Red avanzada de comunicaciones mediante Raspberry Pi para aplicaciones en vehículos aéreos no tripulados

Luis Fernando Hernández Medina

Universidad Autónoma de San Luis Potosí, Lateral Av. Salvador Nava S/N., Col. Lomas. C.P.78290, San Luis Potosí, México

fmedina9008@gmail.com

Marco Aurelio Cárdenas Juárez

Universidad Autónoma de San Luis Potosí, Lateral Av. Salvador Nava S/N., Col. Lomas. C.P.78290, San Luis Potosí, México

cardenas.marco@gmail.com

Enrique Stevens Navarro

Universidad Autónoma de San Luis Potosí, Lateral Av. Salvador Nava S/N., Col. Lomas. C.P.78290, San Luis Potosí, México

enrique.stevens.navarro@gmail.com

Armando Arce Casas

Universidad Autónoma de San Luis Potosí, Lateral Av. Salvador Nava S/N., Col. Lomas. C.P.78290, San Luis Potosí, México

arcecasas@gmail.com

Ulises Pineda Rico

Universidad Autónoma de San Luis Potosí, Lateral Av. Salvador Nava S/N., Col. Lomas. C.P.78290, San Luis Potosí, México

u.pinedarico@gmail.com

Resumen

En este artículo se desarrolló un prototipo para obtener acceso a la información que recopilen un conjunto de tarjetas Raspberry Pi mediante un *script*, y que además puede ser revisada remotamente en una estación base mediante el uso de software. Los resultados arrojaron que es posible crear el escenario de desastre, o incluso recrear un mapa a partir de las imágenes recopiladas. El trabajo a futuro plantea el uso de éstas tarjetas en vehículos aéreos no tripulados en dos modalidades: vehículo a vehículo (vehicle to vehicle, V2V) y vehículo a base (vehicle to base, V2B), de modo que las tarjetas trabajen como enjambre compartiendo la misma información para su uso en tareas de protección civil; realizando un análisis de las imágenes sería posible incluso encontrar a personas en peligro.

Palabra(s) Clave(s): bluetooth, raspberry, UAV, V2V, V2B.

1. Introducción

El interés en los vehículos aéreos no tripulados (UAV, por sus siglas en inglés) se ha incrementado en los años recientes debido, en parte, a la facilidad para adquirirlos y las oportunidades que presentan en términos de investigación y desarrollo tecnológico, pues quedan muchos problemas abiertos por resolver. Por ejemplo, las capacidades de dichos vehículos para mantener un vuelo estable son limitadas y es necesario hacer uso de otros dispositivos para obtener información adicional como la calidad del aire, video, foto, etc. Además, otro de los problemas de gran interés en la actualidad es el que impone la necesidad de comunicación entre UAVs por medio de una plataforma confiable y de bajo costo.

Un UAV es esencialmente una aeronave sin piloto a bordo que puede ser controlados ya sea remotamente por un piloto en tierra en una estación base o, alternativamente, pueden ser autónomos para llevar a cabo una misión previamente planeada o tener un sistema más complejo de automatización dinámica. Los UAVs son actualmente

utilizados para un sinnúmero de aplicaciones que incluyen: reconocimiento, entretenimiento, control de desastres, tareas de protección civil, entre muchos otros.

Existen diferentes tipos de UAVs, y se pueden clasificar de la siguiente manera [1]:

- Objetivo y señuelo: Proporcionan un blanco en tierra o en aire que simula un avión enemigo o un misil.
- Reconocimiento: Utilizados en misiones para proporcionar inteligencia del campo de batalla o un área determinada.
- Combate: Tienen una capacidad de ataque para misiones de alto riesgo.
- Investigación y desarrollo: Utilizados para desarrollar tecnologías UAV o que se integren en los aviones UAV.
- Comerciales y de uso civil: UAVs específicamente diseñados para aplicaciones civiles y comerciales.

Las características de los vehículos no tripulados los hacen perfectos para misiones donde el acceso podría ser un problema para el ser humano y si se maneja una red de sensores entre estos dispositivos las posibilidades de exploración aumentan considerablemente. De este modo, el concepto de “enjambre” o flotilla de vehículos no tripulados permite abordar problemas que serían casi imposibles para un solo vehículo [2]. Sin embargo, es necesario establecer una comunicación eficiente y confiable entre ellos para no perder la información que hayan recopilado. En este contexto, la tecnología Bluetooth se perfila como la solución a este problema de comunicación pues ofrece la posibilidad de crear pequeñas redes inalámbricas entre equipos móviles.

Dentro de estas tendencias, el objetivo de este trabajo persigue el desarrollo de un prototipo que extienda las capacidades individuales de los vehículos no tripulados, aplicando conceptos de comunicaciones para redes de sensores que serán manejados por las tarjetas de desarrollo Raspberry Pi y las cuales, a través de Bluetooth,

establecerán una comunicación entre los diversos vehículos no tripulados a fin de formar una red de comunicación que será monitoreada remotamente.

En diversos proyectos de Raspebrry Pi [3, 4, 5] es posible encontrar un uso con cámara que mayoritariamente tiene que ver con video vigilancia y seguridad doméstica, lo cual permite visualizarlos en tiempo real. A lo anterior se suman aquellos proyectos que realizan capturas de imágenes a ciertos intervalos de tiempo [6] ya sea con propósitos artísticos o de investigación. Finalmente, existen proyectos que tratan de unir a los UAVs con las tarjetas Raspberry Pi en un solo dispositivo, ya sea con fines de exploración [7] o de protección civil, como lo es el proyecto OpenRelief [8] que haciendo uso de esta tarjeta busca desarrollar mejores herramientas de comunicación en conjunto con DroneCode.

2. Desarrollo

El desarrollo del prototipo plantea primeramente la instalación de software en la tarjeta Raspberry Pi que mediante una cámara web tome fotografías en alta resolución y las transmita mediante Bluetooth a otros dispositivos (Ver Figura 1). Además, las imágenes pueden ser monitoreadas remotamente en una estación base donde se pueden adquirir aquellas que reporten una mayor cantidad de información.

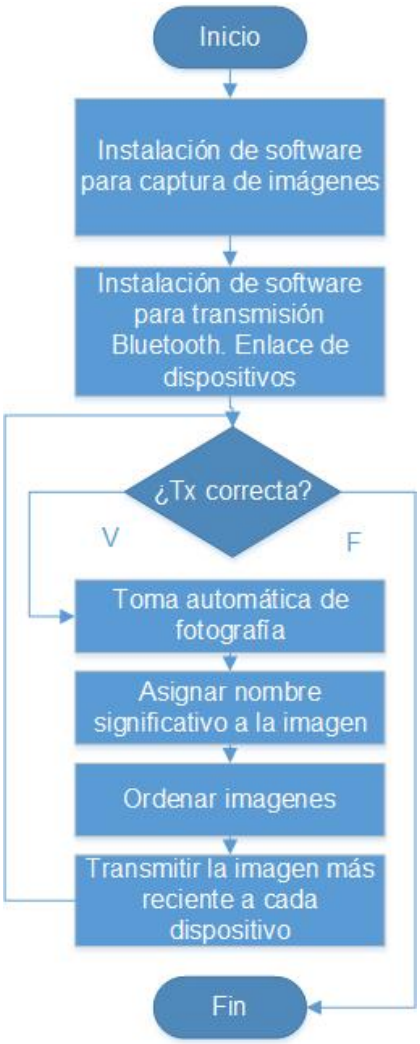

Fig. 1. Cuando se han instalado los programas en la tarjeta, el Script trabajará por su cuenta para enviar la información a los dispositivos si la Tx es exitosa, en caso contrario se asume que el dispositivo no está disponible y da por terminada la Tx a ese dispositivo.

2.1. Objetivo

Se busca hacer uso de la tarjeta Raspberry Pi como una herramienta de transmisión de modo que se pueda montar el prototipo en naves no tripuladas para obtener imágenes y con base en ellas reconstruir el escenario objetivo interior o exterior (bodega, oficina, campo, etc.), creando con ello un mapa actualizado de la zona de desastre para poder realizar un plan de contingencia como se muestra en la Figura 2. Mediante el análisis

de las imágenes es posible incluso encontrar personas que se encuentren atrapadas en alguna zona de desastre y proceder a su rescate.

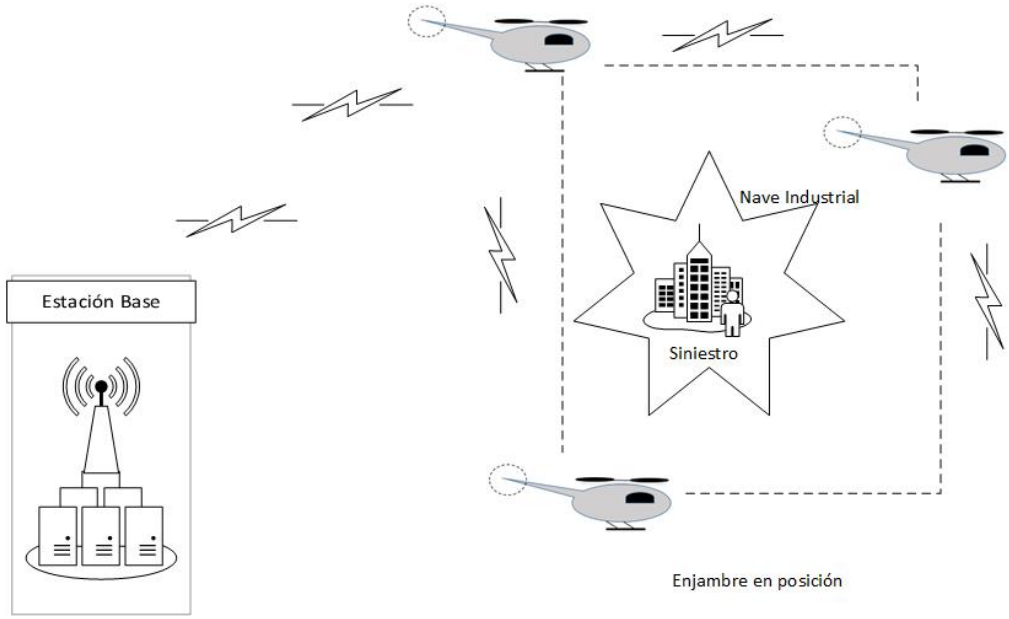

Fig. 2. Al ocurrir un siniestro, los mUAVs entran al recinto con una ruta establecida y comienzan a enviar información entre ellos y la estación base.

2.2. Metodología paso a paso del prototipo

El primer reto consiste en la instalación de un software que a través de una cámara web capture imágenes en alta definición. Para esta tarea ya se han desarrollado diferentes programas, sin embargo, el más sencillo es *Fswebcam* pues no requiere de mucho espacio en la tarjeta.

Lo primero es descargar, instalar y probar el programa mediante el uso de los siguientes comandos:

```
sudo apt-get install fswebcam

fswebcam –r 960x720 –d /dev/video0 imagen1.jpg
```

A continuación se procede a hacer uso de la herramienta *Crontab*, la cual nos permite guardar una serie de instrucciones para que se ejecuten cada determinado intervalo de

tiempo. Lo primero es revisar que no existan rutinas previas mediante el siguiente comando:

```
crontab –l
```

Si no hay rutinas se podrá leer la leyenda *no crontab for pi*, de modo que es posible modificarlo para que capture una imagen cada 5 minutos con la calidad deseada escribiendo el siguiente comando:

```
crontab –e
```

Con lo cual se abrirá el Script en el que la línea final del mismo contiene el texto *#m h dom mon dow command*, donde *m* indica minutos, *h* horas, *dom* día del mes, *mon* mes del año y *dow* día de la semana. Teniendo esto en mente, se puede configurar la captura automática de la siguiente forma:

```
*/5 * * * * fswebcam –r 960x720 –d /dev/video0 /home/pi/imagen2.jpg
```

A partir de este momento la cámara está sobrescribiendo el archivo de imagen cada 5 minutos a una resolución de 960x720.

La siguiente parte del proyecto consiste en instalar un módulo Bluetooth en la tarjeta Raspberry Pi, pues como se mencionó anteriormente se hará uso del mismo para transmitir las imágenes a otros dispositivos y así dotar a las tarjetas con la funcionalidad de comunicación inalámbrica [9]. Para este fin, primero se descarga el programa Bluetooth evitando instalar controladores (*drivers*) de impresoras innecesarios para esta aplicación en particular con el fin de ahorrar espacio. El comando de instalación es como se indica a continuación:

```
sudo apt-get install --no-install-recommends bluetooth
```

Después de esto se realiza la instalación del GUI para Bluetooth de la siguiente manera:

```
sudo apt-get install blues-utils blueman
```

Además, *ObexFTP* es usado para brindar acceso a los archivos en equipos móviles [10], por lo cual resulta muy útil en la tarea de transferencia de archivos entre los dispositivos. Este programa puede instalarse de la siguiente forma:

```
sudo apt-get install obexftp

sudo reboot
```

Una vez reiniciada la tarjeta podemos verificar que el dispositivo USB-Bluetooth sea detectado correctamente mediante la ejecución de la instrucción *lsusb* como se muestra en la Figura 3.

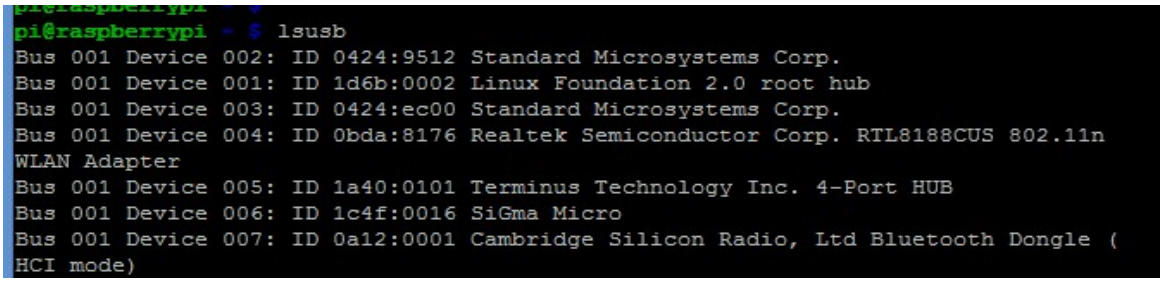

Fig. 3. Verificación del módulo Bluetooth.

Después de que se haya identificado el módulo Bluetooth buscaremos su dirección MAC mediante el comando:

```
hcitool dev
```

En una primera instancia, como fase de pruebas se establece un enlace entre un celular con sistema operativo Android y la tarjeta Raspberry Pi. Para esto, primeramente se debe de verificar que el dispositivo móvil se encuentre *visible* en la conexión Bluetooth. Si se desea omitir este paso y realizar directamente una comunicación entre tarjetas, primero se debe hacer visible el módulo Bluetooth mediante la instrucción

```
hciconfig hci0 piscan
```

Cuando ya se tiene la certeza de que el dispositivo está visible, se procede a escanearlo con la tarjeta con la instrucción:

```
hcitool scan
```

Este comando devolverá el nombre y la dirección MAC de todos los dispositivos Bluetooth como se muestra en la Figura 4.

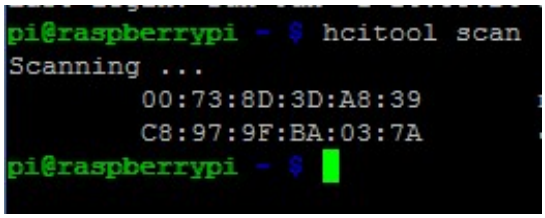

Fig. 4. Direcciones MAC de dispositivos encontrados.

En algunas ocasiones el comando anterior no es detectado y muestra un error, esto puede tener diversas razones, sin embargo la más común se debe a que el servicio debe reiniciarse mediante la instrucción:

```
sudo service bluetooth restart
```

Luego de haber obtenido la dirección MAC del dispositivo destino habrá que emparejarlo haciendo uso de la misma dirección, para este ejemplo se tomará en cuenta la primera dirección mostrada en la Figura 2:

```
sudo bluez-simple-agent hci0 00:73:8D:3D:A8:39
```

Para concretar el enlace, el dispositivo pide teclear un PIN de vinculación, típicamente es 0000 o 1234. Para verificar la conexión es posible realizar un ping de capa 2 entre el dispositivo destino y la tarjeta Raspberry Pi como sigue:

```
sudo l2ping –c 2 00:73:8D:3D:A8:39
```

Cuando se ha logrado un enlace exitoso con el dispositivo móvil, habrá que revisar que servicios tiene disponible el receptor, para ello será necesario hacer uso del control de servidores SDP (Service Delivery Platform) mediante la siguiente instrucción:

```
sdptool browse 00:73:8D:3D:A8:39
```

Esto desplegará un texto con información referente al dispositivo receptor como se muestra en la Figura. 5. Aquí hay que enfocarse en el canal de transmisión pues se hará uso de él para transmitir las imágenes al dispositivo.

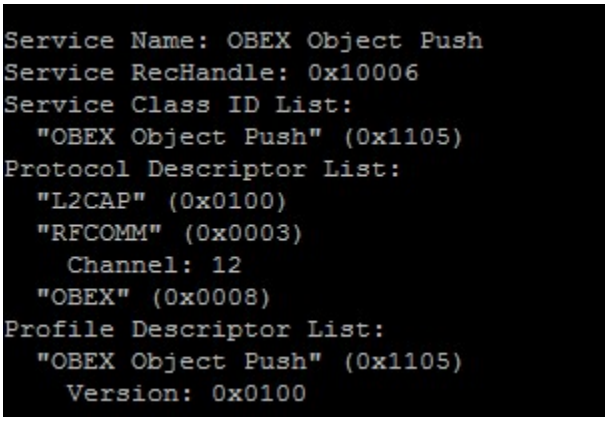

Fig. 5. Información recopilada por sdptool.

Conociendo el canal de transmisión, se procede con el envío de la imagen tecleando la siguiente instrucción:

obexftp --nopath --noconn --uuid none --bluetooth 00:73:8D:3D:A8:39 --channel 12 –p /home/pi/imagen2.jpg

Como se observa habrá que tenerse en cuenta la dirección absoluta donde se encuentra el archivo a enviar, en este caso la dirección fue */home/pi/imagen2.jpg*.

Hasta este punto se logra capturar imágenes y transmitirlas de manera manual pero esto resulta en un gran consumo de tiempo, y más si se considera su uso a gran escala. Para poder solucionar esto, se puede realizar un Script que tome la fotografía y la envíe mediante Bluetooth.

Para iniciar el Script se utiliza el comando seguido del nombre del archivo, en este caso lo llamaremos Tx_Blue:

vi Tx_Blue.sh o **sudo nano Tx_Blue.sh**

Es importante aclarar que no importa en qué editor se realice (ver Figura 6), solamente habrá que verificar que se haya redactado de manera adecuada y que se haga uso del canal que se revisó anteriormente.

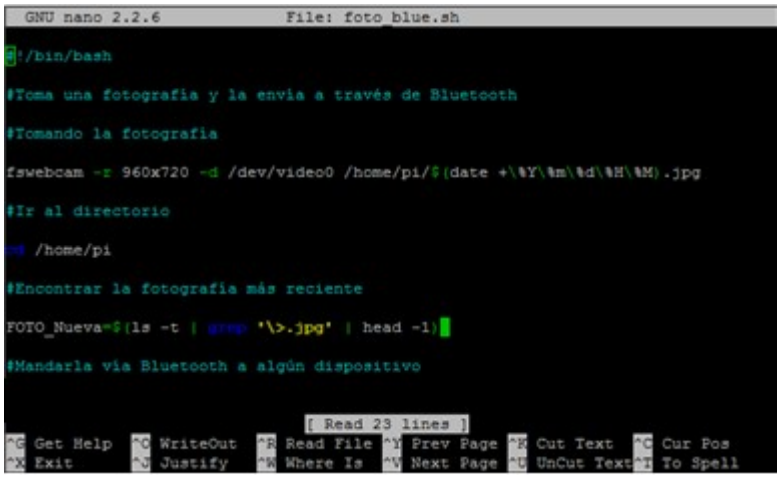

Fig. 6. Script en el editor GNU nano.

A continuación se redactan los comandos con los parámetros y la dirección MAC utilizados anteriormente, los cuales pueden cambiarse de acuerdo a las necesidades:

```
#!/bin/bash
#Toma una fotografía y la envía a través de Bluetooth
#Tomando la fotografía
fswebcam –r 960x720 –d /dev/video0 /home/pi/$(date +\%Y\%m\%d\%H\%M).jpg
#Ir al directorio
cd /home/pi
#Encontrar la fotografía más reciente
FOTO_NUEVA=$(ls –t | grep '\>.jpg'  |  head -1)
#Mandarla vía Bluetooth a algún dispositivo
obexftp --nopath --noconn --uuid none --bluetooth 00:73:8D:3D:A8:39 --channel 12 -p /home/pi/$FOTO_NUEVA
```

De lo anterior, FOTO_NUEVA es una variable que en este caso se asignó para buscar la fotografía más reciente y que ésta se transmita; esto se hace mediante grep para ordenar las imágenes. El comando *$(date +\%Y\%m\%d\%H\%M).jpg* se utiliza para que el nombre de la fotografía sea la fecha y hora en que fue tomada, esto resultará muy útil en la transmisión de las imágenes para su posterior interpretación.

Toda vez que se escribe el Script se procede a modificar el archivo *Crontab*, esta vez se busca que ejecute *Tx_Blue.sh* cada cierto intervalo de tiempo; de este modo tomará una foto y la transmitirá a un dispositivo mediante Bluetooth automáticamente:

```
crontab –e
```

En la parte inferior del archivo se encuentra la línea escrita anteriormente y hay que modificarla de la siguiente manera:

```
*/5 * * * * bash /home/pi/Tx_Blue.sh
```

O cualquiera que sea la ruta absoluta donde se haya guardado el archivo. Tras escribir la línea, se ejecutará el archivo para tomar una fotografía cada 5 minutos y transmitirla a un dispositivo.

Es posible utilizar este Script en un mayor número de tarjetas, de modo que entre ellas se haga un envío de información automático; sin embargo habrá que utilizar eficientemente los tiempos de *Crontab* y no ejecutar instrucciones al mismo tiempo. Teniendo esto en cuenta, no debe existir ningún problema en replicar este procedimiento a múltiples dispositivos.

Cuando toda la transmisión es funcional, es necesario tomar una tarjeta y que transmita la información a una estación base que sea monitoreada remotamente mediante Wi-Fi. Sin embargo, la transmisión de todas las imágenes podría mantener saturado el canal y aquellas que contengan la mayor cantidad de información podrían no ser recibidas en el tiempo requerido.

Es por ello que se ha propuesto la instalación de software para monitoreo remoto, donde se puedan revisar las imágenes a distancia y posteriormente seleccionar las que son de mayor interés para su estudio.

Para este fin, VNC es un sistema para compartir archivos con escritorio gráfico que permite controlar remotamente la interfaz de escritorio de una computadora en otra [11]. Básicamente es otra manera de controlar la tarjeta Raspberry Pi remotamente en modo gráfico, lo cual permite revisar las imágenes que han tomado las diferentes tarjetas y adquirir las que entreguen más información.

El primer paso para la instalación es descargar e instalar la librería como se indica a continuación:

```
sudo apt-get install tightvncserver
```

Una vez que se haya terminado de instalar se procede a ejecutarlo escribiendo el comando:

```
vncserver
```

A continuación se pedirá crear una contraseña (*password)* y confirmarlo. Cuando el programa pida crear una contraseña de *solo vista* seleccionar la opción *No*. Cuando se ejecuta el programa, cada vez mostrará un mensaje con el texto "*New 'X' desktop is raspberrypi:1*". Nótese que la terminación '*:1'* indica la sesión de escritorio creada.

Por el lado de la tarjeta habrán terminado las configuraciones, sin embargo es necesario instalar el programa *VNC Viewer* en la estación base. Al iniciar el programa hay que acceder a la tarjeta con base en la dirección IP que tenga, acompañado del :1 que se mostró anteriormente, hacer clic en conectar y escribir la contraseña (ver Figura 7).

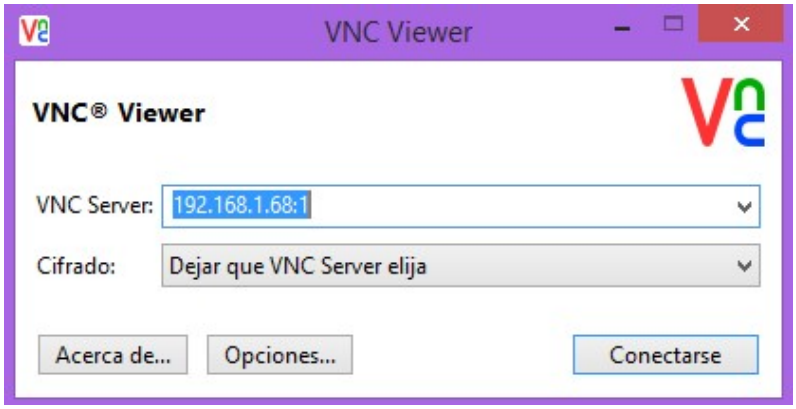

Fig. 7. Interfaz inicial del programa VNC Viewer.

Con esto podremos acceder a la tarjeta remotamente, sin embargo cada vez que se apague habrá que volver a iniciar el servicio por lo cual es necesario que se inicie automáticamente al reiniciar la tarjeta. Para ello habrá que crear un nuevo archivo en el directorio init.d (ver Figura 8).

sudo nano /etc/init.d/tightvncserver

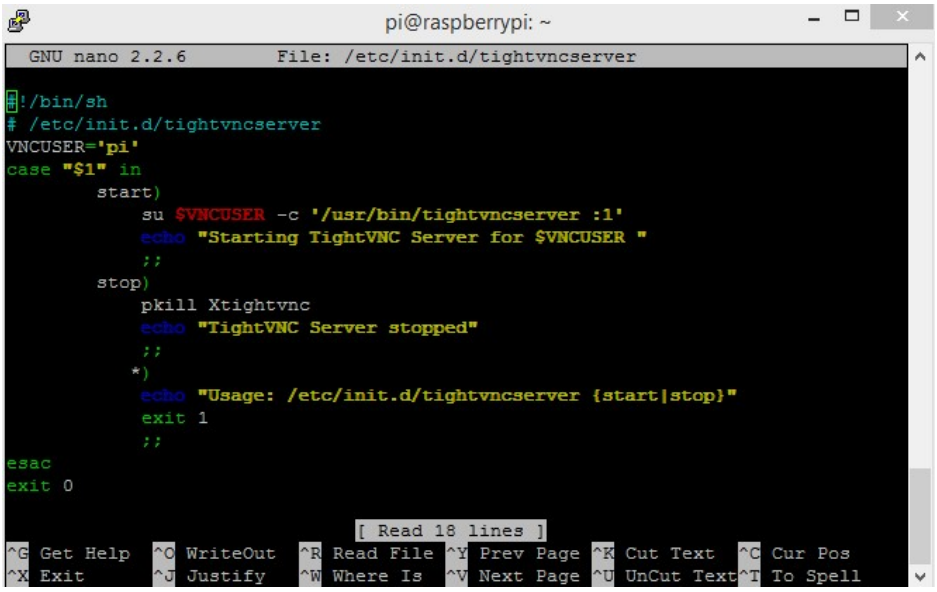

Fig. 8. Modificación del archivo tightvncserver para automatización.

En el editor de texto habrá que escribir el texto siguiente:

```
#!/bin/sh
# /etc/init.d/tightvncserver
VNCUSER=’pi’
case "$1" in
 start)
 su $VNCUSER -c '/usr/bin/tightvncserver :1'
echo "Starting TightVNC Server for $VNCUSER " ;;
stop)
pkill Xtightvnc
echo "TightVNC Server stopped";;
*)
echo "Usage: /etc/init.d/tightvncserver {start|stop}"
exit 1 ;;
esac
exit 0
```

Luego se procede a crear los permisos para ejecución de la siguiente forma:

```
sudo chmod 755 /etc/init.d/tightvncserver
```

Se reinicia el servicio manualmente y se ancla el programa para que inicie al arrancar la tarjeta:

```
sudo /etc/init.d/tightvncserver start

sudo /etc/init.d/tightvncserver stop

sudo update-rc.d tighvncserver defaults
```

Ahora se pueden visualizar las imágenes remotamente, por lo que solo resta tomar las imágenes más útiles. Para poder realizar una transferencia habrá que instalar el programa *WinSCP* (Windows Secure CoPy) el cual mostrará un explorador al clásico estilo Windows donde se podrán transferir archivos con solo jalar y soltar (ver Figura 9).

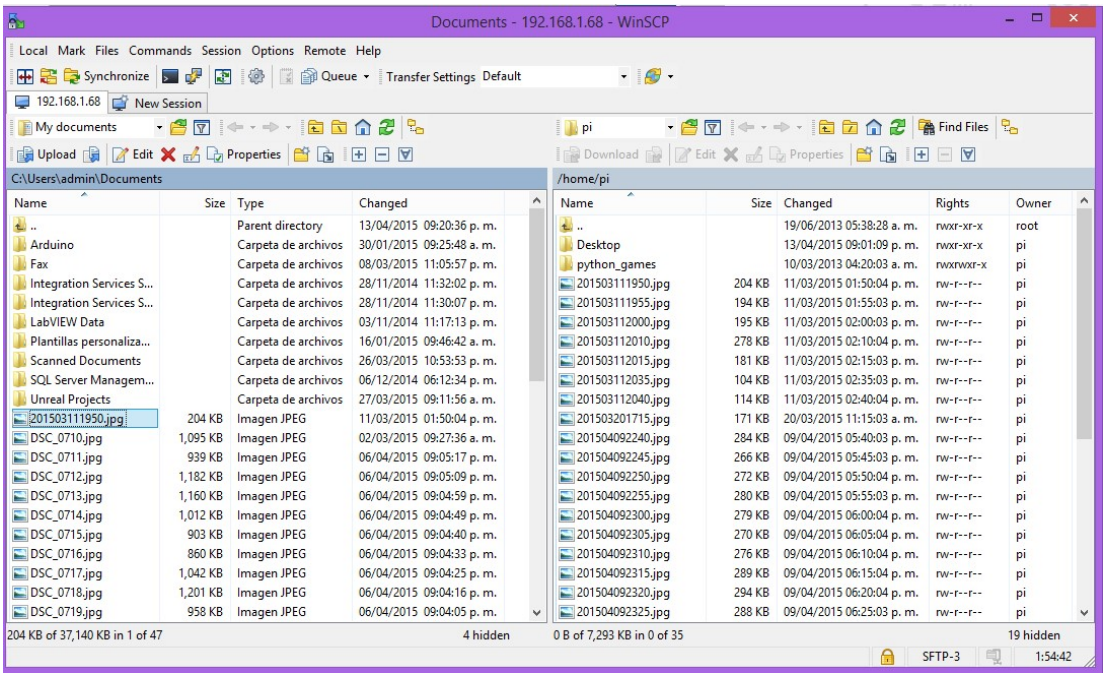

Fig. 9. Interfaz del programa WinSCP.

Al iniciar el programa pedirá la dirección IP, el nombre de usuario y la contraseña de la misma manera que *VNC Viewer*. Tras la instalación de ambos programas se podrán revisar las imágenes y seleccionar aquellas que convengan más de acuerdo a la información que contengan y posteriormente seleccionarlas y llevarlas a la estación base para revisarlas más detenidamente.

3. Resultados

Esta primera etapa aclaró diversas dudas sobre el uso de las tarjetas Raspberry Pi y su posterior uso en vehículos no tripulados como una herramienta para obtención de datos mediante sensores, representados actualmente por fotografías. En la Figura 10 se muestra la cama de pruebas experimental. Raspberry Pi 1 (RB1) mira hacia un horizonte distinto a Raspberry Pi 2 (RB2) y en este caso; por ejemplo: esta diversidad de imágenes podría permitirnos reconstruir el interior de una habitación.

La Figura 11 muestra lo que RB1 y RB2 observan en el horizonte. Tanto RB1 como RB2 tomarán una fotografía, la almacenarán y ésta será replicada a otros dispositivos automáticamente mientras son monitoreadas remotamente en una estación base.

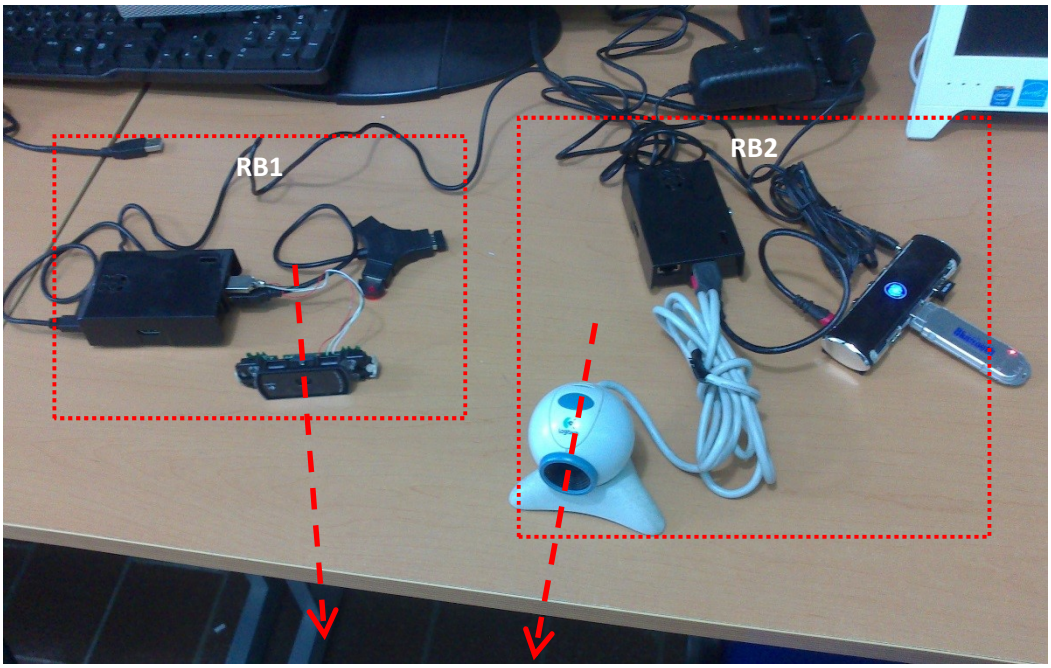

Fig. 10. Configuración de cámaras montadas en Raspberry Pi.

En la Figura 12, mostramos el panorama general de aplicación, cada UAV llevará como parte de su carga útil un RB así como un banco de sensores (en este trabajo una cámara) y su función será de replicar (de manera inalámbrica) al resto del enjambre la información que ha capturado durante su vuelo así como a la estación base.

Fig. 11. Fotografías tomadas por 2 cámaras, a partir de ellas se puede obtener una vista amplia de la zona de desastre.

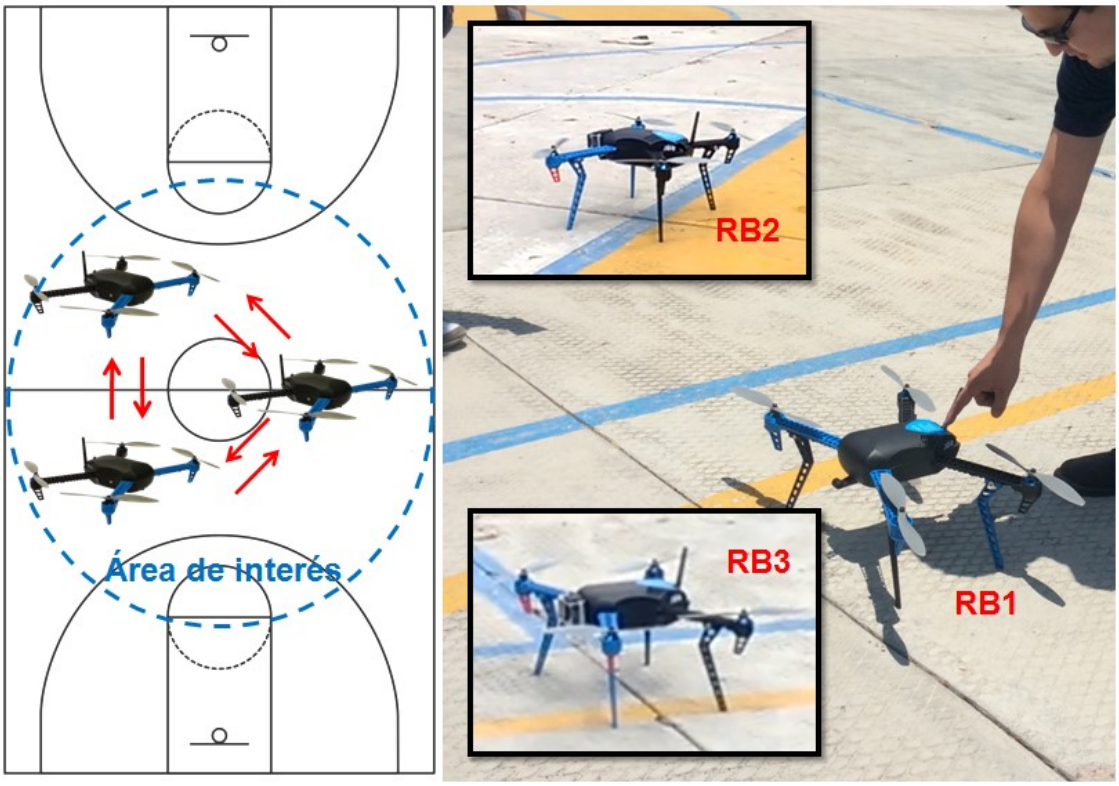

Fig. 12. Panorama general de aplicación.

4. Discusión

A diferencia de OpenRelief nuestro estudio va dirigido a contingencias mucho más pequeñas; presenta una unificación de varios proyectos que van desde la instalación de software para toma de fotografías automáticas y el uso de Bluetooth y Wi-Fi como plataformas de comunicación con otros dispositivos, que puede ser utilizado de inmediato en drones que trabajen en conjunto. De este modo, los resultados obtenidos indican que es posible utilizar la tarjeta Raspberry Pi para la transmisión de información recopilada por sensores a través de un módulo Bluetooth.

Durante el desarrollo se notó que las imágenes deben enviarse a intervalos de tiempo distintos en cada tarjeta para no ocasionar errores por colisión de datos, por lo tanto resultaría más óptimo transmitirlas simultáneamente utilizando diferentes canales de Bluetooth.

5. Conclusiones

Se ha logrado de manera exitosa una transmisión automática de archivos mediante Bluetooth como una red avanzada de comunicaciones, la cual tiene diversas aplicaciones. En este proyecto, la finalidad es montarla en vehículos no tripulados para que exista una red entre ellos y que éstos reporten información a una estación base remota.

Como trabajo futuro se plantea utilizar la red con diversos sensores para monitorear diversas sustancias que puedan existir en un ambiente peligroso para el ser humano, para ser montada en los vehículos no tripulados y enviarlos a misiones de rescate y protección civil.

6. Referencias

[1] The UAV. http://www.theuav.com. Febrero 2015

[2] Kai, D. et al., AirShield: A System-of-Systems MUAV Remote Sensing Architecture for Disaster Reponse. 3rd Annual IEEE International Systems Conference. 2009. IEEE. Vancouver, BC. 196-200 pp.

[3] Raspberry Pi Security Camera with motion pie. https://www.squirrelhosting.co.uk/hosting-blog/hosting-blog-info.php?id=94. Febrero 2015.

[4] Turn a Raspberry Pi into a CCTV Security System. http://www.averagemanvsraspberrypi.com/2014/09/turn-raspberry-pi-into-cctv-security.html. Febrero 2015.

[5] Raspberry Pi as low-cost HD surveillance camera. http://www.instructables.com/id/Raspberry-Pi-as-low-cost-HD-surveillance-camera/. Febrero 2015.

[6] How to capture time-lapse photography with your Raspberry Pi and DSLR or USB Webcam. http://www.makeuseof.com/tag/how-to-capture-time-lapse-photography-with-your-raspberry-pi-and-dslr-or-usb-webcam/. Febrero 2015.

[7] How to build a High-definition FPV UAV using a Raspberry Pi with HD camera, using a high speed Wi-Fi link. http://diydrones.com/profiles/blogs/how-to-build-a-high-definition-fpv-uav-using-a-rasperry-pi-with. Febrero 2015.

[8] Raspberry Pi in the sky: Wallet-sized PC is disaster drone brain. http://www.theregister .co. uk/2012/06/12/raspberry_pi_drone/. Febrero 2015.

[9] How to set up Bluetooth on your Raspberry Pi. http://www.raspberrypi.org/ learning/robobutler/bluetooth-setup.md. Febrero 2015.

[10] Obexftp – Mobile Equipment file transfer tool. http://linux.die.net/man/1/obexftp. Febrero 2015.

[11] VNC (Virtual Networking Computing). http://www.raspberrypi.org/documentation/ remote-access/vnc/. Febrero 2015.

7. Autores

El Sr. Luis Fernando Hernández Medina es pasante en la Licenciatura en Ingeniería en Telecomunicaciones por la Universidad Autónoma de San Luis Potosí.

Marco A. Cárdenas Juárez, Armando Arce Casas, Enrique Stevens Navarro y Ulises Pineda Rico son profesores-investigadores de la Facultad de Ciencias (Universidad Autónoma de San Luis Potosí), perfil PROMEP y pertenecen al Sistema Nacional de Investigadores CONACYT.

8. Agradecimientos

Este trabajo fue financiado mediante el apoyo 215499 del Consejo Nacional de Ciencia y Tecnología (CONACYT), por la Universidad Autónoma de San Luis Potosí mediante el fondo de apoyo a la investigación (FAI) así como por el programa para el desarrollo profesional docente (PRODEP).