

***Universidad Nacional de La
Matanza***

***Trabajo Práctico de
Sistemas Operativos Avanzados***

Portero Inteligente

Curso: *1er Cuatrimestre de 2018*

Alumnos:

Leonardo Pastor

Julián A. Morganella

Ezequiel Valdez

Motivación:

Portero Inteligente: Desarrollar un Dispositivo que se capaz de cumplir con los siguientes objetivos:

- Poder atender de forma remota, a una persona que haya presionado el timbre de nuestra vivienda, y para tal fin transmitir audio y video en tiempo real.
- Realizar el control de acceso a nuestra vivienda por medio de tarjetas de proximidad como así también de forma remota.

Cuando decimos de forma remota, nos referimos a través de una aplicación en nuestros teléfonos inteligentes.

Componentes:

El desarrollo se dividió en 2 módulos funcionales, por un lado, se utilizó una placa Arduino nano y por otro lado se utilizó una placa Raspberry.

¿Por qué el desarrollo se dividió de esta forma?, muy simple uno de los objetivos del proyecto es poder transmitir streaming de audio y video en tiempo real, lo cual requiere contar con hardware idóneo, característica que la placa de desarrollo Arduino nano no proporcionaba.

Además, realizar tal división proporciona un control más preciso en cuanto a tiempo de respuesta se refiere, ya que sobre la plataforma Arduino (al ser un microcontrolador [no posee un sistema operativo]), se puede conectar todo el hardware que requiera de un tiempo de respuesta acotado, brindando además un control más preciso del mismo. Y en la plataforma de desarrollo Raspberry (SBC, single board computer [posee un sistema operativo]), proporciona capacidad de cómputo y bondades que brinda un sistema operativo para poder, por ejemplo, simplificar la tarea de transmisión de audio y video en tiempo real, utilizar un motor de base de datos, etc.

La comunicación entre ambas plataformas se realizó por medio de UART Serial, mediante un protocolo propietario, que más adelante se va a detallar.

La interacción entre la aplicación del teléfono inteligente y el sistema embebido, se realizó por medio de dos canales diferentes:

- Para la transmisión de mensajes desde la aplicación hacia el sistema embebido y viceversa, se utilizó un servicio que brinda Google, Firebase, el cual permite fácil integración en la plataforma de desarrollo Android SDK. En la plataforma Raspberry la interacción con Firebase se implementó mediante scripts (Python).
- Y por otro lado para el streaming de audio y video en tiempo real se utilizaron dos servicios que corren sobre la plataforma Raspberry:
 - UV4L se ocupa del streaming de video, recuerde que para el caso del video es en un solo sentido, es decir uno desde el teléfono móvil va a poder ver quien presiono y está frente a nuestra vivienda, pero la persona que está frente a nuestra vivienda no va a poder ver quien este en el teléfono.

- ffmpeg, en este caso el audio si es bidireccional y la comunicación es de tipo full-dúplex.

En ambos casos la forma de implementación se detallará más adelante.

Entonces, la plataforma Raspberry, como ya adelantamos se va a ocupar de las siguientes actividades:

- Transmisión de audio
- Transmisión de Video
- Comunicación con los servicios de Firebase.
- Validación de los datos provenientes de las tarjetas de control de acceso.
La validación se realiza a través de una base de datos, gestionada por medio De los servicios que brinda SQLite3.
- Interacción con la plataforma de Arduino, Intercambio de mensajes (comandos) provenientes de Firebase (enviados desde la App móvil), como así también mensajes provenientes desde Arduino (solicitud de validación de tarjetas y aviso de eventos, como por ejemplo timbre pulsado.)

Mientras que la placa Arduino va a cumplir las siguientes tareas.

- Control sobre el servo motor, el cual acciona la cerradura de la puerta.
- Gestión del lector de tarjetas de proximidad, lectura de las mismas
- Control del Display, mostrar mensajes de acuerdo a los eventos que ocurran.
- Control sobre eventos que puedan darse lugar en el pulsador (timbre), como así también el final de carrera (usado para determinar el estado de la puerta, y así, poder accionar el servo motor que acciona la cerradura de la puerta.
- Gestión del Relé el cual reacciona en función del estado del sensor de luz LDR.
- Gestión del módulo sensor de luz LDR.
- Control sobre un Buzzer, emite una respuesta sonora ante determinados eventos, por ejemplo, tarjeta sobre el lector.

A continuación, se detalla cómo se conectaron los distintos componentes de hardware, es decir que interfaces se utilizaron:

Raspberry:

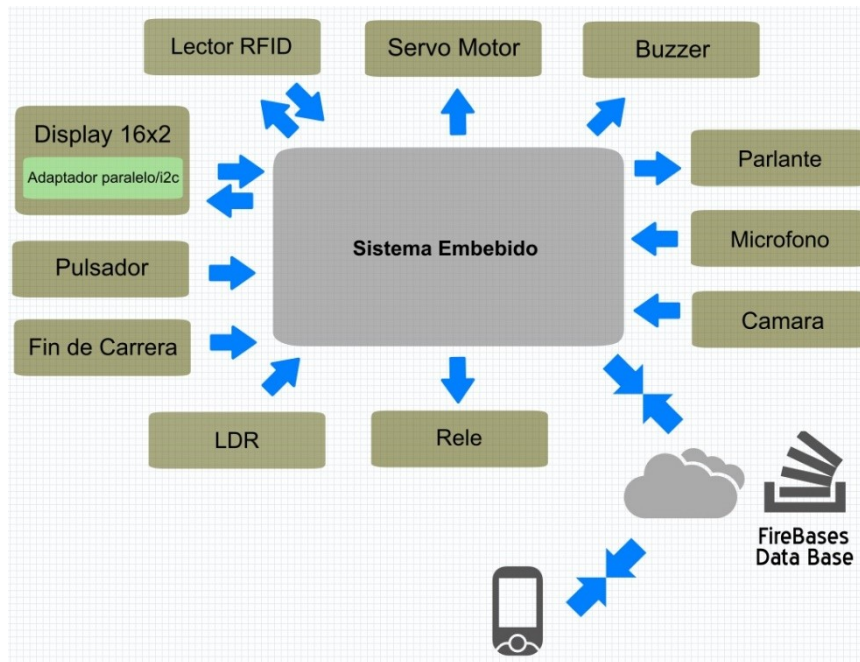
- Cámara Raspberry (Raspi Cam), se utilizó el bus CSI (Camera Serial interface) que dispone la placa Raspberry Pi.
- Para el caso del audio, es decir la conexión del micrófono y el parlante, para que la persona que está en la puerta de nuestra vivienda pueda interactuar con el sistema embebido se utilizó una placa de audio USB externa, ya que, si bien la plataforma posee salida de audio por medio del conector HDMI, o un plug 3,5 mm, la misma no dispone de una entrada de audio para la conexión del micrófono.

- Módulo Wifi, ya integrado en la placa Raspberry, se utiliza para disponer de conectividad, tanto para acceder a los servicios de Firebase, como para la comunicación con la App (transmisión de streaming audio y video) del teléfono móvil.
- Módulo Ethernet, ya integrado en la placa, se utiliza para acceder de forma remota, por medio del servicio SSH, solo con propósitos de debug.
- Puerto Serial UART, para la comunicación con la plataforma Arduino.

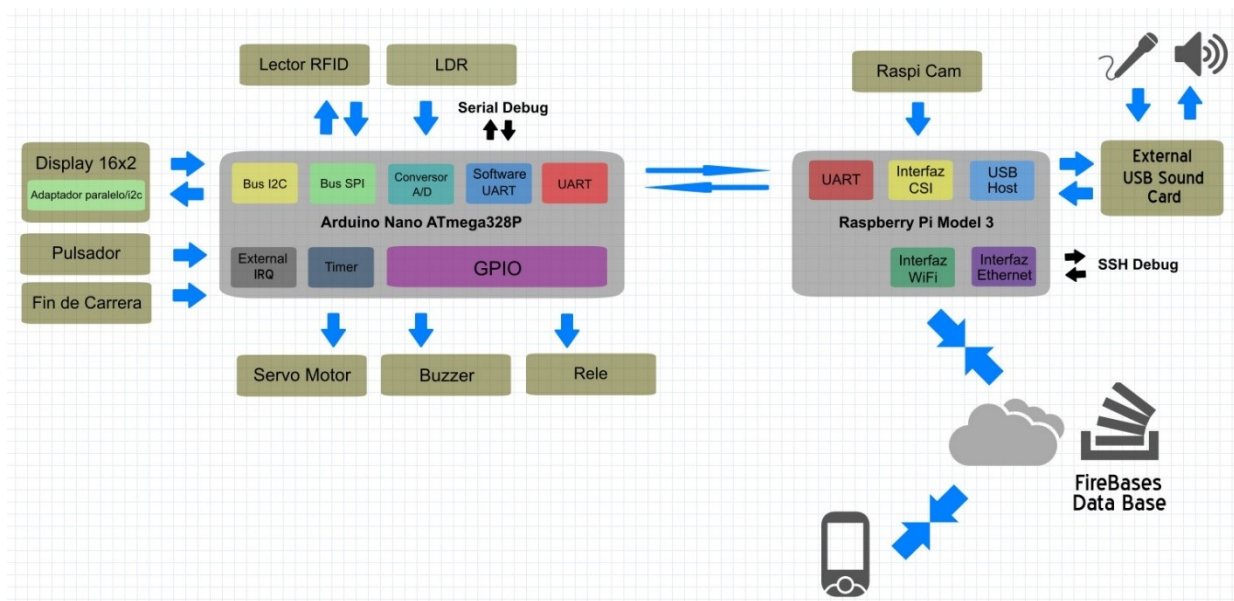
Arduino:

- Display de cristal líquido 16 caracteres x 2 líneas, en este caso se utilizó un adaptador paralelo --> I2C, es decir, entonces que en la placa Arduino se utilizó el bus I2C, ¿para que se utilizó tal adaptador? , Para poder ahorrar la cantidad de pines a utilizar en la placa Arduino.
- Lector de tarjetas de proximidad midFire 13,56 MHz, este dispositivo se conectó al bus SPI, ya que este dispositivo dispone de dicha interfaz.
- Puerto Serial UART para la comunicación con la placa Raspberry.
- Pulsador (timbre) y Final de carrera se conectaron al puerto que es capaz de censar una interrupción de hardware externa (External IRQ).
- LDR, para la lectura del sensor de intensidad de luz se empleó una entrada que con la funcionalidad de conversión analógica-digital.
- Relé y Servomotor fueron conectados a pines dentro del conjunto de GPIO que se disponía.

Diagrama en Bloques:



Despliegue del Diagrama en Bloques:



Android:

Introducción:

La aplicación de Android simplifica y ayuda a la atención del portero desde cualquier sector del mundo, simulando estar en casa o a la atención de la puerta para personas discapacitadas, o de avanzada edad, que tengan movilidad reducida. De esta forma, sin moverse pueden atender y abrir la puerta. Además esta aplicación le permite al usuario, que en caso de que personas malintencionadas toquen el timbre para saber si hay gente en la vivienda, el usuario puede responder como si estuviera presente.

Concepto:

- La comunicación de los estados, como se pulso timbre, se abrió/cerro puerta, se activó el micrófono, son enviados por Firebase desde Arduino a Raspberry e inversa.
- Se necesita un dispositivo con Android 4.2
- La Aplicación hace uso de 3 sensores, proximidad, luz ambiente y acelerómetro. Estos cuando detectan que el celular esta con la pantalla hacia abajo, no tiene nada de luz ambiental, y la proximidad es nula. Se deshabilitan las funciones criticas como activar micrófono, se apaga el video y se deshabilita la posibilidad de abrir la puerta.
- La comunicación de los estados, como se pulso timbre, se abrió/cerro puerta, se activó el micrófono, son enviados por Firebase desde Android a Raspberry e inversa.
- La comunicación multimedia entre el portero y la App puede estar cifrada con SSL, si se configura uv4l.
- El uv4l fue configurado para iniciar automáticamente en la Raspberry.

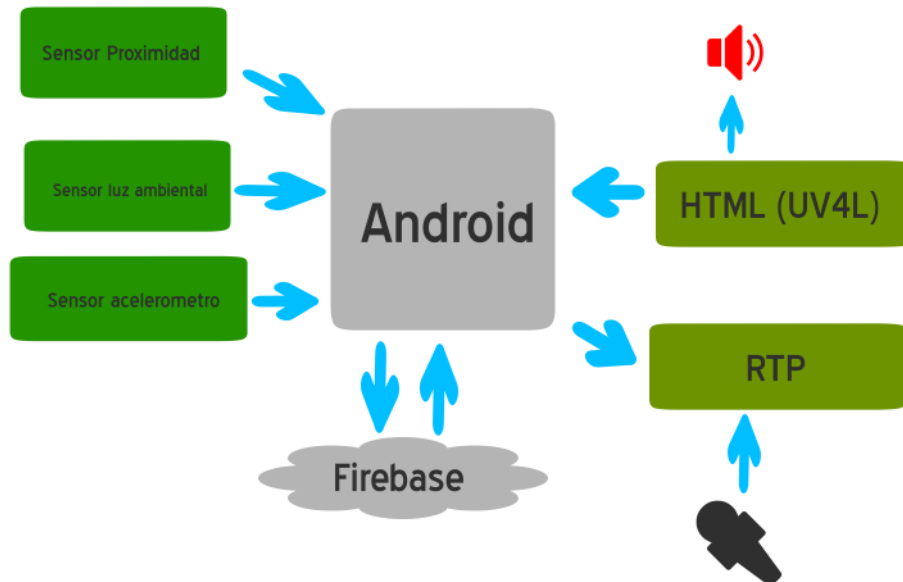
Uv4l es una colección modular de controladores para audio y video que dispositivos de entrada y salida de video real o virtual que permite el streaming de audio y video con latencias y resoluciones aceptables mediante un servidor rtc, se utilizó por las siguientes características:

- Espacio de usuario completo Desde la perspectiva del usuario, los controladores no son diferentes de aquellos en el espacio del kernel.
- Multiplataforma
- Baja probabilidad de crash en el kernel
- Cada módulo se puede cargar de forma dinámica y configurar desde la línea de comandos o el archivo de configuración.
- Hay un proceso de sistema independiente para cada dispositivo manejado (puede haber muchos de ellos funcionando al mismo tiempo)
- Usa una instancia separada por cada comunicación al servidor con lo cual lo hace más segura la comunicación
- El registro y la eliminación del registro del nodo del dispositivo se realizan cuando el proceso se inicia o se cierra respectivamente
- La seguridad de múltiples hilos
- Sin necesidad de privilegios de root

FFmpeg es una colección de software que puede grabar, convertir (transcodificar) y hacer streaming de audio y vídeo. En este proyecto se

utilizó por ser GNU, liviano y poseer la herramienta ffplay, la cual es utilizada para reproducir el audio enviado desde la App

Diagrama en Bloques:



Capturas de la Aplicación:

