ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

"DESARROLLO DE UNA PLATAFORMA GIRATORIA PARA LA INSPECCIÓN REMOTA AGRÍCOLA"

INFORME DE PROYECTO INTEGRADOR

Previo a la obtención del Título de:

INGENIERO EN CIENCIAS COMPUTACIONALES
ORIENTACIÓN SISTEMAS TECNOLÓGICOS

DANIEL ISIDRO SEGURA SALAZAR

GUAYAQUIL – ECUADOR

AÑO: 2015

AGRADECIMIENTOS

Mis más sinceros agradecimientos a…

Mi familia, Ioanna Maria, Isidro, Mariana, Aminta, Francisco. Agradezco al Creador de todas las cosas que nos bendice con el potencial de desarrollar nuevas habilidades y también a las personas que me he encontrado este camino, personas que de alguna manera han contribuido para que se pueda lograr esta meta.

DEDICATORIA

El presente proyecto lo dedico a toda mi familia por su apoyo en este proceso, en especial a mi padre, Isidro; por ser siempre un excelente ejemplo para sus hijos.

TRIBUNAL DE EVALUACIÓN

Ph.D. Dennis Romero	**MSc. Ronald Ponguillo**
PROFESOR EVALUADOR	PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

_____

Daniel Isidro Segura Salazar

RESUMEN

El mercado de cámaras que puedan realizar movimientos pan/tilt (horizontal y vertical) se encuentra en constante crecimiento, razón por lo cual el precio de las cámaras tiende a bajar a medida que el mercado va saturándose. Por lo general, las características y prestaciones de estas cámaras varían, siendo las cámaras más costosas las que pueden tener mayor capacidad de procesamiento.

Este proyecto surge por la necesidad de contar con una camara IP de bajo costo pero a su vez de una potencia de procesamiento mayor que muchas cámaras costosas existentes en el mercado.

ÍNDICE GENERAL

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA

1.1 Introducción del capítulo

El propósito de este capítulo es introducir al lector en el ámbito de los dispositivos embebidos y el uso de estos en aplicaciones de monitoreo remoto. Se hará una breve descripción de los dispositivos existentes, sus ventajas y desventajas, el ámbito donde su uso es más amplio y como pueden estos dispositivos ser introducidos en el área de la monitorización agrícola. Este documento está enfocado en la construcción de un dispositivo de monitorización que use el internet como medio de transmisión, de bajo costo y que use Raspberry pi como sistema embebido.

Para entender mejor la problemática a abordar, a continuación se explica brevemente lo que es un sistema embebido, sus características y como estos dispositivos pueden ser utilizados como medios de monitoreo y video vigilancia en aplicaciones de interés nacional como es el caso de la agricultura y seguridad ciudadana.

1.2 Sistemas embebidos

Un sistema embebido (SE) se define como un sistema electrónico diseñado específicamente para realizar determinadas funciones, generalmente un sistema de computación en tiempo real. Un SE emplea uno o varios procesadores (CPUs) en formato de microprocesador, microcontrolador o DSP.

Al contrario que los computadores de propósito general como son las PC, los cuales están diseñados para cubrir una amplia gama de necesidades, los sistemas embebidos se diseñan para cubrir necesidades específicas. En los sistemas embebidos la mayoría de los elementos se encuentran incluidos en una placa base, por eso muchas veces el resultado final no se parece mucho a una computadora.

La parte central de un SE es el microprocesador, microcontrolador o DSP. Es decir, la CPU o unidad que aporta capacidad de cómputo al sistema, pudiendo incluir memoria interna o externa.

El módulo de reloj es el encargado de generar las diferentes señales de reloj a partir de un único oscilador principal. El tipo de oscilador es importante por varios aspectos: por la frecuencia necesaria, por la estabilidad necesaria y por el consumo de corriente requerido. El oscilador con mejores características en cuanto a estabilidad y coste son los basados en resonador de cristal de cuarzo, mientras que los que requieren menor consumo son los RC [1]. Mediante sistemas PLL (multiplicadores de frecuencia) se obtienen otras frecuencias con la misma estabilidad que el oscilador principal [2].

El consumo de energía puede ser determinante en el desarrollo de algunos sistemas embebidos que necesariamente se alimentan con baterías, con lo que el tiempo de uso del SE suele ser la duración de la carga de las baterías.

En lo que se refiere al software, se tendrán requisitos específicos según la aplicación. En general para el diseño de un SE no se dispone de recursos ilimitados sino que la cantidad de memoria es escasa, la capacidad de cálculo y dispositivos externos es limitada.

Podemos hablar de las siguientes necesidades:

- Optimizar al máximo los recursos disponibles.
- Disponer de un sistema de desarrollo específico para cada familia de microprocesadores empleados.
- Programación en ensamblador, aunque en los últimos años, algunos fabricantes han mejorado la oferta de compiladores que nos permiten trabajar en lenguajes de alto nivel, tales como C, Python entre otros.

El empleo de un sistema operativo determinado o el no empleo de éste dependerá del sistema a desarrollar y es una de las principales decisiones que habrá que tomar en la fase de diseño o elección del SE. Así, en el caso de decidirse por el empleo de microcontroladores y DSP, por lo general no se usará sistema operativo. En caso de emplearse algún microprocesador del tipo ARM, PowerPC, Intel X86, etc. sí lo llevará. La decisión dependerá de los requisitos del sistema entre otros.

1.3 Sistemas de monitorización y vigilancia

Los instrumentos usados para realizar video vigilancia permiten ver una imagen en directo de nuestra casa, negocio u área específica. En el mercado existen muchas opciones para poder lograr este objetivo, los más usados son los dispositivos que usan la comunicación TCP/IP para lograr la transmisión de datos desde una camara hacía un receptor.

Existen también dispositivos más complejos, equipos que pueden soportar una red de cámaras de video vigilancia y gran capacidad de almacenamiento. Usan redes de datos dedicada, muchas veces una red análoga, convirtiendo después estos datos análogos a digitales para su posterior almacenamiento.

En los últimos años gracias al avance de la tecnología y la construcción de circuitos integrados que pueden albergar millones de transistores, los costos de estos se han reducido de una manera considerable y su capacidad de procesamiento ha ido en aumento. Esto ha dado como resultado que muchas de las tareas que en el pasado eran difíciles de realizar usando microprocesadores ahora sean tareas que se pueden realizar en un tiempo relativamente corto.

Actualmente existen en el mercado muchas opciones para poder construir dispositivos de monitoreo remoto para ser usados en diferentes campos como el monitoreo de plantaciones agrícolas, control de procesos, seguridad privada, etc.

El rendimiento de la transmisión de video en este tipo de aplicaciones es de vital importancia ya que permite al usuario saber lo que está pasando, casi en tiempo real, en el área deseada. Este método de transmisión de datos se conoce como *streaming*, lo que quiere decir que la información se transmite de forma instantánea y sin interrupción [3].

A diferencia de la transmisión de datos como texto, voz para transmitir video se usa una gran cantidad de datos, por lo tanto en un sistema embebido se debe tener en cuenta la capacidad de procesamiento. Otro punto muy importante a considerar es la interfaz de usuario, normalmente este tipo de proyectos se desarrollan junto a un software de administración para acceder a las opciones y

configuraciones de los dispositivos, también para visualizar los datos obtenidos de los sensores en caso de que los hubiere.

1.4 Cámara TP-LINK

Un sistema sencillo de monitorización mediante video se lo puede encontrar bajo la marca TP-LINK como vemos en la figura 1.1. La camara de modelo TL-SC4171G cuyo valor en el mercado es de aproximadamente $200 posee las siguientes características:

- Resolución de 640x480, 0.3 megapíxeles.
- 354 grados de giro horizontal (pan) y 125 grados de inclinación vertical (tilt).
- Zoom digital hasta 10x.
- Compresión de video JPEG y MPEG-4.
- Velocidad de imagen hasta 30 cuadros por segundo (fps) con formato MPEG-4, hasta 15 fps en formato JPEG.
- Conectividad wifi e Interface de red con conector RJ-45.
- 512MB de memoria RAM.
- Tarjeta gráfica integrada.
- Procesador Pentium 4 a 1.8 Ghz.
- Detección de movimiento y notificación mediante correo electrónico.
- Micrófono incorporado.
- No posee capacidad de almacenamiento en el dispositivo.

Figura 1.1: Camara TP-LINK, pant/tilt.

La interfaz de usuario está alojada en el dispositivo y se puede acceder mediante un navegador web. El sistema operativo o firmware del dispositivo es propietario, es decir, posee derechos de autor por lo que el usuario tiene pocas posibilidades de modificarlo [4]. Este tipo de cámaras son una buena opción cuando se desea establecer un circuito de cámaras para la vigilancia de áreas específicas tales como monitorización de áreas agrícolas, vigilancia contra intrusos, cuidado de niños, etc.

En cuanto al uso de estos dispositivos en el campo agrícola la asociación de jóvenes agricultores de Murcia, España; en 2011 puso en marcha un proyecto que utiliza cámaras de video vigilancia con el objetivo de bajar los índices de robos que se producen en el campo en áreas agrícolas y ganaderas. Los agricultores estaban expuestos a robos de sus bienes, herramientas, elementos de riego, maquinaria pesada y también destrozos en cultivos. La idea bajo este

proyecto es que los agricultores puedan visualizar las imágenes de sus propiedades en tiempo real de manera remota mediante Internet o en una red local [5].

Una de las desventajas que tienen este tipo de proyectos se dá cuando deben cubrir áreas muy extensas o cuando los dispositivos están a la intemperie porque las cámaras que pueden cubrir distancias grandes, es decir, que cuentan con zoom más potente y que puedan resistir el clima adverso suelen tener un valor elevado en el mercado en comparación con cámaras de uso cotidiano, además del costo del cableado para proporcionar energía a los dispositivos transmisión de datos. No olvidemos también que usualmente los robos ocurren durante la noche, por lo que hay que tener en cuenta cámaras con visión nocturna.

1.5 Cámara con módulo Arduino

Un ejemplo interesante de lo que se puede hacer con sistemas embebidos de bajo coste y código accesible a todo el público ha sido realizado por Sergiy Bogdancev [6], es la construcción de una camara USB como se muestra en la Figura 1.2. Se usan los siguientes elementos:

- Arduino Due.
- Pantalla táctil de 3.2 pulgadas.
- Una camara usb.

La tarjeta electrónica Arduino Due posee un microcontrolador ATMEL SAM3X8E ARM Cortex-M3 de 32 bits [7], opera a una velocidad de reloj de 84 MHz, llegando a alcanzar 105 millones de instrucciones por segundo. El puerto USB de la tarjeta puede alcanzar velocidades de transmisión de datos hasta 480 Mbps, ancho de banda suficiente para la transmisión de datos de una camara USB de gama media.

Figura 1.2: Camara con Arduino.

Aunque este sistema no se centra en usar un servidor web como medio de visualización del video, existen módulos electrónicos que se pueden acoplar a la tarjeta principal Arduino Due y proporcionar la capacidad de tener un servidor web embebido en el sistema y para mostrar las imágenes mediante una página web.

El módulo Arduino no ejecuta un sistema operativo lo que implica que la mayoría de tareas que el procesador principal va a realizar tengan que ser programadas en lenguaje de alto nivel, luego usar un compilador para traducir las instrucciones a lenguaje de máquina. El compilador que se usa para realizar esta tarea es Atmel Studio, es un entorno de desarrollo integrado que facilita la tarea de escribir el programa (código) y luego trasladarlo a una gran variedad de microprocesadores. La alimentación de energía es por el puerto USB de una computadora, como se puede ver en la figura 1.2 y también de una fuente de alimentación externa. El consumo máximo de corriente que tendrá todo el sistema no será mayor de 0.5 amperios a 5 voltios que es el límite de corriente que puede entregar un puerto USB usualmente.

El desarrollo del firmware para la placa Arduino toma más tiempo en comparación con el desarrollo en una placa que ejecuta un sistema operativo.

Aunque la placa posee módulos de comunicación para transferir datos entre la camara y la pantalla táctil, es necesario tener conocimientos de comunicación USB, UVC y formatos de video, para poder implementarlos mediante la programación. En la figura 1.3 podemos apreciar la calidad de video de salida mostrado en la pantalla de 3.2 pulgadas, la resolución de la imagen es de 176x144 pixeles [8].

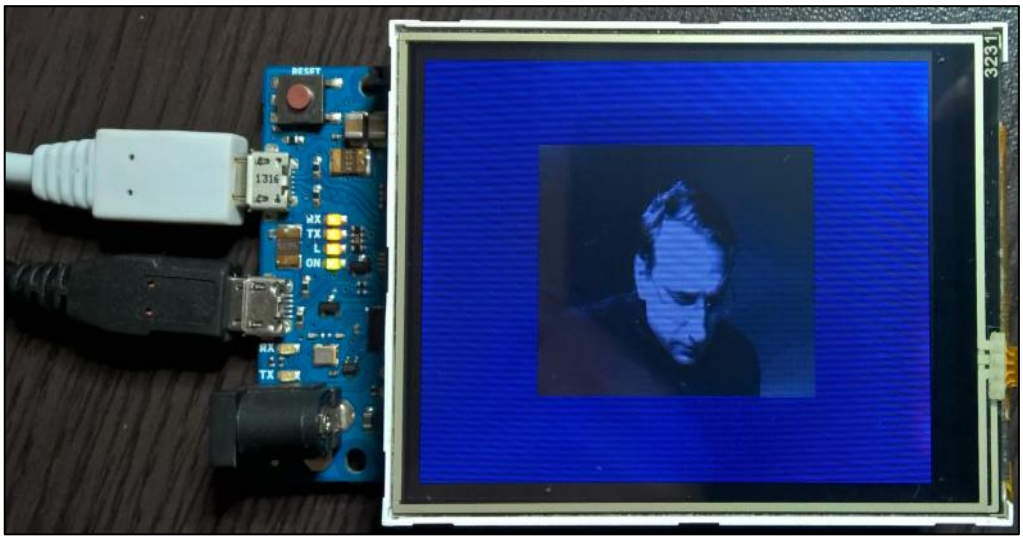

Figura 1.3: Salida de video

Aunque la camara USB se puede capturar video a una resolución de 640x480 pixeles, la resolución de la pantalla es de 320x240 por lo que es necesario configurar la camara a una resolución igual o menor a la de la pantalla. Las pruebas realizadas por Sergiy Bogdancev [8], dieron como resultado que la resolución óptima para este sistema embebido sea de 176x144 pixeles, llegando a la conclusión que este módulo dispone de una capacidad muy limitada en cuanto a poder de procesamiento y memoria para manejar secuencias de video de una resolución de 640x480.

Actualmente los proyectos que se desarrollan con sistemas embebidos para la captura de imágenes o video necesitan contar con una buena calidad de imagen para poder realizar un procesamiento posterior. En el campo de la agricultura cuando se cuenta con buenas imágenes se pueden detectar

diferentes problemas en los cultivos, se pueden tomar decisiones en base a las imágenes obtenidas y resolver distintos tipos de problemas. Las aplicaciones donde se usan sistemas embebidos de bajo costo y consumo de energía como Arduino podrían ser proyectos donde se requiere monitorear las plantaciones a una frecuencia menor en comparación con los sistemas de video vigilancia. Debido al bajo consumo de energía estos dispositivos pueden estar conectados a una batería recargable, inclusive podría estar conectado a un panel solar y así se reducirían aún más los costos.

1.6 Camara IP con Raspberry pi

Mientras avanza la tecnología los dispositivos electrónicos se van haciendo más pequeños y accesibles, un ejemplo de esto es el Raspberry pi. Esta es una placa electrónica que cuenta con todas las partes de una computadora pero en un área muy reducida, como se puede apreciar en la Figura 1.4. Tiene puertos de propósito general para la entrada y salida de señales eléctricas, posee salida de audio y video, conexión TCP/IP, puertos USB, capacidad de comunicación serial, entre otros [9].

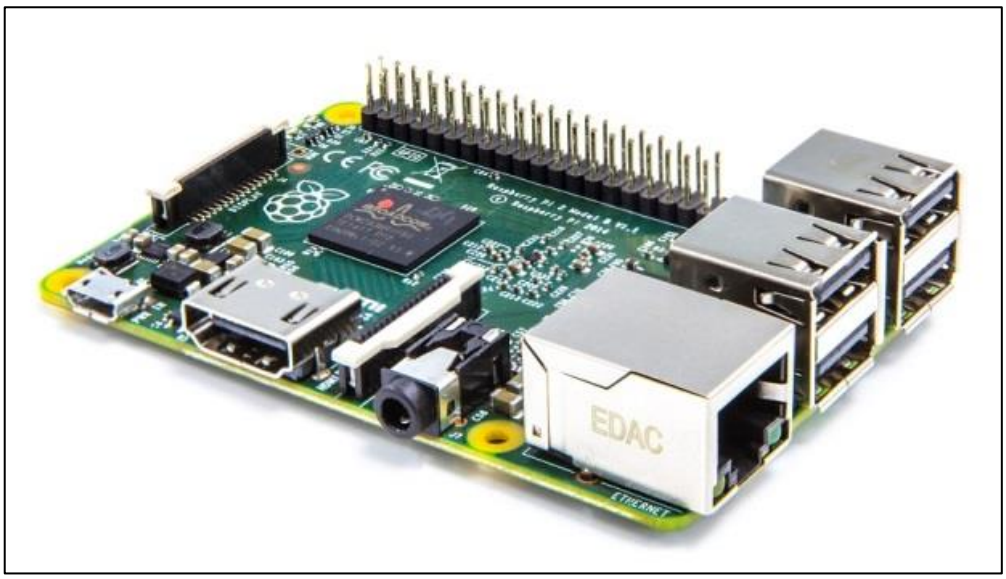

Figura 1.4: Raspberry pi 2.

La última versión de esta mini computadora es Raspberry pi 2 posee un procesador Broadcom de 4 núcleos, con arquitectura ARM v7 y puede ejecutar procesos hasta una velocidad de 900 MHz, también se ha actualizado la memoria RAM que ahora cuenta con 1 Gigabyte de capacidad. Estas características hacen que el Raspberry pi sea ampliamente usado para realizar proyectos de diversa índole. Esta placa puede ejecutar varios sistemas operativos, de los cuales, el que más se acopla a las necesidades para desarrollar una camara IP con control remoto es el sistema operativo Raspbian, debido a que es muy fácil de usar y viene con varios programas útiles preinstalados.

Existen muchos ejemplos y tutoriales de cámaras ip implementadas con Raspberry pi usando diversas herramientas para poder mostrar los datos de la camara en una página web o en un reproductor de video. Entre los programas para realizar streaming de video desde el Raspberry pi tenemos los siguientes:

- Mjpg streamer.
- Motion.
- Gstreamer.

Mjpg streamer es una aplicación para línea de comandos que copia imágenes JPG de una entrada hacia múltiples salidas. Usualmente esta entrada es un espacio de memoria del Raspberry pi donde se están copiando continuamente las imágenes de la cámara y la salida es un simple servidor web donde se podrá visualizar el video. Esta aplicación puede ser usada para realizar streaming de video desde una camara web hasta un navegador como Firefox, Chrome o un visualizador de streams como VideoLan [10]. La última versión es Mjpg streamer experimental y supone mejoras en la transmisión de video. Debido a esta mejora, Mjpg streamer experimental es posiblemente una de las principales opciones para realizar proyectos de monitoreo ambiental en tiempo real, video vigilancia y seguridad.

Existe mucha información acerca de la compilación y puesta en macha de Mjpg streamer y cómo hacerlo funcionar con una camara conectada al Raspberry pi.

Un ejemplo es el proyecto propuesto en Alemania, disponible en la página web especializada en proyectos de robótica usando sistemas embebidos llamada http://www.robosphere.de/. El resultado del desarrollo como se ve en la Figura 1.5, es un robot que puede ser controlado con el Raspberry pi accediendo a una interface desde cualquier navegador web [11].

Figura 1.5: Robot controlado mediante una interface web.

El control del robot mediante una interfaz web supone una ventaja debido a que se puede controlar el robot desde cualquier parte del mundo y desde cualquier computador o dispositivo móvil que pueda ejecutar un navegador web. En la ejecución de la aplicación solo se necesita tomar en cuenta el uso de un navegador web, a diferencia de una aplicación de escritorio o una aplicación móvil donde hay que tomar en cuenta varios factores a la hora de desarrollar una aplicación que sirva de interfaz tales como sistema operativo, computador o dispositivo móvil, configuraciones específicas de cada dispositivo, etc.

Esta tecnología se está usando cada vez más en la agricultura para el monitoreo remoto de fincas pequeñas o parcelas de cultivos. Usualmente los agricultores deben estar presentes en sus campos para poder conocer el estado de los cultivos, temperatura, velocidad del viento, humedad; muchas veces tienen que enfrentar problemas externos a los cultivos como ataques de personas, animales, plagas y maleza. Usando tecnologías de monitoreo remoto el agricultor puede no solo ver en tiempo real lo que está sucediendo en su finca sino que también puede tomar acciones, cuando se usan estos dispositivos de

monitoreo junto con sensores electrónicos para tener un control completo del área de cultivo.

Una aplicación real es el proyecto llamado ROBOTIC URBAN FARM SYSTEM, véase Figura 1.6, el cual propone un sistema de jardinería hidropónico construido con materiales que se pueden encontrar en una ferretería y que están controlados por sensores Arduino y por Raspberry pi. Paul Langdom de BLT Robotics [12] compañía dedicada a mejorar la comunidad a través de proyectos de automatización, es creador de este proyecto. Este sistema aprovecha el espacio vertical y un área de aproximadamente 25 metros cuadrados para cultivar alrededor de 160 plantas.

Con sensores en Arduino y Raspberry pi es posible capturar información sobre las condiciones de temperatura, humedad e iluminación. Aunque estos dispositivos no cuentan con una cámara para el monitoreo remoto en tiempo real; esta es fácilmente adaptable. En la figura 1.6 podemos apreciar cómo está construido el sistema RUFS usando tuberías de PVC.

Figura 1.6: Cultivo hidropónico controlado con Arduino y Raspberry pi.

CAPÍTULO 2

2. IMPLEMENTACIÓN

2.1 Resumen del capítulo

En este capítulo se detalla la implementación de la plataforma giratoria usando Raspberry pi. Se analizan los elementos usados para realizar la construcción del proyecto y una descripción breve de las ventajas de cada uno de estos elementos. Adicionalmente se detalla paso a paso lo que se ha llevado a cabo en la instalación del software, los programas necesarios para lograr el streaming de video y como se logra controlar la estructura giratoria desde un dispositivo para la inspección remota.

2.2 Raspberry pi como sistema embebido

La popularidad de dispositivos como Raspberry pi ha ido aumentando debido a su versatilidad en la realización de proyectos donde se necesita la integración de hardware y software, especialmente software que pueda ejecutar tareas complejas como las que se pueden realizar con una computadora de escritorio.

Entre los elementos principales a considerar antes de realizar la instalación de los programas en Raspberry pi se detallan los siguientes, véase también la Figura 2.1:

- La fuente de alimentación debe ser regulada y que entregue una salida de 5 voltios con 2 amperios de corriente.
- Cable de alimentación con conector micro-usb.
- Camara.
- Teclado.
- Mouse.
- Cable de conexión HDMI.
- Pantalla.
- MicroSD con al menos 4GB de espacio libre.

Es necesario que antes de la instalación se cuente con estos elementos porque el sistema operativo necesita ser configurado.

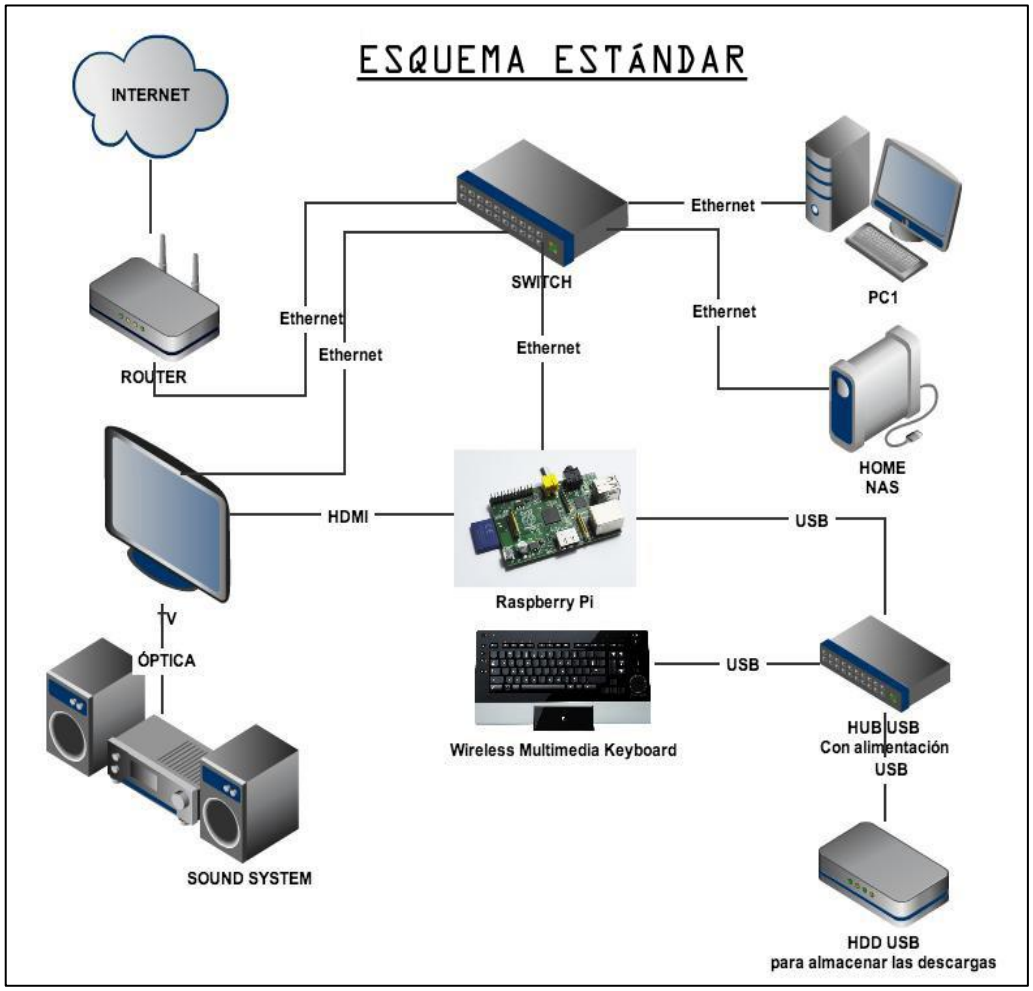

Figura 2.1: Esquema de conexión del Raspberry pi.

Programas a instalar en Raspberry pi:

- Sistema operativo Raspbian
- Servidor web Apache 2.
- Instalación de la camara.
- MJPG streamer-experimental.
- Instalación y pruebas de los motores de paso.

2.3 Pasos para la instalación del sistema operativo

Primero se formatea la tarjeta MicroSD que se insertará en el Raspberry pi, luego se realiza la descarga de los archivos del sistema operativo. Los archivos se descargan desde la siguiente página: https://www.raspberrypi.org/-downloads/noobs/, se escoge la versión 1.4.1, luego se descomprime el archivo en formato RAR y se copian los ficheros a la tarjeta MicroSD.

Posteriormente se procede a encender el Raspberry pi una vez que los periféricos de entrada y salida estén conectados. Al encender el Raspberry pi emergerá una ventana de opciones para escoger el sistema operativo a instalar, como muestra la Figura 2.2.

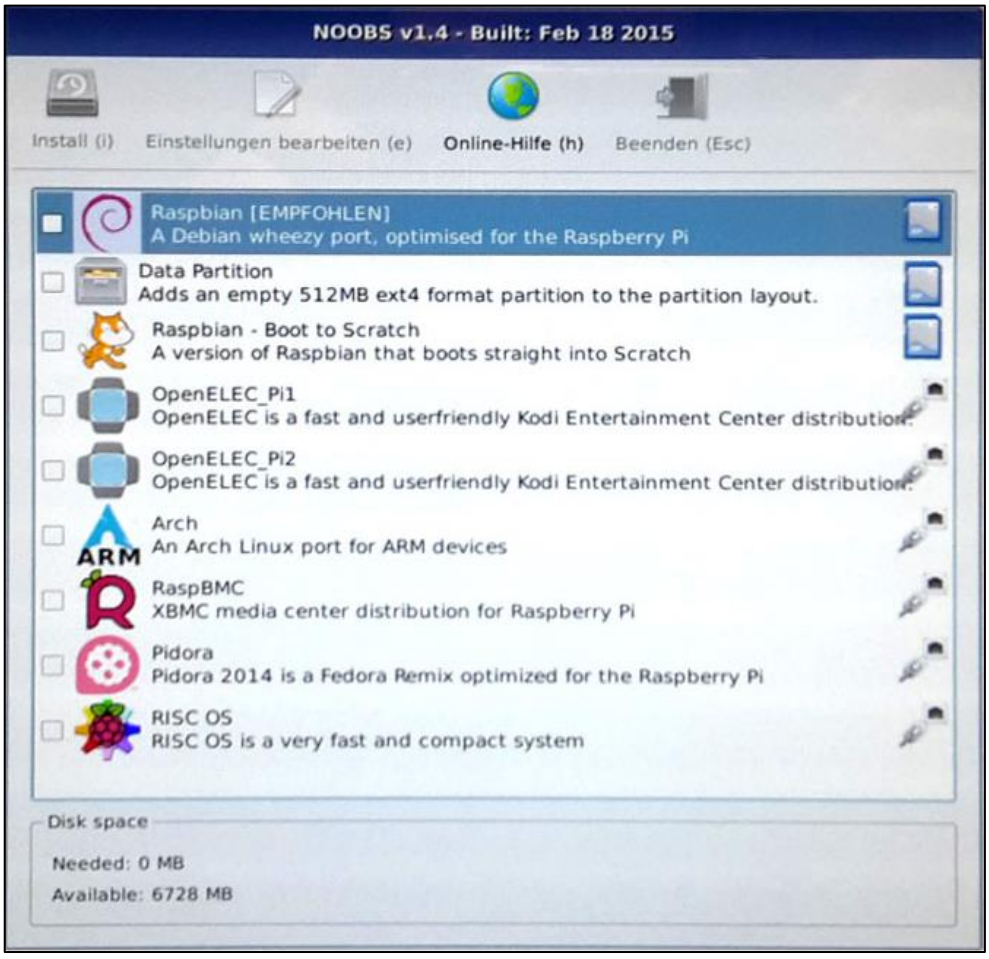

Figura 2.2: Pantalla de inicio por primera vez.

El sistema operativo a utilizar será Raspbian. Tiene algunas ventajas entre ellos la facilidad de instalación, la mayoría de programas compatibles con Raspberry pi son fáciles de instalar desde la terminal, tiene programas preinstalados útiles para la construcción del sistema de control de la plataforma giratoria remota, amplia documentación, gran cantidad de información acerca de las configuraciones necesarias para hacer funcionar los programas instalados, documentación relacionada con errores de compilación de programas, entre otros.

2.4 Instalación del servidor web

El servidor que se usará para alojar la interfaz web donde se controla la estructura giratoria y se ve la secuencia de video, es Apache [13].

Para instalar Apache, en la terminal de Raspberry pi se ejecuta el siguiente comando:

```
sudo apt-get install apache2 -y
```

Cuando ha finalizado la instalación se evalúa si el servidor se ha instalado correctamente. Con la dirección IP del Raspberry pi, desde una computadora que se encuentre dentro de la misma red local que Raspberry pi se comprueba la conectividad, en el terminal de comandos se realiza PING a la dirección IP previamente obtenida. Si al ejecutar el comando PING hay respuesta, entonces se comprueba que la conectividad ha sido exitosa.

Para comprobar que el servidor Apache se ha instalado correctamente, desde una computadora que se encuentre en la misma red que Raspberry pi, en un navegador web se accede a la dirección IP que tiene Raspberry pi. El resultado final se muestra en la Figura 2.3.

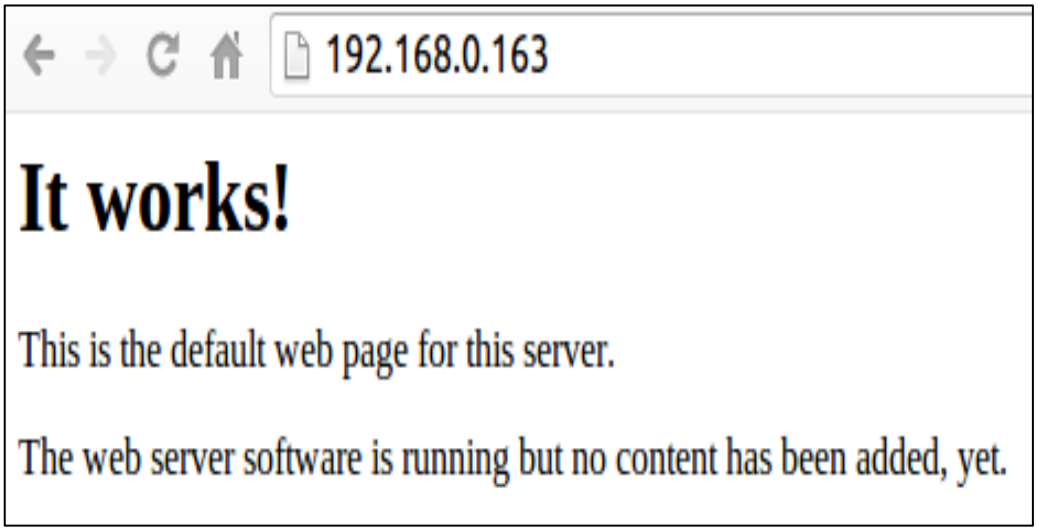

Figura 2.3: Instalación de servidor Apache exitosa.

Obteniendo este resultado se concluye que la instalación del servidor web Apache se ha realizado correctamente.

2.5 Instalación de la cámara web

La camara web es Raspberry Pi Camera Module, véase la Figura 2.4. Posee una resolución de 5 megapíxeles [14]. La ventaja de usar esta camara en vez de una camara USB es que tiene una velocidad de transmisión de datos mayor, mediante comandos se puede configurar varios aspectos de la calidad de video y la tasa de transmisión de datos.

Raspberry pi tiene un puerto CSI usado para la comunicación con el módulo de cámara. Usaremos este puerto conectado a un cable plano de 20 pines que a su vez estará conectado al módulo de cámara.

A diferencia de otros módulos existentes en el mercado, este cuenta con una base para adaptar lentes de diferentes dimensiones. Está fabricado el aluminio y es un adaptador universal al cual se puede acoplar fácilmente lentes de acercamiento y alejamiento, dependiendo del enfoque deseado.

Figura 2.4: Conexión de la camara.

2.6 Instalación de MJPG Streamer Experimental

Para poder realizar la visualización de los datos de la camara en la página web alojada en el servidor se requiere un programa que obtenga los datos de la camara y permita el acceso a esos datos desde una página web.

En Internet hay gran variedad de programas para este propósito, para este proyecto se ha escogido Mjpg Streamer Experimental, es una aplicación para línea de comandos que copia secuencias de imágenes en formato JPEG, desde una entrada hacia múltiples salidas y puede ser usada para enviar secuencias de archivos JPEG sobre una red IP hacia cualquier dispositivo que pueda ejecutar un navegador web [15].

A continuación se muestran los pasos para poder realizar la instalación de Mjpg Streamer Experimental desde la terminal del Raspberry pi:

- Se habilita la camara usando la interface de configuraciones del Raspberry pi.

```
$ sudo raspi-config
```

- Instalación de los paquetes necesarios.

```
$ sudo apt-get update

$ sudo apt-get install libv4l-dev libjpeg8-dev subversion
imagemagick v4l-utils
```

- Obtener Mjpg Streamer Experimental.

```
$ git clone https://github.com/jacksonliam/mjpg-streamer.git
```

- Compilar Mjpg Streamer Experimental.

```
$ cd mjpg-streamer-experimental

$ make USE_LIBV4l2=true clean all
```

- Configuración de la API para usar la camara del Raspberry pi.

```
$ sudo modprobe bcm2835-v4l2

$ sudo vi /etc/modules

# añadir la siguiente linea
  bcm2835-v4l2

$ sudo vi /boot/config.txt
```

- Para probar que la instalación de mjpg streamer fue exitosa, desde la carpeta donde se realizó la instalación se ejecuta el siguiente comando:

```
$ ./mjpg_streamer -i "./input_raspicam.so" -o "./output_http.so –
w       ./www"
```

Luego se accede mediante un navegador web a la dirección de Raspberry pi añadiendo el puerto 8080, por ejemplo: 192.168.1.5:8080. Lo que resulta en la salida de video como se muestra en la Figura 2.5:

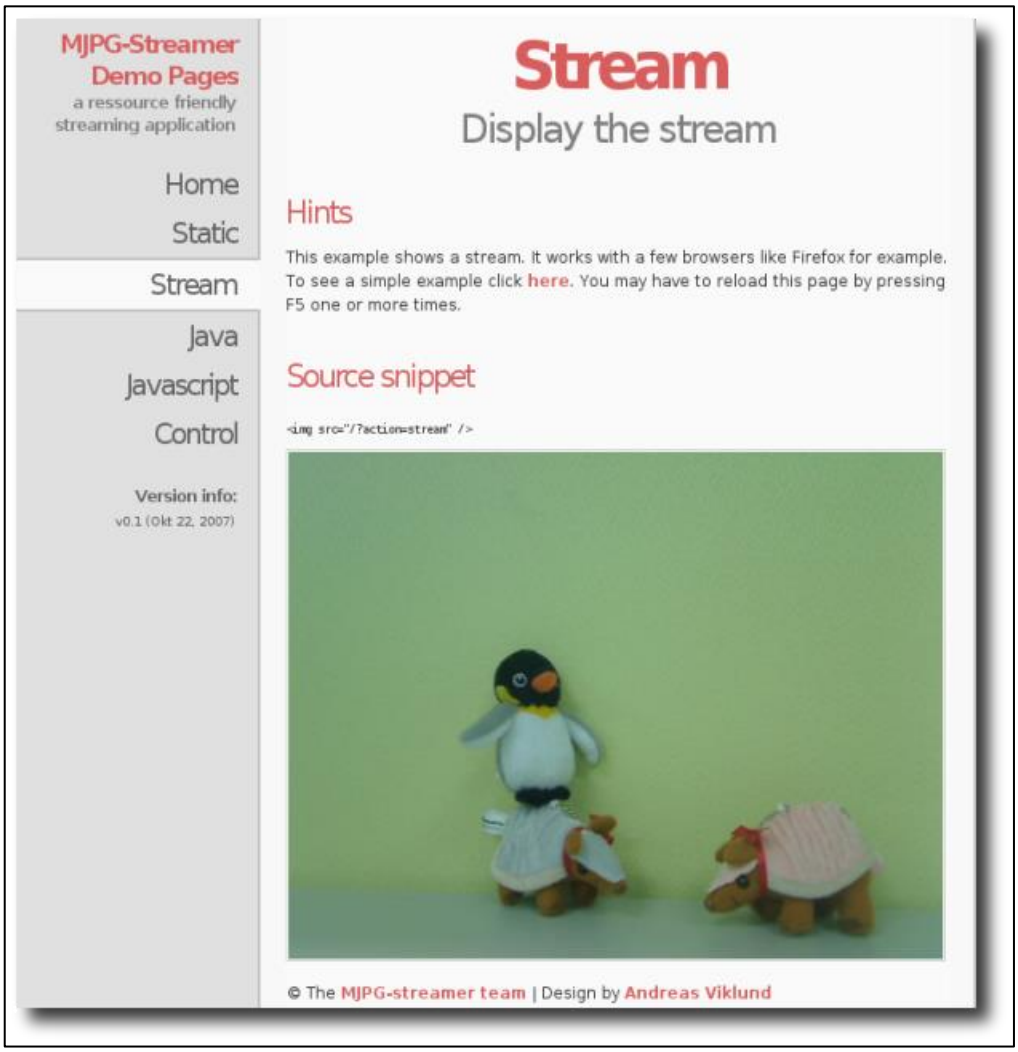

Figura 2.5: Pruebas con Mjpg Streamer Experimental.

2.7 Conexión y configuración de los motores de paso

La función de los motores de paso será el de proporcionar el giro horizontal y vertical, lo que comúnmente se conoce como movimiento pan/tilt. El motor de paso que se utilizará será el modelo 28BYJ-48 que trabaja a 5 voltios, véase la Figura 2.6. Es un motor de pasos unipolar, es decir, todas las bobinas que controlan el giro del motor tienen una conexión en común que es el pin conectado a 5 voltios; siendo 0 voltios el nivel de voltaje para permitir el paso de la corriente en cada una de las bobinas y así el motor gire un paso en la medida que las bobinas activen [16].

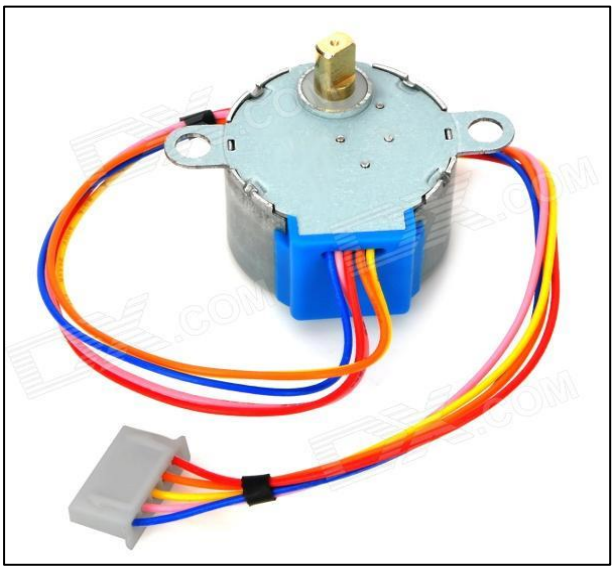

Figura 2.6: Motor de pasos.

Las conexiones de los cables del motor de pasos son las siguientes. Tabla 1:

Color	Función
Rojo	5 voltios - común
Naranja	Bobina 1
Amarillo	Bobina 2
Rosado	Bobina 3
Azul	Bobina 4

Tabla 1: Conexiones de un motor de paso.

Para realizar la conexión del motor de pasos y el Raspberry pi se requiere un circuito integrado controlador (driver) para manejar los impulsos eléctricos del motor de pasos, véase la Figura 2.7. Para esta tarea se usara el circuito integrado ULN2003APG, este circuito realiza la función de separar la alimentación eléctrica del Raspberry pi con los motores de paso, ya que los pines GPIO del Raspberry pi no pueden proveer suficiente corriente para poder alimentar los motores de paso directamente.

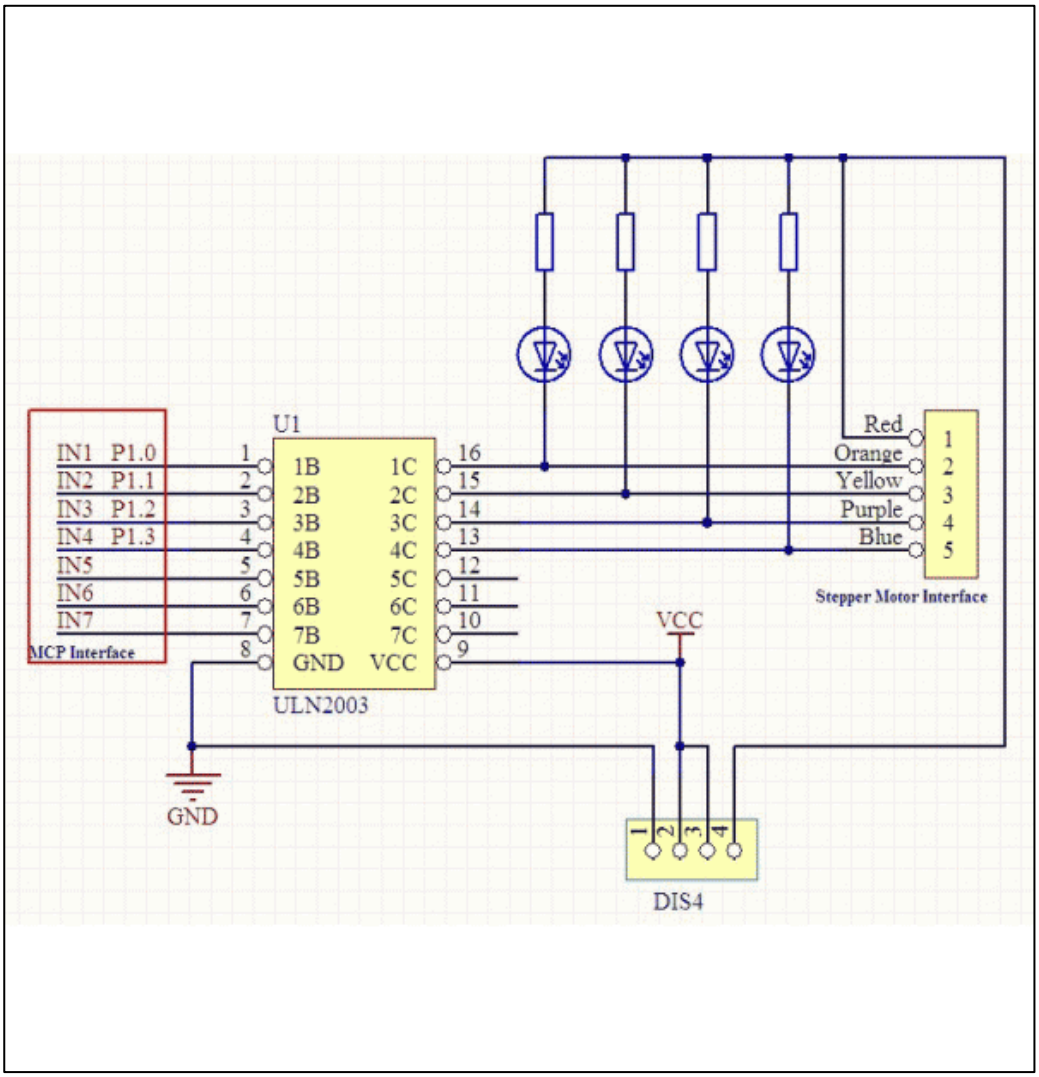

Figura 2.7: Diagrama esquemático del controlador de motor de pasos.

En la figura 2.7 se ven las señales de entrada del driver ULN2003 (IN1-IN7). Para controlar un motor de pasos con los puertos GPIO del Raspberry pi solo se necesitan 4 pines del driver, IN1 a IN4. En la conexión entre los pines del Raspberry pi y el driver no se necesitan elementos adicionales tales como capacitores y resistencias, debido a que se está trabajando con 5 voltios. Como vamos a necesitar 2 motores de paso uno para el movimiento horizontal y otro para el vertical se habilitaran 8 pines GPIO del Raspberry pi, es decir, 4 pines para cada motor. En la Figura 2.8 se muestran los puertos disponibles en el Raspberry pi y cuales se usarán en este proyecto, véase Tabla 2 y Tabla 3.

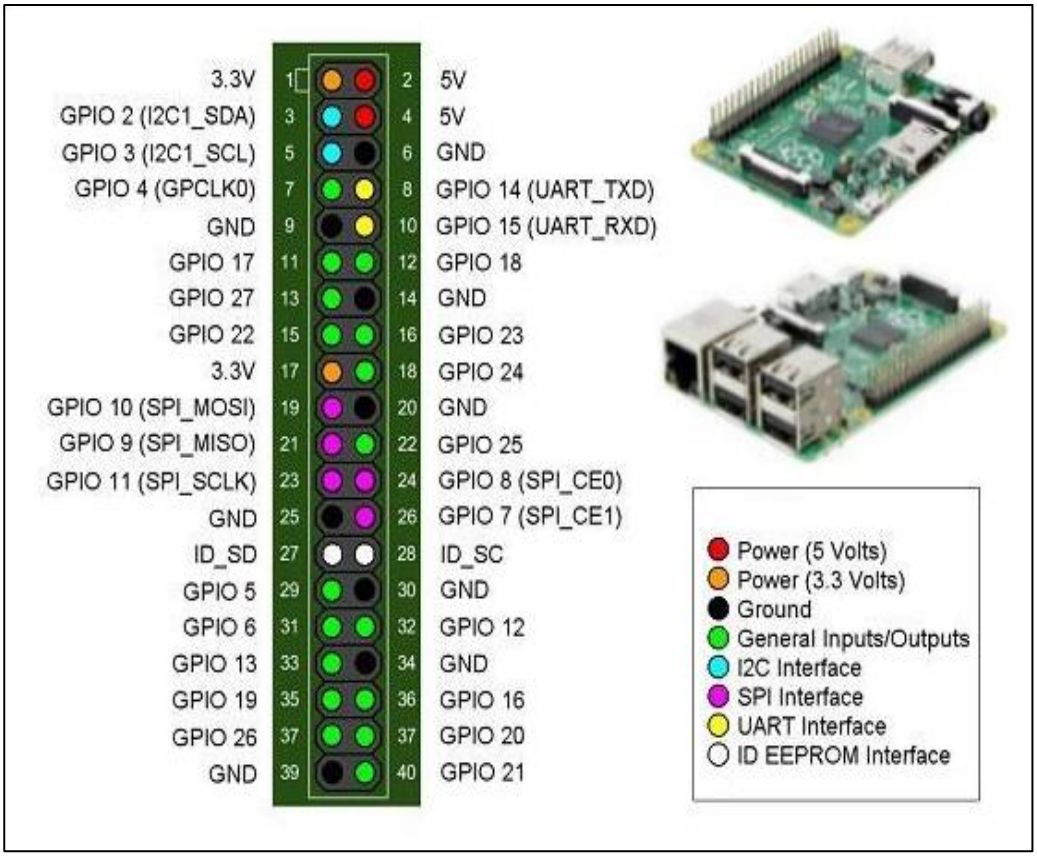

Figura 2.8: Diagrama de pines de Raspberry pi.

Conexiones del motor para el control del movimiento horizontal (pan):

Pin No.	GPIO No.
29	5
31	6
32	12
33	13

Tabla 2: Conexiones del motor A.

Conexiones del motor para el control del movimiento vertical (tilt):

Pin No.	GPIO No.
11	17
12	18
13	27
15	22

Tabla 3: Conexiones del motor B.

El circuito integrado ULN2003APG también cumple otra función, la de invertir la señal de entrada. Como podemos ver en la Figura 2.9, internamente está compuesto por inversores y diodos Darlington. Esto es de utilidad debido a que todas las bobinas del motor de paso tienen una conexión en común a 5 voltios. Para realizar la activación individual de cada bobina se requiere que el nivel de voltaje sea de 0 voltios en la bobina que se desea activar y que las otras tres bobinas tengan 5 voltios para que no exista circulación de corriente. Para que un pin de salida del circuito integrado ULN2003APG sea de 0 voltios, la conexión de entrada relacionada a esa salida tiene que ser de un nivel lógico alto, es decir, 5 voltios en este caso.

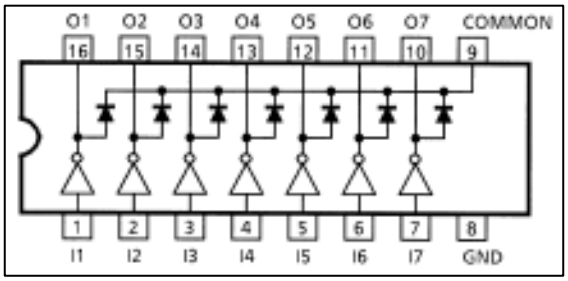

Figura 2.9: Diagrama del driver ULN2003APG.

2.8 Desarrollo del software de control

El programa consta de 2 partes, la primera parte se enfoca en el control de los motores de paso; para esto se ha desarrollado programas en lenguaje Python que permiten girar los dos motores que se usan para este proyecto. La otra parte del software es el desarrollo de un programa que permita visualizar y controlar remotamente sistema embebido.

2.8.1 Desarrollo de los scripts de control en Python

Python es un lenguaje de código abierto que es muy fácil de aprender, es una buena alternativa cuando se empieza en el mundo de la programación por que no se necesita experiencia previa para aprender este lenguaje. Existen muchos programas o módulos que han sido desarrollados por parte de la comunidad de Python para realizar diversas funciones, debido a esto, desarrollar programas en este lenguaje es rápido y fácil [17]. Una gran ventaja al usar el Raspberry pi como dispositivo embebido y su sistema operativo Raspbian Weezy, es que tiene instalado por defecto Python, por lo que no es necesario descargar, instalar y configurar paquetes adicionales.

El programa que controla los motores está dividido en 4 scripts (módulos), dos módulos para cada motor. Se referirá al motor que controla el movimiento horizontal: motor A; el motor que controla el movimiento vertical será: motor B. Los scripts para mover los motores quedarían en el orden que se muestra en la Tabla 4.

Archivo	Motor
stepper-ntclkws.py	Motor A
stepper-clkws.py	Motor A
stepper-up.py	Motor B
stepper-down.py	Motor B

Tabla 4: Módulos de control.

Python posee una modulo llamado RPIO.GPIO, las clases de este paquete permiten trabajar con los puertos de entrada y salida GPIO (General Purpouse Input Output) [18]. Se debe tener en cuenta que esta librería no puede ser usada para aplicaciones en tiempo real o aplicaciones que tengan un estricto control del tiempo.

Los scripts que se usan para controlar los motores tienen los siguientes elementos en común:

- Declaración de los módulos.

  ```
  import sys

  import time

  import RPi.GPIO as GPIO

  from select import select
  ```

- Configuración del modo en que se referencian los pines.

  ```
  GPIO.setmode(GPIO.BCM)
  ```

- Configuración de los puertos GPIO a usar.

  ```
  StepPins = [5,6,12,13]

  for pin in StepPins:

  print "Setup pins"

  GPIO.setup(pin,GPIO.OUT)

  GPIO.output(pin, False)
  ```

- Inicialización de la matriz de secuencia.

  ```
  Seq = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]]
  ```

- Inicialización de la variable de dirección del giro.

  ```
  StepDir = 1 # Set to 1 for clockwise
  ```

```
# Set to -1 for anti-clockwise
```

- Configuración del tiempo de espera entre pasos.

```
if len(sys.argv)>1:
  WaitTime = int(sys.argv[1])/float(1000)
else:
  WaitTime = 10/float(1000)
```

- Hilo donde se ejecutan las secuencias infinitamente hasta que se detenga el proceso.

```
for i in range(0, 50):
 try:
  for pin in range(0, 4):
   xpin = StepPins[pin]
   if Seq[StepCounter][pin]!=0:
     GPIO.output(xpin, True)
   else:
     GPIO.output(xpin, False)
  StepCounter += StepDir
  # If we reach the end of the sequence
  # start again
  if (StepCounter>=StepCount):
   StepCounter = 0
  if (StepCounter<0):
```

```
StepCounter = StepCount -1
```

- La función cleanup() devuelve el estado a entrada a todos los puertos que han sido configurados como salida durante la ejecución del programa.

```
GPIO.cleanup()
```

Existen dos maneras para hacer referencia a los pines de Raspberry pi, usando la librería RPIO.GPIO: GPIO.BOARD y GPIO.BCM. En la primera librería se refiere a los pines del Raspberry pi por el número del pin en el conector de 40 pines; usando la segunda librería se refiere a los pines del Raspberry pi mediante el “Broadcom SOC Channel - BCM”, que es el numero definido por el sistema [19].

Para elegir los pines a configurar como salidas que van conectadas a los motores de paso, hay que cambiar los valores de la matriz SetPins Los valores para el motor A y B son los siguientes:

```
Motor A          SetPins = [17,18,27,22]

Motor B          SetPins = [5,6,12,13]
```

La secuencia de giro que se usa para cada motor es diferente. El motor B que controla el movimiento vertical necesita un mayor torque debido principalmente al peso de la camara. Debido a esto se usará la configuración Half-Step, quiere decir que tendrá una menor velocidad pero con un torque mayor. La configuración Full-Step la usará en el motor A porque no se necesita un torque considerable para mover la camara en el eje horizontal, además de que el movimiento horizontal debe ser rápido en comparación con el movimiento vertical [20].

La secuencia del movimiento para cada motor está en la matriz Seq.

```
Motor A Seq = [[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]]

Motor B  Seq=[[1,0,0,0],[1,1,0,0],[0,1,0,0],[0,1,1,0],[0,0,1,0],[0,0,1,1],
```

[0,0,0,1],[1,0,0,1]]

Para realizar el movimiento en sentido contrario del motor A y B, hay que cambiar los valores de la variable StepDir. Los valores de esta variable pueden ser 1 y -1. Para los scripts stepper-antclkws.py y stepper-up.py el valor de la variable será -1, quiere decir movimiento en sentido contrario a las manecillas del reloj y movimiento vertical hacia arriba de la camara. Para los scripts stepper-clkws.py y stepper-down.py el valor de la variable StepDir será de 1.

2.8.2 Desarrollo de la interface para la monitorización y control

Una de las partes importantes en el desarrollo de un sistema de monitorización es la interfaz entre el usuario y el dispositivo embebido. El usuario no controla directamente los motores que son la base del mecanismo de movimiento de la camara, tampoco accede directamente a los datos de la camara conectada a Raspberry pi. El usuario envía señales al sistema embebido y este los traducirá a instrucciones que serán ejecutadas una vez que hayan sido validadas. Por lo tanto, la interfaz debe tener todos controles para que el usuario pueda enviar las señales al dispositivo embebido y este a su vez entregue la información que corresponda al usuario.

Se ha desarrollado una aplicación web alojada en el dispositivo embebido. Esta aplicación web consta de dos partes principales: interface web y scripts php. El desarrollo de la interfaz fue realizado usando la terminal del Raspberry pi, mediante el editor de textos nano que viene instalado por defecto en el sistema operativo. La Figura 2.10 muestra la interfaz que se ve el usuario.

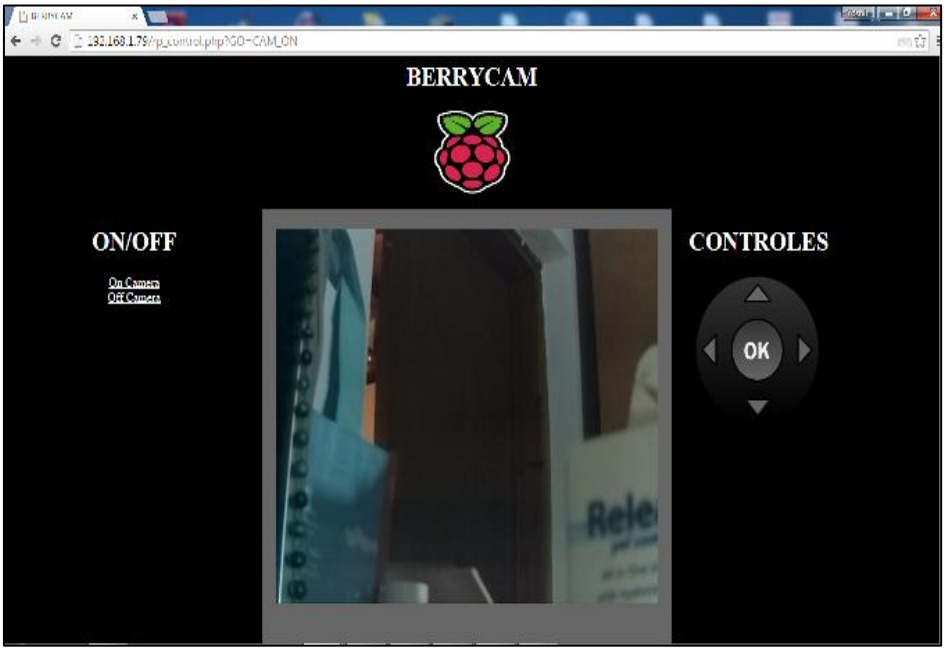

Figura 2.10: Interfaz de usuario.

La otra parte que compone la interfaz es el script escrito en lenguaje de programación PHP, este programa ejecuta una tarea cada vez que el usuario presiona un botón en la interfaz. El script es llamado rp_control.php y realiza las siguientes tareas:

- Ejecutar las rutinas para encender y apagar la camara.
- Ejecutar las rutinas para mover la camara en el eje vertical y horizontal.

En este punto cabe recalcar que el servidor web apache necesita ser instalado en el Raspberry pi con los permisos de seguridad correspondientes para que pueda ejecutar los scripts escritos en Python usando el comando sudo.

En la Figura 2.11 se presenta una parte del código que realiza la tarea de llamar a los diferentes procedimientos escritos en Python.

```
        }
print_r($_GET);
switch ($action)
{

        case "CAM_ON":
                $message= shell_exec('sudo ./run-mjpgstreamer.sh');
                break;
        case "CAM_OFF":
                $message = shell_exec('sudo killall mjpg_streamer');
                break;
        case "ARRIBA":
                exec('sudo python stepper-down.py 10 > /dev/null &');
                break;
        case "ABAJO":
                $message = shell_exec('sudo python stepper-up.py 10 > /dev/null &');
                break;
        case "IZQ":
                $message = shell_exec('sudo python stepper-antclkws.py 10 > /dev/null &');
                break;
        case "DER":
                $message = shell_exec('sudo python stepper-clkws.py 10 > /dev/null &');
                break;
        case "RELEASE":
                exec('sudo killall python');
                break;
}
```

Figura 2.11: Procedimientos para controlar la camara.

La aplicación web está alojada en el servidor apache que fue instalado previamente. El directorio por defecto es /var/www. Para poder crear, copiar y editar archivos en este directorio hay que tener privilegios de súper usuario. Para acceder a la aplicación habrá que ejecutar cualquier navegador web y escribimos la dirección IP de Raspberry pi haciendo referencia también al archivo html donde se ha diseñado la interfaz de usuario. Por ejemplo 192.168.1.10/control.html, siendo control.html el archivo de la interfaz.

2.9 Desarrollo de la plataforma giratoria

Para diseñar una camara usando un dispositivo embebido, como lo es Raspberry pi, los materiales que deben escogerse y la estructura en donde va a instalarse el sistema deben ser resistentes al ambiente hostil, como un clima muy húmedo o muy caluroso. Por ejemplo, este sistema podría ser instalado en la intemperie donde la lluvia o la luz del sol pudieran causar daños en los componentes del sistema embebido y también de sus periféricos.

La base de la estructura y el soporte de la camara son de acero inoxidable como muestra la Figura 2.12. Entre el soporte de la camara y la base se ha instalado una pequeña base circular de nylon la cual cumple la función de estabilizar el soporte de la camara y de conexión del motor A, para que pueda realizar el giro horizontal. Es importante destacar que gracias al nylon existirá un menor rozamiento entre la base y el soporte de la camara lo cual permite que el motor B trabaje con una carga menor de torque. A continuación se presenta mediante imágenes las diferentes etapas de la construcción de la plataforma giratoria.

El motor A esta fijado mediante tornillos a la base de acero inoxidable. Esta base de acero posee un pequeño orificio que sirve para que el acople del motor pueda ser adherido con el nylon que estabiliza la camara.

Figura 2.12: Motor A.

El material que sirve de contenedor para el sistema embebido y los periféricos de entrada y salida es acrílico transparente de 3 milímetros de grosor. La Figura 2.13 muestra la maquina donde se realizaron los cortes del acrílico. Para proceder, primero se debe realizar un diseño de los cortes que se harán en el material. El resultado de este diseño es un archivo que puede ser interpretado por la maquina cortadora. Cabe destacar que el error en los cortes fue de un margen muy reducido, casi imperceptible.

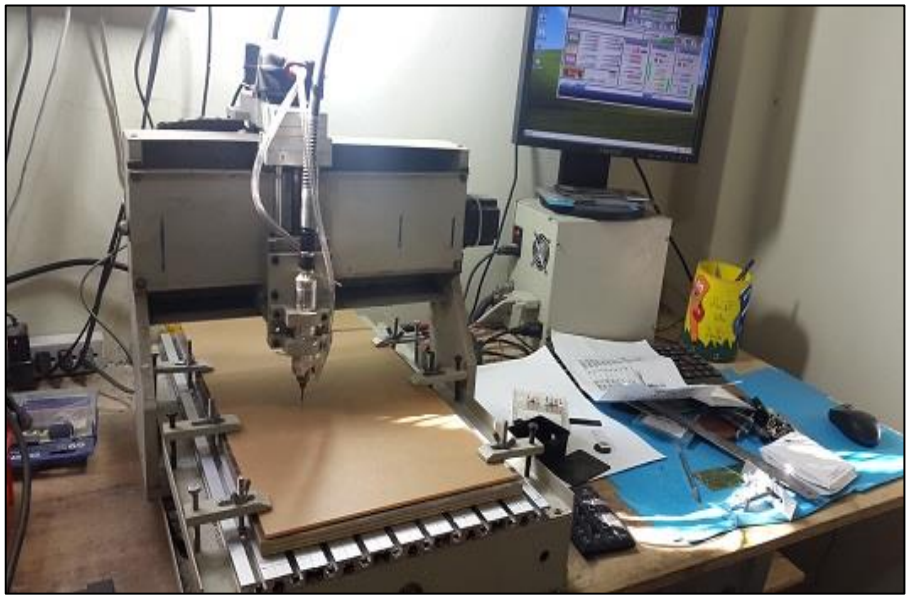

Figura 2.13: Maquina cortadora.

Una vez terminado el proceso de corte, el siguiente paso es ensamblar las partes usando pegamento acrílico, este es un pegamento muy fuerte lo que proporciona una resistencia y durabilidad a la caja donde van a estar instalados los elementos. Como muestra la Figura 2.14, los elementos quedan en orden dentro de la caja de acrílico.

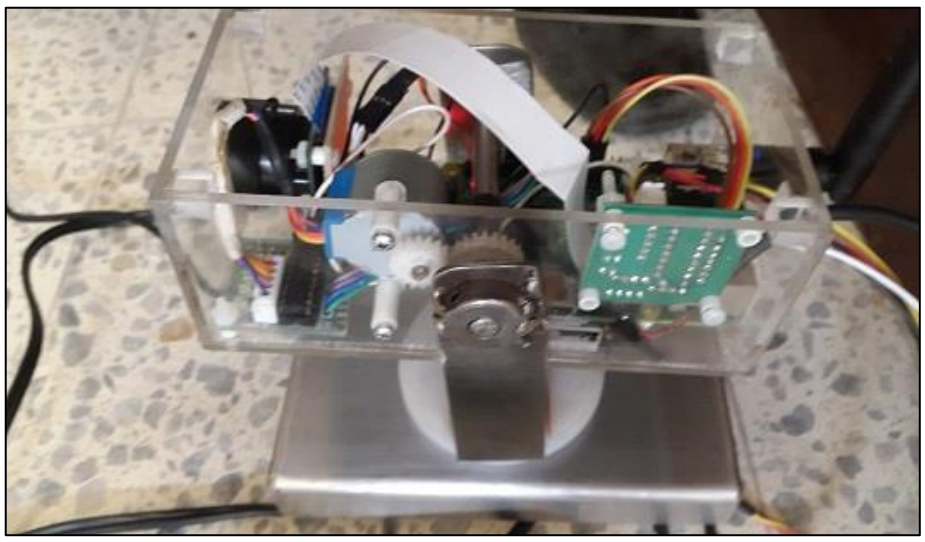

Figura 2.14: Ensamble de elementos.

CAPÍTULO 3

3. Resultados obtenidos y análisis

3.1 Introducción del capítulo

El uso del Raspberry pi como sistema embebido ofrece muchas ventajas en comparación con las cámaras que se encuentran en el mercado. Por ejemplo, el usuario puede cambiar la capacidad de almacenamiento de la camara de acuerdo a sus necesidades. Existen 4 puertos usb los cuales pueden ser conectados a dispositivos de almacenamiento masivo o almacenamiento de la tarjeta microsd.

El producto final que se aprecia en la figura 2.18 ha sido desarrollado con materiales reciclados, por ejemplo los engranes del motor que produce el movimiento vertical y el eje de este mismo fueron obtenidos de una impresora que no funcionaba. El acero inoxidable fue obtenido en talleres de metalurgia que tenían piezas pequeñas de este material y no se le podía dar un uso específico.

Se ha tratado de reducir al máximo los costos de producción para que una cámara con estas características sea accesible al usuario y también para competir en el mercado de las cámaras IP. El enfoque de esta cámara es que sea usada para mejorar la vida de la comunidad de productores agrícolas.

3.2 Análisis de las características de transmisión de video

Una característica importante de las cámaras IP es la velocidad de transmisión de los datos. En lo posible, se requiere que el usuario pueda ver la transmisión en cualquier lugar con el menor tiempo de retardo posible. Es aquí donde la rapidez de la aplicación Mjpg Streamer Experimental muestra sus ventajas.

Se ha realizado una comparación con otras aplicaciones que pueden ser instaladas en Raspberry pi y que se usan para transmitir video vía internet. Las aplicaciones son las siguientes:

- Mjpg Streamer.
- Gstreamer.

- Netcad.
- FFmpeg
- VLC streaming.
- Mjpg Streamer Experimental.

Mjpg Streamer Experimental, ha resultado ser la opción con menos latencia en la transmisión de los datos. Siendo todas las aplicaciones probadas bajo el mismo ambiente. El video obtenido tiene una resolución de 320x240 pixeles y a una frecuencia de 15 frames por segundo lo que resulta en una imagen nítida pero que no es de mucha ayuda cuando se desea ver detalles de la zona donde se va a realizar la monitorización.

En el caso de las plantaciones agrícolas muchas veces se usan las cámaras IP de control remoto para poder realizar procesamiento de las imágenes con programas avanzados donde se usan imágenes de alta resolución. Afortunadamente Mjpg Streamer Experimental posee opciones para obtener imágenes en alta resolución, sin embargo hay que tomar en cuenta que mientras el video tenga más resolución mayor será los datos que se van a transmitir y esto puede ser un problema si la red de datos donde está instalado el sistema no tiene un ancho de banda considerable, dando como resultado mayor tiempo de retardo en la transmisión de video.

Hay que tomar en cuenta también que los recursos del Raspberry pi son limitados en cuanto a la cantidad de imágenes por segundo que el sistema embebido podrá procesar debido a la cantidad reducida de memoria RAM y velocidad del CPU.

3.3 Análisis mecánico de la plataforma giratoria

Se puede considerar a la plataforma giratoria, como una plataforma robusta y apta para funcionar en ambientes con mucha humedad, como suelen ser las plantaciones agrícolas. La base del soporte de la cámara sufrió cambios en la etapa de construcción, Figura 3.1. Las pruebas que se realizaron demostraron que no ofrecía una buena estabilidad y ocasionaba que el eje del motor soporte el peso de estructura. Esto podía resultar en daños del motor, dejando de proporcionar el torque suficiente para realizar el movimiento horizontal.

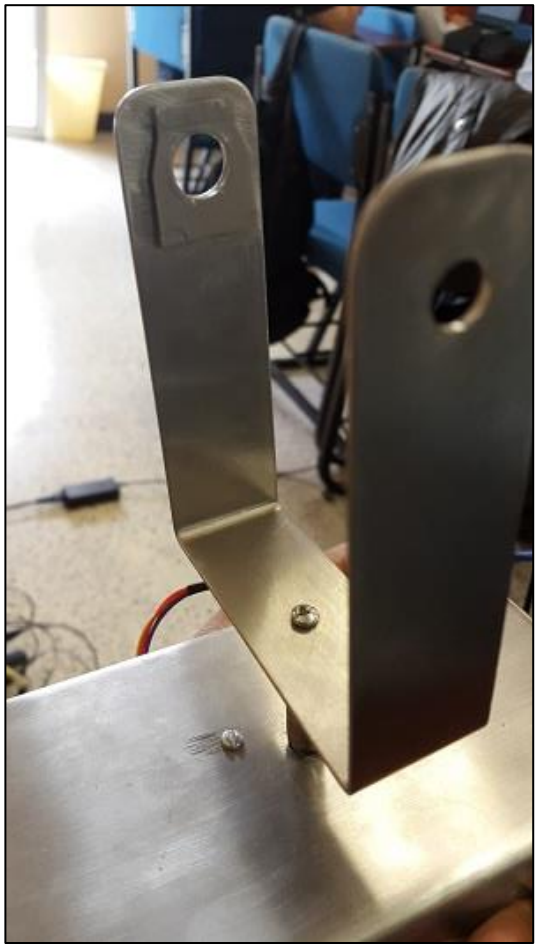

Figura 3.1: Soporte de la camara.

Como se puede ver en la Figura 3.2, la nueva base fue diseñado en nylon, un material que produce poca fricción al rozar el acero inoxidable, esto proporcionó estabilidad al soporte de la cámara. El nuevo material permitió que el movimiento horizontal se realice con mayor velocidad en comparación con la base anterior y que el peso de la cámara más la estructura de acero se distribuya en la base para que el motor no sufra avería.

Figura 3.2: Base circular de nylon.

3.4 Análisis de las características de la camara.

Gracias a los cortes precisos de la maquina cortadora computarizada, el ensamblaje y ordenamiento de los elementos se realizó sin mayores complicaciones. Los elementos y las conexiones están dispuestos de tal manera que se ahorra espacio y la versión final de la cámara IP es de menor dimensión que el prototipo. La capacidad de la camara del Raspberry pi es de 2592x1944 pixeles para tomar fotos y para reproducir videos es de 1080p/30fps, 720p/60fps y 640x480/60-90fps, por esta razón es necesario la configuración del dispositivo cuando se va a realizar el streaming de video. Los engranes del motor y del eje que permiten realizar el movimiento vertical son lo suficientemente fuertes para resistir el peso de la cámara y mantener el ángulo de inclinación, aunque hayan sido fabricados en plástico.

Como muestra la Figura 3.3 la ubicación de la camara tiene un acabado preciso que no se lograría sin el uso de herramientas de corte profesionales. Aunque hay que tomar en cuenta que el uso de estas herramientas incide en el precio final del producto.

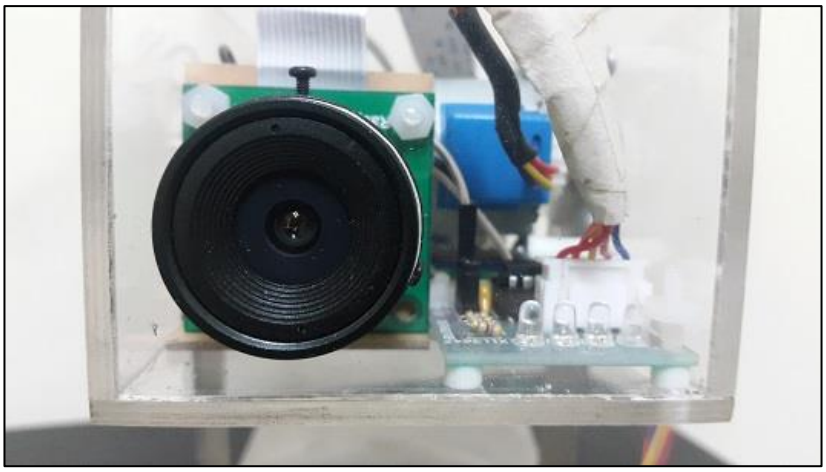

Figura 3.3: Vista de la camara.

El resultado final como se puede apreciar en la Figura 3.4 es una camara robusta, que cumple con las funciones especificadas a un costo mucho menor que las cámaras que generalmente encontramos en el mercado. Al usar el Raspberry pi como un sistema embebido podemos usar los periféricos de salida para tener mayor almacenamiento de datos.

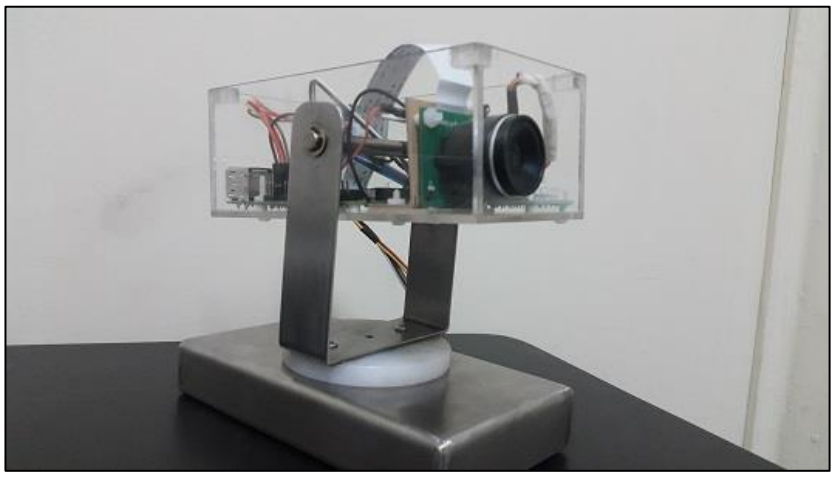

Figura 3.4: Producto final.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

1. Raspberry pi es una herramienta muy útil cuando se trata de tener un sistema embebido, su bajo costo lo hace muy accesible para proyectos donde se necesite procesar datos, controlar motores, encender luces y almacenamiento de datos. Aunque existen en el mercado módulos como Arduino que pueden funcionar como sistemas embebidos para realizar tareas sencillas, se necesita comprar varios elementos para tener un sistema que pueda ser usado como servidor que aloje una página web y que a su vez sirva para realizar streaming de video y controlar motores de paso. En este proyecto fue una ventaja tener en una sola placa todos los elementos que se necesitaban, como consecuencia fue más fácil de implementar.

2. La implementación de una plataforma giratoria para la inspección de áreas agrícolas puede tener un buen impacto en la vida de los agricultores, debido a que gastarían menos recursos trasladándose a sus fincas para evaluar el crecimiento y la seguridad de los cultivos.

3. El uso de materiales reciclados como los engranes de impresoras y piezas de acero desechadas contribuyó en el bajo costo del producto final.

4. Aunque este trabajo se centra en resolver la problemática de monitoreo remoto con una plataforma giratoria en el sector agrícola, este mismo principio podría aplicarse a la monitorización del área ganadera, acuícola, control de incendios forestales, observación de aves y demás áreas relacionadas.

5. En este proyecto se encontró que la mejor opción para realizar streaming de video es Mjpg Streamer, debido a que en las pruebas realizadas presentaba la menor latencia durante la transmisión de video.

Recomendaciones

1. Después de la construcción de este proyecto se puede apreciar que los recursos del Raspberry pi no están siendo usados completamente. Se puede usar el Raspberry pi para el almacenamiento y procesamiento de datos de los cultivos. Mediante el uso de sensores se puede obtener información más detalla

sobre el área agrícola y esa información puede ser útil al agricultor para que la producción de sus cultivos sea más eficiente.

2. El producto final como se puede ver en la figura 3.3 no es impermeable, por lo tanto, no se puede instalar en un área agrícola que está expuesta a la lluvia o a maquinaria de riego.
3. Este proyecto no cuenta con visión nocturna, por lo que para realizar la monitorización nocturna es necesario contar con una fuente de iluminación externa.
4. Esta cámara no cuenta con módulos de detección de movimiento, pero en Raspberry pi existe la posibilidad de instalar librerías de Opencv especializadas en procesamiento de imágenes y de esta manera aprovechar mejor el potencial que nos ofrece este sistema embebido.

.

BIBLIOGRAFÍA

[1] Boylestad y Nashelsky, Teoría de Circuitos y Dispositivos Electrónicos, Prentice Hall, 2009.

[2] Angelfire, El principio de funcionamiento del PLL, http://www.angelfire.com/al3/PLL/pllfunc.html, fecha de consulta Julio 2015.

[3] Blog Tecnos, streaming en informática definición, http://blogtecnos.blogspot.fr/2010/12/streaming.html, fecha de consulta Julio 2015.

[4] Asociación para el progreso de las comunidades, software propietario, https://www.apc.org/es/glossary/term/241, fecha de consulta Julio 2015.

[5] D-link, Videovigilancia contra robos del campo, http://www.videovigilanciadlink.es/Blog/Aplicaciones/Videovigilancia-contra-los-robos-del-campo.html, fecha de consulta Agosto 2015.

[6] Sergiy Bogdancev, Getting video stream from USB web-camera on Arduino Due Part 1, http://www.codeproject.com/Articles/863938/Getting-video-stream-from-USB-web-camera-on-Arduin, fecha de consulta Septiembre 2015.

[7] Atmel, SAM3X/A SMART ARM-based MCU datasheet, http://www.atmel.com/Images/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf, fecha de consulta Septiembre 2015.

[8] Sergiy Bogdancev, Getting video stream from USB web-camera on Arduino Due Part 8, http://www.codeproject.com/Articles/1012944/Getting-Video-Stream-from-USB-Web-camera-on-Ardu, fecha de consulta Septiembre 2015.

[9] Raspberri pi, Raspberri Pi 2 model b, https://www.raspberrypi.org/products/raspberry-pi-2-model-b/, fecha de consulta Septiembre 2015.

[10] Google, mjpg-streamer, https://code.google.com/p/mjpg-streamer/, fecha de consulta Septiembre 2015.

[11] Ronald Fiala, Raspberry pi con robot en tiempo real a través de una web camara con mjpg-streamer, http://www.robosphere.de/raspberry-pi-roboter-mit-echtzeit-webcam-via-mjpg-streamer/#more-660, fecha de consulta Septiembre 2015.

[][12] BLT Robotics, Better Living Through Robotics, http://www.bltrobotics.com/#, fecha de consulta Septiembre 2015.

[13] Apache, HTTP Server Project, https://httpd.apache.org/ABOUT_APACHE.html, fecha de consulta Agosto 2015.

[14] Raspberry pi Foundation, Camera module, https://www.raspberrypi.org/products/camera-module/, fecha de consulta Junio 2015.

[15] GitHub, mjpg-streamer-experimental, https://github.com/jacksonliam/mjpg-streamer/tree/master/mjpg-streamer-experimental, fecha de consulta Agosto 2015.

[16] Instructables, BYJ48 stepper motor, http://www.instructables.com/id/BYJ48-Stepper-Motor/, fecha de consulta agosto 2015.

[17] Python, About, https://www.python.org/about/, fecha de consulta Agosto 2015.

[18] Python Software Foundation, RPi GPIO 5.11, https://pypi.python.org/pypi/RPi.GPIO, fecha de consulta Agosto 2015.

[19] State Exchange, What is the difference between BOARD and BCM for GPIO pin numbering?, http://raspberrypi.stackexchange.com/questions/12966/what-is-the-difference-between-board-and-bcm-for-gpio-pin-numbering, fecha de consulta Agosto 2015.

[20] Anaheim Automation, Stepper Motor: Full step vs. Half step excitation, https://www.youtube.com/watch?v=dmk6zIkj7WM, fecha de consulta Agosto 2015.