

# Introducción a los Proyectos

Si el lector ha saltado hasta este apartado sin siquiera hojear la parte introductoria de este manual se entenderá que el lector ya tiene un conocimiento medio sobre el uso de la Raspberry y tiene herramientas suficientes para llevar a cabo cada uno de los proyectos aquí presentes. Si no es el caso se sugiere al lector tomar un tiempo para regresar al apartado introductorio ya que en los próximos proyectos no se explicara a detalle funcionalidades y características básicas.

Ahora bien, finalmente nos ensuciaremos las manos y empezaremos con la practica, es preciso comentar que en cada practica se puntualizara sobre el hardware que es necesario, ademas de ciertas características que son fundamentales para la realización del proyecto.

Antes de empezar con cualquiera de los siguientes proyectos es necesario verificar que la Raspberry, el hardware adicional y la conexión a Internet estén funcionando perfectamente.

Para la facilidad del lector los proyectos que a continuación se presentan se guiaran por una misma estructura:

- Introducción al proyecto
- Hardware y software requeridos
- Esquema del circuito
- Código
- Resultados

El lector debe tener en consideración que los procedimientos pueden manipularse a su gusto y tomar a estos proyectos como punto de partida, con esto aventurarse a crear ideas originales y de mayor magnitud.



## Chapter 1

# Controlador de motores a distancia

En caso de que sea la primer experiencia para el lector en el uso de la Raspberry el título del proyecto puede parecer algo complicado, sin embargo es un proyecto que requiere conocimientos básicos en programación en Python y tener una vaga idea sobre el uso de la librería RPi.GPIO, el hardware adicional es sencillo de conseguir y no se requiere de una pre-configuración. En la parte introductoria de este manual se explica minuciosamente el uso de la librería RPi.GPIO.

Básicamente el proyecto consiste en la elaboración de un circuito que pueda manipular un par de motores RF370C15370 a través de un Puente Doble L298, todas las instrucciones se montan sobre la Raspberry y el control remoto controlado desde un sistema con soporte de JAVA.

No es necesario que el control remoto este programado en JAVA, puede estar elaborado en cualquier otro lenguaje e incluso trabajarlo dentro de la misma Raspberry y usar el escritorio remoto en el otro dispositivo, pero se pretende que el lector identifique la forma en que se trabaja con socket, que funciona con una conexión mediante las ip de los dispositivos. Mas adelante se explicara a detalle el funcionamiento de este tipo de conexión.

Es bastante evidente que el lector identifique la estructura del modelo con un carro a control remoto. El uso de motores, su disposición dentro del diagrama y la interfaz del control en JAVA se pueden asociar a que la construcción de cierto vehículo es el objetivo principal, sin embargo el lector puede usar dicho proyecto con diversos usos. Por tanto no se mencionara acerca de algún tipo de montaje de llantas o de alguna carcasa protectora.

## Requerimientos de hardware y librerías

- ◇ RF370C15370 Mabuchi Original DC Motor (x2)
- ◇ Controlador de motores doble puente H - L298
- ◇ Librería net para python
- ◇ Librería net para java

## Esquema y circuito

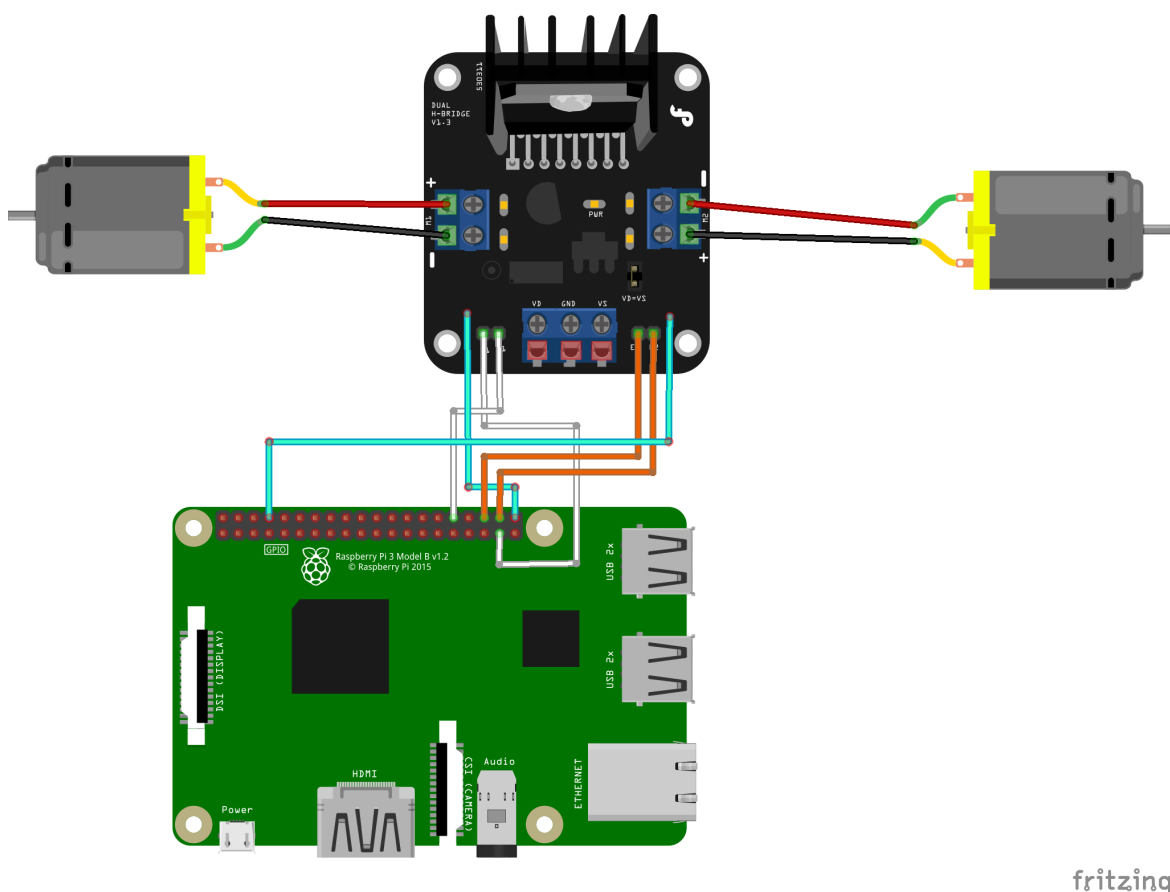


Figure 1.1: *Esquema 1, Controlador de motores a distancia*

Tal como se aprecia en el esquema Figura:1.1 las conexiones entre los componentes es poco compleja pero resulta de suma importancia identificar la posición de los motores si no queremos que estos giren en sentido contrario.

Los motores RF370C15370 Mabuchi Original DC Motor no cuentan con polaridad por tanto es imprescindible preocuparse del orden de la carga. Pero se aconseja tener en cuenta que conector lleva la carga positiva y de ser posible tener el mismo color de conector para la misma carga en ambos motores.

Ahora bien, en cuanto a la conexión entre la Raspberry Pi y el Controlador de motores doble puente H - L298 es aconsejable al menos para este primer proyecto seguir las indicaciones tal como se especifican en este manual, aunque como bien sabrá el lector es posible tomar, en las conexiones, pines diferentes a los que se especifican tanto aquí como en la captura de código que se encontrara adjunto en la sección de anexos de este manual. *Vease: Anexo:\**

Como ya se menciona es importante saber que pines en la Raspberry Pi conectaran con los pines del motor que controlaran el motor derecho y cuales lo harán en el motor del lado opuesto, por tanto en la Tabla:1.1 se muestra explícitamente el nombre del pin tanto en la Raspberry Pi como en el Controlador L298. Estos datos coinciden con el esquema 1.1. La referencia del diagrama de los pines de la Raspberry Pi se encuentra en la parte introductoria de este manual.

Nota: Se le recuerda al lector que la referencia de los pines de la Raspberry Pi que se manejaran, a menos que se establezca el caso contrario, será con Board

Table 1.1: *Controlador de motores a distancia: Conexión de los pines*

Raspberry Pi	Controlador L298
37	in4
32	in3
40	enb
36	in2
38	in1
7	ena

En la parte introductoria de este manual se da una breve explicación acerca del funcionamiento de un circuito y consideraciones que se deben tener presente para la alimentación de corriente tanto para este como para los siguientes proyectos.

## Programación

Tanto Java como Python traen por defecto la librería 'net' la cual ocuparemos en el socket que nos servirá para conectar java con python y hacer funcionar el control remoto que estaremos ocupando en otro dispositivo diferente a la Raspberry Pi.

Ahora bien, comenzaremos con el diseño de la interfaz gráfica que tendrá nuestro control. En la Figura ?? se muestra la disposición de botones que se ocupan, se recomienda usar un nombre específico en la etiqueta de cada botón para evitar confusiones en el código.

La conexión del socket y las librerías las encontraremos en el Anexo?? al final de este manual. Como puede notarse se ocuparon 6 botones, 4 de ellos mandaran la indicación del movimiento (Run, Right, Left y Reverse), otro botón para indicar que los motores se detengan (Stop) y un ultimo botón que es de suma importancia, que indicara el corte del proceso, hay que tener en cuenta que el botón Stop no cortara el proceso ya que se pretende que el botón Stop solo sea un frenado sin intención de terminar de interactuar con los motores. El código en java se muestra en el Anexo ??.

Para el uso del socket es indispensable abrir un canal, en este ejemplo se uso el canal

'9999' pero se puede ocupar cualquier otro canal, siempre y cuando se tenga la referencia en todo el código que ese canal es el que se está ocupando.

En el Anexo ?? se muestra como se invocara cada uno de los 6 botones, para este ejemplo la etiqueta se dejó con el nombre predeterminado, pero como ya se comentó anteriormente es recomendable cambiarlos a nombres más particulares.



Figure 1.2: Esquema 2, Interfaz Grafica

Prácticamente con esto hemos terminado la parte del control con el que se manejarán los motores a distancia, el cual por obvias razones estará en otro dispositivo. A continuación empezaremos a programar los pines de la Raspberry Pi con Python.

Lo inicial es llamar las librerías que ocuparemos, es muy importante la librería RPi.GPIO, la cual como se muestra en el Anexo ?? usaremos el seudónimo GPIO para referenciarla más fácilmente y la librería socket que como se explico antes con la net en Java, se utilizara para la conexión entre ambos programas. En este mismo Anexo se da la instrucción de como nombraremos los pines para su referencia, y adicionalmente agregamos 'GPIO.setwarnings(False)' que se usara para evitar los warnings que aparecen indicando que los pines están ocupados.

Ahora que ya tenemos preparadas las librerías empezamos a referencia los pines que estarán conectados de la Raspberry Pi y el Controlador de Motores L298, en el Anexo ?? se muestran las conexiones de acuerdo a la Tabla 1.1 tanto para motor izquierdo y motor derecho, se vuelve a hacer incapie en poner atención la orientación de los motores para evitar algún fallo.

Bien, ahora recordemos la conexión que le pusimos al socket en java, ahora usaremos ese numero de conexión mas la ip del dispositivo desde donde estaremos usando el control y activamos conjuntamente el socket. Tanto esto como las indicaciones que mandaremos a los motores se encuentran en el Anexo ??, es importante tener en cuenta que GPIO.HIGH maracará la dirección hacia donde girara el motor, mientras que GPIO.LOW sera la dirección en reposo del motor. En cuanto a pulso.start se refiere a los pulsos a los que girara el motor al que se le indica. Como podemos observar al final del codigo el apretar una tecla cualquiera interrumpira el proceso, poniendo los motores en 0.

Es buena practica usar GPIO.cleanup() para dejar limpios los pines que se programarán cuando el programa deje de ejecutarse.

## Chapter 2

# Automatización de actividades por Whatsapp

¿Ocupar una aplicación de mensajería para controlar las luces de tu hogar, estes donde estes? ¿Alimentar a tus mascotas con un horario exacto aun sin estar cerca de ellas?

Al proyecto se le pueden añadir muchas ideas más, como abrir ventanas, cerrar puertas con seguridad e incluso ambientar tu hogar con un solo toque en la pantalla del smartphone.

Con este proyecto el lector comprenderá la forma en como montar un cliente de servicio de mensajería instantánea como en este caso Whatsapp, aunque el lector debe tener en cuenta que hacerlo con otros proveedores como Telegram también es posible y que lleva un proceso similar, sobre la Raspberry Pi. Como se podrá observar esta es la parte más significativa y rigurosa del proceso ya que depende mucho de las actualizaciones tanto en el sistema Raspbian como en las políticas que varían con cada actualización de Whatsapp.

Con esta primera fase del proyecto el lector tendrá la posibilidad de usar la terminal de Raspberry Pi para enviar mensajes de texto, mensajes multimedia, agregar contactos y modificar ciertas configuraciones de Whatsapp.

Ahora bien para facilitar el uso crearemos un menú con todas las instrucciones posibles que la Raspberry Pi será capaz de efectuar. Por tanto podemos ampliar las instrucciones tanto como deseemos. Dicho menú será enviado al smartphone y la Raspberry estará a la espera de recibir el número de instrucción que se desea realizar.

Para finalizar, esta instrucción recibida en el cliente de Whatsapp de Raspberry creará un nuevo proceso que maneje el hardware necesario para cumplir el objetivo final. Aquí únicamente presentaremos como ejemplo el encendido de una lámpara.

## Requerimientos de hardware y librerías

◇

**Esquema y circuito**

**Programación**



## Chapter 3

# Automatización de riego

¿Automatizar una tarea que no posee mayor complicación como proveer de agua a una planta? De acuerdo, pero ahora pensemos que nuestra tarea se convierte en proveer de agua a todo un invernadero o a un huerto con dimensiones considerables. Mas que eso, pensemos que nuestra Raspberry Pi contara con la información sobre la humedad del suelo y con ello hidratara la tierra cuando realmente sea necesario. Suena a un trabajo mas eficiente ¿no?. Y si ademas le añadimos información sobre que tipo de planta que pretendemos crezca en cierta area ¿Ya tiene el suficiente merito usar nuestra Raspberry Pi? Seguro que si.

Bien, como podras notar ademas de nuestra Raspberry necesitaremos un sensor de humedad que se encontrara emergido en el suelo cuyo objetivo será proporcionarnos información sobre el estado de hidratación con la que cuenta la tierra, enviando una señal a la Raspberry Pi cuando el suelo requiera agua para mantener hidratada a la planta.

La Raspberry Pi por su parte la programaremos de modo que, al recibir la señal del sensor de humedad, detectara a que sensor corresponde esta señal y reconocera a que segmento de nuestra plantación dirigira la salida de agua, teniendo en cuenta que tipo de plantación se encuentra en dicho segmento, determinando asi el volumen de agua que abastecera a dicha area.

En este manual se le dará al lector la información necesaria para la automatización del riego de una sola planta. Aunque extendiendo el codigo y los diagramas el lector podra magnificar el ejersisio a proyectos mas ambiciosos y de mayor dimension.

## Requerimientos de hardware y librerías

◇

## Esquema y circuito

## Programación

## Chapter 4

# Controlador de la montura de un telescopio

Cuanto tiempo tardamos en lograr enfocar un objetivo y con el minimo error, sobre todo con telescopios de montura altacimuntal, nos desviamos distancias enormes, al grado de preferir espiar a los vecinos. Bueno antes de encontrarnos con problemas con la ley, mejor usemos nuestra Raspberry Pi y demos una presición casi exacta al astro que decidimos contemplar a travez de nuestro telescopio.

Es claro que necesitaremos un tipo de montura que podamos manipular con un ordenador, que en este caso será nuestra Raspberry Pi, mas adelante se daran algunos modelos de montura que se pueden ocupar.

Sabemos que es facil ubicar objetos en la boveda celeste gracias a coordenadas, asi que estas serán nuestras herramientas para enfocar con precisión optima a nuestro objetivo. Las coordenadas las manipularemos usando el software matématico 'Matlab'. En este manual nos limitaremos a explicar ciertas configuraciones, mas no la instalación del software.

No es indispensable el uso de un software de astronomia, aunque este claro, facilitara bastante las tareas de mapeo y busqueda de objetos astronomicos, por lo cual se recomienda al lector descargar previamente Stellarium o Sky Guide. El primero es de codigo abierto y su intalación es posible en Windows, Mac Os X y Linux. El segundo esta disponible para Android y iOS.



## Chapter 5

