



Tec@



**Kit de desarrollo
Sigfox RCZ4**

**UFOX
Manual Técnico**

Versión 1.2 - 2020

**El futuro
es simple.**

INDICE

I.	Introducción.....	2
II.	Imagen de UFOX.....	2
III.	Especificaciones Técnicas.....	3
IV.	Aplicaciones.....	3
V.	Descripción General.....	3
	5.1 Resumen.....	3
	5.2 Descripción de Pines.....	4
	5.3 Diagrama de Bloques.....	4
VI.	Conexión de UFOX a una PC.....	5
	6.1 Reconocimiento en IDE Arduino.....	5
	6.2 Cargando un sketch a UFOX.....	6
	6.3 Comunicación serial con radio Sigfox.....	6
	6.4 Enviar un mensaje Sigfox.....	7
	6.5 Librería UFOX.....	8
	6.6 Enviar un mensaje Sigfox.....	8
VII.	Backend Sigfox.....	10
VIII.	Lista de comando AT disponibles.....	12
IX.	Certificación UFOX en Sigfox Ready.....	18
X.	Diagrama Electrónico.....	20

I. Introducción

Ufox es un kit de desarrollo para Internet de las Cosas (IoT) con comunicación Sigfox RCZ4 (Radio configuración zona 4). Es programable con lenguaje basado en Arduino. Incluye microcontrolador con comunicación USB y antena integrada de tipo helicoidal.

Fabricante:	Teca Perú SAC
Dirección:	Av. Alberto Alexander 2244, Lince - Lima, Perú
País:	Perú
Partida Registral, objeto social:	Partida registral TECA PERÚ S.A.C. Nro. 14419112, Número de Asiento B0001, Página 8 de 14. Objeto Social: "(...) Diseño y construcción de sistemas (...) comercialización e importación (...) de Tecnología, Telecomunicaciones y afines". Zona Registral Nro. IX - Sede Lima - Oficina registral Lima.
Modelo:	UFox
Marca:	Teca
Descripción:	Adaptador Inalámbrico
Función	Dispositivo inalámbrico para transmisión de datos (a título secundario)
Notas:	El dispositivo incorpora un módulo RF de marca WiSOL, modelo SFM10R4 (Comunicación Sigfox 920.1375 - 922.6625 Mhz). Con certificado FCC ID: 2ABA2SFM10R4 y código de homologación MTC - TRSS48222

II. Imagen de UFox



III. Especificaciones Técnicas

Microcontrolador:	ATMEGA 32U4-AU a 16MHZ
Voltaje de Operación:	3.3 VDC
Voltaje de Entrada (VIN):	3.7 - 9VDC
Interfaces:	USB, I2C, SPI, UART
Total Pines:	28
Pines Digitales I/O:	20 (5 PWM)
Pines Analógicos:	11
Temperatura de Operación:	-20 °C a +85 °C
Estándar de Comunicación:	Sigfox 54 canales (9 macro-canales x 6 micro-canales)
Tipos de Modulación:	DBPSK
Técnica de Modulación:	FHSS
Rango de Frecuencia:	920.1375 - 922.6625 MHz
Potencia de Transmisión Máxima:	0,2045 W máx.
Sensibilidad en Recepción:	-129 dBm
Antena:	Helicoidál 3dBi
Tamaño:	70mm Largo x 26mm Ancho x 19.5mm Alto

IV. Aplicaciones

- Educación y prototipos.
- Automatización de viviendas y edificios.
- Sistemas de alarma y seguridad.
- Entorno inteligente e industrial.
- Rastreo
- Telemetría de sensores.

V. Descripción General

5.1 Resumen

Ufox es un kit de desarrollo para Internet de las Cosas. Permite a los estudiantes y profesionales crear soluciones IoT en base a hardware y conectividad inalámbrica. La distribución de sus pines es adecuado para el uso en protoboard. Además de poseer conexión USB, esto facilita el proceso de prototipado y desarrollo de una solución IoT. Ufox se conecta a internet mediante la red 0G de Sigfox. Que habilita el funcionamiento de dispositivos transmitiendo datos por muchos años.

Se ha desarrollado una librería exclusiva para el devkit Ufox compatible con Arduino que facilita el proceso de desarrollo, disponible en Github:

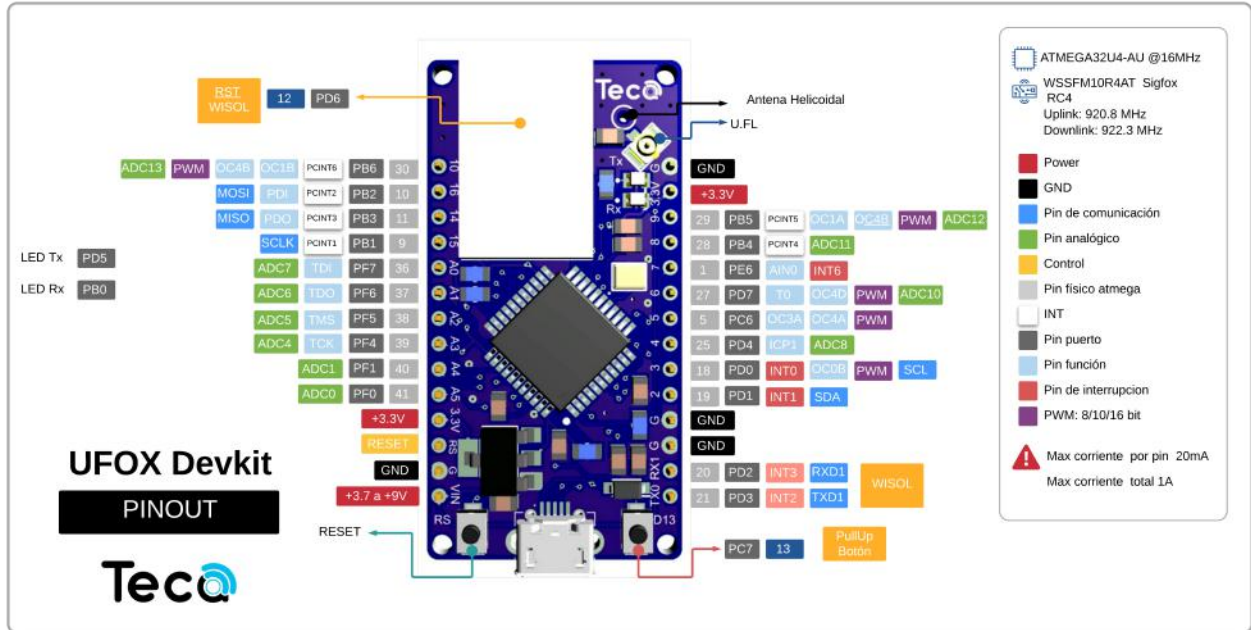
<https://github.com/TECA-IOT/Ufox>

Los datos son administrados directamente desde el backend de Sigfox:

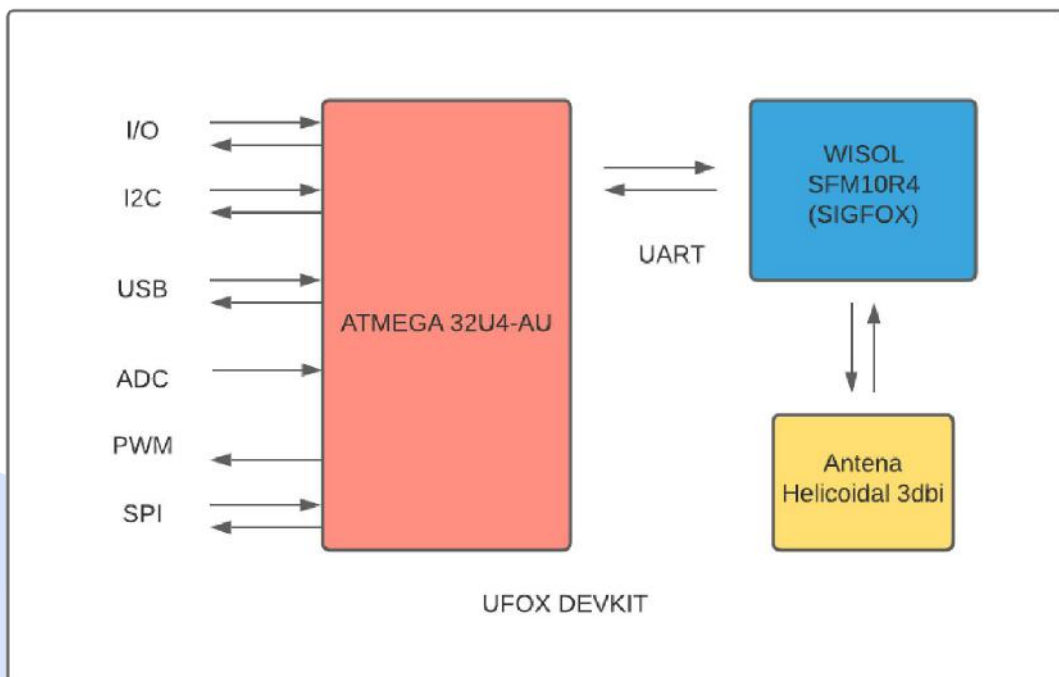
<https://backend.sigfox.com>

La información puede ser reenviada a otros servidores o también recibir la información directo a nuestro celular mediante un email.

5.2. Descripción de Pines

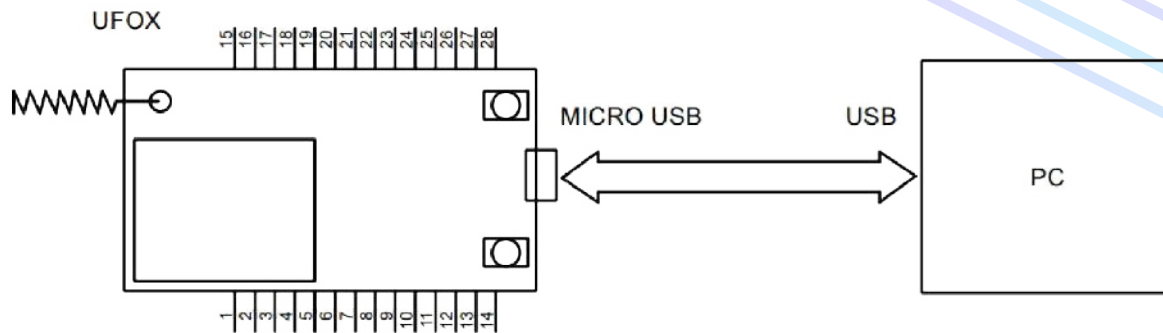


5.3. Diagrama de Bloques



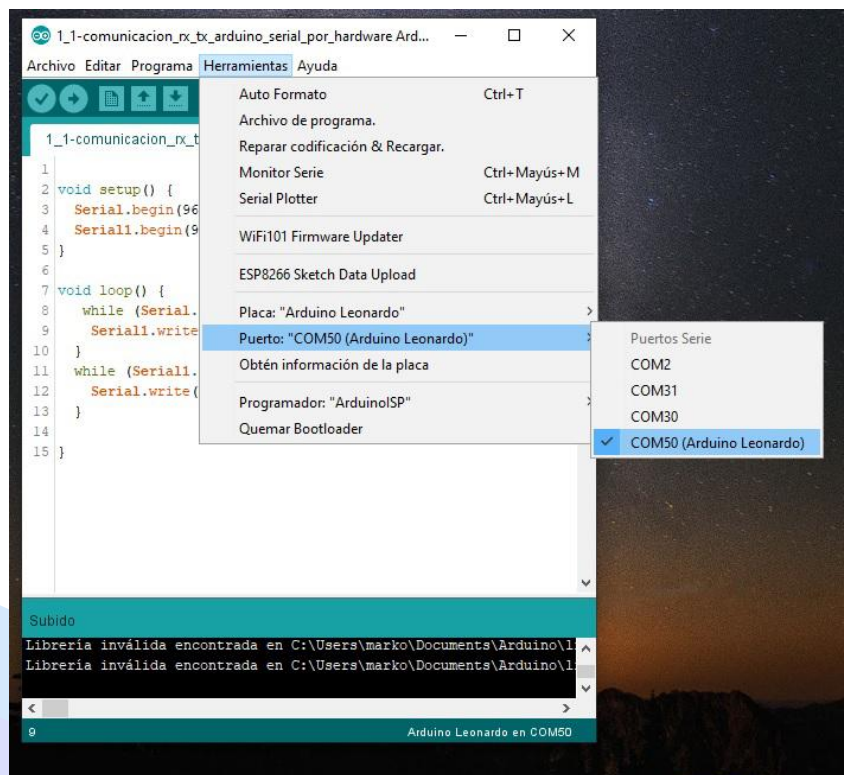
VI. Conexión de Ufox a una PC

Para empezar a programar el kit Ufox se debe conectar hacia una PC por medio de un cable USB, la conexión también servirá para energizar a la tarjeta. No es necesario instalar drivers, la instalación es inmediata, una vez reconocido se creará un puerto COM.



6.1 Reconocimiento en IDE Arduino

- Conecte el cable USB con el conector micro USB al devkit Ufox, luego conectar al puerto USB del PC.
- Espere a que la PC reconozca al devkit Ufox (el reconocimiento es instantáneo para SO Windows (7, 8, 10).
- Abra el IDE Arduino. Si no tiene instalado, descargar directamente desde este enlace: <https://www.arduino.cc/en/software>
- Ufox será reconocido con un puerto COM (Arduino Leonardo) como se muestra en la imagen.



6.2 Cargando un sketch a Ufox

Una vez que es el devkit Ufox y enumerado con un puerto COM, seleccione el menú "Herramienta" y luego la opción "Placa", elija la tarjeta "Arduino Leonardo". Puede cargar un ejemplo básico como el siguiente código.

```
int RXLED = 17; // El RXLED (AZUL) tiene un pin Arduino definido con numero 17

void setup()
{
  pinMode(RXLED, OUTPUT); //LED RX como salida
}

void loop()
{
  digitalWrite(RXLED, LOW); // encender RXLED
  TXLED0;                  // apagar TXLED
  delay(500);              // Esperar un segundo
  digitalWrite(RXLED, HIGH); // apagar el RXLED
  TXLED1;                  //TXLED macro, encender TXLED
  delay(500);              // Esperar un segundo
}
```

6.3 Comunicación Serial con Radio Sigfox

Ufox posee un radio de comunicación inalámbrico WiSol SFM10R4 que por medio de comunicación UART físico (Serial1 en Arduino) a 9600 baudios es posible entablar conexión la interacción con el radio, el manejo es por medio de comandos "AT", el radio trabaja a una frecuencia de 920MHZ Uplink (mensajes de subida) y 922 Dowlink (mensajes de bajada).

Puede usar este ejemplo para interactuar con el radio Sigfox SFM10R4 directamente.

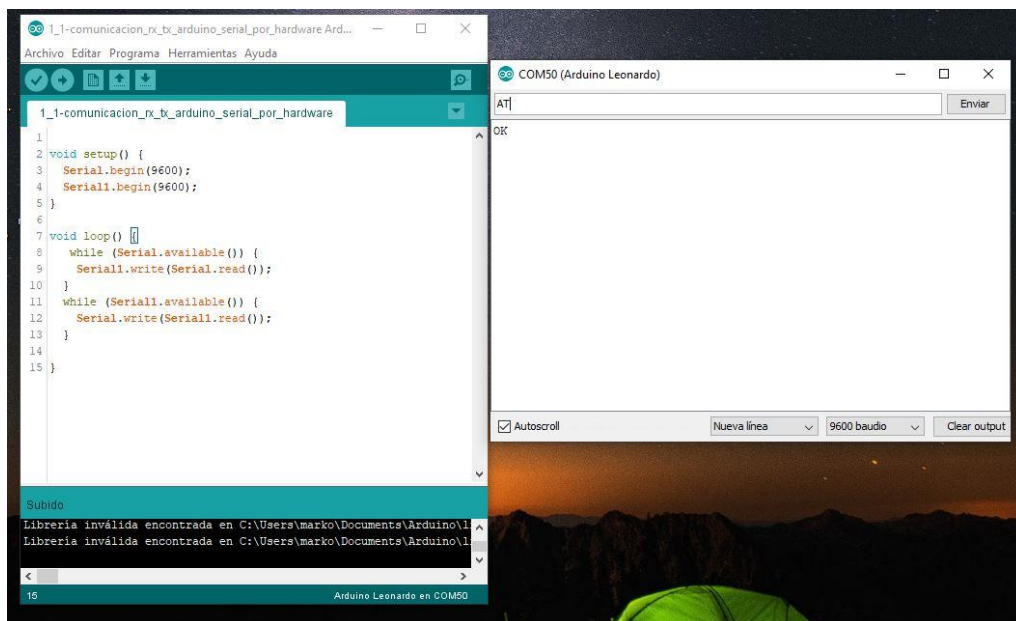
```
void setup() {
  Serial.begin(9600);
  Serial1.begin(9600);
}

void loop() {
  while (Serial.available()) {
    Serial1.write(Serial.read());
  }
  while (Serial1.available()) {
    Serial.write(Serial1.read());
  }
}
```

- Cargado el sketch abrir la terminal serial a 9600 baudios, seleccionar la opción "Nueva línea".
- Escribir el comando "AT", presionar enter, y a continuación el radio WiSol responderá con un "Ok".
- El radio está listo para recibir cualquier comando.

Comandos de interés:

- ID: AT\$I=10
- PAC: AT\$I=11
- Voltaje: AT\$V?
- Temperatura: AT\$T?



6.4 Enviar un mensaje Sigfox

Enviar un mensaje Sigfox (Uplink), como máximo se puede enviar 12 bytes en un mensaje, ejemplo: enviar el siguiente valor 255 (formato decimal) equivale a FF en hexadecimal.

Siga la siguiente secuencia de comando:

1. AT\$G1?
2. AT\$RC
3. AT\$SF=FF

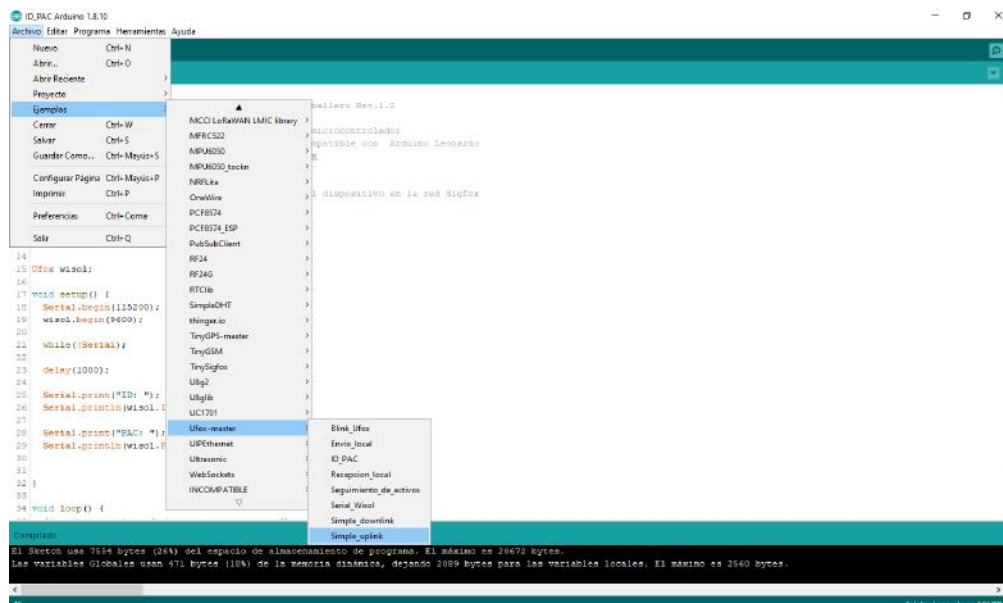
Si requiere recibir información desde la nube (Dowlink), ejecute el comando AT\$SF=FF,1

6.5 Librería Ufox

La librería Ufox incorpora una serie de funciones y métodos que simplifican la conexión a la red de Sigfox. Instale la librería Ufox en el IDE de Arduino descargando este repositorio, existen diversos modos de instalar librería externas.

Link: <https://github.com/TECA-IOT/Ufox>

- Opción 1: descomprima este repositorio .ZIP y copie la carpeta dentro de: [Unidad]:\Users[usuario]\Documents\Arduino\librarles
- Opción 2: desde el mismo IDE Arduino diríjase a "Programa, Incluir Librería, Añadir biblioteca .ZIP" localice el repositorio descargado y abrir.
- Una vez instalado recargue el Ide Arduino y podrá ver que la librería y ejemplo se muestran:



6.6 Enviar un mensaje Sigfox

En este ejemplo de la librería Ufox.h se envía un mensaje Sigfox con un valor random de 0 a 15 (formato decimal) cada vez que el botón D13 sea presionado. Tener en cuenta que el mensaje Sigfox que es indicado para ser transmitido, será convertido valores de codificación hexadecimal, es posible enviar un máximo de 12 bytes por mensaje. Todos los valores numéricos introducido en la función ufox.SEND() serán convertidos a valores hexadecimales, salvo que implícitamente envíe valores en formato hexadecimales, para este último caso la librería no convertirá a ningún formato adicional.

```

#include <Ufox.h>

#define btn 13
#define RXLED 17

Ufox wisol;

void setup() {
  pinMode(btn, INPUT);
  pinMode(RXLED, OUTPUT);

  Serial.begin(115200);
  wisol.begin(9600);

  while(!Serial); //comentar si usará una fuente de energía externa

  Serial.println("-Presione botón 13-");
}

void loop() {

  if(digitalRead(btn)==0){

    digitalWrite(RXLED, LOW);
    wisol.RST();

    uint32_t valor = random(0,15);
    Serial.println(wisol.SEND(valor));

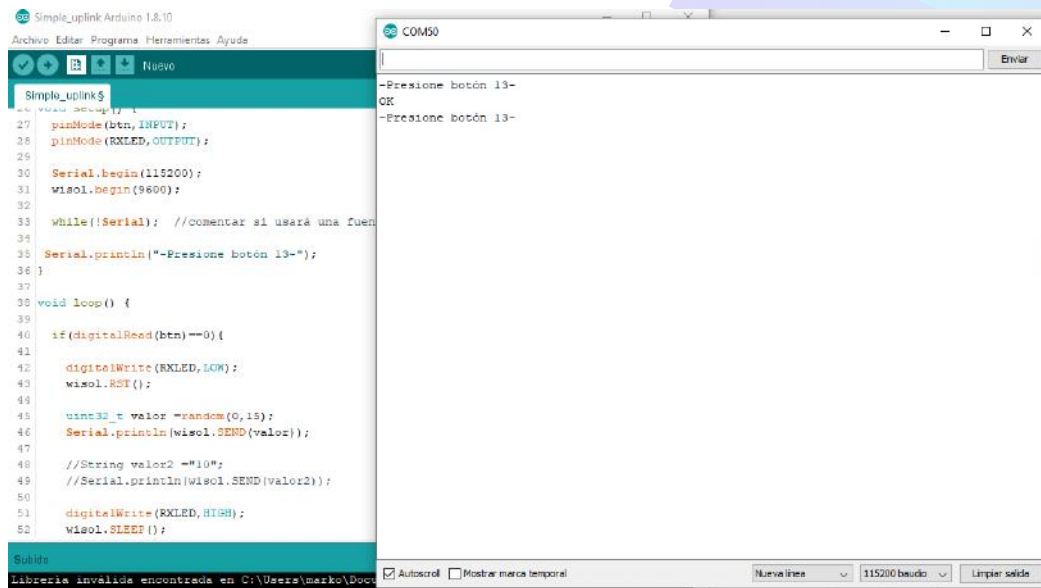
    //String valor2 = "10";
    //Serial.println(wisol.SEND(valor2));

    digitalWrite(RXLED, HIGH);
    wisol.SLEEP();
    delay(3000);
    Serial.println("-Presione botón 13-");
  }

}

```

Al abrir la terminal serial, se mostrará el mensaje de "Presionar botón 13", cada vez que se requiera enviar un mensaje Sigfox se deberá presionar el botón D13 de Ufox, y este automáticamente enviara el mensaje. La transmisión tarda en promedio 3 Segundos.



VII. Backend Sigfox

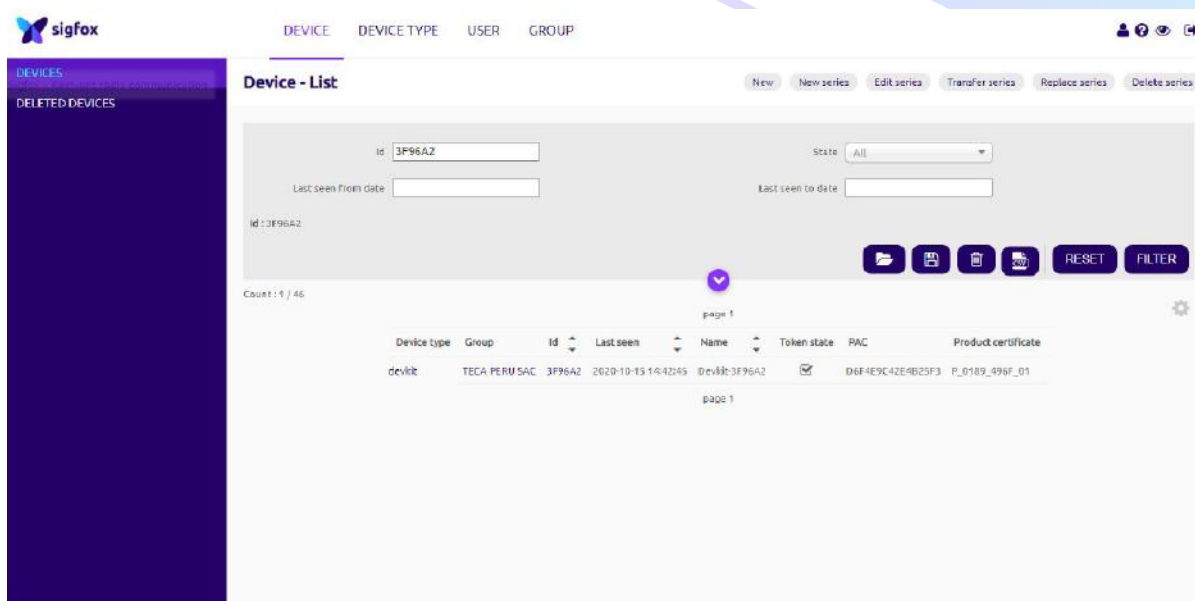
Es la plataforma de gestión de datos de los dispositivos previamente suscritos, en él se puede configurar y redirigir los mensajes hacia otras plataformas. Para iniciar sesión en su cuenta de Sigfox, ingrese a este enlace:

<https://backend.sigfox.com/auth/login>

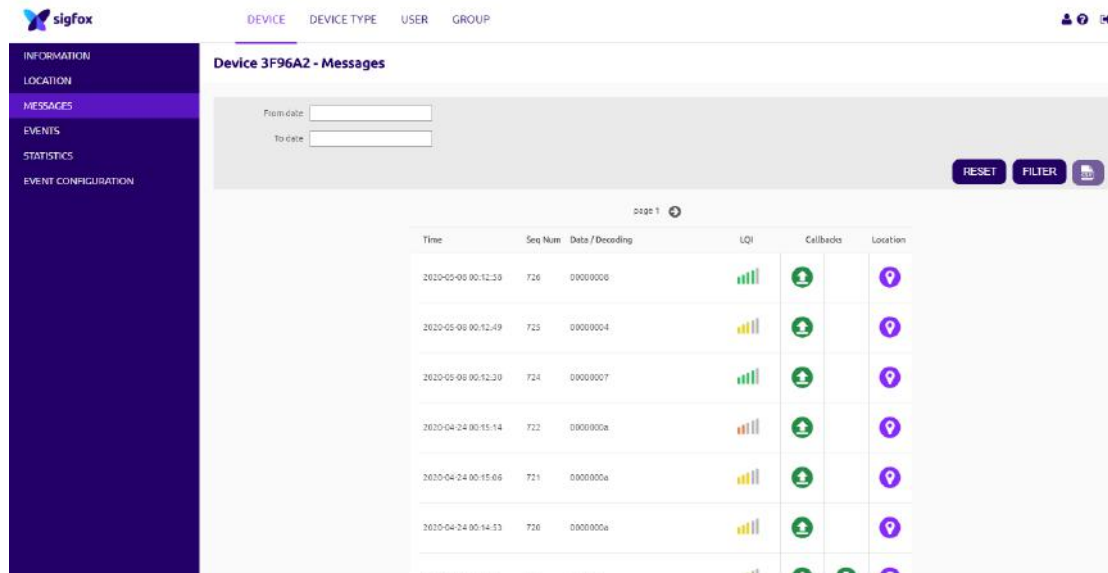
The image shows the login interface of the Sigfox backend. At the top is the Sigfox logo. Below it are two input fields: 'Email address' with an envelope icon and 'Password' with a lock icon. Under the password field is a checkbox labeled 'Remember me for one week'. At the bottom right is a 'Sign in' button.

La información que emitió Ufox se visualizará en el backend de Sigfox. Previamente debe tener asociado a su cuenta el kit Ufox.

Es necesario extraer el ID y PAC previamente para su activación.



Diríjase al ID del Ufox y dar click sobre encima de él, se abrirá una nueva ventana, para visualizar los mensajes, seleccione el menú "Messages".



La Información puede reenviarse a un servidor externo como AWS, AZURE, GOOGLE CLOUD, etc. Por medio de los Callbacks. Además también se pueden redirigir mensajes hacia servidores personalizados por medio de URLs mediante GET o POST.

Se debe definir la trama de datos a enviarse, puede ser en formulario o mediante un JSON, además se puede agregar algunos metadatos propios de la plataforma de Sigfox, como TIME, DEVICE, DATA, SEQNUMBER, entre otros.

Además es posible reenviar la Información a un Correo electrónico.
Se pueden crear múltiples Callbacks.

The screenshot shows the 'Device type devkit - Callback edition' page in the Sigfox web interface. The left sidebar contains navigation links: INFORMATION, LOCATION, ASSOCIATED DEVICES, DEVICES BEING REGISTERED, STATISTICS, EVENT CONFIGURATION, CALLBACKS (highlighted), and BULK OPERATIONS. The main content area has tabs for DEVICE, DEVICE TYPE, USER, and GROUP. The 'Callbacks' section is active, showing a form with the following fields: Type (DATA/UPLINK), Channel (EMAIL), Custom payload config (datafox: uint32), Recipient (marko.antonio.1.16.92@gmail.com), Subject (Devkit Ufox [device]), and Message (se ha recibido información de : {device} , data: {data}). The form also includes a 'Message' field and 'Ok'/'Cancel' buttons.

VIII. Lista de comando AT disponibles

Lista de comandos admisibles por el radio Sigfox integrado en Ufox, además de sus respectivos parámetros. Verificar previamente que el radio se encuentre encendido con un comando AT simple, si responde con OK, el radio se encuentra listo.

Comando	Nombre	Descripción
AT	Comando Básico	Simplemente devuelve "OK" y no hace nada más. Se puede utilizar para comprobar la comunicación.
ATSSB=bit[,bit]	Enviar un bit	Envíe un estado de bit (0 o 1). La bandera de bit opcional indica si AX-SFAZ debe recibir una trama de enlace descendente.
AT\$SF=frame[,bit]	Enviar trama	Envíe datos de carga útil, de 1 a 12 bytes. La bandera de bit opcional indica si AX-SFAZ debe recibir una trama de enlace descendente.
AT\$S0	Enviar manualmente mensaje fuera de banda	Envíe el mensaje fuera de banda
AT\$TR?	Obtener la repetición de transmisión.	Devuelve el número de repeticiones de transmisión. Predeterminado: 2

AT\$TR=?	Obtener rango de transmisión.	Devuelve el rango permitido de repeticiones de transmisión.									
AT\$TR=uint	Obtener transmisión repetida.	Establece la repetición de transmisión.									
AT\$uint?	Obtener registro.	Consultar el valor de un registro de configuración específico. Consulte el capítulo "Registros" para obtener una lista de registros.									
AT\$uint=uint	Establecer registro.	Cambiar un registro de configuración.									
AT\$uint=?	Obtener rango de registro.	Devuelve el rango permitido de repeticiones de transmisión.									
AT\$IF=uint	Establecer Frecuencia TX.	Establezca el canal de macro de portadora de salida para tramas Sigfox.									
AT\$IF?	Obtener frecuencia TX.	Obtenga la frecuencia TX elegida actualmente.									
AT\$DR=uint	Establecer frecuencia RX.	Configure el canal de macro de portadora de recepción para tramas Sigfox.									
AT\$DR?	Obtener frecuencia RX.	Obtenga la frecuencia RX elegida actualmente.									
ATSCW=uint,bit[,uint_opt]	Onda continua	<p>Para ejecutar pruebas de emisión para la certificación Sigfox es necesario enviar una onda continua, es decir, solo la frecuencia base sin ninguna modulación.</p> <p>Parámetros:</p> <table> <tr> <td>Nombre Frecuencia</td><td>Rango 800000000-999999999,0</td><td>Descripción Frecuencia de onda continua en Hz. Utilice 902200000 para Sigfox o 0 para mantener la frecuencia anterior.</td></tr> <tr> <td>Mode</td><td>0,1</td><td>Habilita o deshabilita la onda portadora.</td></tr> <tr> <td>Potencia</td><td>24</td><td>dBm de señal Predeterminado: 24</td></tr> </table>	Nombre Frecuencia	Rango 800000000-999999999,0	Descripción Frecuencia de onda continua en Hz. Utilice 902200000 para Sigfox o 0 para mantener la frecuencia anterior.	Mode	0,1	Habilita o deshabilita la onda portadora.	Potencia	24	dBm de señal Predeterminado: 24
Nombre Frecuencia	Rango 800000000-999999999,0	Descripción Frecuencia de onda continua en Hz. Utilice 902200000 para Sigfox o 0 para mantener la frecuencia anterior.									
Mode	0,1	Habilita o deshabilita la onda portadora.									
Potencia	24	dBm de señal Predeterminado: 24									
AT\$CB=uint_opt,bit	Modo test: TX byte constante	<p>Para las pruebas de emisión, es útil enviar un patrón de bits específico. El primer parámetro especifica el byte a enviar. Utilice '-1' para un patrón (pseudo-) aleatorio.</p> <p>Parámetros:</p> <table> <tr> <td>Nombre Patrón</td><td>Rango 0-255,-1</td><td>Descripción Byte para enviar. Utilice "-1" para (pseudo patrón aleatorio).</td></tr> <tr> <td>Modo</td><td>0,1</td><td>Activa o desactiva el modo de prueba de patrón.</td></tr> </table>	Nombre Patrón	Rango 0-255,-1	Descripción Byte para enviar. Utilice "-1" para (pseudo patrón aleatorio).	Modo	0,1	Activa o desactiva el modo de prueba de patrón.			
Nombre Patrón	Rango 0-255,-1	Descripción Byte para enviar. Utilice "-1" para (pseudo patrón aleatorio).									
Modo	0,1	Activa o desactiva el modo de prueba de patrón.									

AT\$RC	Reset de macrocanal	Este comando restablece el macrocanal. Las operaciones de transmisión posteriores (AT \$ SO, AT \$ SB, AT \$ SF) pueden pausar hasta 20 s para garantizar el cumplimiento.
AT\$T?	Obtener temperatura	Mida la temperatura interna en grados Celsius.
AT\$V?	Obtener Voltajes	Devuelve el voltaje actual y el voltaje medidos durante la última transmisión en mV.
AT\$I=uint	Información	<p>Muestra información diversa del producto:</p> <p>0: Nombre y versión del software Respuesta de ejemplo: AX - Sigfox 1.2 - RCZ4</p> <p>1: Detalles de contacto Respuesta de ejemplo: support@axsem.com</p> <p>2: Byte inferior de revisión de silicio Respuesta de ejemplo: 8F</p> <p>3: Byte superior de revisión de silicio Respuesta de ejemplo: 51</p> <p>4: Versión de firmware principal Respuesta de ejemplo: 1</p> <p>5: versión de firmware menor Respuesta de ejemplo: 1</p> <p>7: Variante de firmware (banda de frecuencia, etc. (UE / EE. UU.)) Respuesta de ejemplo: RCZ4</p> <p>9: Versión de la biblioteca SIGFOX Respuesta de ejemplo: UDL1-1.8.9</p> <p>10: ID de dispositivo Ejemplo de respuesta: 00012345</p> <p>11: PAC Ejemplo de respuesta: 0123456789ABCDEF</p>
AT\$P=uint	Establecer modo de energía	<p>Para ahorrar energía, el AX - SFAZ se puede poner en reposo manualmente. Dependiendo del modo de energía, será responsable de reactivar el AX - SFAZ nuevamente.</p> <p>0: restablecimiento del software (la configuración se restablecerá a los valores en flash)</p> <p>1: dormir (enviar un descanso para despertar)</p>

		2: sueño profundo (alternar el pin GPIO9 o RESET_N para despertar; el AX - SFAZ no se está ejecutando y todas las configuraciones se restablecerán)
AT\$WR	Guardar configuración	Escriba todas las configuraciones en flash (frecuencias RX / TX, registros) para que sobrevivan al reinicio / sueño profundo o pérdida de energía. Utilice AT \$ P = 0 para restablecer el AX - SFAZ y cargar la configuración desde la memoria flash.
AT:Pn?	Obtener pin GPIO	Devuelve la configuración del GPIO Pin n; n puede oscilar entre 0 y 9. Se devuelve una cadena de caracteres que describe el modo del pin, seguida del valor real. Si el pin está configurado como pin analógico, se devuelve el voltaje (rango 0... 1 V). Los caracteres de modo tienen el siguiente significado: <div> <div>Modo</div> <div>Descripción</div> </div> <div> <div>0</div> <div>Pin conduce bajo</div> </div> <div> <div>1</div> <div>Pin conduce alto</div> </div> <div> <div>Z</div> <div>Pin es una entrada de alta impedancia</div> </div> <div> <div>U</div> <div>Pin es entrada con pull-up</div> </div> <div> <div>A</div> <div>Pin es una entrada analógica (GPIO pin 0... 3 solamente)</div> </div> <div> <div>T</div> <div>Pin es impulsado por reloj o DAC (solo pin GPIO 0 y 4)</div> </div> El modo predeterminado después de salir del reinicio es U en todos los pines GPIO.
AT:Pn=?	Obtener rango de pin GPIO	Imprime una lista de posibles modos para un pin. La siguiente tabla muestra la respuesta. <div> <div>Pin</div> <div>Modos</div> </div> <div> <div>P0</div> <div>0, 1, Z, U, A, T</div> </div> <div> <div>P1</div> <div>0, 1, Z, U, A</div> </div> <div> <div>P4</div> <div>0, 1, Z, U, T</div> </div> <div> <div>P5</div> <div>0, 1, Z, U</div> </div> <div> <div>P6</div> <div>0, 1, Z, U</div> </div> <div> <div>P7</div> <div>0, 1, Z, U</div> </div> <div> <div>P8</div> <div>0, 1, Z, U</div> </div> <div> <div>P9</div> <div>0, 1, Z, U</div> </div>
AT:Pn=mode	Establecer pin GPIO	Configure el modo de pin GPIO. Para obtener una lista de los modos, consulte el comando AT: Pn?
AT:ADC Pn[-Pn[(1V 10V)]]?	Establecer pin GPIO voltaje analógico	Mide el voltaje aplicado a un pin GPIO. El comando también permite medir la diferencia de voltaje entre dos pines GPIO. En el modo diferencial, el rango de escala completa también se puede especificar como 1 V o 10 V. Sin embargo, tenga en cuenta que los voltajes de entrada de los pines no deben exceder el rango 0..VDD _IO.

		El comando devuelve el resultado como una fracción del rango de escala completa (1 V si no se especifica ninguno). Los pines GPIO a los que se hace referencia deben inicializarse en modo analógico antes de emitir este comando.															
AT: SPI[(A B C D)]=bytes	Transacción SPI	<p>Este comando registra los bytes en el puerto SPI. La frecuencia del reloj es de 312,5 kHz. El comando devuelve los bytes leídos en MISO durante la salida. Opcionalmente, se puede especificar el modo de reloj (el predeterminado es A).</p> <table> <tr> <td>Mode</td><td>Clock Inversion</td><td>Clock Phase</td></tr> <tr> <td>A</td><td>normal</td><td>normal</td></tr> <tr> <td>B</td><td>normal</td><td>alternate</td></tr> <tr> <td>C</td><td>inverted</td><td>normal</td></tr> <tr> <td>D</td><td>inverted</td><td>alternate</td></tr> </table>	Mode	Clock Inversion	Clock Phase	A	normal	normal	B	normal	alternate	C	inverted	normal	D	inverted	alternate
Mode	Clock Inversion	Clock Phase															
A	normal	normal															
B	normal	alternate															
C	inverted	normal															
D	inverted	alternate															
AT:CLK=freq,refreq	Establecer reloj generador	Emite una onda cuadrada en los pines configurados en modo T. La frecuencia de la onda cuadrada es $(\text{freq} / 2^{16}) \times \text{refreq}$. Los valores posibles para refreq son 20000000, 10000000, 5000000, 2500000, 1250000, 625000, 312500, 156250. Los valores posibles si freq son 0... 65535.															
AT:CLK=OFF	Apagar reloj generador	Apague el generador de reloj.															
AT:CLK?	Establecer reloj generador	Devuelve la configuración del generador de reloj. Se devuelven dos números, freq y refreq.															
AT:DAC=value	Establecer sigma delta DAC	Genere un valor DAC en los pines configurados en modo T. El valor del parámetro puede estar en el rango -32768... 32767. El voltaje de salida promedio es $(1/2 + \text{valor} / 2^{17}) \times \text{VDD}$. Se necesita un filtro de paso bajo externo para obtener voltajes de salida suaves. La frecuencia de modulación es de 20 MHz. Una posible elección de filtro de paso bajo es un filtro de paso bajo RC simple con $R = 10 \text{ k}$ y $C = 1 \text{ uF}$.															
AT:DAC=OFF	Apagar sigma delta DAC	Apague el DAC.															
AT:DAC?	Obtener Sigma delta DAC	Devuelve el valor DAC.															

AT\$TM=mode,config	Activar el modo Test de Sigfox	<p>Modos de prueba disponibles:</p> <p>0. TX BPSK:</p> <p>Enviar solo BPSK con bit de sincronismo + trama sincronizada + secuencia PN: sin saltos centrados en la frecuencia TX. Los bits de configuración 0 a 6 definen el número de repeticiones. El bit 7 de la configuración define si se aplica un retraso o no en el bucle.</p> <p>1. Protocolo TX:</p> <p>Modo Tx con protocolo completo con clave Sigfox: envíe tramas de protocolo Sigfox con bandera de inicio de enlace descendente = Verdadero. Config define el número de repeticiones.</p> <p>2. Protocolo RX:</p> <p>Este modo prueba el protocolo de enlace descendente completo solo en enlace descendente. Config define el número de repeticiones.</p> <p>3. RX GFSK:</p> <p>Modo RX con patrón conocido con SB + SF + Patrón en RX_frequency (comparación interna con la trama recibida patrón conocido = AA AA B2 27 1F 20 41 84 32 68 C5 BA AE 79 E7 F6 DD 9B. Config define el número de repeticiones. Config. define el número de repeticiones.</p> <p>4. Sensibilidad RX:</p> <p>Hace enlace ascendente + enlace descendente con clave Sigfox y tiempos específicos. Esta prueba es específica de los equipos de prueba y software de SIGFOX.</p> <p>5. Síntesis de TX:</p> <p>Hace un marco de enlace ascendente en cada canal Sigfox para medir el paso de síntesis de frecuencia.</p>
AT\$SE	Inicia AT\$TM- 3,255 indefinidamene	Comando de conveniencia para pruebas de sensibilidad.
AT\$SL[=frame]	Enviar trama local	Envía una trama de bucle local con una carga útil opcional de 1 a 12 bytes. Carga útil predeterminada: 0x84, 0x32, 0x68, 0xC5, 0xBA, 0x53, 0xAE, 0x79, 0xE7, 0xF6, 0xDD, 0x9B.
AT\$RL	Recepcionar trama local	Empieza a escuchar un tramalocal.

Tabla de Registros:

Número	Nombre	Descripción	Por defecto	Rango	Unidades
300	Periodo fuera de banda	AX - SFAZ envía mensajes estáticos periódicos para indicar que están vivos. Establecer en 0 para deshabilitar.	24	0-24	Horas
400	Mascara macrocanal	La máscara de macrocanales a utilizar.	<00000000> <c0000000> <0000007f>,63		
410	Configuración de la clave de cifrado	Establecer en cero para funcionamiento normal. Establecer en uno para usar con el kit de emulador de red SIGFOX (SNEK)	0	0-1	0: llave privada 1: llave pública

IX. Certificación Ufox en Sigfox Ready

El devkit Ufox posee un código de certificación P_0189_496F_01 otorgado por Sigfox Francia, que identifica Ufox en la red de Sigfox OG.

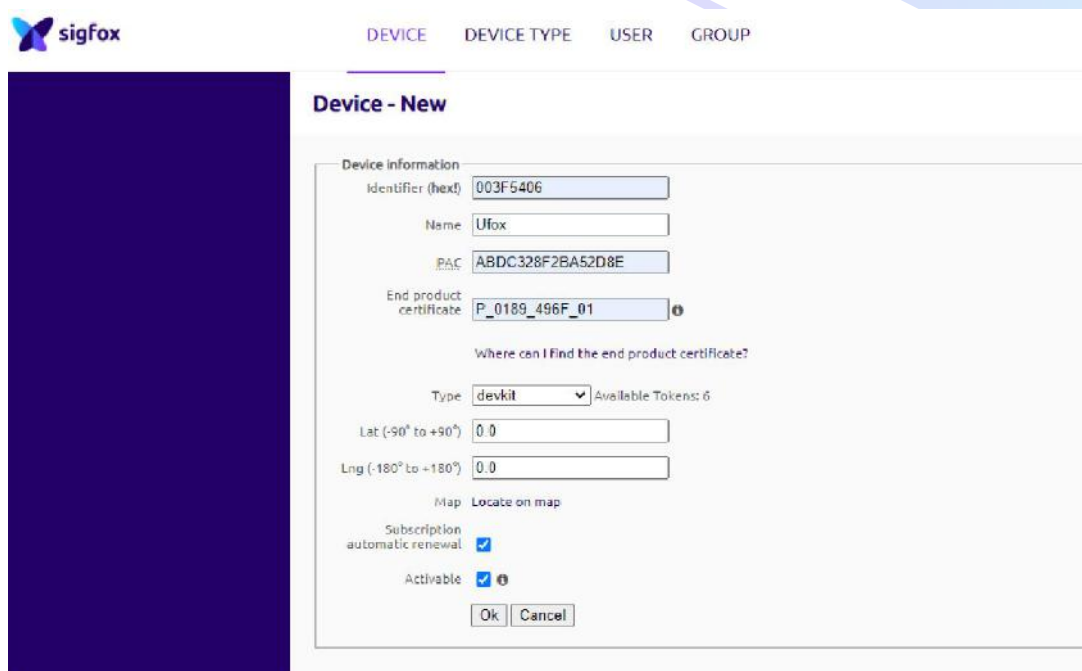
Certification id: P_0189_496F_01
Product manufacturer: TECA Perú
Product name: UFox
Product version: 1.0
Radio configurations:

RC4

Maximum radiated power EIRP: N/A
Uplink class: N/A

Modes: Downlink, Uplink
DevKit:
Payload encryption:
Input sensitivity: N/A
Message modulo: M_4096
Qualification date: 2020-02-13
Report number:
Description: Devkit
External Id:
Creation date: 2020-02-13 11:48:15
Created by: Brigitte Ray
Last edition date: 2020-02-13 11:48:15
Last edited by: Brigitte Ray

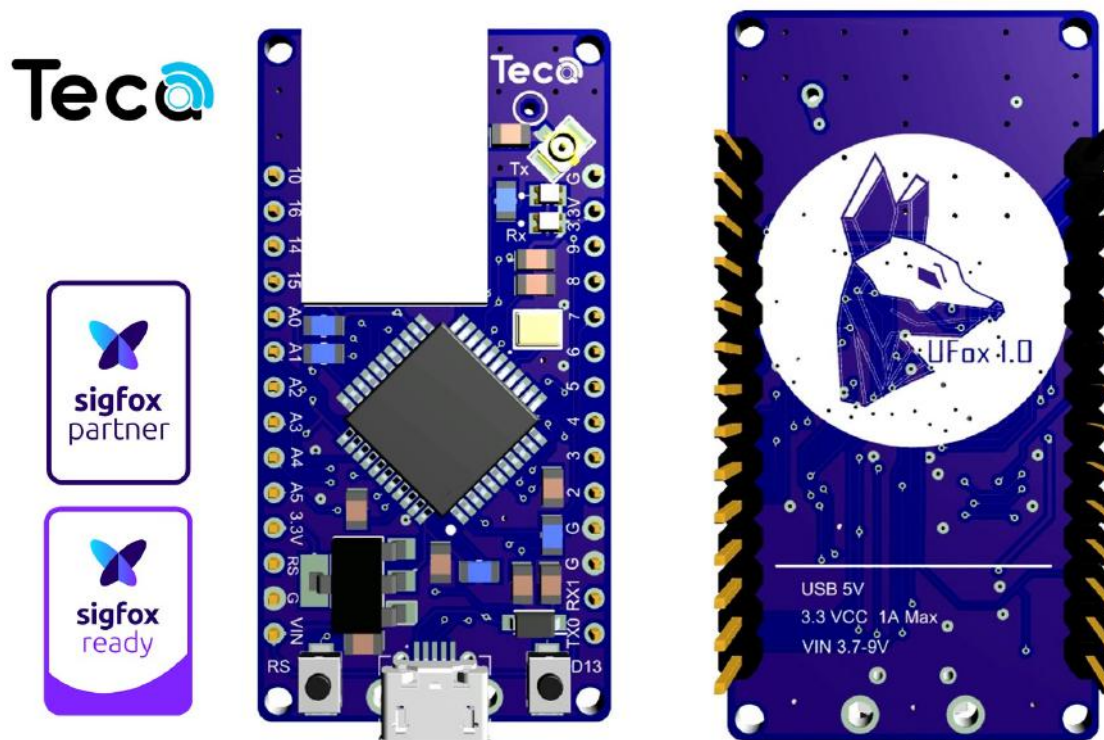
El código de certificación es necesario al momento de activar el kit Ufox en la plataforma Backend Sigfox.



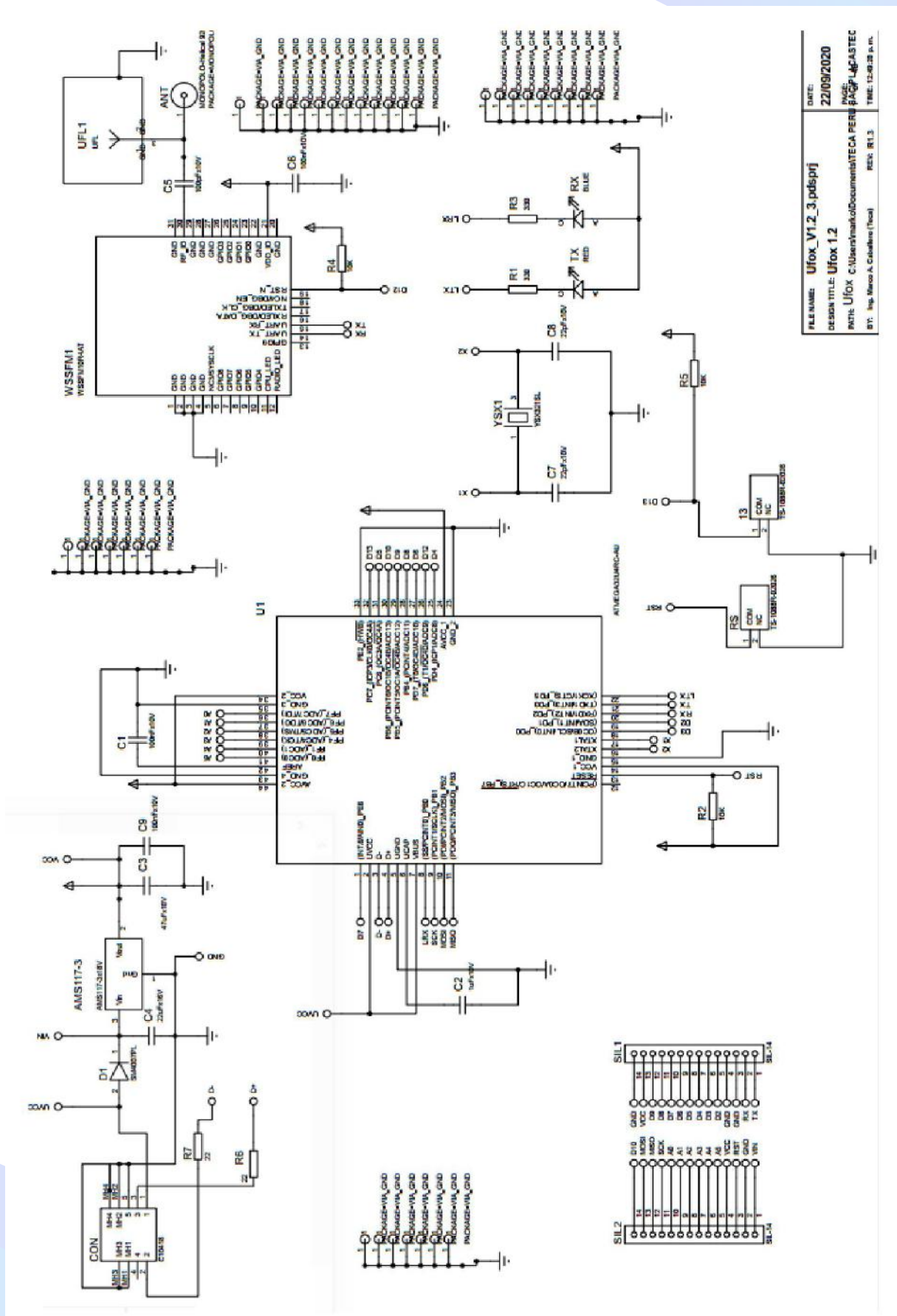
The screenshot shows the 'Device - New' form in the Sigfox Backend. The form is titled 'Device - New' and has tabs for 'DEVICE', 'DEVICE TYPE', 'USER', and 'GROUP'. The 'DEVICE' tab is selected. The form contains the following fields and options:

- Device information:**
 - Identifier (hex): 003F5406
 - Name: Ufox
 - PAC: ABDC328F2BA52D8E
 - End product certificate: P_0189_496F_01
- Where can I find the end product certificate?**
 - Type: devkit (dropdown menu)
 - Available Tokens: 6
- Location:**
 - Lat (-90° to +90°): 0.0
 - Lng (-180° to +180°): 0.0
- Map:** Locate on map
- Subscription:**
 - automatic renewal: ☒
 - Activable: ☒
- Buttons:** Ok, Cancel

La certificación Sigfox Ready es concedido a los dispositivos que han sido validados por Sigfox en Francia, sede central. Este certificado respalda que el dispositivo cumple con requisitos de protocolo y rendimiento en radiofrecuencia Sigfox.



X. Diagrama Electrónico



FILE NAME:	Ufox_V1.2_3.pdspri	DATE:	22/09/2020
DESIGN TITLE:	Ufox 1.2	PAGE:	3
PATH:	C:\Users\hank\Documents\TECA PERE\Ufox\Ufox1.2	PROJECT:	Ufox1.2
BY:	Ing. Marco A. Calabrese (Tutor)	REV:	R1.3
		TIME:	12:48:28 p.m.



Teca Perú S.A.C
Av. Alberto Alexander 2244,
Dist. Lince - Lima

Tel. 998481553
<https://teca.pe>
administracion@teca.pe