:</b> <b>DIRECTION: TRUN-RIGHT</b>	

---

## Módulo de Manipulación

---

### Visio general del Modulo

El módulo de manipulación de AIO, se encarga de expresar emociones por medio de una pinza. De este depende de un sistema intermediario encargado de comunicarse con el modulo por medio de un puerto serial y con el módulo de procesamiento con WebSockets esta lógica está desarrollada en C++11 usando el framework de AIO para este lenguaje. El código fuente del proyecto se encuentra publicado en [GitHub](#).

### Dependencias

- **Software:**
  - [Scoket.io-client-cpp](#)
  - [LibBoost](#)
  - [WiringPi](#)
- **Hardware:**
  - Odroid-C2
  - Módulo de manipulación AIO

### Librerías

Las dependencias externas incluidas en el repositorio del código, se listan a continuación:

- **Gestión de periféricos**
  - Perifericos: [WiingPi](#)
- **Canal de comunicación:**
  - WebSockets: [Scoket.io-client-cpp](#)

### Ejecutar Modulo

Para ejecutar el modulo tener instalada la librería de [Wiring](#) especial para la ODROID-C2, y la herramienta CMAKE para compilar el proyecto. Teniendo esto en cuenta, primero debemos instalar las dependencias con el siguiente comando:

```
$ sudo apt-get install cmake libblkid-dev e2fslibs-dev libboost-all-dev libaudit-dev
```

Luego se ejecutar los siguientes comandos:

```
$ cmake  
$ make  
$ ./handling_module
```

El modulo está configurado para conectarse con el módulo de procesamiento como local host y el puerto 9090. Si desea cambiarlo al ejecutarlo agregando la propiedad -h seguido la dirección del módulo procesamiento y -p seguido del puerto. Por ejemplo:

```
$ ./handling_module -h 192.168.0.7 -p 8182
```

## Servicios Ofrecidos

El módulo ofrece los siguientes servicios:

<b>Nombre:</b>	SHUT-GRIP
<b>Descripción:</b>	Indicar que la pinza debe cerrarse.
<b>Parámetros</b>	
<b>Ejemplo en el Script</b>	
<b>SHUT-GRIP:</b>	

<b>Nombre:</b>	OPEN-GRIP
<b>Descripción:</b>	Indicar que la pinza debe abrirse.
<b>Parámetros</b>	
<b>Ejemplo en el Script</b>	
<b>OPEN-GRIP:</b>	



<b>Nombre:</b>	CALIBRATING-SENDOR
<b>Descripción:</b>	Este servicio es para calibrar los sensores de línea, inicialmente el modulo al activarse se calibra. Pero si es necesario se puede indicar a este que se calibre de nuevo.
<b>Parámetros</b>	
<b>Nombre:</b>	MODE
<b>Descripción:</b>	<p>Tiene dos modos de calibración:</p> <ul style="list-style-type: none"> <li>• <b>Automática(AUTOMATIC):</b> que el modulo simplemente se mueve y se calibra.</li> <li>• <b>Manual(MANUAL):</b> donde requiere de un usuario quien mueva el modulo sobre la línea.</li> </ul>
<b>Ejemplo en el Script</b>	
<b>CALIBRATING-SENDOR:</b> <b>MODE: MANUAL</b>	



