

# Interfaz de comunicación visual bidireccional mediante códigos QR

Indalecio Martínez Bourial, Sergio Antequera Manrique, Alberto Peinado Domínguez.

indaleciomartinez@uma.es, seryiosam28@gmail.com, apeinado@ic.uma.es

E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga, Campus de Teatinos 29071 Málaga

**Abstract**-This paper presents the development of a bidirectional communication interface between two devices whose sole link is through the use of QR Codes. In this way, the original QR functionality is extended with a file transfer service, while a new secure link, save from radiofrequency interference and eavesdropping, is provided for an easy and secure personal device synchronization. The implemented communication system has been designed to connect devices equipped with screens and cameras. Two versions have been developed. One, as a proof of concept, on two Raspberry Pi units, and the other, focused on the end user, on Android smartphones, that can be download and installed from the official Store site.

## I. INTRODUCCIÓN

Actualmente, las nuevas tecnologías se han convertido en una herramienta fundamental que nos permite agilizar, optimizar y perfeccionar algunas de las actividades que realizamos en nuestra vida. A partir de la pandemia de COVID19 vivida en 2020, una tecnología muy simple se popularizó de manera drástica, ya que permitió a todas las empresas tener un medio para presentar sus servicios o productos a los clientes guardando el distanciamiento físico necesario para evitar la propagación del virus: los códigos QR (Quick Response Code) [1], una tecnología creada en 1994 como evolución de los códigos de barras unidimensionales.

Aprovechando que se puede considerar que el uso de los códigos QR es universal y la práctica totalidad de los usuarios de smartphone los ha utilizado en alguna ocasión, se plantea en este trabajo la implementación de un canal de comunicación para transferir ficheros entre dispositivos basado exclusivamente en este tipo de códigos. Esto permitirá realizar transferencias de ficheros en situaciones de nula cobertura de datos móviles o wifi y sin las debilidades de seguridad propias de los sistemas de comunicaciones a través de radiofrecuencia, como Bluetooth.

La propuesta plantea dar un paso más en la utilización de estos códigos que, hasta la fecha, se han empleado como mecanismo de transmisión unidireccional, donde una vez codificado el mensaje a transmitir en un código QR, éste se libera y se ubica estratégicamente hasta que algún usuario decide leerlo, momento en el que se lleva a cabo la recepción del mensaje, sin interacción alguna entre ambos extremos de la comunicación.

El sistema que se propone vendrá caracterizado por la capacidad de enviar archivos de distinto tamaño a través de sucesivas imágenes de códigos QR que deben ser captadas por la cámara del dispositivo receptor y, posteriormente, procesadas para obtener la información transmitida.

En las secciones 2 y 3, se describen las principales características de los códigos QR y de los protocolos de enlace de datos, respectivamente. Las secciones 4, 5 y 6 describen las especificaciones del sistema propuesto, los aspectos propios de la implementación y el funcionamiento y validación, respectivamente. Por último, la sección 7 incluye las conclusiones.

## II. CÓDIGOS QR

Los códigos QR fueron desarrollados por la empresa japonesa Denso-Wave en 1994. El estándar japonés fue publicado en 1999 [1] y su correspondiente estándar internacional ISO aprobado en 2000 [2]. Los QR son códigos bidimensionales formados por una matriz de propósito general diseñada para un escaneo rápido de información. Los códigos QR tienen una relación de aspecto cuadrada y se identifican fácilmente porque contienen tres cuadros más pequeños ubicados en tres de las cuatro esquinas, lo que permite identificar la posición del código al lector.

El estándar contempla 40 versiones con el fin de adaptarse a la cantidad de datos que se deseen transmitir. El fundamento de todas las versiones es el mismo. La única diferencia reside en el tamaño, que aumenta con el número de la versión. La capacidad de corrección de errores puede configurarse eligiendo uno de los cuatro niveles disponibles. El nivel L representa menor capacidad de corrección y el nivel H la máxima.

Las principales características de estos códigos son la velocidad de decodificación y la resistencia frente a errores conseguida mediante la utilización de códigos Reed-Solomon. Por todo ello, se pueden considerar estos códigos como un contenedor de información o, empleando terminología de comunicaciones, como un mecanismo de modulación para adaptar la información a las especiales características del canal de comunicación visual que se establece entre transmisor y receptor a través de la cámara que captura el código.

## III. PROTOCOLO DE ENLACE DE DATOS

La utilización habitual de los códigos QR se limita a la recepción de un único mensaje sin posibilidad de informar al emisor sobre un posible fallo o error en la decodificación. Cuando el objetivo que se plantea es la transmisión de ficheros, debido a la limitada capacidad de los códigos QR, se hace necesario utilizar más de un código. Del mismo modo en que un fichero debe fragmentarse para su envío por Internet mediante una determinada cantidad de paquetes, en este caso deberá fragmentarse para enviarse utilizando tantos códigos

QR como sean necesarios. Por lo tanto, siguiendo la analogía de las redes IP, la transmisión de ficheros mediante códigos QR necesita implementar un protocolo de control de enlace de datos que controle el flujo de la transmisión de cada paquete en forma de código QR.

Para mantener la simplicidad del sistema y dado el reducido tiempo de propagación, se ha seleccionado como protocolo ARQ el mecanismo de parada y espera [3] (Fig.1).

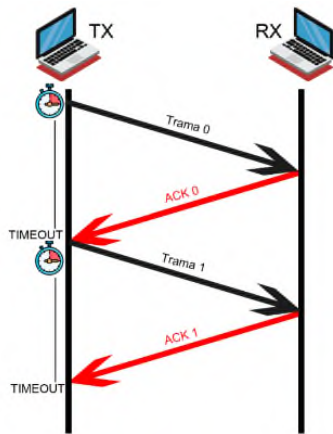


Fig. 1. Control de flujo parada y espera utilizado en el sistema

Este control de flujo requiere el envío de tramas de confirmación (ACK) que también se enviarán codificadas como códigos QR. Siguiendo el esquema de los protocolos de enlace de datos más habituales, se ha diseñado una cabecera para indicar al receptor el número de secuencia de cada QR y el número total de paquetes que componen el fichero a transmitir (Fig. 2).

#### IV. ESPECIFICACIONES Y DISEÑO DEL SISTEMA

Con el propósito de cumplir los objetivos marcados en la introducción se hace uso, inicialmente, de dos microordenadores Raspberry Pi dotados de pantalla LCD y cámara que serán capaces de captar y procesar los datos enviados a través de los códigos QR. Una segunda versión del sistema ha sido desarrollada para Android a partir del análisis del comportamiento del sistema inicial sobre RaspBerry Pi.

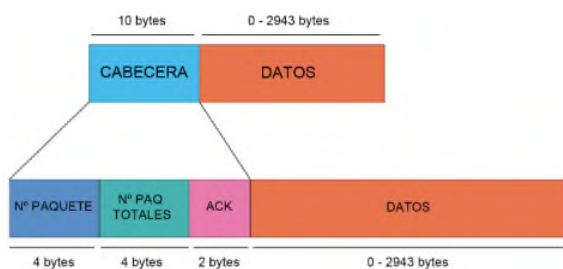


Fig. 2. Formato de la trama incluida en el código QR.

En la primera versión, se ha utilizado un tamaño de código QR fijo debido a que se trataba de una prueba de concepto en la que testear el funcionamiento del sistema de transmisión. Sin embargo, en la versión para Android, el objetivo es la utilidad práctica. En consecuencia, se ha planteado como prioritario que el tamaño del código QR sea calculado al inicio mediante un proceso de autocalibración entre los dispositivos que se van a conectar.

A partir de estas premisas, el sistema propuesto deberá proporcionar la siguiente funcionalidad: calibración inicial para determinar el tamaño (versión) óptimo del código QR; transmisión o recepción de un fichero de cualquier tamaño; detección de fallos durante la transmisión y cancelación de la misma, si procede.

#### V. IMPLEMENTACIÓN

Para la primera versión, se han utilizado una Raspberry Pi 3 Model B/B+ y una Raspberry Pi 4 Model B. Mediante la herramienta oficial Raspberry Pi Imager se ha instalado el sistema operativo en una tarjeta MicroSD y se ha activado el protocolo SSH para poder operar en remoto y realizar una adecuada monitorización del sistema.

Se ha utilizado una pantalla de 3.5" LCD que, al no ser Plug and Play, ha sido necesario hacer uso de los pines GPIO como se muestra en la Fig. 3

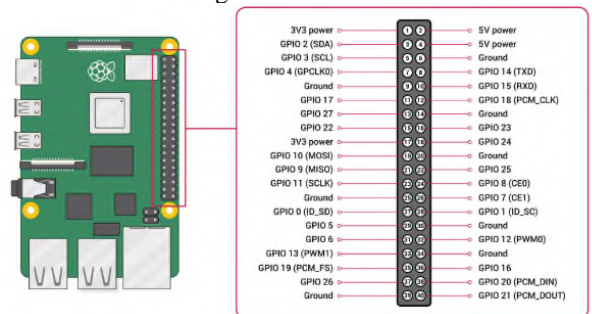


Fig. 3. Configuración de LCD en Raspberry Pi

Para la detección de los QR se ha utilizado una cámara digital que se integra perfectamente con la placa de la Raspberry Pi y permite brindar al sistema la capacidad de captar las imágenes de los códigos QR que posteriormente serán procesados.

Gracias a la activación del protocolo SSH en la configuración previa, se puede acceder cómodamente de manera remota a través de la herramienta PuTTY, pudiendo ejecutar las líneas de código que permiten la instalación y actualización de drivers de los distintos accesorios para el correcto funcionamiento.

Tras esto, se obtiene ambos sistemas funcionales y listo para poder ejecutar el interfaz de comunicación como se muestra en la Fig. 4.



Fig. 4. Prototipo basado en Raspberry Pi

El interfaz de comunicación basado en códigos QR ha sido desarrollado en Python, haciendo uso del módulo OpenCV [], una biblioteca de código abierto muy utilizada en el ámbito de la visión artificial y el procesamiento de imágenes. El principal uso de esta tecnología es brindar al dispositivo de la

capacidad para la detección y decodificación de códigos QR directamente de imágenes o video. También permite localizar y posicionar los códigos QR en una imagen para poder obtener los datos codificados de ellos.

Para una mejor disposición de los dispositivos, se ha diseñado, a través del software Fusion360, un soporte como se observa en la Fig. 5 que permite sujetar el módulo de la cámara justamente encima de la Raspberry Pi para así poder realizar la ejecución del interfaz de manera eficiente. Este diseño se ha procesado en la herramienta UtiMaker Cura, consiguiendo una laminación por capas correcta para poder imprimir en una impresora 3D con filamento PLA.

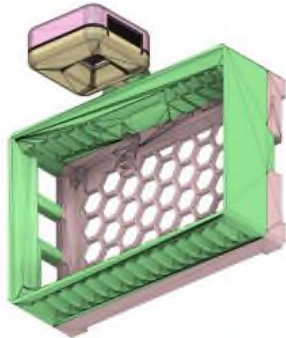


Fig. 5. Soporte, para impresión 3D, utilizado en pruebas con Raspberry Pi

La versión para Android, QRShareExpress, es completamente funcional y cumple con todas las especificaciones. La versión definitiva se puede descargar y utilizar libremente desde el repositorio Google Play (Fig. 6) [5] en cualquier dispositivo gracias a la calibración inicial. Mediante este proceso se calcula el tamaño óptimo del código QR. Este aspecto permite a la aplicación ser utilizada por cualquier pareja de dispositivos que se quieran conectar, puesto que el tamaño del código QR depende, además de las condiciones lumínicas del entorno en el que se realice la transmisión, de los siguientes aspectos:

- resolución de la cámara frontal, que suele ser menor que la cámara principal.
- tamaño de la pantalla
- brillo de la pantalla
- resolución de la pantalla
- distancia a la que se encuentren los dispositivos
- ángulo entre la pantalla del transmisor y la cámara del receptor

El proceso de calibración se realiza en ambos sentidos de la comunicación y consiste en el aumento progresivo de la versión de los códigos hasta encontrar el límite del escenario particular en el que se realiza el proceso; si bien, el tamaño calculado solo es necesario cuando el dispositivo actúa como transmisor puesto que los códigos QR que contienen las tramas ACK pueden utilizar versiones más pequeñas. Esta funcionalidad, equivalente a las fases de negociación de los protocolos inalámbricos, permite también reajustar los parámetros entre dos dispositivos previamente calibrados cuando las condiciones ambientales han sufrido cambios considerables.

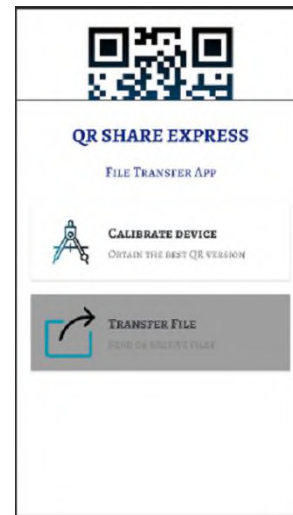


Fig. 6. App QRShareExpress

## VI. FUNCIONAMIENTO DEL SISTEMA Y VALIDACIÓN

El funcionamiento del interfaz consiste en una sincronización de los distintos módulos que las componen. Principalmente se parte en un estado base donde ambas Raspberry Pi se encuentran en estado de reposo de captura ejecutado por la librería OpenCV.

Para ajustar el tamaño de ventana de captura a la resolución de la pantalla de 3.5" LCD se han reajustado los fotogramas obtenidos a la resolución de 480 x 320 px para que pueda contenerse en las dimensiones de la pantalla de manera correcta como se muestra en la Fig. 6.



Fig. 7. Escenario para el modelo del sistema

Partiendo de este estado, pueden suceder dos cosas. La primera es que una de las dos Raspberry pi quiera transferir un archivo y para ello interrumpa su proceso de captación de código QR para poder procesar el archivo que quiere transmitir. Para que ocurra esta interrupción se puede utilizar distintos métodos, que van desde la simple utilización de un pulsador para indicarle al sistema que se quiere transmitir un fichero, hasta la utilización de señales, multiprocesamiento o hilos para causar dicha interrupción. Para las pruebas, se han utilizado periféricos convencionales, como son un teclado y ratón, haciendo que en este caso se pueda utilizar la interrupción vía teclado con una simple tecla y pudiendo proporcionar el nombre del archivo que se desee transmitir. El sistema procederá a preparar el paquete con la cabecera correspondiente a los datos que se van a transmitir junto a su

payload correspondiente. Esto se transformará en bytes consecutivos e introducidos en el código QR configurado previamente con la versión que se desee para mostrarlo por pantalla.



Fig. 8. Pruebas de ejecución de la versión Raspberry

Como se muestra en la Fig. 7, a la izquierda se dispondría del código QR con la trama preparada y el contenido de un archivo de texto. A la derecha se dispone la decodificación del receptor tras el procesamiento correcto del código QR captado. Tras esto, la Raspberry Pi receptora generará otro código QR donde solamente tendremos una trama en la que el campo de la cabecera estará indicado el número de paquete correspondiente recibido y un '1' en el campo de ACK. Esto permitirá al transmisor saber que el paquete que ha enviado se ha recibido correctamente sin errores y proceder a la transmisión del siguiente paquete.

Volviendo al inicio, el segundo caso que puede suceder es que estando en el estado ocioso de captación, una de las dos Raspberry Pi capte un código QR de la otra. Se debe principalmente recibir el nombre del archivo que se va a recibir en un primer código QR, donde el número de paquete será el '0'. Una vez recibido y almacenado el nombre del archivo, se procede a la decodificar los códigos QR recibidos, almacenando los distintos payload binarios en el archivo creado con el nombre del archivo previamente recibido. Este procede a confirmar por cada código QR recibido su ACK correspondiente. Una vez finalizada la captación de los códigos QR no es necesario que se emita nada más, ya que se entiende que la transmisión y recepción del archivo se ha finalizado correctamente.

En cuanto a la versión de Android, se ha diseñado un interfaz gráfico de usuario muy simple, con dos únicas opciones al inicio: calibración o transmisión. Para realizar la transmisión, una vez calibrado los equipos, solo hay que seleccionar el fichero a transmitir y acercar los dos dispositivos del modo en que se ha realizado la calibración.

Las pruebas realizadas para validar el funcionamiento han utilizado ficheros de distinto tamaño y formato. Se han considerado ficheros de texto, audio, imagen, binarios y muy distintos tamaños que han determinado la fragmentación en códigos QR en cantidades que han oscilado desde 1 hasta 120. La máxima velocidad obtenida está en torno a los 3 códigos por segundo, lo que permite transmitir ficheros de un tamaño moderado, sobre todo teniendo en cuenta que la transmisión se realiza manteniendo enfrentados los dispositivos manualmente, por lo que, aunque el protocolo soporta transmisiones por tiempo indefinido, no resulta práctico ni cómodo realizar transmisiones muy largas.

## VII. CONCLUSIONES

Las pruebas realizadas permiten asegurar un alto nivel de aplicabilidad del enlace bidireccional mediante códigos QR, tanto de forma integrada en nuevos desarrollos orientados a dotar a dispositivos embebidos, como por ejemplo sensores, de una interfaz de comunicación inalámbrica independiente y segura, como a través de aplicaciones software que ofrezcan a los usuarios finales la posibilidad de intercambiar información entre sus dispositivos sin más que enfrentar sus correspondientes cámaras y pantallas, como se puede comprobar a través de la app QRShareExpress – Transfer File [5] desarrollada en este trabajo como demostrador, que puede descargarse y utilizarse gratuitamente desde el repositorio Google Play.

El sistema desarrollado, especialmente, la versión Android, es susceptible de mejora en cuanto a la velocidad de transmisión mediante la aplicación de técnicas de compresión de la información. No obstante, se presenta como un caso de éxito en la utilización de una nueva vía de comunicación inalámbrica que incorpora seguridad implícita, esto es, seguridad vinculada directamente al método de transmisión de la información que reduce los riesgos de interferencia e interceptación radioeléctrica que sufren otros sistemas de comunicación personal como Bluetooth, WLAN e incluso NFC, éste en menor medida.

## AGRADECIMIENTOS

Esta publicación es parte del proyecto de I+D+i P2QProMeTe (PID2020-112586RB-I00), financiado por MCIN/ AEI/10.13039/501100011033.

## REFERENCIAS

- [1] Denso Wave. QR-Codes specification, <http://www.denso-dave.com/qrcode/qrstandard-e.html>
- [2] International Organization for Standardization/International Electrotechnical Commission, "Information technology. Automatic identification and data capture techniques. QR Code 2005 bar code symbology specification", ISO/IEC 18004:2006, 2007
- [3] Stallings, W., Comunicaciones y redes de computadores. Pearson Education, 2008.
- [4] OpenCV. <https://opencv.org/>
- [5] Google Play. QRShareExpress – Transfer Files. [https://play.google.com/store/apps/details?id=com.srdidimau.test\\_to\\_qr&hl=es\\_AR&pli=1](https://play.google.com/store/apps/details?id=com.srdidimau.test_to_qr&hl=es_AR&pli=1) (último acceso: 07/04/2024)