



Universidad
Politécnica
de Cartagena

UPCT

FINAL YEAR PROJECT

**Diseño y desarrollo de un
sistema de presencia y
medición de ruido ambiente
empleando dispositivos IoT**

Ricardo Carrillo García

Directores:

RAFAEL ASOREY CACHEDA,
JOSÉ MARTÍNEZ MONTORO

23 de julio de 2021

Índice

1	Introducción	7
2	Objetivos	8
2.1	Estado del arte	8
3	Desarrollo	9
3.1	Herramientas Utilizadas	9
3.1.1	LoPy4	9
3.1.2	Expansion Board 3.1	10
3.1.3	Sensor RK300-06A	10
3.1.4	Convertor DC ajustable 4-12V	11
3.1.5	Convertor Sparkfun MAX232	12
3.1.6	Wireshark OUI Lookup Tool	13
3.1.7	NetAnalyzer	13
3.2	Fases del desarrollo	14
3.2.1	Instalación del entorno de trabajo con Pycom	14
3.2.2	Implementación de los módulos necesarios	16
3.2.3	Desarrollo de scripts	17
3.2.4	Ajuste de los parámetros de los scripts	18
3.2.5	Recogida de datos y graficación	18
4	Flujograma del script	19
4.1	Establecimiento de la comunicación LoRa	19
4.2	Recopilación de datos para la construcción de los diccionarios OUI	20
4.2.1	Fabricantes que destinan OUIs para dispositivos móviles	20
4.2.2	Fabricantes que destinan OUIs para dispositivos móviles	21
4.3	Configuración WLAN	21
4.4	Rutina de atención a los paquetes	23
4.5	Definición de la clase micrófono	24
4.6	Representación gráfica del flujograma del diseño	25
4.7	Aternativa de trabajo con Bluetooth	27
5	Resultados obtenidos y validaciones gráficas	28
5.1	Funcionamiento general	28
5.2	Aumento de personas sin aumento de ruido	29
5.3	Aumento elevado del ruido	30
6	Presentes limitaciones	31

6.1	Estándares de comunicación	31
6.1.1	Tecnología WLAN	31
6.1.2	Tecnología Bluetooth	31
6.2	Tecnología MAC	31
6.3	Sensorización del sonido	32
7	Conclusiones y futuras líneas de trabajo	33
7.1	Deducciones del trabajo	33
7.2	Futuros desarrollos	33
8	Bibliografía	35
8.1	Wireshark OUI Lookup Tool	35
8.2	IEEE Lookup Standard	35
8.3	Documentación oficial de Pycom	35
8.4	Foro oficial de usuarios de Pycom	35
8.5	Página oficial de LoRa Alliance	35
8.6	Artículos relativos de Wikipedia	35
9	Anexo	37
9.1	main.py	37
9.2	helper.py	56
9.3	Carpeta lib	62
9.3.1	microfono.py	62

Índice de figuras

Figura 1	Procesador LoPy4	9
Figura 2	Expansion Board 3.1	10
Figura 3	Sensor RK300-06A	11
Figura 4	Convertor DC ajustable 4-12V	11
Figura 5	Convertor de RS	12
Figura 6	Convertor de RS	12
Figura 7	Disposición de la consola de Atom	15
Figura 8	Cableado del convertor DC	16
Figura 9	Cableado del convertor RS-232 to TTL	17
Figura 10	Instancia de los parámetros LoRa	19
Figura 11	Envío de los datos por LoRa	19
Figura 12	Muestra de ejemplo de los diferentes grupos de LG	20
Figura 13	Flujograma de la recogida de MACs	25
Figura 14	Flujograma de la estimación de número de personas	26
Figura 15	Monitorización de personas por la mañana	28
Figura 16	Monitorización del ruido por la mañana	28
Figura 17	Monitorización de personas por la tarde	29
Figura 18	Monitorización de personas por la tarde	29
Figura 19	Monitorización de personas por la tarde	30
Figura 20	Monitorización de personas por la tarde	30

Índice de cuadros

Tabla 1	Esquema de conexionado de los cables/pines del con-	
	versor TTL	17
Tabla 2	Tipos de paquetes distinguidos por la WLAN.	22
Tabla 3	Tabla Estimación personas en base a dB	24

1. Introducción

Con el actual auge que están viviendo las diferentes tecnologías IoT, con tantas opciones en cuanto a sensorización para recabar datos, con tantas facilidades para la transmisión de datos (Wifi, Bluetooth, LoRa...) cada vez más estandarizadas, es inevitable buscar soluciones a situaciones que se dan en el día a día y pueden facilitar o agilizar labores que aún se continúan llevando a cabo de formas, de algún modo obsoletas, y que merece la pena replantearse con la finalidad de impulsar este tipo de tecnologías.

Relativamente novedosos, los dispositivos Pycom tienen la oportunidad de revolucionar los diversos sistemas implementados en cualquier infraestructura objetivo con el fin de desarrollar novedosos sistemas IoT.

Este trabajo propone un sistema en el que se contabiliza el número de personas que se encuentran en un determinado habitáculo y sus proximidades, solución que puede resultar de utilidad en diversas situaciones y presentar una alternativa más moderna a situaciones típicas como el pasar lista o tener que firmar la asistencia en clases prácticas de esta nuestra universidad, hasta reemplazar un sistema de ficha para controlar la entrada y salida de los trabajadores de una empresa.

Cabe por supuesto mencionar, en los actuales tiempos de covid, la necesidad que ha surgido de distribuir mejor el espacio interpersonal y controlar las aglomeraciones de gente que también pueden encontrar en el desarrollo de este TFG posibles soluciones a los problemas ya planteados.

Relativamente novedosos, los dispositivos Pycom tienen la oportunidad de revolucionar los diversos sistemas implementados en cualquier infraestructura objetivo con el fin de desarrollar novedosos sistemas IoT.

2. Objetivos

El principal objetivo del trabajo es el de monitorizar y contabilizar el número de personas que encuentran en un determinado habitáculo.

Para ello se ha optado por diferentes métodos que serán discutidos más adelante, pero el estudio se fundamenta en la realización de un 'wifi sniffer' que tendrá en cuenta las MACs de los dispositivos que encuentre conectados a las diferentes redes disponibles que estén al alcance del dispositivo Pycom.

Inicialmente se implementará un sensor de ruido (modelo RK300-06A de la marca RIKA) con el objetivo de poder medir el ruido ambiente en tiempo real, lo que complementará el posterior desarrollo de software dedicado a la monitorización de las MACs.

2.1. Estado del arte

Hasta el momento se han desarrollado funcionalidades similares mediante diferentes módulos de RaspBerry aunque sin mucho éxito.

La capacidad de contabilizar y, principalmente diferenciar el tipo de dispositivos que identifican a las personas que los portan. Esto supone la principal casuística del proyecto, que será explicado en las próximas secciones.

En gran parte dependemos de la propia organización de la empresa en cuanto a distribución de sus OUI, así como el comportamiento de los dispositivos en la red.

El reto principal consiste en diferenciar y separar los dispositivos que pueden ser identificados como personas físicas de los que no, atendiendo a diferentes parámetros relacionados con el propio comportamiento de una persona para un caso de estudio enfocado a la actividad correspondiente a la vida en una universidad o una empresa.

3. Desarrollo

El hecho de usar un dispositivo Pycom es debido a que se trata de una tecnología más accesible, por su precio, su facilidad de programación (se emplea el lenguaje MicroPython) ya que es bastante fácil de aprender si se conoce Python y se emplea un número más reducido de librerías aunque suficiente para la potencial de los chips de Pycom.

Para ello se ha desarrollado un script que constantemente evalúa la dirección MAC de los diferentes dispositivos de la red conforme envían paquetes. Paralelamente, se ha implementado un sonómetro que actuara de apoyo para concretar de manera más exacta el número de personas que se encuentran en la habitación.

3.1. Herramientas Utilizadas

En esta sección se discutirá sobre las diferentes tecnologías, así como los programas y sitios web que han sido necesarios en el desarrollo del proyecto.

3.1.1. LoPy4

Desarrollado por Pycom, es un chip programable, en lenguaje MicroPython, capaz de funcionar con Bluetooth, LoRa, Sigfox y/o Wifi. De este modo es procesador más que capaz para llevar a acabo la recepción y el envío de datos desde el sensor hasta el servidor LoRa además de acarrear con la ejecución del conteo de las MACs vía Wifi.

Viene además con un puerto para acoplar una antena LoRa con el fin de evitar problemas de cobertura.

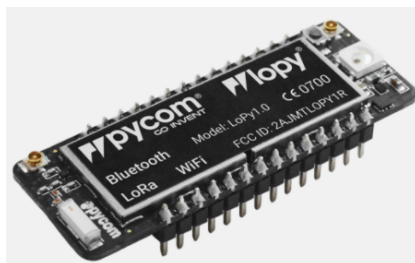


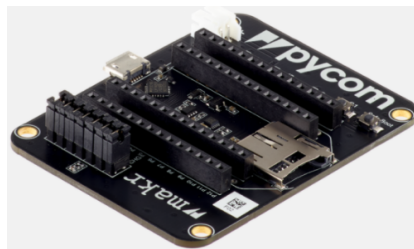
Figura 1: Procesador LoPy4

3.1.2. Expansion Board 3.1

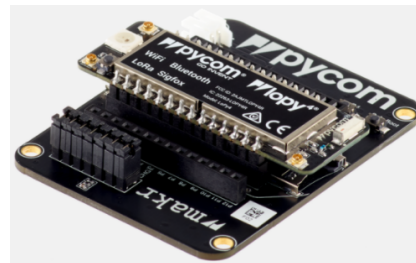
Otro producto de la empresa del IoT, Pycom, se trata de la placa a la que irá conectado el procesador, supone la interfaz entre el LoPy y tanto el ordenador como el sensor.

La comunicación con el ordenador se hará mediante cable USB, al puerto microUSB que se dispone en la parte frontal de la placa. De este modo se cargan los scripts en la memoria del procesador directamente desde el entorno de programación.

Para la comunicación con el sensor se utilizarán pines designados para la conexión del cableado relativo a la UART, y los pines de conector tierra (*GND*) y 5 voltios (*VCC*) para la alimentación del dispositivo y el conexaso del circuito.



[Expansion Board sola]



[Expansion Board con LoPy4 acoplado]

Figura 2: Expansion Board 3.1

3.1.3. Sensor RK300-06A

Unidad desarrollada por la empresa China RIKA, formado por un medidor de sonido modular/digital con multi-función (leer medición del ruido ambiente, leer dirección del dispositivo, cambiar dirección del dispositivo, etc).

Se caracteriza por su tecnología de detección y procesado digital de la señal, lo que le otorga un elevado rango de actuación; se considera un aparato

diseñado para exteriores por lo que su implementación en aulas y naves de gran capacidad no supondría problema alguno.

Puede emplear tanto el protocolo MODBUS RS-485 como el RS-232 propio de la UART. Debido a que el protocolo MODBUS está menos generalizado en cuanto a la tecnología de Pycom, ya que pese a disponer de su propia librería y repositorio en GitHub de manera oficial, aún sigue en vías de desarrollo; se ha decidido optar por funcionar con la UART que supone una manera bastante más sencilla de configurar la comunicación.



Figura 3: Sensor RK300-06A

3.1.4. Conversor DC ajustable 4-12V

Atendiendo a los requerimientos de alimentación del dispositivo sonómetro, facilitados en el datasheet, de entre 12 y 24 voltios, fue necesario incorporar al diseño este conversor porque de la placa solo podían sacarse 5 voltios.

El circuito es bastante sencillo, consiste en un potenciómetro que con el que se puede ajustar el voltaje de salida a partir del de entrada enroscando o desenroscando un tornillo situado en la parte central.

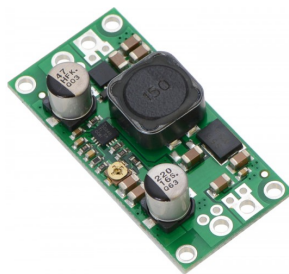


Figura 4: Conversor DC ajustable 4-12V

3.1.5. Conversor Sparkfun MAX3232

Pese a que el protocolo RS-232 es completamente compatible con la UART provista por el Pycom, si que ha sido necesario incorporar este módulo al diseño con la finalidad de que el Pycom se 'entienda' con el sensor de ruido ya que los datos que el sonómetro a 12 voltios no podían ser correctamente procesados por la placa, la cual funciona solamente a 5 voltios.

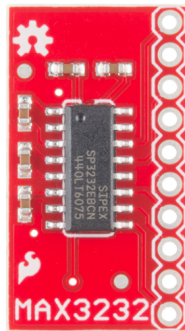


Figura 5: Conversor de RS

Finalmente, todo el dispositivo desarrollado habiendo instalado los dos módulos y con el sensor de ruido completamente implementado, el sistema quedaría configurado de la siguiente manera:

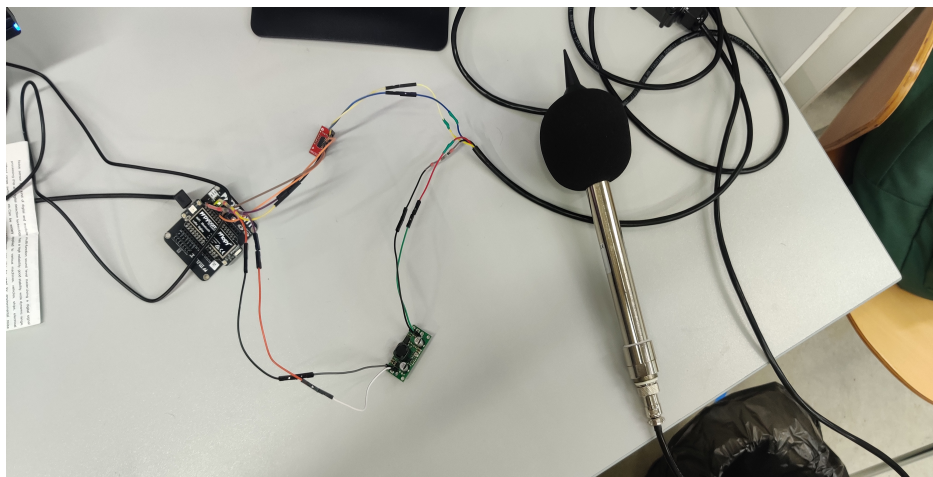


Figura 6: Conversor de RS

3.1.6. Wireshark OUI Lookup Tool

Se trata de una extensión configurable en el software de Wireshark que también cuenta con su propio cliente online, lo que agiliza su uso.

Consiste en una web que realiza consultas a una BBDD, que ciertamente se encuentra algo desactualizada pero contiene información más que suficiente para desarrollar la funcionalidad.

Permite tanto consultar las OUI específicas de la dirección MAC de un dispositivo, en cuyo caso se devuelve el fabricante al que pertenece. También es posible consultar el nombre del fabricante deseado para obtener todas las OUI correspondientes a dicho 'vendor'.

3.1.7. NetAnalyzer

Aplicación utilizada en mi dispositivo Android, su versión de iOS está más capada en cuanto a reconocimiento de direcciones MAC, que permite recoger bastante información sobre la red a la que estamos conectados, así como del resto de dispositivos que también se encuentran conectados a la misma red que nosotros.

La principal funcionalidad que ha servido como apoyo a este TFG ha sido la de 'LAN Scan', como su propio nombre indica, nos permite hacer un barrido de todas las direcciones IP de la red, facilitándonos: evidentemente su dirección IP, el nombre del dispositivo en caso de ser visible para los demás, también incluye su dirección MAC y lo que es de gran utilidad, nos facilita el fabricante asociado al OUI de la MAC.

3.2. Fases del desarrollo

3.2.1. Instalación del entorno de trabajo con Pycom

Con la finalidad de poder programar y depurar el funcionamiento de nuestra unidad Pycom, será necesaria la instalación del entorno de programación de VS Code o de Atom, como hemos mencionado antes, se ha seleccionado Atom por su gestión de los archivos parchivos de código fuente principalmente.

En primer lugar será necesario asegurarnos de que nuestro equipo reconoce el dispositivo Pycom, conectándolo con un cable de micro-USB a USB accedemos al administrador de dispositivos y localizamos nuestro dispositivos en el apartado de dispositivos serie (COM), que es por donde se realizará la comunicación entre el PC y el LoPy.

A la hora de instalar Atom desde su web oficial (<https://atom.io/>), por la experiencia de éste TFG se conoce que la última versión hasta la fecha, la 1.57, no funciona correctamente con un módulo que es necesario instalar para poder trabajar con el Pycom. La versión de trabajo óptima para este fin es la 1.55.

Una vez instalado, será necesario acceder a la pestaña de 'Install a Package' y buscamos el módulo 'Pymkr', desarrollado por el equipo de Pycom y que incluye todas las librerías esenciales para trabajar con nuestro dispositivo.

Hecho ésto, ya podemos disponernos a programar nuestro dispositivo, seleccionamos la pestaña 'File' y clicamos en 'Open Folder' y seleccionamos la carpeta en la que se encuentren los archivos de nuestro proyecto, será estrictamente necesario que todos los archivos tengan la extensión '.py' y además que el archivo que se va a ejecutar se llame 'main.py', el resto de scripts que se desarrollen pueden colocarse en una carpeta interna y ser importados como paquetes externos para utilizar las funciones definidas en ellos.

También es necesario conocer las funcionalidades que ofrece la consola instalada por el módulo Pymkr:

-pestaña '*Connect device*': Pestaña que nos permite seleccionar el puerto serie por el que queremos enviar el programa y consecuentemente programar la unidad, será necesario conocer que puerto le corresponde en caso de tener varios Pycoms conectados al mismo ordenador.

-botón *'Connect/Disconnect'*: Botón que sirve para habilitar el dispositivo seleccionado, sirve tanto como para parar la ejecución como para conectar el dispositivo en otra ventana y poder ejecutar otro script.

-botón *'Run selected file'*: Sirve para ejecutar el script que esté seleccionado y desplegado en la ventana, el script no se guarda en la propia memoria del Pycom.

-botón *'Upload project to device'*: Sirve para cargar todo el conjunto de scripts que hay en nuestra carpeta para el funcionamiento del proyecto. Todos los archivos son almacenados en la memoria del Pycom y comienza su ejecución, cada vez que se inicie el dispositivo Pycom comenzará la ejecución de su programa en memoria.

Aquí reside una de las mayores ventajas de esta tecnología, una vez esté completamente programado y funcionando, es posible emplazarlo en la ubicación deseada y, siempre y cuando sea correctamente alimentado con cualquier tipo de batería, continuará ejecutando su programa.

Cada vez que se carga un software, se sobrescribe la memoria del Pycom.

-botón *'Download from device'*: Sirve para descargar los archivos contenidos en la memoria del dispositivo Pycom. Es útil para no perder los contenidos en memoria de un Pycom.

-botón *'Get device info'*: Utilizado para obtener información del dispositivo tal como su dirección MAC, su LoRa ID, la versión de su firmware, etc.

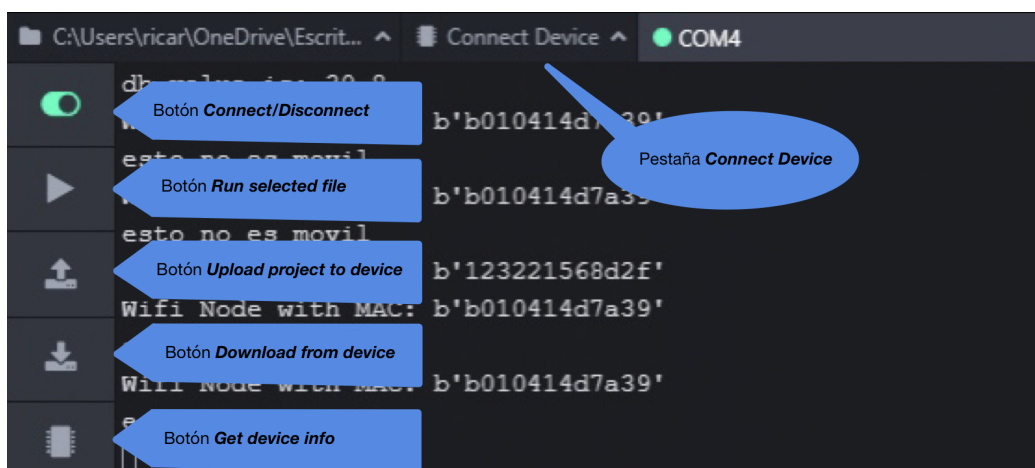


Figura 7: Disposición de la consola de Atom

3.2.2. Implementación de los módulos necesarios

El primer módulo implementado fue el Conversor DC con la finalidad de poder alimentar el sensor de ruido, se ajusta al valor máximo que puede alcanzar de 12 voltios ya que el sonómetro funciona entre 12 y 24 voltios. Se conectarán el pin de *GND* y *VCC* de la placa en la entrada del módulo y la salida se conecta con los cables correspondientes propios del sensor.

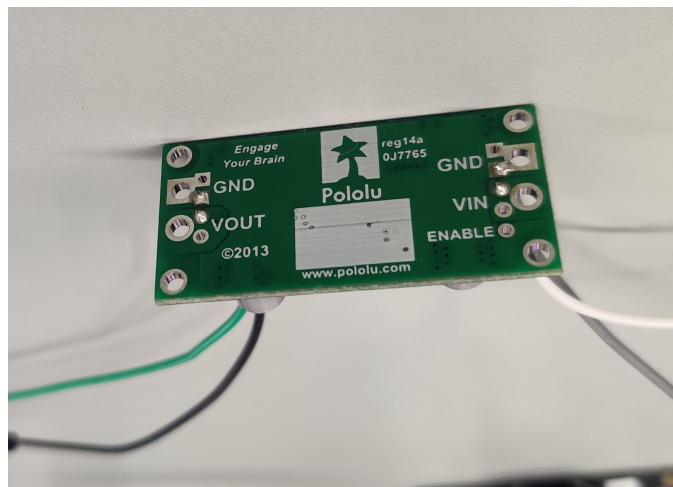


Figura 8: Cableado del conversor DC

Como se comprueba en la *Figura 8*, a los pines de *GND* van las correspondiente conexiones a tierra de la placa y el sensor, y al pin de *Vin* se conectan los 5 voltios procedentes de la placa, mientras que el pin *Vout* se conectará el cable de alimentación del sensor.

Para el segundo módulo, que adapta los datos para ser procesados por la placa. El cableado para conseguir tal fin es el mostrado en la *Figura 9*:

Cuadro 1: Esquema de conexionado de los cables/pines del conversor TTL

Cable	Función
Amarillo	Transmisión de datos del sensor de sonido.
Azul	Recepción de datos del sensor de sonido.
Naranja (1)	Alimentación VCC de la placa. De 5 a 3.3 voltios.
Negro	Conectado a GND de la placa.
Naranja (2)	Transmisión de datos de Pycom.
Marrón	Recepción de datos de Pycom.

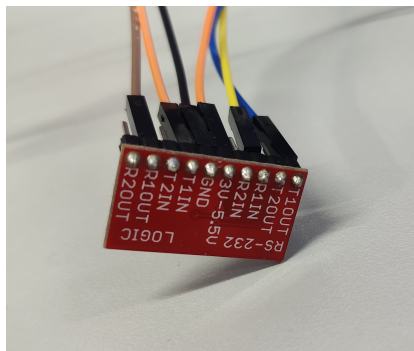


Figura 9: Cableado del conversor RS-232 to TTL

3.2.3. Desarrollo de scripts

Una vez con los dispositivos en funcionamiento y la comunicación siendo posible, se pasó al desarrollo de los programas de software que llevarían a cabo la función deseada.

En primer lugar se trabajó en el script relativo a la lectura del valor del ruido ambiente, lo que fue bastante sencillo una vez se dispuso de todos los elementos necesarios para ello.

A continuación, se continuó con el script correspondiente al conteo de las direcciones MAC de los dispositivos encontrados en la red.

Para finalizar, la funcionalidad del primer script se implementó como funciones que el segundo script utiliza para poder obtener valores de ruido ambiente y actualizar el número de personas que se encuentran en la sala.

3.2.4. Ajuste de los parámetros de los scripts

Como veremos más adelante, los scripts dependen de una serie de variables que condicionan su funcionamiento (tiempo para la medición del ruido, número supuesto de personas por db en ruido ambiente...).

El valor actual con el que transcurre la ejecución del software en su versión actual, ha sido determinada por la experiencia de las sucesivas pruebas que se han llevado a cabo, teniendo en cuenta las limitaciones de los dispositivos como sus capacidades de medida y de procesamiento y también las limitaciones impuestas por el actual panorama covid que evidentemente ha hecho imposible testear los dispositivos, principalmente el sensor de ruido, hasta los límites de su capacidad.

3.2.5. Recogida de datos y graficación

En la propia configuración del software se dedica un apartado para la transmisión de los datos mediante la red LoRa instalada en los laboratorios, con el Pycom recogiendo datos por las sucesivas ejecuciones durante las horas de trabajo destinadas a ello, se obtuvieron los datos de la presencia en los laboratorios que han sido representados en los paneles desarrollados con grafana.

4. Flujograma del script

A continuación se discutirá sobre las diferentes funciones y funcionalidades necesarias para conseguir la implementación objetivo.

4.1. Establecimiento de la comunicación LoRa

Para esto es necesario definir la dirección del gateway LoRa y las 'keys' para habilitar el acceso a la red LoRa del laboratorio.

Seguidamente es necesario iniciar la conexión con los métodos determinados para la clase LoRa y se crean los 'sockets' necesarios para la transmisión de los datos.

```
dev_addr = struct.unpack(">I", binascii.unhexlify('0030a086'))[0]
nwk_swkey = binascii.unhexlify('4a38a1b7c0fa8502d4864d6045b2d1a6')
app_swkey = binascii.unhexlify('80da10d9e61c157e57d2f366e315e596')
g_tx_power = 14
g_coding_rate = 1
g_data_rate = 4

lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868, adr = False, tx_retries=1, device_class=LoRa.CLASS_A)
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))

s = create_socket(data_rate = g_data_rate)

s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, True)
```

Figura 10: Instancia de los parámetros LoRa

Una vez acabada la ejecución de todo el script y la consiguiente recogida de los datos y la estimación de personas, se envía por la red LoRa todo el conjunto de los datos y las estadísticas telemétricas de la red.

```
data = struct.pack("id", personitas, db)

try:
    s.send(data)
    print(lora.stats()) #Con esto puedo ver el RSSI y SNR y podria ser interesante para lo de la red de sensores en malla
except Exception as e:
    print("")
```

Figura 11: Envío de los datos por LoRa

4.2. Recopilación de datos para la construcción de los diccionarios OUI

El OUI (organizationally unique identifier) consiste en los primeros tres octetos de una dirección MAC, 24 bits que identifican a cada empresa o 'vendedor' responsable de la fabricación de un dispositivo determinado. Todos y cada uno de estos OUIs están reglados y distribuidos por la IEEE.

Cada empresa dispone de diferentes OUIs, que no son más que prefijos identificativos, y las diferentes combinaciones para los octetos restantes con sus determinados OUIs.

Con la herramienta 'Wireshark OUI Lookup' junto con 'NetAnalyzer' y los documentos disponibles en la página web de la IEEE, se han desarrollado los diccionarios usados para comprobar la cabecera de las MACs que va detectando el Pycom la finalidad de identificar el fabricante y el tipo de dispositivo del que se trata.

Con el fin de identificar a las personas se ha optado por identificar los dispositivos móviles 'smartphones', el problema es que se diferencian dos tipos de 'vendedores' en cuanto a la distribución de las OUIs que tienen para con los móviles.

4.2.1. Fabricantes que destinan OUIs para dispositivos móviles

Xiaomi o **LG** son marcas comerciales que destinan sus OUIs a diferentes dispositivos por lo que existen grupos diferenciados de los que podemos conocer sus respectivas cabeceras, por lo que podemos estar seguros de que se trata de un dispositivo móvil.

```
B0:37:95 LG Electronics
B4:B2:91 LG Electronics
B4:F1:DA LG Electronics (Mobile Communications)
B4:F7:A1 LG Electronics (Mobile Communications)
B8:1D:AA LG Electronics (Mobile Communications)
BC:F5:AC LG Electronics (Mobile Communications)
C0:41:F6 Lg Electronics Inc
```

Figura 12: Muestra de ejemplo de los diferentes grupos de LG

Si bien es cierto que para otros fabricantes como **Samsung**, diferencia sus grupos según su lugar de fabricación o el sector industrial al que pertenecen, el problema es que el grupo correspondiente a los dispositivos móviles en lo que nos centramos en este estudio, es muy amplio y abarca una variedad mayor de dispositivos, lo que dificulta en gran medida la labor de conteo.

4.2.2. Fabricantes que destinan OUIs para dispositivos móviles

En contrapunto, hay diversas marcas que no distinguen OUIs para diferentes tipos de dispositivos. En el caso de una marca como **OnePlus** carece de sentido hacer tal distinción ya que la marca solo comercializa con teléfonos móviles o 'smartphones'.

Sin embargo para otros fabricantes con una gama de productos más amplias y sobretodo, más populares y generalizados como por ejemplo **Apple**.

El principal problema que plantea ésta situación, a parte de no poder diferenciar el tipo de dispositivo, para algunos dispositivos 'smatphones' de esta marca, los 'iPhones', se desconoce su OUI, es decir, ni las herramientas de 'NetAnalyzer' y 'Wireshark' reconocen tal cabecera, ni la propia IEEE las dispone en la documentación disponible.

4.3. Configuración WLAN

Para que el dispositivo Pycom pueda reconocer la información que circula por las redes próximas, es necesario programar la los ajustes de WLAN mediante el módulo de nombre homónimo.

Será necesario configurarlo en modo 'Station', con lo que su comportamiento se asemejará a un repetidor de señal, por lo cuál será posible procesar y monitorizar los sucesivos paquetes que generan los dispositivos conectados a la red.

De igual manera se definirá el evento de recepción de paquetes con los que se activará la monitorización, así como la función que se utilizará para atender a los paquetes.

Con tal fin, pasamos a reconocer que tipo de paquetes pueden ser monitorizados y cuál es más interesante de cara a nuestro estudio; teniendo en

cuenta que será necesario trabajar en modo promiscuo con la red, para poder monitorizar los paquetes desde nuestro dispositivo, éstos son los tipos de paquetes que podemos encontrar:

Cuadro 2: Tipos de paquetes distinguidos por la WLAN.

Tipo de paquete	Descripción
Administración	Capturará los paquetes relativos a la administración de la red.
Control	Contabilizará paquetes propios de control de la red (flujo, admisión, etc.).
Datos	Dedicado a los paquetes de datos que circulan por la red.
MPDU	Tramas Ethernet juntas en la misma trama con cabecera MAC.
AMPDU	Paquetes que incluyen en su cabecera la MAC de origen y/o destino.
Miscelania	Tipo de paquetes con su información cifrada y oculta de cara a terceros.
Cualquiera	De este modo, se capturarán cualquier tipo de paquetes de los anteriores.

Como se aprecia, lo más lógico sería decantarse por los paquetes de **Administración** ya que en el momento que un dispositivo solicita acceder a la WLAN, generará ese tipo de paquete. En ese instante podemos recoger la dirección MAC de dicho paquete y tratarla en nuestro sistema.

Desafortunadamente, en la práctica con el resto de opciones de paquetes no se han conseguido resultados satisfactorios, desde no poder monitorizarlos aún teniendo un uso normal de la red sin ningún percance hasta no poder extraer información relevante del contenido de los paquetes debido a su configuración.

Consecuentemente, con la selección de tipo **Cualquiera**, lo que se hace es abarcar todos y cada uno de los paquetes sin distinción; con las dificultades que presentan el resto de paquetes por separado, se acaba resultando en la monitorización de los paquetes de tipo **Administración**.

4.4. Rutina de atención a los paquetes

Se trata de una función que reconoce el paquete recibido y extrae la información del mismo; estos paquetes contienen varios campos como el 'rssi', 'channel' con el canal por el que se ha recibido, etc.

En nuestro caso, nos interesa con la información que se encuentra en los bits del 10 al 16 del campo 'data', contenido en el paquete. Si bien usaremos la información de paquete 'rssi' con la finalidad de mejorar la fiabilidad de la aplicación y ceñirnos a una zona de cobertura lo más próxima posible a nuestro dispositivo.

Acto seguido, se extraerán los 6 primeros bytes de la dirección MAC, los correspondientes a lo que sería el OUI, y se comprobará si dicho valor está incluido en alguno de los diccionario, especialmente elaborados con este propósito.

Si efectivamente coincide con el valor de alguno de los diccionarios, se incluirá el valor completo de la MAC, junto con el valor del momento en el que se incluyó la MAC como válida en la lista de direcciones a contar, a modo de 'timestamp'.

Finalmente y con el objeto de poder determinar cuando un dispositivo ha abandonado la sala, se va revisando periódicamente la lista de direcciones y se va comparando la marca de tiempo de cada una con el instante actual para determinar el tiempo que lleva conectado a la red y ayudarnos así a determinar así si se trata de un dispositivo móvil o estático como puede ser un PC que lleva conectado varias horas, más de las que lo estaría un 'smartphone'.

En ese supuesto caso, se retiraría la dirección en cuestión de la lista, cabe mencionar que una vez se determina por duración de la conexión que no procede tener en cuenta a ese determinado dispositivo, se expulsa de la lista y no formará parte de la lista hasta una próxima ejecución del sistema en su totalidad.

A continuación se realiza una comprobación idéntica pero en función a otros valores de tiempo, ya que no se trata de una diferenciación discriminatoria entre un tipo de dispositivo u otro, sino que hay que atender al comportamiento del dispositivo en cuanto a las conexiones a la red.

4.5. Definición de la clase micrófono

Una vez concretado el número de dispositivos en base al reconocimiento de su dirección MAC, se hace una llamada a un método de la clase micrófono con la que se medirá el valor medio de ruido ambiente en dBs durante un tiempo especificado en segundos y pasado como argumento a la función.

Esta función realizará una medición cada 10 segundos, la almacena y una vez ha transcurrido el tiempo especificado como argumento, calcula el valor medio correspondiente.

Con este valor medio de ruido, constituyendo un argumento de una nueva función que junto con el número de MACs reconocidas y aceptadas, que determinará un número de personas final de acuerdo a la siguiente tabla:

Cuadro 3: Tabla Estimación personas en base a dB

<i>MACs</i> \ <i>dBs</i>	30	40	50	60	70	80	90	100	110	120	130
1	1	1	1	2	3	4	5	6	7	8	9
2	2	2	2	2	3	4	5	6	7	8	9
3	3	3	3	3	3	4	5	6	7	8	9
4	4	4	4	4	4	4	5	6	7	8	9
5	5	5	5	5	5	5	5	6	7	8	9
6	6	6	6	6	6	6	6	6	7	8	9
7	7	7	7	7	7	7	7	7	7	8	9
8	8	8	8	8	8	8	8	8	8	8	9
9	9	9	9	9	9	9	9	9	9	9	9

Como se observa se ha establecido un criterio por el cuál, pese a haber contabilizado un número menor de personas previamente, ante un elevado ruido ambiente mantenido durante el tiempo especificado, se estima como resultado un mayor número de personas real, causante del aumento de los decibelios.

Los límites de 30 y 130 dBs son los propios del sonómetro, por lo que aún que haya un nivel de ruido menor, el sensor reportará un valor alrededor de 30 y de igual manera, aún superando los 130 decibelios, el sensor mostrará tal valor.

La estimación de tal incremento en el número de personas se ha hecho de forma lineal aunque lo que de verdad se intenta plasmar en cierto modo es un 'nivel de número de personas'. Suponiendo un aumento de 2/3 personas

por nivel podemos concretar el número de personas en la sala.

Si bien se hace prevalecer el número de MACs detectadas sobre el número de personas que corresponderían por decibelios, esto es debido a diferentes situaciones silenciosas que se pueden dar, como en las clases, a las que asiste un número elevado de alumnos pero es el profesor el único que habla y por lo tanto marca el nivel de ruido en la mayoría de ocasiones.

4.6. Representación gráfica del flujograma del diseño

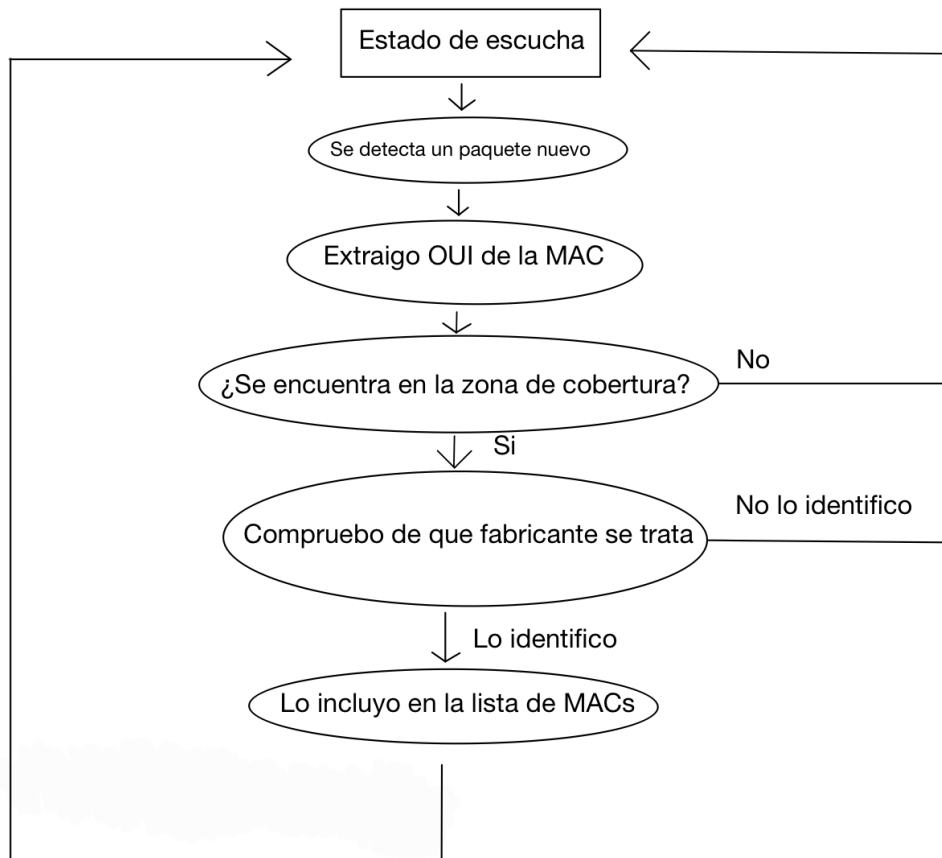


Figura 13: Flujograma de la recogida de MACs

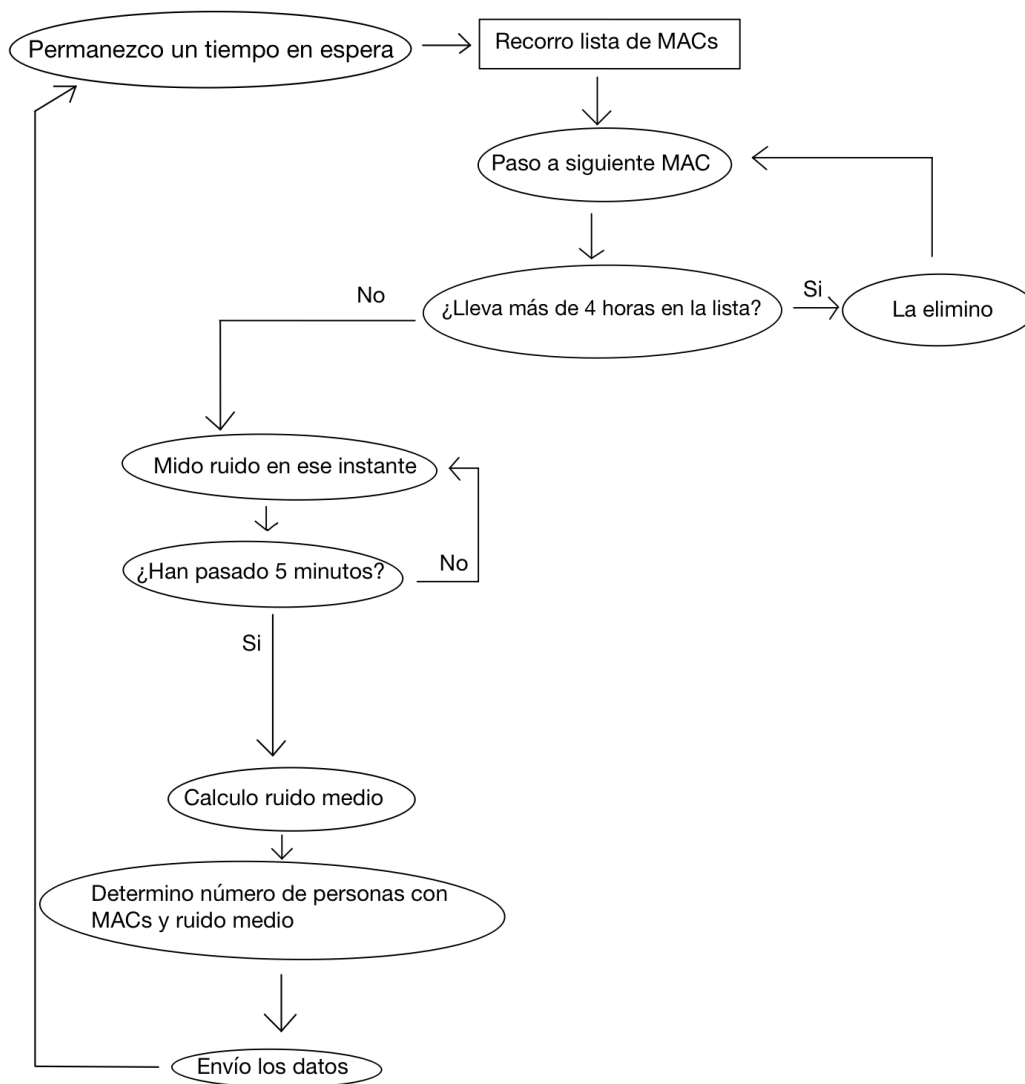


Figura 14: Flujograma de la estimación de número de personas

4.7. Alternativa de trabajo con Bluetooth

Tras algunos inconvenientes presentados por la metodología empleada en el desarrollo mediante el uso de un 'WiFi sniffer' en redes WLAN, se optó por configurar la aplicación haciendo uso del módulo Bluetooth de nuestro LoPy.

Las ventajas que proponía esta alternativa parecían suponer una óptima solución frente a las limitaciones de la propuesta anterior. Con la tecnología Bluetooth es mucho más fácil concretar el tipo de dispositivo ante el que nos encontramos ya que se puede conocer el nombre del dispositivo y los servicios que tiene activos.

Como contrapunto, suele ser más común que la gente tenga activado el WiFi y no el Bluetooth por lo que la mayoría de dispositivos serían inaccesibles para nuestro procesador. Surgió entonces la cuestión de si sería posible combinar ambos métodos para ganar una mayor fiabilidad frente a dispositivos reconocidos.

Finalmente se tuvo que desechar la idea de trabajar con Bluetooth, debido a que el módulo correspondiente de cualquier procesador de la tecnología Pycom, trabaja con la funcionalidad básica del BLE.

El BLE (Bluetooth Low Energy) es una modalidad del Bluetooth convencional introducido en la versión 4.2 del estándar. Como su propio nombre indica, se caracteriza por tener un menor consumo energético a causa de soportar únicamente transmisiones bastante básicas.

El inconveniente encontrado es que el dispositivo LoPy solo es capaz de comunicarse vía Bluetooth con otros dispositivos que dispongan de la susodicha versión Bluetooth 4.2, ya que solo trabaja con el BLE más básico, el de su versión de lanzamiento. Es por esto que no admite ni siquiera versiones posteriores. Desafortunadamente, es hasta donde llega esta tecnología, aún en fase de desarrollo por Pycom.

Con la mayoría de dispositivos móviles actuales disponiendo de la versión 5.0 de la comunicación Bluetooth, no es la opción que mejor se ajuste a nuestro cometido.

5. Resultados obtenidos y validaciones gráficas

Para la representación de los datos obtenidos se han desarrollado unos paneles con "Grafana" donde se han incluido los datos de la estimación de personas resultante de la ejecución del script y el valor medio de del ruido en decibelios que se ha extraído con el uso del sensor.

5.1. Funcionamiento general

A continuación observaremos como se monitoriza el número de personas presentes en la habitación atendiendo a la hora en la que se han recibido los datos procedentes del Pycom. Pasamos a comparar ahora la evolución del tránsito de personas a lo largo del día:



Figura 15: Monitorización de personas por la mañana

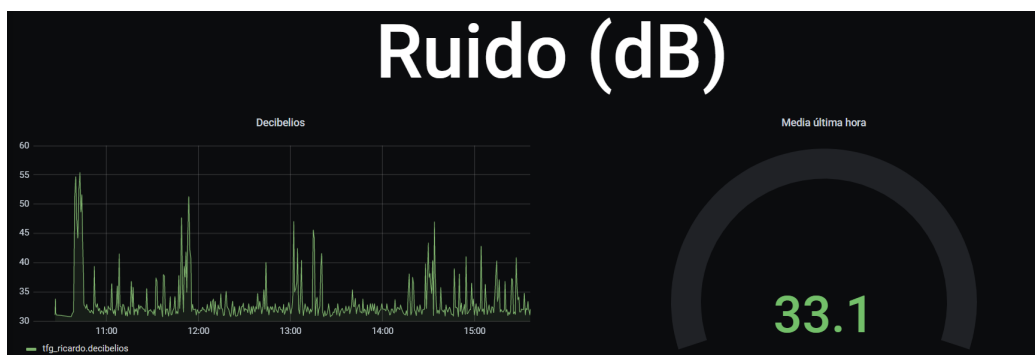


Figura 16: Monitorización del ruido por la mañana

5.2. Aumento de personas sin aumento de ruido

Seguidamente observamos el flujo de personas en avanzadas horas de la tarde donde se demuestra un mayor número de personas en la habitación y sus alrededores:



Figura 17: Monitorización de personas por la tarde

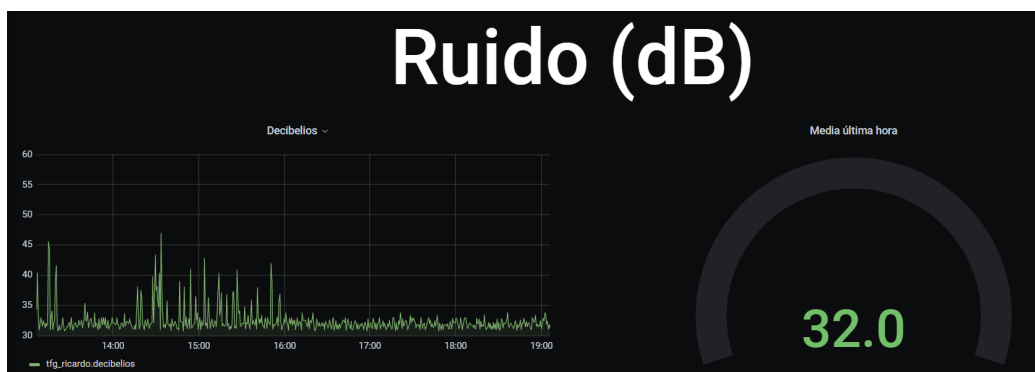


Figura 18: Monitorización de personas por la tarde

En este caso la situación que se observa consiste en el aumento de dispositivos móviles en los laboratorios colindantes, lo que explica que se haya incrementado el número de personas pero el nivel de decibelios se mantengan en niveles mínimos detectados por el sonómetro.

5.3. Aumento elevado del ruido

Se presenta a continuación un caso en el elevado numero de personas está correlacionado con el aumento del nivel de ruido de la sala, información que se encuentra plasmada en el siguiente conjunto de gráficos:



Figura 19: Monitorización de personas por la tarde

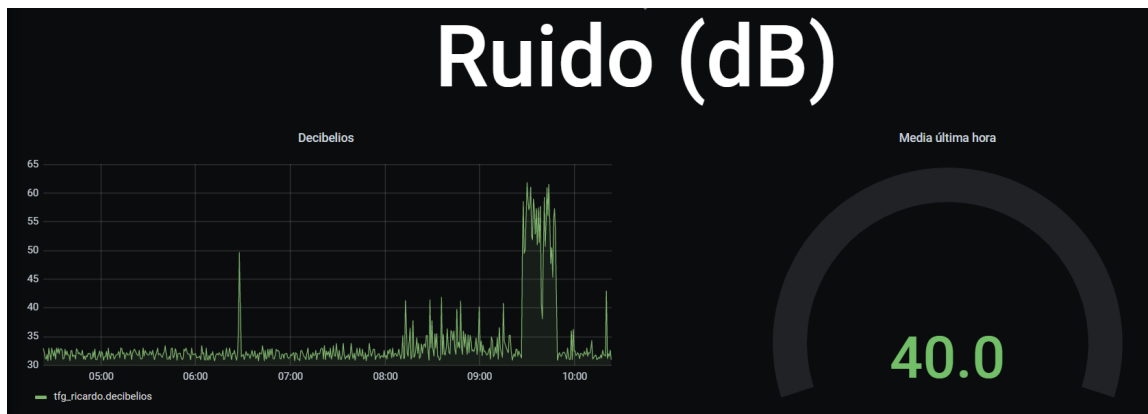


Figura 20: Monitorización de personas por la tarde

6. Presentes limitaciones

6.1. Estándares de comunicación

6.1.1. Tecnología WLAN

Al configurar la conexión WLAN, es necesario especificar el tipo de paquetes que queremos monitorizar el tipo de paquetes que nosotros queramos.

Para este trabajo se ha seleccionado monitorizar los paquetes de gestión 'management', el resto de configuraciones no ofrecen resultados del todo, la única configuración que devuelve paquetes monitorizados es la hacerlo con cualquier tipo de paquete 'any', lo que resulta en la monitorización de los paquetes 'management' y ninguno otro ya que como hemos visto por separado, no funcionan.

Principalmente, y debido a incompatibilidades con determinadas marcas, enfundadas en otorgar una mayor privacidad sus dispositivos, es posible que algunos paquetes sean inadvertidos por nuestro dispositivo.

6.1.2. Tecnología Bluetooth

Como hemos visto en el caso de la tecnología Bluetooth, solo funciona con el BLE más básico por lo que no es apropiado por la incompatibilidad con la mayoría de dispositivos móviles actuales, lo que imposibilita su reconocimiento

Sería óptimo poder comunicarse con dispositivos más modernos y con oferta de posibilidades mayor por lo que sería posible implementar diferentes dispositivos con todas y cada una de sus facetas.

6.2. Tecnología MAC

Debido a la continúa producción de dispositivos y el desarrollo de nuevos tipos de dispositivos (smartphones, smartwatches, smartTVs...) y unido a la extrema diversidad de marcas fabricantes de tales productos, hace crecer exponencialmente el número de OUIs a tener en cuenta. Siendo éstos el eje central de la investigación hace necesario disponer de una extensa 'biblioteca' con la que relacionar cada OUI con su correspondiente 'vendor'.

En algunos casos incluso, no es posible conocer el fabricante responsable del dispositivo meramente al obtener la cabecera de su MAC; y mucho me-

nos determinar el tipo de dispositivo con lo que concretar si se trata de una persona o no.

Sería favorable poder utilizar otro parámetro como apoyo, y como hemos visto se intentó emplear la comunicación Bluetooth aunque sin resultados positivos debido a las presentes limitaciones de Pycom.

Para poder obtener una mayor fiabilidad con el desarrollo actual, se precisaría una constante actualización y revisión de los medios que facilitan las relaciones de las marcas para con los OUI.

6.3. Sensorización del sonido

Apoyar la detección de direcciones MAC con la sensorización del sonido, ha constituido resultados muy positivos, si bien para un número de personas muy elevado, la fiabilidad del producto se ve reducida para tal fin.

Por lo que es necesario tener en cuenta que para cierto umbral de ruido, el número de personas puede dispararse en relación a las personas contabilizadas por nuestro sistema.

7. Conclusiones y futuras líneas de trabajo

Con los extractos de las presentes limitaciones y los resultados dibujados durante el desarrollo del estudio, se ha dado lugar a un sistema fiable que permite llevar un conteo actualizado del número de personas que se encuentran en una determinada sala.

Si bien es cierto que las presentes faltas en el sistema son producto de las limitaciones ya comentadas. Por lo que el ajuste de parámetros y elecciones de diseño atienden a un criterio regido por la coetánea investigación que ha conllevado el trabajo.

7.1. Deducciones del trabajo

Conociendo la situación actual de desarrollo de las diferentes tecnologías empleadas, se concibe como necesaria la inversión en el desarrollo con la resultante evolución de las plataformas usadas. Todo ello confluirá en el un entorno más favorecedor para nuevos y más complejos desarrollos que supondrán estudios y nuevas tecnologías con una serie de fines tan numerosos como se pueda plantear.

El objeto de éste trabajo, paralelamente al desarrollo de un sistema para la monitorización de personas con dispositivos IoT para sus ya mencionados propósitos; es el de conocer y familiarizarse con estos, personalmente, nuevos dispositivos a los que no se ha podido acceder durante los estudios de grado y que satisfactoriamente puede dar a conocer en su justa medida, el potencial que tienen los dispositivos utilizados.

7.2. Futuros desarrollos

Con el fin de mejorar el sistema desarrollado en este estudio y de cara al desarrollos futuros, sería más que interesante proponer una adaptación el emplazamiento u organismo donde se pretenda incorporar nuestro sistema.

La adaptación consistiría en hacer un reconocimiento de las direcciones MAC de los dispositivos y equipos que se conoce, serán utilizados por las personas que se encuentren en el emplazamiento en cuestión.

De ésta manera en vez de tener que evaluar cualquier paquete percibido por el dispositivo y necesitar de comprobar la marca y el tipo de dispositivo en la documentación para ello desarrollada, bastaría simplemente con incluir las cabeceras MAC de equipos mencionados.

Haciendo esta información accesible al dispositivo no supondría problema relacionados con la seguridad y integridad de los datos disponibles en los dispositivos ya que, únicamente sería necesario conocer un conjunto de OUIs determinado por el conglomerado de dispositivos a contabilizar. Con esta información no se puede ver comprometida la integridad de los aparatos.

Alternativamente podría complementarse nuestro sonómetro con algún otro medio de sensorización complementario al ruido con el fin de establecer otro dato/parámetro que nos ayude a concretar de manera más precisa el número de personas ubicadas en la sala.

También sería posible sustituir el sensor actual con, tal vez uno de rango más preciso, o bien se podría optar por una solución distinta de sensorización como puede resultar ser un sensor de presencia; que cuente el tránsito de personas a la entrada de la sala, lo que plantearía una casuística diferente con sus correspondientes limitaciones y cuestiones.

las implicaciones

8. Bibliografía

8.1. Wireshark OUI Lookup Tool

<https://www.wireshark.org/tools/oui-lookup.html>

8.2. IEEE Lookup Standard

<http://standards-oui.ieee.org/oui/oui.txt>

8.3. Documentación oficial de Pycom

<https://docs.pycom.io/>

<https://docs.pycom.io/firmwareapi/pycom/machine/uart/>

<https://docs.pycom.io/firmwareapi/pycom/network/wlan/>

<https://docs.pycom.io/firmwareapi/pycom/network/lora/>

<https://docs.pycom.io/firmwareapi/pycom/network/bluetooth/>

8.4. Foro oficial de usuarios de Pycom

<https://forum.pycom.io/>

8.5. Página oficial de LoRa Alliance

<https://lora-alliance.org/>

8.6. Artículos relativos de Wikipedia

<https://es.wikipedia.org/wiki/Modbus>

<https://es.wikipedia.org/wiki/RS-232>

https://es.wikipedia.org/wiki/Bluetooth_de_baja_energ%C3%ADa

9. Anexo

9.1. main.py

```
import machine
import pycom
import time
from network import Bluetooth
from network import LoRa
from network import WLAN
import socket
import binascii
import ubinascii
from helper import *
from time import sleep
from machine import UART
from machine import I2C
from machine import SPI,Pin
import struct
import _thread
from microfono import *

dev_addr = struct.unpack(">l", binascii.unhexlify('0030a086'))[0]
nwk_swkey = binascii.unhexlify('4a38a1b7c0fa8502d4864d6045b2d1a6')
app_swkey = binascii.unhexlify('80da10d9e61c157e57d2f366e315e596')
g_tx_power = 14
g_coding_rate = 1
g_data_rate = 4

lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868, adr = False, tx_retries=1, dev
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))

s = create_socket(data_rate = g_data_rate)

s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, True)

# def lora_cb(lora):
#     events = lora.events()
#     if events & LoRa.RX_PACKET_EVENT:
```

```

#         print('Lora packet received')
#     if events & LoRa.TX_PACKET_EVENT:
#         print('Lora packet sent')
#     if events & LoRa.TX_FAILED_EVENT:
#         print('Lora packet failed')
#
# lora.callback(trigger=(LoRa.RX_PACKET_EVENT | LoRa.TX_PACKET_EVENT | LoRa.TX_F

##Presencia#####
macs = []

dict_xiaomi = {
1:b'009ec8',2:b'00ec0a',3:b'04b167',4:b'04c807',5:b'04d13a',6:b'04e598',
7:b'081c6e',8:b'082525',9:b'0c1daf',10:b'0c9838',
11:b'0cf346',12:b'102ab3',13:b'103f44',14:b'14f65a',15:b'1801f1',16:b'185936',
17:b'188740',18:b'18f0e4',19:b'1cccd6',20:b'2034fb',
21:b'2047da',22:b'2082c0',23:b'20a60c',24:b'20f478',25:b'241145',26:b'28167f',
27:b'28e31f',28:b'2cd066',29:b'341cf0',30:b'3480b3',
31:b'38a4ed',32:b'38e60a',33:b'482ca0',34:b'48fda3',35:b'4c0220',36:b'4c49e3',
37:b'4c6371',38:b'4cf202',39:b'503dc6',40:b'508e49',
41:b'508f4c',42:b'509839',43:b'50a009',44:b'582059',45:b'584498',46:b'5cd06e',
47:b'606ee8',48:b'60ab67',49:b'640980',50:b'64a200',
51:b'64b473',52:b'64cc2e',53:b'64dde9',54:b'68dfdd',55:b'703a51',56:b'705fa3',
57:b'70bbe9',58:b'741575',59:b'742344',60:b'7451ba',
61:b'7802f8',62:b'7c035e',63:b'7c03ab',64:b'7c1dd9',65:b'7c2adb',66:b'7cd661',
67:b'7cfd6b',68:b'8035c1',69:b'80ad16',70:b'884604',
71:b'8caace',72:b'8cbebe',73:b'8cd9d6',74:b'9078b2',75:b'941700',76:b'9487e0',
77:b'98f621',78:b'98fae3',79:b'9c28f7',80:b'9c2ea1',
81:b'9c5a81',82:b'9c99a0',83:b'9cbcf0',84:b'a086c6',85:b'a44519',86:b'a44bd5',
87:b'a45046',88:b'a45590',89:b'a89ced',90:b'acc1ee',
91:b'acf7f3',92:b'b0e235',93:b'b4c4fc',94:b'b83bcc',95:b'bc7fa4',96:b'c40bcb',
97:b'c46ab7',98:b'c83ddc',99:b'd09c7a',100:b'd4970b',
101:b'd832e3',102:b'd86375',103:b'd8ce3a',104:b'dcb72e',

105:b'e01f88',106:b'e06267',
107:b'e0ccf8',108:b'e0dcff',109:b'e446da',110:b'e85a8b',
111:b'ecd09f',112:b'f0b429',113:b'f4308b',114:b'f460e2',
115:b'f48b32',116:b'f4f5db',
117:b'f8a45f',118:b'fc1999',119:b'fc64ba',120:b'fcd908',
121:b'bc6ad1',122:b'8c7a3d'
}

```

```

dict_oneplus = {
1:b'4801c5',2:b'4c4fee',3:b'5c17cf',4:b'64a2f9',5:b'8c64a2',6:b'94652d',
7:b'9809cf',8:b'a091a2',9:b'ac5fea',10:b'acd618',
11:b'c0eefb',12:b'd0497c',13:b'e44122'
}

dict_LG = {
1:b'001c62',2:b'001e75',3:b'001f6b',4:b'001fe3',5:b'0021fb',6:b'0022a9',
7:b'002483',8:b'0025e5',9:b'0026e2',10:b'0034da',
11:b'003de8',12:b'0057c1',13:b'00aa70',14:b'041b6d',15:b'08d46a',16:b'0c4885',
17:b'10683f',18:b'10f1f2',19:b'10f96f',20:b'2021a5',
21:b'2c54cf',22:b'2c598a',23:b'30766f',24:b'30fceb',25:b'344df7',26:b'34fcef',
27:b'3830f9',28:b'40b0fa',29:b'485929',30:b'48605f',
31:b'48902f',32:b'505527',33:b'583f54',34:b'58a2b5',35:b'5c70a3',36:b'5caf06',
37:b'603cee',38:b'60e3ac',39:b'640d22',40:b'64899a',
41:b'64bc0c',42:b'64c2de',43:b'6cd68a',44:b'700514',45:b'74a722',46:b'78f882',
47:b'7cf31b',48:b'805a04',49:b'88074b',50:b'88365f',
51:b'88c9d0',52:b'8c3ae3',53:b'98b8ba',54:b'98d6f7',55:b'a039f7',56:b'a04f85',
57:b'a09169',58:b'a816b2',59:b'a8922c',60:b'a8b86e',
61:b'ac0d1b',62:b'acf6f7',63:b'b4f1da',64:b'b4f7a1',65:b'b81daa',66:b'bcf5ac',
67:b'c4438f',68:b'c49a02',69:b'c8f319',70:b'ccfa00',
71:b'd013fd',72:b'dc0b34',73:b'e892a4',74:b'f01c13',75:b'f80cf3',76:b'f895c7',
77:b'f8a9d0'
}

dict_BQ = {
1:b'4c7403',2:b'b49d0b'
}

dict_s = {
1:b'b01041', 2:b'08beac', 3:b'542aa2', 4:b'00ad24', 5:b'7c7a91', 6:b'08beac'
}

dict_iphone = {
1:b'9ef9ae', 2:b'feda80', 3:b'165dac'
}

dict_apple = {
1:b'000393',2:b'000a27',3:b'000a95',4:b'000d93',5:b'0010fa',6:b'001124',
7:b'001451',8:b'0016cb',9:b'0017f2',10:b'0019e3',

```


11:b'001b63',12:b'001cb3',13:b'001d4f',14:b'001e52',15:b'001ec2',16:b'001f5b',
17:b'001ff3',18:b'0021e9',19:b'002241',20:b'002312',
21:b'002332',22:b'00236c',23:b'0023df',24:b'002436',25:b'002500',26:b'00254b',
27:b'0025bc',28:b'002608',29:b'00264a',30:b'0026b0',
31:b'0026bb',32:b'003065',33:b'003ee1',34:b'0050e4',35:b'0056cd',36:b'005b94',
37:b'006171',38:b'006d52',39:b'007d60',40:b'008865',
41:b'00b362',42:b'00c610',43:b'00cdfc',44:b'00db70',45:b'00f39f',46:b'00f4b9',
47:b'00f76f',48:b'040cce',49:b'041552',
51:b'041e64',52:b'042665',53:b'04489a',54:b'044bed',55:b'0452f3',56:b'045453',
57:b'0469f8',58:b'047295',59:b'04d3cf',60:b'04db56',
61:b'04e536',62:b'04f13e',63:b'04f7e4',64:b'080007',65:b'082cb6',66:b'086698',
67:b'086d41',68:b'087045',69:b'087402',70:b'0887c7',
71:b'08c729',72:b'08e689',73:b'08f4ab',74:b'08f69c',75:b'08f8bc',76:b'08ff44',
77:b'0c1539',78:b'0c19f8',79:b'0c3021',80:b'0c3b50',
81:b'0c3e9f',82:b'0c5101',84:b'0c74c2',85:b'0c771a',86:b'0cbc9f',87:b'0cd746',
88:b'0ce441',89:b'101c0c',90:b'102959',
91:b'103025',92:b'1040f3',93:b'10417f',94:b'1093e9',95:b'1094bb',96:b'109add',
97:b'10cee9',98:b'10ddb1',99:b'14109f',100:b'14205e',
101:b'145a05',102:b'1460cb',103:b'147dda',104:b'14876a',
105:b'1488e6',106:b'148fc6',
107:b'1495ce',108:b'149877',109:b'1499e2',110:b'149d99',
111:b'14bd61',112:b'14c213',113:b'14c88b',114:b'14d00d',
115:b'14d19e',116:b'182032',
117:b'183451',118:b'183eef',119:b'1855e3',120:b'1856c3',
121:b'186590',122:b'187eb9',123:b'18810e',124:b'189efc',
125:b'18af61',126:b'18af8f',
127:b'18e7f4',128:b'18ee69',129:b'18f1d8',130:b'18f643',
131:b'1c1ac0',132:b'1c36bb',133:b'1c5cf2',134:b'1c9148',
135:b'1c9180',136:b'1c9e46',
137:b'1caba7',138:b'1ce62b',139:b'203cae',140:b'206980',
141:b'20768f',142:b'2078f0',143:b'207d74',144:b'209bcd',
145:b'20a2e4',146:b'20ab37',
147:b'20c9d0',148:b'20e2a8',149:b'20e874',
150:b'20ee28',151:b'241b7a',152:b'241eeb',153:b'24240e',
154:b'245ba7',155:b'245e48',
156:b'24a074',157:b'24a2e1',158:b'24ab81',159:b'24d0df',
160:b'24e314',161:b'24f094',162:b'24f677',163:b'280b5c',
164:b'283737',165:b'285aeb',
166:b'286ab8',167:b'286aba',168:b'2877f1',169:b'28a02b',
170:b'28c709',171:b'28cfda',172:b'28cfe9',173:b'28e02c',
174:b'28e14c',175:b'28e7cf',

177:b'28ec95',178:b'28ed6a',179:b'28f033',180:b'28f076',
181:b'28ff3c',182:b'2c1f23',183:b'2c200b',184:b'2c3361',
185:b'2c61f6',186:b'2cb43a',
187:b'2cbc87',189:b'2cbe08',190:b'2cf0a2',
191:b'2cf0ee',192:b'3010e4',193:b'3035ad',194:b'305714',
195:b'30636b',196:b'309048',
197:b'3090ab',198:b'30d9d9',199:b'30f7c5',200:b'3408bc',
201:b'341298',202:b'34159e',203:b'34318f',204:b'34363b',
205:b'344262',206:b'3451c9',
207:b'347c25',208:b'34a395',209:b'34a8eb',210:b'34ab37',
211:b'34c059',212:b'34e2fd',213:b'34fd6a',214:b'380f4a',
215:b'38484c',216:b'38539c',
217:b'3865b2',218:b'3866f0',219:b'3871de',220:b'38892c',
221:b'38b54d',222:b'38c986',223:b'38cada',224:b'38ec0d',
225:b'38f9d3',226:b'3c0630',
227:b'3c0754',228:b'3c15c2',229:b'3c22fb',230:b'3c2ef9',
231:b'3c2eff',232:b'3c4dbe',233:b'3c7d0a',234:b'3ca6f6',
235:b'3cab8e',236:b'3cbf60',
237:b'3ccd36',238:b'3cd0f8',239:b'3ce072',240:b'402619',
241:b'403004',242:b'40331a',243:b'403cfc',244:b'404d7f',
245:b'406c8f',246:b'4070f5',
247:b'40831d',248:b'4098ad',249:b'409c28',250:b'40a6d9',
251:b'40b395',252:b'40bc60',253:b'40c711',254:b'40cbc0',
255:b'40d32d',256:b'40e64b',
257:b'40f946',258:b'440010',259:b'4418fd',260:b'442a60',
261:b'443583',262:b'444adb',263:b'444c0c',264:b'4490bb',
265:b'44a8fc',266:b'44c65d',
267:b'44d884',268:b'44e66e',269:b'44f09e',270:b'44f21b',
271:b'44fb42',272:b'48262c',273:b'483b38',274:b'48437c',
275:b'484baa',276:b'4860bc',
277:b'48746e',278:b'48a195',279:b'48a91c',280:b'48b8a3',
281:b'48bf6b',282:b'48d705',283:b'48e9f1',284:b'4c20b8',
285:b'4c3275',286:b'4c569d',
287:b'4c57ca',288:b'4c6be8',289:b'4c74bf',290:b'4c7c5f',
291:b'4c7cd9',292:b'4c8d79',293:b'4cab4f',294:b'4cb199',
295:b'4cb910',296:b'501fc6',
297:b'503237',298:b'507a55',299:b'507ac5',300:b'5082d5',
301:b'50a67f',302:b'50bc96',303:b'50de06',304:b'50ead6',
305:b'50ed3c',306:b'50f4eb',
307:b'540910',308:b'542696',309:b'542b8d',310:b'5433cb',
311:b'544e90',312:b'5462e2',313:b'54724f',314:b'549963',

315:b'549f13',316:b'54ae27',
317:b'54e43a',318:b'54e61b',319:b'54eaa8',320:b'581faa',
321:b'58404e',322:b'5855ca',323:b'586b14',324:b'587f57',
325:b'58b035',326:b'58d349',
327:b'58e28f',328:b'58e6ba',329:b'5c0947',330:b'5c1dd9',
331:b'5c5948',332:b'5c7017',333:b'68a86d',334:b'5c8730',
335:b'5c8d4e',336:b'5c95ae',
337:b'5c969d',338:b'5c97f3',339:b'5cadcf',340:b'5cf5da',
341:b'5cf7e6',342:b'5cf938',343:b'600308',344:b'6006e3',
345:b'6030d4',346:b'60334b',
347:b'606944',348:b'6070c0',349:b'607ec9',350:b'608373',
351:b'608b0e',352:b'608c4a',353:b'609217',354:b'609ac1',
355:b'60a37d',356:b'60bec4',
357:b'60c547',358:b'60d9c7',359:b'60f445',360:b'60f81d',
361:b'60facd',362:b'60fb42',363:b'60fec5',364:b'640bd7',
365:b'64200c',366:b'645aed',
367:b'647033',368:b'6476ba',369:b'649abe',370:b'64a3cb',
371:b'64a5c3',372:b'64b0a6',373:b'64b9e8',374:b'64c753',
375:b'64d2c4',376:b'64e682',
377:b'680927',378:b'682f67',379:b'685b35',380:b'68644b',
381:b'68967b',382:b'689c70',383:b'68ab1e',384:b'68ae20',
385:b'68d93c',386:b'68dbca',
387:b'68ef43',388:b'68fb7e',389:b'68fef7',390:b'6c19c0',
391:b'6c3e6d',392:b'6c4008',393:b'6c4a85',394:b'6c4d73',
395:b'6c709f',396:b'6c72e7',
397:b'6c8dc1',398:b'6c94f8',399:b'6c96cf',400:b'6cab31',
401:b'6cc26b',402:b'6ce85c',403:b'701124',404:b'7014a6',
405:b'703c69',406:b'703eac',
407:b'70480f',408:b'705681',409:b'70700d',410:b'7073cb',
411:b'7081eb',412:b'70a2b3',413:b'70cd60',414:b'70dee2',
415:b'70e72c',416:b'70ea5a',
417:b'70ece4',418:b'70ef00',419:b'70f087',420:b'741bb2',
421:b'74428b',422:b'74650c',423:b'748114',424:b'748d08',
425:b'748f3c',426:b'749eaf',
427:b'74b587',428:b'74e1b6',429:b'74e2f5',430:b'7831c1',
431:b'783a84',432:b'784f43',433:b'7864c0',434:b'7867d7',
435:b'786c1c',436:b'787b8a',
437:b'787e61',438:b'78886d',439:b'789f70',440:b'78a3e4',
441:b'78ca39',442:b'78d162',443:b'78d75f',444:b'78e3de',
445:b'78fd94',446:b'7c0191',
447:b'7c04d0',448:b'7c11be',449:b'7c5049',450:b'7c6d62',

451:b'7c6df8',452:b'7c9a1d',453:b'7ca1ae',454:b'7cab60',
455:b'7cc3a1',456:b'7cc537',
457:b'7cd1c3',458:b'7cf05f',459:b'7cfadf',460:b'80006e',
461:b'800c67',462:b'804971',463:b'804a14',464:b'805fc5',
465:b'808223',466:b'80929f',
467:b'80b03d',468:b'80be05',469:b'80d605',470:b'80e650',
471:b'80ea96',472:b'80ed2c',473:b'842999',474:b'843835',
475:b'844167',476:b'846878',
477:b'84788b',478:b'848506',479:b'8489ad',480:b'848c8d',
481:b'848e0c',482:b'84a134',483:b'84ab1a',484:b'84ac16',
485:b'84ad8d',486:b'84b153',
487:b'84fcac',488:b'84fcfe',489:b'881908',490:b'881fa1',
491:b'885395',492:b'8863df',493:b'886440',494:b'88665a',
495:b'8866a5',496:b'886b6e',
497:b'88a479',498:b'88a9b7',499:b'88ae07',500:b'88b291',
501:b'88c08b',502:b'88c663',503:b'88cb87',504:b'88e87f',
505:b'88e9fe',506:b'8c006d',
507:b'8c2937',508:b'8c2daa',509:b'8c5877',510:b'8c7b9d',
511:b'8c7c92',512:b'8c8590',513:b'8c861e',514:b'8c8ef2',
515:b'8c8fe9',516:b'8cec7b',
517:b'8cfaba',158:b'8cfe57',519:b'9027e4',520:b'903c92',
521:b'9060f1',522:b'907240',523:b'90812a',524:b'908158',
525:b'90840d',526:b'908c43',
527:b'908d6c',528:b'909c4a',529:b'90a25b',530:b'90b0ed',
531:b'90b21f',532:b'90b931',533:b'90c1c6',534:b'90dd5d',
535:b'90e17b',536:b'90fd61',
537:b'940c98',538:b'941625',539:b'945c9a',540:b'949426',
541:b'94b01f',542:b'94bf2d',543:b'94e96a',544:b'94ea32',
545:b'94f6a3',546:b'94f6d6',
547:b'9800c6',548:b'9801a7',549:b'9803d8',550:b'9810e8',
551:b'98460a',552:b'985aeb',553:b'9860ca',554:b'989e63',
555:b'98b8e3',556:b'98ca33',
557:b'98d6bb',558:b'98e0d9',559:b'98f0ab',560:b'98fe94',
561:b'9c04eb',562:b'9c207b',563:b'9c28b3',564:b'9c293f',
565:b'9c35eb',566:b'9c4fda',
567:b'9c583c',568:b'9c648b',569:b'9c760e',570:b'9c84bf',
571:b'9c8ba0',572:b'9ce33f',573:b'9ce65e',574:b'9cf387',
575:b'9cf48e',576:b'9cfc01',
578:b'9cfc28',579:b'a01828',580:b'a03be3',
581:b'a04ea7',582:b'a04ecf',583:b'a056f3',584:b'a07817',
585:b'a0999b',586:b'a0d795',

587:b'a0edcd',588:b'a0fbc5',589:b'a43135',590:b'a45e60',
591:b'a46706',592:b'a483e7',593:b'a4b197',594:b'a4b805',
595:b'a4c361',596:b'a4d18c',
597:b'a4d1d2',598:b'a4d931',599:b'a4e975',600:b'a4f1e8',
601:b'a82066',602:b'a85b78',603:b'a85c2c',604:b'a860b6',
605:b'a8667f',606:b'a8817e',
607:b'a886dd',608:b'a88808',609:b'a88e24',610:b'a8913d',
611:b'a8968a',612:b'a8bbcf',613:b'a8be27',614:b'a8fad8',
615:b'ac15f4',616:b'ac1d06',
617:b'ac1f74',618:b'ac293a',619:b'ac3c0b',620:b'ac49db',
621:b'ac61ea',622:b'ac7f3e',623:b'ac87a3',624:b'ac88fd',
625:b'ac9085',626:b'acbc32',
627:b'accf5c',628:b'ace4b5',629:b'acfddec',630:b'b019c6',
631:b'b03495',632:b'b035b5',633:b'b0481a',634:b'b065bd',
635:b'b0702d',636:b'b08c75',
637:b'b09fba',638:b'b0ca68',639:b'b0e5f9',640:b'b418d1',
641:b'b41bb0',642:b'b440a4',643:b'b44bd2',644:b'b456e3',
645:b'b485e1',646:b'b48b19',
647:b'b49cdf',648:b'b4f0ab',649:b'b4f61c',650:b'b4fa48',
651:b'b8098a',652:b'b817c2',653:b'b82aa9',654:b'b841a4',
655:b'b844d9',656:b'b853ac',
657:b'b85d0a',658:b'b8634d',659:b'b8782e',660:b'b87bc5',
661:b'b881fa',662:b'b88d12',663:b'b89047',664:b'b8b2f8',
665:b'b8c111',666:b'b8c75d',
667:b'b8e856',668:b'b8f12a',669:b'b8f6b1',670:b'b8ff61',
671:b'bc0963',672:b'bc3baf',673:b'bc4cc4',674:b'bc52b7',
675:b'bc5436',676:b'bc6778',
677:b'bc6c21',678:b'bc926b',679:b'bc9fef',680:b'bca5a9',
681:b'bca920',682:b'bc8b863',683:b'bce143',684:b'bcec5d',
685:b'bcfed9',686:b'c01ada',
687:b'c06394',688:b'c0847a',689:b'c09ad0',690:b'c09f42',
691:b'c0a53e',692:b'c0a600',693:b'c0b658',694:b'c0ccf8',
695:b'c0cecd',696:b'c0d012',
697:b'c0e862',698:b'c0f2fb',699:b'c40b31',700:b'c41234',
701:b'c41411',702:b'c42ad0',703:b'c42c03',704:b'c4618b',
705:b'c48466',706:b'c4910c',
707:b'c49880',708:b'c4b301',709:b'c4c36b',710:b'c81ee7',
711:b'c82a14',712:b'c8334b',713:b'c83c85',714:b'c869cd',
715:b'c86f1d',716:b'c88550',
717:b'c8b1cd',718:b'c8b5b7',719:b'c8bcc8',720:b'c8d083',
721:b'c8e0eb',722:b'c8f650',723:b'cc088d',724:b'cc08e0',

725:b'cc20e8',726:b'cc25ef',
727:b'cc29f5',728:b'cc2db7',729:b'cc4463',730:b'cc660a',
731:b'cc69fa',732:b'cc785f',733:b'ccc760',734:b'ccc95d',
735:b'ccd281',736:b'd0034b',
737:b'd023db',738:b'd02598',739:b'd02b20',740:b'd03311',
741:b'd03faa',742:b'd04f7e',743:b'd06544',744:b'd0817a',
745:b'd0a637',746:b'd0c5f3',
747:b'd0d23c',748:b'd0d2b0',749:b'd0e140',750:b'd446e1',
751:b'd4619d',752:b'd461da',753:b'd4909c',754:b'd49a20',
755:b'd4a33d',756:b'd4dccd',
757:b'd4f46f',758:b'd8004d',759:b'd81c79',760:b'd81d72',
761:b'd83062',762:b'd84c90',763:b'd88f76',764:b'd89695',
765:b'd89e3f',766:b'd8a25e',
767:b'd8bb2c',768:b'd8cf9c',769:b'd8d1cb',770:b'd8dc40',
771:b'd8de3a',772:b'dc080f',773:b'dc0c5c',774:b'dc2b2a',
775:b'dc2b61',776:b'dc3714',
777:b'dc415f',778:b'dc5285',779:b'dc56e7',780:b'dc86d8',
781:b'dc9b9c',782:b'dca4ca',783:b'dca904',784:b'dcd3a2',
785:b'e02b96',786:b'e0338e',
787:b'e05f45',788:b'e06678',789:b'e06d17',790:b'e0897e',
791:b'e0925c',792:b'e0accb',793:b'e0b52d',794:b'e0b55f',
795:b'e0b9ba',796:b'e0c767',
797:b'e0c97a',798:b'e0eb40',799:b'e0f5c6',800:b'e0f847',
801:b'e425e7',802:b'e42b34',803:b'e450eb',804:b'e47684',
805:b'e48b7f',806:b'e490fd',
807:b'e498d6',808:b'e49a79',809:b'e49adc',810:b'e4b2fb',
811:b'e4c63d',812:b'e4ce8f',813:b'e4e0a6',814:b'e4e4ab',
815:b'e8040b',816:b'e80688',
817:b'e81cd8',818:b'e83617',819:b'e87865',820:b'e87f95',
821:b'e8802e',822:b'e88152',823:b'e8854b',824:b'e88d28',
825:b'e8a730',826:b'e8b2ac',
827:b'e8fbe9',828:b'ec2651',829:b'ec2ce2',830:b'ec3586',
831:b'ec852f',832:b'ecadb8',833:b'ecced7',834:b'f01898',
835:b'f02475',836:b'f02f4b',
837:b'f05cd5',838:b'f0766f',839:b'f07807',840:b'f07960',
841:b'f0989d',842:b'f099b6',843:b'f099bf',844:b'f0a35a',
845:b'f0b0e7',846:b'f0b3ec',
847:b'f0b479',848:b'f0c1f1',849:b'f0c371',850:b'f0cba1',
851:b'f0d1a9',852:b'f0dbe2',853:b'f0dbf8',854:b'f0dce2',
855:b'f0f61c',856:b'f40616',
857:b'f40e01',858:b'f40f24',859:b'f41ba1',860:b'f431c3',

```
861:b'f434f0',862:b'f437b7',863:b'f45c89',864:b'f465a6',
865:b'f4afe7',866:b'f4beec',
867:b'f4d488',868:b'f4dbe3',869:b'f4f15a',870:b'f4f951',
871:b'f80377',872:b'f81093',873:b'f81edf',874:b'f82793',
875:b'f82d7c',876:b'f83880',
877:b'f84e73',878:b'f86214',879:b'f8665a',880:b'f86fc1',
881:b'f887f1',882:b'f895ea',883:b'f8b1dd',884:b'f8e94e',
885:b'f8ffc2',886:b'fc183c',
887:b'fc1d43',888:b'fc253f',889:b'fc2a9c',890:b'fc4ea4',
891:b'fc66cf',892:b'fcb6d8',893:b'fcd848',894:b'fce998',
895:b'fcfc48'
}
```

```
dict_nosesabe = {
1:b'7e600d',2:b'9ae3da',4:b'ca3759',5:b'b28d8b',6:b'12aabb',
7:b'067fdc',8:b'02f991',9:b'0678c2',10:b'ee063c',
11:b'2e5f05',12:b'3ab739',13:b'8a6015',14:b'267997',15:b'6ac445',15:b'76a68f',
17:b'ae7baa',18:b'32158d',19:b'6e9120',20:b'6a1e63',
21:b'd23f8d',22:b'0e5baa',23:b'ca01cd',24:b'e6cf42',25:b'6284fb',26:b'a23a83',
27:b'a23872',28:b'1e5a5c',29:b'9a607b',30:b'a2e489',
31:b'06d725'
}
```

```
dict_google = {
1:b'001a11',2:b'00f620',3:b'089e08',4:b'08b4b1',5:b'14223b',6:b'14c14e',
7:b'1cf29a',8:b'201f3b',9:b'20dfb9',10:b'240588',
11:b'28bd89',12:b'30fd38',13:b'388b59',14:b'3c286d',15:b'3c5ab4',16:b'3c8d20',
17:b'44070b',18:b'44bb3b',19:b'48d6d5',20:b'546009',
21:b'582429',22:b'58cb52',23:b'60b76e',24:b'703acb',25:b'7c2ebd',26:b'7cd95c',
27:b'883d24',28:b'88541f',29:b'900cc8',30:b'9495a0',
31:b'94eb2c',32:b'98d293',33:b'a47733',34:b'ac6784',35:b'b02a43',36:b'b0e4d5',
37:b'cca7c1',38:b'ccf411',39:b'd4f547',40:b'd86c63',
41:b'd88c79',42:b'd8eb46',43:b'daa119',44:b'e45e1b',45:b'e4f042',46:b'f05c77',
47:b'f072ea',48:b'f0ef86',49:b'f40304',50:b'f4f5d8',
51:b'f4f5e8',52:b'f80ff9',53:b'f81a2b',54:b'f88fca'
}
```

```
dict_qualcomm = {
1:b'00a0c6',2:b'649c81',3:b'88124e',4:b'8cfd0'
}
```

```
dict_intel = {
1:b'0012f0',2:b'001302',3:b'001320',4:b'0013ce',
5:b'0013e8',6:b'001500',7:b'001517',8:b'00166f',9:b'001676',10:b'0016ea',
11:b'0016eb',12:b'0018de',13:b'0019d1',14:b'0019d2',15:b'001b21',
16:b'001b77',17:b'001cbf',18:b'001cc0',19:b'001de0',20:b'001de1',
21:b'001e64',22:b'001e65',23:b'001e67',24:b'001f3b',25:b'001f3c',
26:b'00215c',27:b'00215d',28:b'00216a',29:b'00216b',30:b'0022fa',
31:b'0022fb',32:b'002314',33:b'002315',34:b'0024d6',35:b'0024d7',
36:b'0026c6',37:b'0026c7',38:b'00270e',39:b'002710',40:b'0028f8',
41:b'004238',42:b'00919e',43:b'00bb60',44:b'00c2c6',45:b'00d76d',
46:b'00dbdf',47:b'00e18c',48:b'0433c2',49:b'0456e5',50:b'046c59',
51:b'04d3b0',52:b'04ea56',53:b'04ecd8',54:b'04ed33',55:b'081196',
56:b'085bd6',57:b'086ac5',58:b'087190',59:b'08d23e',60:b'08d40c',
61:b'0c5415',62:b'0c7a15',63:b'0c8bfd',64:b'0c9a3c',65:b'0cd292',
66:b'0cdd24',67:b'1002b5',68:b'100ba9',69:b'103d1c',70:b'104a7d',
71:b'105107',72:b'10f005',73:b'1418c3',74:b'144f8a',75:b'14857f',
76:b'14abc5',77:b'14f6d8',78:b'181dea',79:b'182649',80:b'183da2',
81:b'185680',82:b'185e0f',83:b'18cc18',84:b'18ff0f',85:b'1c1bb5',
86:b'1c4d70',87:b'1c9957',88:b'1cc10c',89:b'2016b9',90:b'201e88',
91:b'207918',92:b'20c19b',93:b'24418c',94:b'247703',95:b'24ee9a',
96:b'2811a8',97:b'2816ad',98:b'287fcf',99:b'28b2bd',100:b'28c63f',
101:b'28d0ea',102:b'28dfef',103:b'2c6dc1',104:b'2c6e85',105:b'2c8db1',
106:b'2cdb07',107:b'302432',108:b'303a64',109:b'30e37a',110:b'340286',
111:b'3413e8',112:b'342eb7',113:b'34415d',114:b'347df6',115:b'34c93d',
116:b'34cff6',117:b'34de1a',118:b'34e12d',119:b'34e6ad',120:b'34f39a',
121:b'34f64b',122:b'380025',123:b'386893',124:b'3887d5',125:b'38baf8',
126:b'38dead',127:b'38fc98',128:b'3c58c2',129:b'3c6aa7',130:b'3c9c0f',
131:b'3ca9f4',132:b'3cf011',133:b'3cf862',134:b'3cfdfe',135:b'401c83',
136:b'4025c2',137:b'4074e0',138:b'40a3cc',139:b'40a6b7',140:b'40ec99',
141:b'44032c',142:b'448500',143:b'44af28',144:b'44e517',145:b'484520',
146:b'4851b7',147:b'4851c5',148:b'48684a',149:b'4889e7',150:b'48a472',
151:b'48f17f',152:b'4c1d96',153:b'4c3488',154:b'4c77cb',155:b'4c796e',
156:b'4c79ba',157:b'4c8093',158:b'4ceb42',159:b'502da2',160:b'502f9b',
161:b'5076af',162:b'50e085',163:b'50eb71',164:b'5414f3',165:b'548d5a',
166:b'586c25',167:b'5891cf',168:b'58946b',169:b'58961d',170:b'58a023',
171:b'58a839',172:b'58fb84',173:b'5c514f',174:b'5c5f67',175:b'5c80b6',
176:b'5c879c',177:b'5cc5d4',178:b'5ccd5b',179:b'5cd2e4',180:b'5ce0c5',
181:b'5ce42a',182:b'6036dd',183:b'605718',184:b'606720',185:b'606c66',
186:b'60a5e2',187:b'60dd8e',188:b'60e32b',189:b'60f262',190:b'60f677',
191:b'6432a8',192:b'644c36',193:b'645d86',194:b'646ee0',195:b'6479f0',
```


196:b'648099',197:b'64bc58',198:b'64d4da',199:b'6805ca',200:b'680715',
201:b'681729',202:b'683e26',203:b'68545a',204:b'685d43',205:b'68ecc5',
206:b'6c2995',207:b'6c6a77',208:b'6c8814',209:b'6c9466',210:b'6ca100',
211:b'6cfe54',212:b'701ce7',213:b'709cd1',214:b'70a6cc',215:b'70cd0d',
216:b'70cf49',217:b'7470fd',218:b'74d83e',219:b'74e50b',220:b'74e5f9',
221:b'780cb8',222:b'782b46',223:b'78929c',224:b'78ff57',225:b'7c2a31',
226:b'7c5079',227:b'7c5cf8',228:b'7c67a2',229:b'7c70db',230:b'7c7635',
231:b'7c7a91',232:b'7cb0c2',233:b'7cb27d',234:b'7cccb8',235:b'80000b',
236:b'801934',237:b'803253',238:b'8038fb',239:b'8045dd',240:b'8086f2',
241:b'809b20',242:b'80b655',243:b'84144d',244:b'841b77',245:b'843a4b',
246:b'845cf3',247:b'84683e',248:b'84a6c8',249:b'84c5a6',250:b'84ef18',
251:b'84fdd1',252:b'88532e',253:b'887873',254:b'88b111',255:b'8c1d96',
256:b'8c554a',257:b'8c705a',258:b'8c8d28',259:b'8ca982',260:b'8cc681',
261:b'902e1c',262:b'9049fa',263:b'9061ae',264:b'907841',265:b'90ccdf',
266:b'90e2ba',267:b'94659c',268:b'94b86d',269:b'94e23c',270:b'94e6f7',
271:b'94e70b',272:b'982cbc',273:b'983b8f',274:b'9843fa',275:b'984fee',
276:b'98541b',277:b'988d46',278:b'98af65',279:b'9c2976',280:b'9c4e36',
281:b'9cda3e',282:b'9cfce8',283:b'a0369f',284:b'a0510b',285:b'a08869',
286:b'a088b4',287:b'a0a4c5',288:b'a0a8cd',289:b'a0afbd',290:b'a0c589',
291:b'a0d37a',292:b'a0e70b',293:b'a402b9',294:b'a434d9',295:b'a4423b',
296:b'a44e31',297:b'a46bb6',298:b'a4b1c1',299:b'a4bf01',300:b'a4c3f0',
301:b'a4c494',302:b'a864f1',303:b'a86daa',304:b'a87eea',305:b'ac1203',
306:b'ac2b6e',307:b'ac5afc',308:b'ac675d',309:b'ac7289',310:b'ac74b1',
311:b'ac7ba1',312:b'ac8247',313:b'aced5c',314:b'acfdce',315:b'b0359f',
316:b'b06088',317:b'b07d64',318:b'b0a460',319:b'b40ede',320:b'b46921',
321:b'b46bfc',322:b'b46d83',323:b'b49691',324:b'b4b676',325:b'b4d5bd',
326:b'b80305',327:b'b808cf',328:b'b88198',329:b'b88a60',330:b'b89a2a',
331:b'b8b81e',332:b'b8bf83',333:b'bc091b',334:b'bc0f64',335:b'bc17b8',
336:b'bc542f',337:b'bc7737',338:b'bca8a6',339:b'bcf171',340:b'c03c59',
341:b'c0b6f9',342:b'c0b883',343:b'c42360',344:b'c48508',345:b'c4d0e3',
346:b'c4d987',347:b'c809a8',348:b'c82158',349:b'c8348e',350:b'c858c0',
351:b'c8b29b',352:b'c8e265',353:b'c8f733',354:b'cc1531',355:b'cc2f71',
356:b'cc3d82',357:b'ccd9ac',358:b'ccf9e4',359:b'd03c1f',360:b'd0577b',
361:b'd07e35',362:b'd0abd5',363:b'd0c637',364:b'd4258b',365:b'd43b04',
366:b'd4548b',367:b'd46d6d',368:b'd4d252',369:b'd83bbf',370:b'd8f2ca',
371:b'd8f883',372:b'd8fc93',373:b'dc1ba1',374:b'dc2148',375:b'dc215c',
376:b'dc41a9',377:b'dc5360',378:b'dc7196',379:b'dc8b28',380:b'dca971',
381:b'dcfb48',382:b'e02be9',383:b'e09467',384:b'e09d31',385:b'e0d464',
386:b'e0d4e8',387:b'e4029b',388:b'e442a6',389:b'e45e37',390:b'e470b8',
391:b'e4a471',392:b'e4a7a0',393:b'e4b318',394:b'e4f89c',395:b'e4fafd',
396:b'e4fd45',397:b'e82aea',398:b'e884a5',399:b'e8b1fc',400:b'e8f408',

```
401:b'ec63d7',402:b'f0421c',403:b'f057a6',404:b'f077c3',405:b'f09e4a',
406:b'f0b61e',407:b'f0d5bf',408:b'f40669',409:b'f42679',410:b'f44637',
411:b'f44ee3',412:b'f47b09',413:b'f48c50',414:b'f49634',415:b'f4a475',
416:b'f4b301',417:b'f4d108',418:b'f81654',419:b'f83441',420:b'f85971',
421:b'f85ea0',422:b'f8633f',423:b'f894c2',424:b'f8ac65',425:b'f8e4e3',
426:b'f8f21e',427:b'fc4482',428:b'fc7774',429:b'fcb3bc',430:b'fcf8ae'
}
```

```
dict_samsung = {
1:b'0000f0',2:b'0007ab',3:b'001247',4:b'0012fb',5:b'001377',
6:b'001599',7:b'0015b9',8:b'001632',9:b'00166b',10:b'00166c',
11:b'0016db',12:b'0017c9',13:b'0017d5',14:b'0018af',15:b'001a8a',
16:b'001b98',17:b'001c43',18:b'001d25',19:b'001df6',20:b'001e7d',
21:b'001ee1',22:b'001ee2',23:b'001fcc',24:b'001fcd',25:b'00214c',
26:b'0021d1',27:b'0021d2',28:b'002339',29:b'00233a',30:b'002399',
31:b'0023d6',32:b'0023d7',33:b'002454',34:b'002490',35:b'002491',
36:b'0024e9',37:b'002566',38:b'002567',39:b'00265d',40:b'00265f',
41:b'006f64',42:b'0073e0',43:b'007c2d',44:b'008701',45:b'00b5d0',
46:b'00bf61',47:b'00c3f4',48:b'00e3b2',49:b'00f46f',50:b'00fa21',
51:b'04180f',52:b'041bba',53:b'04b1a1',54:b'04b429',55:b'04b9e3',
56:b'04ba8d',57:b'04bdbf',58:b'04fe31',59:b'0808c2',60:b'0821ef',
61:b'08373d',62:b'083d88',63:b'087808',64:b'088c2c',65:b'08aed6',
66:b'08bfa0',67:b'08d42b',68:b'08eca9',69:b'08ee8b',70:b'08fc88',
71:b'08fd0e',72:b'0c1420',73:b'0c2fb0',74:b'0c715d',75:b'0c8910',
76:b'0c8dca',77:b'0ca8a7',78:b'0cb319',79:b'0cdfa4',80:b'0ce0dc',
81:b'1007b6',82:b'101dc0',83:b'1029ab',84:b'102b41',85:b'103047',
86:b'103917',87:b'103b59',88:b'1077b1',89:b'1089fb',90:b'108ee0',
91:b'109266',92:b'10d38a',93:b'10d542',94:b'10e4c2',95:b'10ec81',
96:b'140152',97:b'141f78',98:b'1432d1',99:b'14568e',100:b'1489fd',
101:b'1496e5',102:b'149f3c',103:b'14a364',104:b'14b484',105:b'14bb6e',
106:b'14f42a',107:b'1816c9',108:b'1819d6',109:b'181eb0',110:b'182195',
111:b'18227e',112:b'182666',113:b'183a2d',114:b'183f47',115:b'184617',
116:b'184e16',117:b'184ecb',118:b'1854cf',119:b'185bb3',120:b'1867b0',
121:b'188331',122:b'18895b',123:b'18ab1d',124:b'18ce94',125:b'18e2c2',
126:b'1c232c',127:b'1c3ade',128:b'1c5a3e',129:b'1c62b8',130:b'1c66aa',
131:b'1caf05',132:b'1ce57f',133:b'1ce61d',134:b'2013e0',135:b'202d07',
136:b'20326c',137:b'205531',138:b'205ef7',139:b'206e9c',140:b'20d390',
141:b'20d5bf',142:b'244b03',143:b'244b81',144:b'245ab5',145:b'2468b0',
146:b'24920e',147:b'24c696',148:b'24dbed',149:b'24f5aa',150:b'24fce5',
151:b'2802d8',152:b'2827bf',153:b'28395e',154:b'288335',
156:b'28987b',157:b'28bab5',158:b'28cc01',159:b'2c4053',160:b'2c4401',
```

161:b'2cae2b',162:b'2cbaba',163:b'301966',164:b'306a85',
165:b'3096fb',167:b'30c7ae',168:b'30cbf8',169:b'30cda7',170:b'30d587',
171:b'30d6c9',172:b'34145f',173:b'342d0d',174:b'343111',
175:b'3482c5',176:b'348a7b',177:b'34aa8b',178:b'34be00',179:b'34c3ac',180:b'3801
181:b'380a94',182:b'380b40',183:b'3816d1',184:b'382dd1',
185:b'382de8',186:b'3868a4',187:b'386a77',188:b'389496',189:b'389af6',190:b'38d4
191:b'38ece4',192:b'3c0518',193:b'3c195e',194:b'3c20f6',
195:b'3c576c',196:b'3c5a37',197:b'3c6200',198:b'3c8bfe',199:b'3ca10d',200:b'3cbb
201:b'3cdcbc',202:b'3cf7a4',203:b'40163b',204:b'40d3ae',
205:b'444e1a',206:b'445ce9',207:b'446d6c',208:b'44783e',209:b'44f459',210:b'4813
211:b'4827ea',212:b'4844f7',213:b'4849c7',214:b'485169',
215:b'48794d',216:b'489dd1',217:b'48c796',218:b'4c3c16',219:b'4ca56d',220:b'4cbc
221:b'4cc95e',222:b'4cdd31',223:b'5001bb',224:b'503275',
225:b'503da1',226:b'5049b0',227:b'5050a4',228:b'5056bf',229:b'507705',230:b'5085
231:b'5092b9',232:b'509ea7',233:b'50a4c8',234:b'50b7c3',
235:b'50c8e5',236:b'50f0d3',237:b'50f520',238:b'50fc9f',239:b'54219d',240:b'543a
241:b'5440ad',242:b'5492be',243:b'549b12',245:b'54b802',
246:b'54bd79',247:b'54d17d',248:b'54f201',249:b'54fa3e',250:b'54fcf0',
251:b'58a639',252:b'58b10f',253:b'58c38b',254:b'58c5cb',
255:b'5c10c5',256:b'5c2e59',257:b'5c3c27',258:b'5c497d',259:b'5c5181',260:b'5c86
261:b'5c9960',262:b'5cc1d7',263:b'5ccb99',264:b'5ce8eb',
265:b'5cf6dc',266:b'603aaf',267:b'60684e',268:b'606bbd',269:b'6077e2',270:b'608e
271:b'608f5c',272:b'60a10a',273:b'60a4d0',274:b'60af6d',
275:b'60c5ad',276:b'60d0a9',277:b'64037f',278:b'6407f6',279:b'641cae',280:b'641c
281:b'646cb2',282:b'647791',283:b'647bce',284:b'6489f1',
285:b'64b310',286:b'64b853',287:b'64e7d8',288:b'680571',289:b'682737',290:b'6848
291:b'685acf',292:b'6872c3',293:b'687d6b',294:b'68bfc4',
295:b'68e7c2',296:b'68ebae',297:b'6c006b',298:b'6c2f2c',299:b'6c2f8a',300:b'6c83
301:b'6cb7f4',302:b'6cddbc',303:b'6cf373',304:b'701f3c',
305:b'70288b',306:b'702ad5',307:b'705aac',308:b'70b13d',309:b'70ce8c',310:b'70f9
311:b'70fd46',312:b'74458a',313:b'749ef5',314:b'74eb80',
315:b'78009e',316:b'781fdb',317:b'782327',318:b'7825ad',319:b'783716',320:b'7840
321:b'7846d4',322:b'78471d',323:b'78521a',324:b'78595e',
325:b'789ed0',326:b'78a873',327:b'78abbb',328:b'78bdbc',629:b'd0667b',330:b'78c3
331:b'78f7be',332:b'7c0a3f',333:b'7c0bc6',334:b'7c1c68',
335:b'7c2302',336:b'7c2edd',337:b'7c38ad',338:b'7c6456',339:b'7c787e',340:b'7c89
341:b'7c8bb5',342:b'7c9122',343:b'7cf854',344:b'7cf90e',
345:b'8018a7',346:b'8020fd',347:b'8031f0',348:b'804786',349:b'804e70',350:b'804e
351:b'805719',352:b'80656d',353:b'807b3e',354:b'8086d9',
355:b'808abd',356:b'809ff5',357:b'80ceb9',358:b'84119e',359:b'8425db',360:b'842e
361:b'8437d5',362:b'845181',363:b'8455a5',364:b'849866',

365:b'84a466',366:b'84b541',367:b'84c0ef',368:b'88299c',369:b'887598',370:b'8883
371:b'889b39',372:b'889f6f',373:b'88a303',374:b'88add2',
375:b'88bd45',376:b'8c1abf',377:b'8c71f8',378:b'8c7712',379:b'8c79f5',380:b'8c83
381:b'8cbfa6',382:b'8cc8cd',383:b'8cdee6',384:b'8ce5c0',
385:b'8cea48',386:b'9000db',387:b'900628',388:b'90633b',389:b'9097f3',390:b'90b1
391:b'90eec7',392:b'90f1aa',393:b'9401c2',394:b'942ddc',
395:b'94350a',396:b'945103',397:b'9463d1',398:b'9476b7',399:b'947be7',400:b'948b
401:b'94b10a',402:b'94d771',403:b'98063c',404:b'981dfa',
405:b'98398e',406:b'9852b1',407:b'9880ee',408:b'988389',409:b'98b8bc',410:b'9c02
411:b'9c2a83',412:b'9c3aaf',413:b'9c5fb0',414:b'9c65b0',
415:b'9c8c6e',416:b'9ca513',417:b'9cd35b',418:b'9ce063',419:b'9ce6e7',420:b'a007
421:b'a01081',422:b'a02195',423:b'a027b6',424:b'a06090',
425:b'a07591',426:b'a0821f',427:b'a0ac69',428:b'a0b4a5',429:b'a0cbfd',430:b'a0d7
431:b'a407b6',432:b'a4307a',433:b'a46cf1',434:b'a48431',
435:b'a49a58',436:b'a4d990',437:b'a4ebd3',438:b'a80600',439:b'a816d0',440:b'a82b
441:b'a830bc',442:b'a8346a',443:b'a8515b',444:b'a87650',
445:b'a87c01',446:b'a88195',447:b'a887b3',448:b'a89fba',449:b'a8f274',450:b'ac1e
451:b'ac3613',452:b'ac5a14',453:b'acafb9',454:b'acc33a',
455:b'acee9e',456:b'b047bf',457:b'b06fe0',458:b'b0c4e7',459:b'b0c559',460:b'b0d0
461:b'b0df3a',462:b'b0ec71',463:b'b41a1d',464:b'b43a28',
465:b'b46293',466:b'b47443',467:b'b49d02',468:b'b4bff6',469:b'b4ce40',470:b'b4ef
471:b'b857d8',472:b'b85a73',473:b'b85e7b',474:b'b86ce8',
475:b'b8bbaf',476:b'b8bc5b',477:b'b8c68e',478:b'b8d9ce',479:b'bc1485',480:b'bc20
481:b'bc4486',482:b'bc4760',483:b'bc5451',484:b'bc72b1',
485:b'bc765e',486:b'bc79ad',487:b'bc7abf',488:b'bc7e8b',489:b'bc851f',490:b'bca5
491:b'bcb1f3',492:b'bcd11f',493:b'bce63f',494:b'c01173',
495:b'c0174d',496:b'c0238d',497:b'c03d03',498:b'c048e6',499:b'c06599',500:b'c087
501:b'c08997',502:b'c0bdc8',503:b'c0d2dd',504:b'c0d3c0',
505:b'c0dcda',506:b'c418e9',507:b'c44202',508:b'c45006',509:b'c4576e',510:b'c45d
511:b'c462ea',512:b'c4731e',513:b'c488e5',514:b'c493d9',
515:b'c4ae12',516:b'c81479',517:b'c819f7',518:b'c83870',519:b'c85142',520:b'c87e
521:b'c8a823',522:b'c8bd69',523:b'c8d7b0',524:b'cc051b',
525:b'cc07ab',526:b'cc2119',527:b'cc464e',528:b'cc6ea4',529:b'ccb11a',530:b'ccf9
531:b'ccfe3c',532:b'd003df',533:b'd0176a',534:b'd03169',
535:b'd059e4',536:b'd07fa0',537:b'd087e2',538:b'd0b128',539:b'd0c1b1',540:b'd0c2
541:b'd0d003',542:b'd0dfc7',543:b'd0fccc',544:b'd411a3',
545:b'd47ae2',546:b'd487d8',547:b'd48890',548:b'd48a39',549:b'd49dc0',550:b'd4ae
551:b'd4e6b7',552:b'd4e8b2',553:b'd80831',554:b'd80b9a',
555:b'd831cf',556:b'd85575',557:b'd857ef',558:b'd85b2a',559:b'd868c3',560:b'd890
561:b'd8a35c',562:b'd8c4e9',563:b'd8e0e1',564:b'dc44b6',
565:b'dc6672',566:b'dc74a8',567:b'dc8983',568:b'dccf96',569:b'dcdce2',570:b'dcf7

```

571:b'e09971',572:b'e0aa96',573:b'e0c377',574:b'e0cbee',
575:b'e0d083',576:b'e0db10',577:b'e4121d',578:b'e432cb',579:b'e440e2',580:b'e458
581:b'e458e7',582:b'e45d75',583:b'e47cf9',584:b'e47dbd',
585:b'e492fb',586:b'e4b021',587:b'e4e0c5',588:b'e4f3c4',589:b'e4f8ef',590:b'e4fa
591:b'e8039a',592:b'e81132',593:b'e83a12',594:b'e84e84',
595:b'e86dcb',596:b'e87f6b',597:b'e89309',598:b'e8b4c8',599:b'e8e5d6',600:b'ec10
601:b'ec7cb6',602:b'ecaa25',603:b'ece09b',604:b'f008f1',
605:b'f03965',606:b'f05a09',607:b'f05b7b',608:b'f06bca',609:b'f0728c',610:b'f08a
611:b'f0e77e',612:b'f0ee10',613:b'f0f564',614:b'f40e22',
615:b'f4428f',616:b'f47190',617:b'f47b5e',618:b'f47def',619:b'f49f54',120:b'f4c2
621:b'f4d9fb',622:b'f4fefb',623:b'f83f51',624:b'f877b8',
625:b'f884f2',626:b'f88f07',627:b'f8d0bd',628:b'f8e61a',629:b'f8f1e6',630:b'fc03
631:b'fc1910',632:b'fc4203',633:b'fc643a',634:b'fc8f90',
635:b'fca13e',636:b'fca621',637:b'fcaab6',638:b'fcc734',639:b'fcde90',640:b'fcf1
}

```

```

def pack_cb(pack):
    mac = bytearray(6)
    pk = wlan.wifi_packet()
    control = pk.data[0]
    subtype = (0xF0 & control) >> 4
    type = 0x0C & control
    #print("Control:{}, subtype:{}, type:{}".format(control, subtype, type))
    if subtype == 4:
        for i in range (0,6):
            mac[i] = pk.data[10 + i]
            print ("Wifi Node with MAC: {}".format(ubinascii.hexlify(mac)))
            ya_esta=1

        if -95 < pk.rssi < 0 :
            if ubinascii.hexlify(mac)[0:6] in dict_xiaomi.values():
                ya_esta=0
                print('xiaomi')

            elif ubinascii.hexlify(mac)[0:6] in dict_oneplus.values():
                ya_esta=0
                print('oneplus')

            elif ubinascii.hexlify(mac)[0:6] in dict_BQ.values():
                ya_esta=0
                print('BQ')

```

```

elif ubinascii.hexlify(mac)[0:6] in dict_LG.values():
    ya_esta=0
    print('LG')

elif ubinascii.hexlify(mac)[0:6] in dict_samsung.values():
    ya_esta=0
    print('samsung')

elif ubinascii.hexlify(mac)[0:6] in dict_iphone.values():
    ya_esta=0
    print('iphone')

elif ubinascii.hexlify(mac)[0:6] in dict_apple.values():
    ya_esta=1
    print('apple')

elif ubinascii.hexlify(mac)[0:6] in dict_google.values():
    ya_esta=1
    print('ok google')

elif ubinascii.hexlify(mac)[0:6] in dict_nosesabe.values():
    ya_esta=1
    print('nosesabe')

elif ubinascii.hexlify(mac)[0:6] in dict_s.values():
    ya_esta=1
    print('esto no es movil')

else:
    ya_esta = 1
else:
    ya_esta = 1

for i in range(0,len(mac)):
    if macs[i][0] == ubinascii.hexlify(mac):
        timestamp = time.time()
        ya_esta=1
        macs[i][1]=timestamp
if ya_esta == 0:

```

```

        #print(pk)
        timestamp = time.time()
        macs.append([ubinascii.hexlify(mac), timestamp, timestamp])
        #print(macs)

wlan = WLAN()
wlan.init(mode=WLAN.STA)
#wlan = WLAN(mode=WLAN.STA, antenna=WLAN.EXT_ANT)
wlan.callback(trigger=WLAN.EVENT_PKT_MGMT, handler=pack_cb)
wlan.promiscuous(True)
#####

while True:
    try:
        wlan.promiscuous(False)
        time.sleep(1)
        wlan.promiscuous(True)
    except Exception as e:
        print(e)

    terminado = 0
    actual_time=time.time()
    personas=0
    try:
        while terminado==0:
            while True:
                for i in range(0,len(macs)):
                    if actual_time-macs[i][1] >= 14400:
                        macs.pop(i)
                        break
                terminado=1
                break

            for i in range(0,len(macs)):
                if actual_time-macs[i][2] <= 14400:
                    personas = personas+1
    except Exception as e:

```

```

        personas=1
        print("Error",e)

segundos_medir = 300

db = leer_valor_db(segundos_medir)
print(db)

personitas = get_personitas(db, personas)

print("Macs ",macs)
print("Numero de personitas ",personitas)

data = struct.pack("id", personitas, db)

try:
    s.send(data)
    print(lora.stats()) #Con esto puedo ver el RSSI y SNR y podria ser inter
except Exception as e:
    print("")

```


9.2. helper.py

```
from network import LoRa
import socket
import time
import binascii
import machine
import pycom
import struct
import math

hash_table = [247, 146, 42, 23, 21, 143, 201, 47, 100, 80, 12, 153, 58, 34, 238,

def factor():
    pycom.nvs_set('order', 0x00)
    pycom.nvs_set('hash', 0x00)
    return

def two_bytes_to_float(two_bytes):
    assert len(two_bytes) == 2
    raw = struct.unpack('H', two_bytes)
    raw = HalfToFloat(raw[0])
    return struct.unpack('f', struct.pack('I', raw))[0]

# http://forums.devshed.com/python-programming-11/converting-half-precision-floa
def HalfToFloat(h):
    s = int((h >> 15) & 0x00000001)    # sign
    e = int((h >> 10) & 0x0000001f)    # exponent
    f = int(h & 0x000003ff)            # fraction

    if e == 0:
        if f == 0:
            return int(s << 31)
        else:
            while not (f & 0x00000400):
                f <<= 1
                e -= 1
            e += 1
            f &= ~0x00000400
            print(s,e,f)
```

```

elif e == 31:
    if f == 0:
        return int((s << 31) | 0x7f800000)
    else:
        return int((s << 31) | 0x7f800000 | (f << 13))

e = e + (127 - 15)
f = f << 13

return int((s << 31) | (e << 23) | f)

def compute_over_the_air_time(payload_length, sf, cr):
    # lora.stats()[7] / 1000 # <- alternativamente
    payload_length += 13
    BW = 125e3
    preamble_symbols = 8
    header_length = 0
    explicit_header = 0

    if sf == 0 and cr == 0:
        return 0

    assert 7 <= sf <= 12
    assert 5 <= cr <= 8
    de = 1 if sf >= 11 else 0
    # http://forum.thethingsnetwork.org/t/spreadsheet-for-lora-airtime-calculation
    t_sym = pow(2, sf) / BW * 1000 # symbol time in ms
    t_preamble = (preamble_symbols + 4.25) * t_sym # over the air time of the preamble
    payload_symbol_number = 8 + max([(math.ceil(
        (8 * (payload_length + header_length) - 4 * sf + 28 + 16 - 20 * (1 -
            4 * (sf - 2 * de))) * cr), 0)]) # number of symbols of the payload
    t_payload = payload_symbol_number * t_sym # payload time in ms
    t_packet = t_preamble + t_payload
    return t_packet / 1000 # expressed in seconds

def craft_packet(tx_power, data_rate, coding_rate, counter, req_ack):
    assert 0 <= counter <= 255
    paq_bytes = bytes([data_rate, coding_rate, tx_power, counter & 0xff, req_ack])
    return paq_bytes

```

```

# Pearson hashing
def compute_hash(message):
    hash = len(message) % 256
    for i in message:
        hash = hash_table[(hash+i) % 256]

    return hash

def fill_bytes(counter, hash, num_bytes):
    assert isinstance(num_bytes, int)
    assert 2 < num_bytes <= 230
    counter = counter % 256
    return bytes([counter]) + bytes([hash]) + struct.pack('B'* (num_bytes - 2),

def accum_vector(vector):
    v = 0
    ret = []
    for _ in range(len(vector)):
        v += vector[_]
        ret.append(v)

    assert abs(ret[-1] - 1) < 1e-3
    ret[-1] = 1 # force 1
    return ret

def get_dr_cr_from_index(index):
    tx_power = [2, 6, 10, 14][index // 12]
    cr = 3 if ((index % 12) // 6) else 1
    dr = 12 - ((index % 6) + 7)
    return dr, cr, tx_power

def get_config_from_vector(config):
    r = get_random_0_1()
    acc_config = accum_vector(config)

    cfg = 0
    while r > acc_config[cfg]:
        cfg += 1

    return cfg

```

```

def get_config_from_accum_vector(acc_config):
    r = get_random_0_1()
    print("Random number", r)

    cfg = 0
    while r > acc_config[cfg]:
        cfg += 1

    return cfg

def get_random_0_1():
    return (machine.rng() % 16777215) / 16777215

def get_inter_arrival(lambda_):
    return - math.log(1.0 - get_random_0_1()) / lambda_

def create_lora_adr(handler, app_eui, app_key):
    lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868, adr = False)
    lora.callback(trigger=LoRa.RX_PACKET_EVENT, handler=handler)
    lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0, dr=0)

    while not lora.has_joined():
        time.sleep(2.5)
        print("Waiting for lora to join")

    return lora

def create_lora(handler, app_eui, app_key, tx_power = 14, coding_rate = 1):
    assert 2 <= tx_power <= 14
    assert 1 <= coding_rate <= 4

    lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868, adr = False)
    lora.tx_power(tx_power) # from 2 to 14
    lora.coding_rate(coding_rate) # 1 = 4/5, 2 = 4/6, 3 = 4/7, 4 = 4/8
    lora.callback(trigger=LoRa.RX_PACKET_EVENT, handler=handler)
    lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0, dr=0)

    while not lora.has_joined():
        time.sleep(2.5)
        print("Waiting for lora to join")

```

```

return lora

def create_lora_abp(tx_power = 14, coding_rate = 1):
    assert 2 <= tx_power <= 14
    assert 1 <= coding_rate <= 4

    lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868, adr = False)
    lora.tx_power(tx_power) # from 2 to 14
    lora.coding_rate(coding_rate) # 1 = 4/5, 2 = 4/6, 3 = 4/7, 4 = 4/8
    lora.callback(trigger=LoRa.RX_PACKET_EVENT, handler=update_tx_params_cb)
    lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))
    return lora

def create_socket_adr():
    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, False)
    s.setblocking(True)
    return s

def create_socket(data_rate = 5):
    assert 0 <= data_rate <= 5
    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
    # SF7 (data rate = 5)
    # SF12 (data rate = 0)
    s.setsockopt(socket.SOL_LORA, socket.SO_DR, data_rate)
    s.setsockopt(socket.SOL_LORA, socket.SO_CONFIRMED, False)
    s.setblocking(True)
    return s

def set_tx_power(lora, tx_power = 14):
    global g_tx_power
    assert 2 <= tx_power <= 14
    lora.tx_power(tx_power)
    # lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))
    g_tx_power = tx_power

def set_coding_rate(lora, coding_rate = 1):
    global g_coding_rate

```

```

    assert 1 <= coding_rate <= 4
    lora.coding_rate(coding_rate)
    # lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))
    g_coding_rate = coding_rate

def set_data_rate(s, lora, data_rate = 5):
    global g_data_rate
    assert 0 <= data_rate <= 5
    s.setsockopt(socket.SOL_LORA, socket.SO_DR, data_rate)
    # lora.tx_power(g_tx_power)    # dirty trick
    # lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))    #
    g_data_rate = data_rate

def data_rate_to_sf(data_rate):
    return (5 - data_rate) + 7

def coding_rate_to_cr(coding_rate):
    return coding_rate + 4

def get_off_period(payload_length = 4):
    sending_time = 3
    return max((
        compute_over_the_air_time(payload_length,
            data_rate_to_sf(g_data_rate),
            coding_rate_to_cr(g_coding_rate)) / 0.01) - sending_time,
        0)

```

9.3. Carpeta lib

9.3.1. microfono.py

```
from machine import UART
import utime
import time
from machine import Pin
import pycom
import ustruct
import machine

def leer_valor_db(segundos_medir):
    uart = UART(1, baudrate=9600, pins=('P21', 'P20'), parity=None, stop=1, bits=8)
    cont = 0
    valores = []
    segundos_10 = segundos_medir/10

    while cont != segundos_10:
        # pycom.rgbled(0xFF0000)
        uart.write(b'\x01\x03\x00\x00\x00\x01\x84\x0A') #consultar valor del ru
        # uart.write(b'\x00\x20\x00\x68') --consultar dirección del micro
        # uart.write(b'\x00\x10\x01\xBD\xC0') --cambiar la dirección del micro
        time.sleep(0.5)

        bytes_disp = uart.any()
        r = uart.read(bytes_disp)
        # pycom.rgbled(0x00FF47)

        print("result is: " + str(r))
        # if bytes_disp != 0 and bytes_disp > 5:
        dbs = ustruct.unpack(">h", r[3:5])[0]
        print("db value is: " + str(dbs/10))

        valores.append(dbs/10)

        time.sleep(10)
        cont += 1
```

```

        print(valors)
        print(cont)

    print('terminooooo')
    media = sum(valors)/len(valors)
    return media

def get_personitas(db, personas):
    if db < 48:
        personitas = personas

    elif 48 < db < 57:
        if personas <= 2:
            personitas = 2;
        else:
            personitas = personas

    elif 57 < db < 65:
        if personas <= 3:
            personitas = 3
        else:
            perosnitas = personas

    elif 65 < db < 75:
        if personas <= 4:
            personitas = 4
        else:
            personitas = personas

    elif 75 < db < 85:
        if personas <= 5:
            personitas = 5
        else:
            personitas = personas

    elif 85 < db < 95:
        if personas <= 6:
            personitas = 6
        else:
            perosnitas = personas

```



```
elif 95 < db < 105:
    if personas <= 7:
        personitas = 7
    else:
        personitas = personas

elif 105 < db < 115:
    if personas <= 8:
        personitas = 8
    else:
        personitas = personas
else:
    if personas >= 9:
        personitas = personas
    else:
        personitas = 9

print(personitas)
return personitas
```

