

---

# PROYECTO MOTAM

## GUÍA DE INSTALACIÓN Y DESPLIEGUE DE LA PLATAFORMA

### TABLA DE CONTENIDO

<b>Guía de instalación y despliegue de la plataforma.....</b>	<b>1</b>
<b>Control de versiones del documento.....</b>	<b>2</b>
<b>Instalación de la pasarela – Raspberry Pi.....</b>	<b>3</b>
Conexión de los componentes hardware .....	3
Instalación del sistema operativo (Raspbian) .....	4
Configuración de la pantalla tactil .....	4
Instalación de Node.JS .....	4
Instalación de las dependencias (paquetes software) .....	5
Copia de los ficheros/componentes de la plataforma motam.....	5
Permisos necesarios para la ejecución.....	6
Instalación de los módulos Node.JS .....	6
Instalación de los paquetes Python .....	6
Creación de acceso directo en escritorio .....	6
<b>Instalación de la plataforma hardware Arduino/Genuino 101.....</b>	<b>6</b>
Conexión de los componentes hardware .....	7
Instalación del entorno de desarrollo Arduino .....	8
Programación del microcontrolador .....	8
Poner en funcionamiento los actuadores .....	9
<b>Instalación de los sensores RFduino .....</b>	<b>9</b>
Instalación y configuración del entorno de desarrollo para RFduino .....	9
Programación del microcontrolador de los sensores RFduino .....	10
Sensor de climatología .....	10
Sensor silla de bebés.....	11
Sensor calidad del aire .....	12
Sensor de luces de semáforo .....	12
Sensor de presencia de bicicleta .....	13
Modificación de sensores RFduino .....	14
<b>Instalación del módulo NB-IoT .....</b>	<b>15</b>
Programación de la placa Arduino con la shield NB-IoT .....	16

## **CONTROL DE VERSIONES DEL DOCUMENTO**

30 nov. 2017	Añadido apartado “Modificación de sensores RFduino” e índice
8 ene. 2018	Añadidas instrucciones para evitar conflicto con Wifi Direct surgido tras actualizar los paquetes en “Instalación de las Dependencias”
5 feb. 2018	Añadido apartado “Instalación del módulo NB-IoT”. Actualizada URL del repositorio de Sensores MOTAM.
15 feb. 2018	Actualizada información placas Arduino compatibles con el módulo SODAQ NB-IOT SHIELD.

En esta guía se describen los pasos a seguir para instalar y desplegar los distintos componentes -software y hardware- que conforman la plataforma MOTAM.

## INSTALACIÓN DE LA PASARELA – RASPBERRY PI

La pasarela, consistente en la plataforma hardware *Raspberry Pi*, es la base sobre la que se despliegan tanto los sensores MOTAM como la pasarela de comunicación entre puerto serie y BLE.



ILUSTRACIÓN 1: PLATAFORMA HARDWARE RASPBERRY PI (PASELA) CON ACTUADORES CONECTADOS POR PUERTO SERIE

A continuación, se presentan los pasos necesarios para llevar a cabo la configuración y el despliegue de la pasarela y del resto de componentes MOTAM a ejecutar sobre la misma.



ILUSTRACIÓN 2: ELEMENTOS DE LA PASARELA: PLATAFORMA HARDWARE RASPBERRY PI

## CONEXIÓN DE LOS COMPONENTES HARDWARE

La conexión del receptor GPS y del lector OBD-II, dispositivos USB en los que se basan los sensores de posición y velocidad de la plataforma MOTAM, es tan simple como conectarlos a los puertos USB libres de la Raspberry Pi. El orden en el que se conecten es **indiferente**, y pueden ser conectados tanto con la Raspberry Pi apagada como encendida.

La transmisión mediante BLE desde la pasarela se realiza mediante un chip integrado en el actual modelo de la Raspberry Pi (Model 3B), por lo que no tenemos que preocuparnos en conectarlo ni configurarlo.

## INSTALACIÓN DEL SISTEMA OPERATIVO (RASPBIAN)

Se deberán seguir los siguientes pasos:

1. Descargar la última versión del sistema operativo *Raspbian* desde la web (<https://www.raspberrypi.org/downloads/raspbian/>). Elegir la opción “*Raspbian with Desktop*” para poder tener un escritorio en nuestro entorno. La versión usada en nuestro caso ha sido la “*Raspbian Stretch with Desktop*”.
2. Descargar el programa “*Win32DiskImager*”: <https://sourceforge.net/projects/win32diskimager/>
3. Extraer el fichero .zip una vez descargado el archivo.
4. Formatear la tarjeta SD. Se recomienda el uso de la herramienta SD Formatter 5.0 ([https://www.sdcard.org/downloads/formatter\\_4/](https://www.sdcard.org/downloads/formatter_4/))
5. Volcar la imagen de Raspbian en la tarjeta SD. Para ello introducimos la tarjeta SD en el PC, abrimos Win32DiskImager, seleccionamos el fichero .img descomprimido, seleccionamos la tarjeta SD en “Device” y pulsamos en “Write”.
6. Conectamos teclado, ratón y monitor. Enchufamos la Raspberry Pi y la instalación se realizará automáticamente, mostrándonos por pantalla el escritorio Raspbian. Ahora podemos conectarnos a la red mediante un cable de red o mediante Wifi haciendo clic en el icono correspondiente.

NOTA: Las credenciales para entrar son “pi:raspberry”.

## CONFIGURACIÓN DE LA PANTALLA TACTIL

Es posible que cuando se inicie el sistema operativo de Raspberry Pi, la pantalla se encuentre rotada 180º. Para solucionar esto, debemos ejecutar:

```
$ sudo su -c 'echo lcd_rotate=2 >> /boot/config.txt'
```

## INSTALACIÓN DE NODE.JS

Para la instalación de Node.JS, entorno sobre el cual se apoya el software de la pasarela, se deben seguir los siguientes pasos en la terminal:

1. El primer paso es actualizar los repositorios y el software de nuestra versión de Raspbian. Para ello ejecutamos:  

```
$ sudo apt-get update && sudo apt-get upgrade
```
2. Añadimos el repositorio Debian de “nodesource.com” en el sistema. Para ello hay que descargar el script y ejecutarlo en el sistema:  

```
$ curl -sL https://deb.nodesource.com/setup_6.x | sudo -E bash -
```
3. Instalamos Node.JS versión 6 en el sistema:  

```
$ sudo apt-get install nodejs
```

NOTA: Se ha elegido la versión 6 de Node.JS ya que está destinada a ser la versión actual LTS.

### INSTALACIÓN DE LAS DEPENDENCIAS (PAQUETES SOFTWARE)

Para instalar y configurar correctamente cada una de las dependencias necesarias para la ejecución de todos los componentes de la plataforma MOTAM, deberemos seguir los siguientes pasos:

1. En primer lugar deberemos instalar los paquetes software que gestionan el receptor GPS.  
`$ sudo apt-get install gpsd gpsd-clients`
2. El paquete instala un servicio que escucha continuamente un socket local y ejecuta gpsd cuando algún cliente se conecta al socket. Dicho servicio interfiere con otras ejecuciones de gpsd (como las del software MOTAM), por lo que se hace necesario desactivarlo.  
`$ sudo systemctl stop gpsd.socket`  
`$ sudo systemctl disable gpsd.socket`
3. Instalar el simulador de interfaz OBDII. Para ello ejecutamos:  
`$ sudo apt-get install obdgpslogger`
4. Instalar la utilidad “socat”, necesario para usar puertos serie virtuales para el simulador de la interfaz OBDII.  
`$ sudo apt-get install socat`
5. Instalar udhcpd, necesario para crear la red de Wifi Direct con la que comunicaremos pasarela y terminal móvil.  
`$ sudo apt-get install udhcpd`
6. Deshabilitar el servicio cliente de DHCP para el correcto funcionamiento de Wifi Direct. Con la actualización 6.11.5 del paquete DHCPCD5 se produce un conflicto entre la creación de una interfaz virtual de Wifi Direct y el cliente DHCPCD. Por tanto, es necesario deshabilitarlo.  
`$ sudo systemctl disable dhcpcd.service`
7. Reiniciar el sistema. Es necesario reiniciar para que los cambios surtan efecto –en especial la deshabilitación de DHCPCD.  
`$ sudo reboot`

NOTA: En las nuevas versiones de Raspbian, ya vienen instalados los paquetes software relativos a Bluetooth, por lo que no es necesario instalarlos.

NOTA 2: Si se necesita conexión a internet, la mejor opción es por Ethernet, puesto que el interfaz WiFi integrado en la Raspberry Pi se usará para crear la conexión Wifi Direct. Si se requiere asignar una IP fija a la pasarela para esto, no se hará a través del cliente DHCP, puesto que ha sido deshabilitado. Se realizará mediante el fichero de configuración `/etc/network/interfaces`.

NOTA 3: (solo para probar el funcionamiento). Si queremos ejecutar un cliente GPS, deberemos introducir los siguientes comandos, donde “`/dev/ttyUSB0`” es el puerto asignado al receptor GPS.

```
$ sudo gpsd /dev/ttyUSB0 -F /var/run/gpsd.sock  
$ cgps -s
```

### COPIA DE LOS FICHEROS/COMPONENTES DE LA PLATAFORMA MOTAM

Todos los ficheros/componentes de la plataforma MOTAM a ejecutar sobre la plataforma Raspberry Pi se encuentran en el siguiente repositorio Git:

<https://github.com/nicslabdev/MOTAM-Gateway>

Por lo tanto, tan sólo hace falta copiar el contenido de este repositorio al directorio que vamos a crear, y en el cual estarán todos los ficheros relacionados con el proyecto MOTAM “/home/pi/MOTAM”

```
$ git clone https://github.com/nicslabdev/MOTAM-Gateway /home/pi/MOTAM
```

#### PERMISOS NECESARIOS PARA LA EJECUCIÓN

Algunos scripts en los que se apoya el código principal necesitan permisos de ejecución para poder ser ejecutados. Para ello ejecutamos:

```
$ sudo chmod +x /home/pi/MOTAM/util/usbDiscovery  
$ sudo chmod +x /home/pi/MOTAM/util/startGateway.sh  
$ sudo chmod +x /home/pi/MOTAM/util/zenity
```

#### INSTALACIÓN DE LOS MÓDULOS NODE.JS

Los módulos necesarios para el funcionamiento de nuestro código de Node.JS se encuentran especificados en el fichero “*package.json*”. Por tanto, para instalarlos, solo es necesario ejecutar:

```
$ cd /home/pi/MOTAM  
$ npm install
```

#### INSTALACIÓN DE LOS PAQUETES PYTHON

Será necesario instalar los paquetes necesarios para el funcionamiento de los scripts auxiliares de Python. Los instalaremos para el super-usuario (*sudo*), ya que el código Node.JS se ejecutará como super-usuario.

```
$ sudo pip install obd bluepy
```

#### CREACIÓN DE ACCESO DIRECTO EN ESCRITORIO

Se procede a colocar un acceso directo en el escritorio con el objetivo de que no sea necesario el uso de ratón y teclado para ejecutar el programa principal. Una vez colocado, bastará con pulsar dos veces desde la pantalla táctil de la Raspberry Pi para ejecutarlo.

```
$ mv /home/pi/MOTAM/MOTAM /home/pi/Desktop
```

#### INSTALACIÓN DE LA PLATAFORMA HARDWARE ARDUINO/GENUINO 101

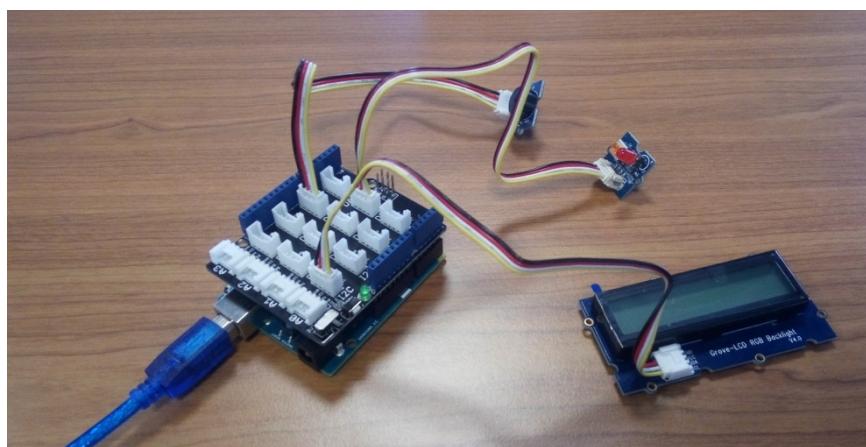
La plataforma hardware Genuino 101 es la base sobre la que se despliegan los actuadores visual, luminoso y sonoro de la plataforma MOTAM. La conexión entre esta plataforma y la pasarela se realiza mediante puerto serie a través de un cable USB, aunque se ha proyectado que en futuras versiones esta conexión sea realizada mediante tecnologías inalámbricas.

Los componentes empleados en esta plataforma son los siguientes:



**ILUSTRACIÓN 3: COMPONENTES DE LA PLATAFORMA HARDWARE ARDUINO/GENUINO 101**

A continuación, se presentan los pasos necesarios para llevar a cabo la instalación, configuración y despliegue de los componentes MOTAM a ejecutar en esta plataforma hardware.



**ILUSTRACIÓN 4: PLATAFORMA HARDWARE ARDUINO/GENUINO 101**

#### CONEXIÓN DE LOS COMPONENTES HARDWARE

La conexión de los distintos actuadores de la plataforma MOTAM acoplados a la placa de desarrollo Arduino/Genuino 101 se ha llevado a cabo a través de la plataforma Grove (de Seeed-Studio), que ofrece un sistema de conexión modular (y listo para usar) con una gran variedad de componentes electrónicos.

La citada interfaz Grove va insertada en los pines de la placa de desarrollo Arduino/Genuino 101.

El esquema de conexión usado para conectar los distintos actuadores a la plataforma hardware Arduino/Genuino 101

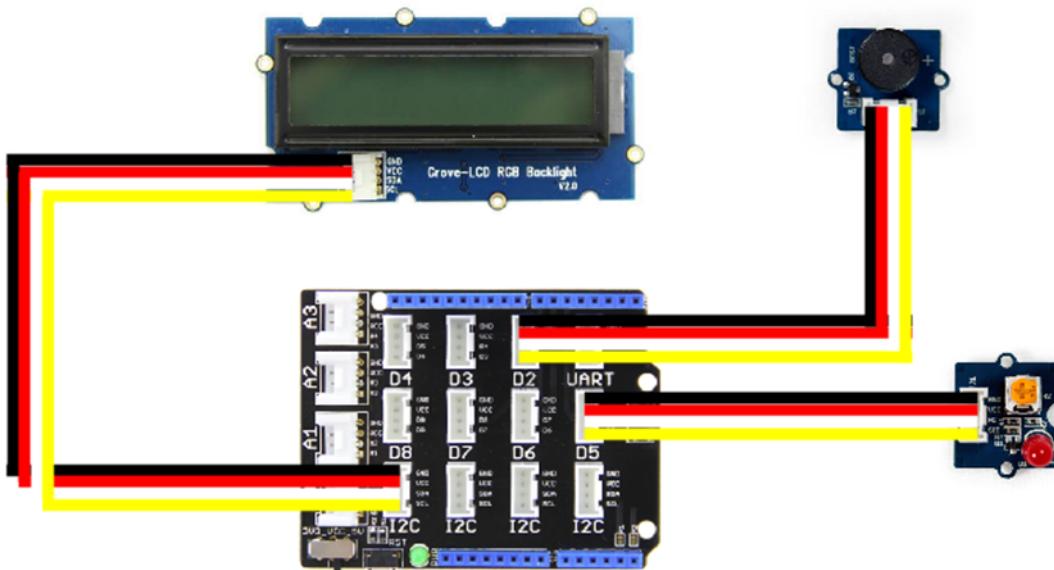


ILUSTRACIÓN 5: ESQUEMA DE CONEXIÓN ENTRE LOS COMPONENTES GROVE Y LA PLATAFORMA ARDUINO/GENUINO 101

#### INSTALACIÓN DEL ENTORNO DE DESARROLLO ARDUINO

Para la programación en Arduino de nuestra plataforma, será necesario instalar el Arduino IDE

1. Descargar e instalar el IDE oficial de Arduino desde la web de Arduino. En este caso hemos empleado la versión 1.8.4. <https://www.arduino.cc/en/main/software>
2. Descargamos los *sketchs* y las librerías necesarias desde el repositorio de MOTAM-Arduino\_RFduino:  
[https://github.com/nicslabdev/MOTAM-Arduino\\_RFduino](https://github.com/nicslabdev/MOTAM-Arduino_RFduino)
3. Instalamos las librerías descargadas en nuestro entorno de desarrollo Arduino. Para ello, copiamos el contenido de la carpeta “*libraries*” en la carpeta de las librerías instaladas de nuestro Arduino IDE, normalmente situada en “C:\Users\usuario\Documents\Arduino\libraries”
4. Instalamos los drivers para Arduino/Genuino 101. Para ello, arrancamos el Arduino IDE y nos vamos a “Herramientas – Placa – Gestor de tarjetas...”. En el buscador ponemos “Intel Curie Boards” y seleccionamos e instalamos el paquete que incluye las tarjetas Arduino/Genuino 101.

#### PROGRAMACIÓN DEL MICROCONTROLADOR

1. Conectamos por USB el Arduino/Genuino 101 al PC.
2. Abrimos el IDE de Arduino haciendo doble click en el sketch que queremos instalar. En nuestro caso “*Display\_Light\_Sound\_Actuators\_Genuino101.ino*”
3. Configuramos el IDE. Es posible que esto tengamos que hacerlo cada vez que conectemos la placa por USB al PC:
  - a. Nos vamos a Herramientas/Placa y seleccionamos “*Arduino/Genuino 101*”.

- b. Nos vamos a Herramientas/Puerto y seleccionamos aquel en el que se haya detectado "Arduino/Genuino 101".
- 4. Pulsamos sobre el botón "Subir" para iniciar la compilación y subida del software al microcontrolador de la placa de desarrollo.

#### PONER EN FUNCIONAMIENTO LOS ACTUADORES

Para poner en funcionamiento esta plataforma hardware, basta con conectarla mediante un cable USB a un puerto USB libre de la pasarela. Cuando se ejecute el programa principal de la pasarela, se establecerá automáticamente una conexión mediante puerto serie.

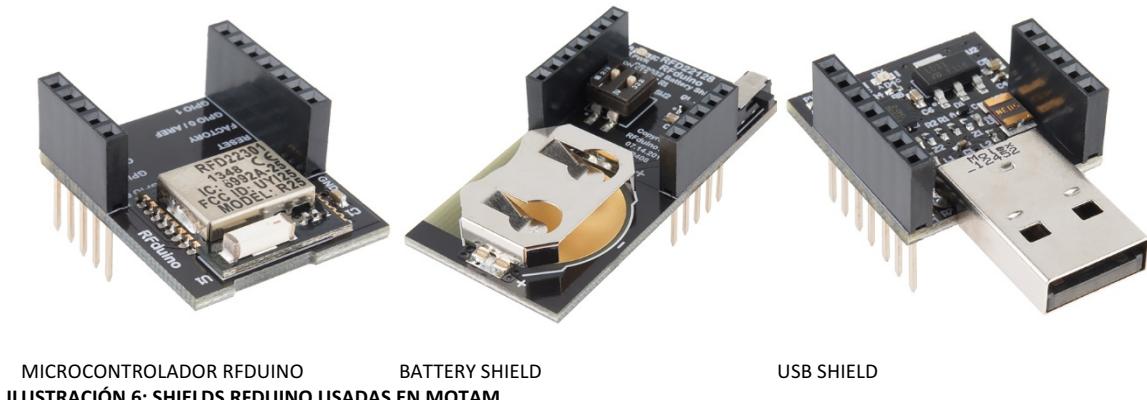
### INSTALACIÓN DE LOS SENSORES RFDUINO

La plataforma hardware RFduino es la base sobre la que se despliegan el resto de sensores y balizas de la plataforma MOTAM.

En estos momentos se encuentran desarrollados 5 tipos de sensores diferentes: sensor de climatología, sensor para silla de bebés, sensor de calidad del aire, sensor de luces de semáforo y sensor de presencia de bicicleta.

Todos los sensores y balizas están basados en la plataforma de desarrollo RFduino. Este tipo de placas se basan en "shields", de forma que se pueden ir agregando unas shields encima de otras para añadir funcionalidad.

Existen tres shields básicas usadas en MOTAM:



MICROCONTROLADOR RFDUINO      BATTERY SHIELD      USB SHIELD  
ILUSTRACIÓN 6: SHIELDS RFDUINO USADAS EN MOTAM

### INSTALACIÓN Y CONFIGURACIÓN DEL ENTORNO DE DESARROLLO PARA RFDUINO

Para poder llevar a cabo la programación del microcontrolador RFduino, es necesario configurar previamente el equipo mediante estos pasos:

1. Descargar e instalar los controladores correspondientes para el sistema operativo del PC desde el que se va a realizar la programación: <http://www.ftdichip.com/Drivers/VCP.htm>
2. Abrir el IDE de Arduino. Ir a Archivo, Preferencias. Añadir la siguiente dirección en "Gestor de URLs adicionales de tarjetas": [http://rfduino.com/package\\_rfduino\\_index.json](http://rfduino.com/package_rfduino_index.json)

3. Ir a Herramientas, Placa, Gestor de tarjetas... Instalar el paquete RFduino Boards by RFduino.

#### PROGRAMACIÓN DEL MICROCONTROLADOR DE LOS SENSORES RFDUINO

Para realizar la programación de los RFduino deberemos acoplar el USB Shield a el shield del microcontrolador RFduino. El conector USB del USB Shield se deberá conectar a un puerto USB del equipo desde el que se vaya a realizar la programación del microcontrolador.

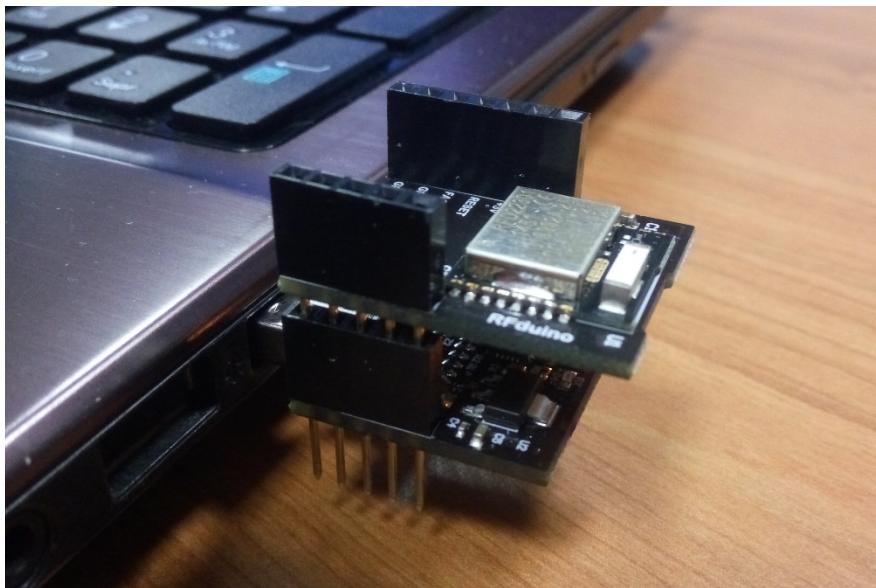


ILUSTRACIÓN 7: SHIELD USB CONECTADA AL MICROCONTROLADOR RFDUINO PARA PROGRAMAR

Deberemos seguir los siguientes pasos para realizar la programación del microcontrolador:

1. Conectamos por USB el RFduino al PC.
2. Abrimos el IDE de Arduino haciendo doble click en el sketch que queremos instalar. Hay un sketch por cada tipo de baliza-sensor.
3. Configuramos el IDE. Es posible que esto tengamos que hacerlo cada vez que conectemos la placa por USB al PC:
  - a. Nos vamos a Herramientas/Placa y seleccionamos "RFduino".
  - b. Nos vamos a Herramientas/Puerto y seleccionamos aquel en el que se haya detectado.
4. Pulsamos sobre el botón "Subir" para iniciar la compilación y subida del software al microcontrolador de la placa de desarrollo.

#### SENSOR DE CLIMATOLOGÍA

El sensor de climatología se encarga de detectar la temperatura, humedad y la presencia de agua. Está compuesto por una placa de desarrollo RFduino, un sensor de temperatura y humedad, un sensor de agua y una placa de conexión elaborada por la UMA.

El sensor de agua se conecta al pin 5 de RFduino, el sensor de temperatura y humedad se conecta al pin 6 de RFduino.



ILUSTRACIÓN 8: COMPONENTES DEL SENSOR DE CLIMATOLOGÍA



ILUSTRACIÓN 9: SENSOR DE CLIMATOLOGÍA MONTADO EN CAJA DE PROTECCIÓN

### SENSOR SILLA DE BEBÉS

El sensor de silla de bebés es capaz de detectar si un niño está sentado en la silla y si el cinturón se encuentra abrochado. Está compuesto por una placa de desarrollo RFduino, un interruptor magnético, un sensor de presión y una placa de conexión elaborada por la UMA. La citada placa de conexión consta de un potenciómetro que sirve para ajustar la sensibilidad del sensor de presión.

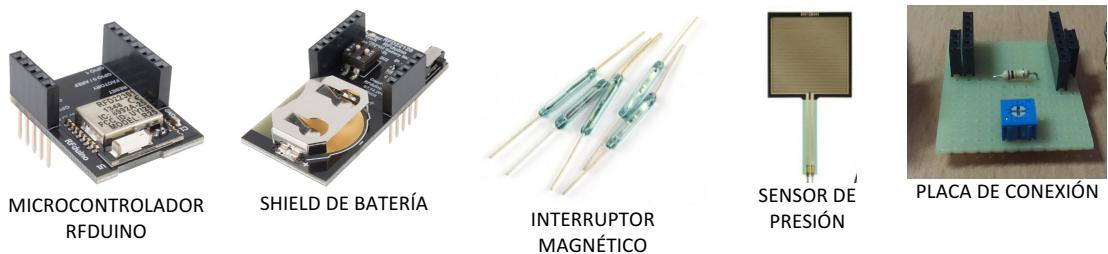


ILUSTRACIÓN 10: COMPONENTES DEL SENSOR SILLA DE BEBÉS

El sensor de presión se conecta al pin 5 de RFduino, el sensor magnético se conecta al pin 6 de RFduino. Las conexiones tanto el sensor de presión como el magnético con la caja de protección del sensor se realiza mediante un conector RJ11 con el fin de facilitar la instalación.



**ILUSTRACIÓN 11: SENSOR DE SILLA DE BEBÉS MONTADO EN CAJA DE PROTECCIÓN**

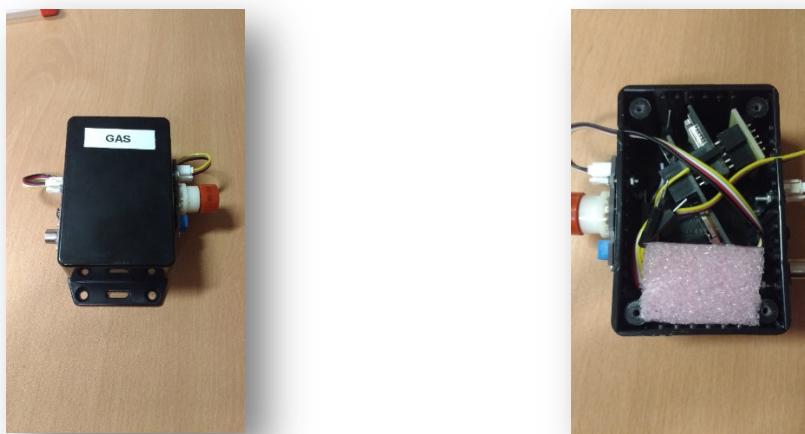
#### SENSOR CALIDAD DEL AIRE

El sensor de climatología detecta la calidad del aire y la presencia de gases. Está compuesto por una placa de desarrollo RFduino, un sensor de calidad del aire, un sensor de gases y una placa de conexión elaborada por la UMA.



**ILUSTRACIÓN 12: COMPONENTES DEL SENSOR DE CALIDAD DEL AIRE**

El sensor de calidad del aire se conecta al pin 5 de RFduino y el sensor de gas se conecta al pin 6 de RFduino.



**ILUSTRACIÓN 13: SENSOR DE CALIDAD DEL AIRE MONTADO EN SU CAJA DE PROTECCIÓN**

#### SENSOR DE LUCES DE SEMÁFORO

El sensor de luces de semáforo detecta el estado actual de un semáforo y está compuesto por una placa de desarrollo RFduino, tres fotorresistencias y una placa de conexión elaborada por la UMA. La placa de conexión cuenta a su vez con tres potenciómetros para ajustar la sensibilidad de las fotorresistencias.

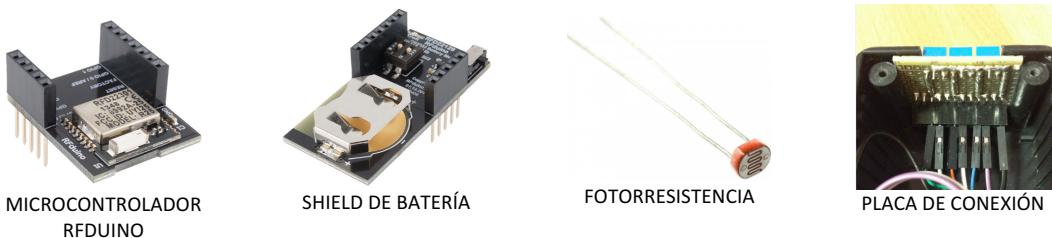


ILUSTRACIÓN 14: COMPONENTES DEL SENSOR DE LUCES DE SEMÁFORO

Cada una de las fotorresistencias se encuentra conectada a los pines 4, 5 y 6 de RFduino.

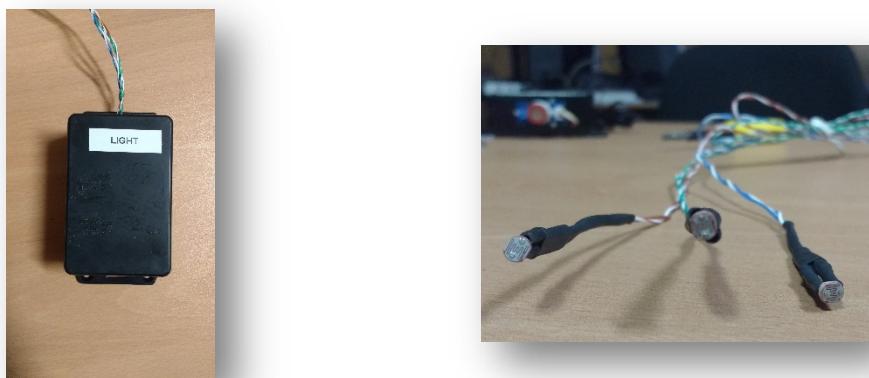


ILUSTRACIÓN 15: SENSOR DE LUCES DE SEMÁFORO MONTADO EN SU CAJA DE PROTECCIÓN

#### SENSOR DE PRESENCIA DE BICICLETA

El sensor de presencia de bicicleta es capaz de detectar la presencia de una bicicleta y su estado: bicicleta en marcha o bicicleta accidentada. Está compuesto por una placa de desarrollo RFduino, una placa de conexión elaborada por la UMA, dos sensores de inclinación y un sensor magnético. El sensor magnético irá colocado en la horquilla, de forma que sea activado mediante el paso de rueda con un imán situado en un radio.



ILUSTRACIÓN 16: COMPONENTES DEL SENSOR DE PRESENCIA DE BICICLETA

Los sensores de inclinación se conectan a los pines 4 y 5. El sensor magnético se conecta al pin 6.

La conexión del sensor de paso de rueda con la caja de protección del sensor se realiza mediante un conector RJ11 con el fin de facilitar la instalación. Los sensores de inclinación se encuentran embutidos dentro de la caja de protección.



ILUSTRACIÓN 17: SENSOR DE PRESENCIA DE BICICLETA MONTADO EN SU CAJA DE PROTECCIÓN

### MODIFICACIÓN DE SENSORES RFDUINO

Es posible modificar el comportamiento de los sensores y la información que transmiten. Esto puede resultar útil para, por ejemplo, definir la ubicación GPS exacta y/o el sentido de circulación para el que debe funcionar el sensor que se va a desplegar.

Una vez configurado el PC para poder compilar y subir los sketches a los RFduino, solo queda modificar el código de estos sketches para poner la ubicación y el rumbo que se quiera.

Para ello, modificamos el sketch de la siguiente forma:

```
uint8_t advdata[] =
{
    0x11, // Length
    0xFF, // Manufacturer data type

    // DEPhisit BEacon identifier
    0xBE,
    0xDE,

    // Latitude of the TrafficLightDetectorBeacon, in LE float format (28.481768)
    0xA9,
    0xDA,
    0xE3,
    0x41,

    // Longitude of the TrafficLightDetectorBeacon, in LE float format (-16.324577)
    0xBC,
    0x98,
    0x82,
    0xC1,

    0x01, // Type of DEPhisit BEacon (0x01 -> Traffic Sign Beacon, 0x02 -> Weather Beacon, 0x03 -> Bicycle Beacon)

    // DEPhisit BEacon data (different structure for each beacon type, and dynamic values)
    STOP_SIGN, // Default state of traffic light -> Unknown

    0x23, // From direction that applies (35)
    0x00,

    0x18, // To direction that applies (280)
    0x01,
};
```

ILUSTRACIÓN 18: ARRAY DONDE SE SITÚA LA INFORMACIÓN QUE TRANSMITE EL SENSOR DE SEÑAL STOP

En el array advData se sitúa la trama del paquete advertising que transmiten periódicamente los RFduino.

En primer lugar tenemos que obtener las coordenadas del lugar donde vamos a situar el sensor. Para ello, se recomienda el uso de Google Maps. Haciendo click sobre el lugar del mapa en el que se va a desplegar el sensor, se mostrarán las coordenadas.

En Bluetooth Low Energy, los bytes de información se transmiten en formato Little-endian. Además, en la trama BLE de MOTAM, las coordenadas se transmiten pasando a binario dos números en formato float (latitud y longitud). Para hacer más fácil el manejo de estos valores, en el sketch se encuentran expresados en formato hexadecimal. Para facilitar la tarea de incluir en la trama estos dos valores (latitud y longitud), se emplea la siguiente herramienta online.

<https://gregstoll.dyndns.org/~gregstoll/floattohex/>

En ella, hay que hacer click sobre “Swap Endiannes” para activar la representación de los valores en Little-endian. Como podemos ver en la ilustración 18, la precisión de los valores viene determinada por que la trama reserva 4 bytes para la latitud y 4 bytes para la longitud.

Para poner el rumbo o sentido de la marcha para la que está desplegado el sensor (campo “from direction that applies” en el sketch) no es válido el uso de la herramienta anterior, pues este nuevo valor se expresa como un entero. En su lugar, se recomienda el uso de la siguiente herramienta:

<http://www.binaryhexconverter.com/decimal-to-hex-converter>

No olvidar que este número también se expresa en Little Endian, por lo que los bytes más significativos van después.

**Ejemplo:** 35 (decimal) = 0x0023 (hexadecimal) = 0x2300 (hexadecimal little endian)

280 (decimal) = 0x0118 (hexadecimal) = 0x1801 (hexadecimal little endian)

## INSTALACIÓN DEL MÓDULO NB-IoT

El proyecto MOTAM incluye soporte para dotar a la plataforma de conectividad Narrow Band Internet of Things (NB-IoT). Esta conectividad se realiza mediante una shield compatible con el factor de forma habitual en Arduino.

Hasta el momento, la red NB-IoT de España solo ha sido desplegada por el operador Vodafone. Para el acceso a esta red es necesario tener una tarjeta SIM compatible con NB-IoT de este operador.

La shield utilizada en el proyecto MOTAM es la SODAQ NB-IoT Shield, la cual incorpora dos módulos y varios sensores, los cuales se describen a continuación:

- Módulo NB-IoT: Se trata de un módulo del fabricante ublox, modelo SARA-N211. Proporciona conectividad NB-IoT en las bandas 8 y 20. Cabe destacar que la red NB-IoT de Vodafone España utiliza la banda 20.
- Módulo GPS: También fabricado por ublox, modelo SAM-M8Q. Se trata de un receptor GPS.
- Sensor de temperatura y humedad: La shield incorpora el sensor HTS221.
- Sensor de presión atmosférica: Modelo LPS22HB.

- Sensor de aceleración y magnetómetro: Modelo LSM303AGR.



**ILUSTRACIÓN 19: SHIELD NB-IOT MONTADA SOBRE PLACA ARDUINO LEONARDO CON UNA TARJETA SIM VODAFONE NB-IOT**

#### PROGRAMACIÓN DE LA PLACA ARDUINO CON LA SHIELD NB-IoT

Es conveniente conocer un aspecto de esta shield. Es frecuente que la comunicación entre microcontroladores se haga a través del puerto serie. Y es así como ocurre entre el módulo NB-IoT de esta shield y Arduino. Es por esto que, dependiendo del Arduino utilizado, puede existir un conflicto entre el puerto USB utilizado para conectar el PC al Arduino (y que usa este puerto serie) y el puerto serie con el que se comunican Arduino y NB-IoT. Por tanto:

Con modelos de Arduino cuyo microcontrolador tenga más de un UART y pueda elegirse usar cada UART en un puerto serie distinto no existe problema. Simplemente a nivel de software (sketch Arduino) deberá definirse un puerto serie para la conexión con el módulo NB-IoT y otro para la conexión USB con el PC. Esta configuración hace que sea posible el debug mediante un monitor de puerto serie en el PC, como normalmente se hace con Arduino. Algunos modelos Arduino que cumplen estos son: **Arduino Leonardo, Arduino 101, Arduino Zero, Arduino M0, SODAQ ExpLoRer.**

Con modelos de Arduino cuyo microcontrolador solo tenga un UART o solo un puerto Serie no es posible la conexión mediante USB al PC mientras se encuentre montada la shield NB-IoT. Los sketches de Arduino deberán ser cargados en la placa con la shield desconectada, pues de lo contrario la comunicación entre PC y Arduino no será correcta y el IDE de Arduino no lo cargaría, notificándonos de un error de conexión. Tras cargar el sketch ya podremos conectar la shield. El debug mediante un monitor de puerto serie en el PC no es posible de forma nativa. Algunos modelos de placa Arduino con esta limitación son: **Arduino UNO, Arduino MEGA o Intel Galileo.**

Una vez conocido esto, los pasos a seguir para cargar un sketch es el siguiente:

1. Conectamos por USB la placa Arduino al PC.
2. Abrimos el IDE de Arduino haciendo doble click en el sketch que queremos instalar.
3. Configuramos el IDE. Es posible que esto tengamos que hacerlo cada vez que conectemos la placa por USB al PC:

- a. Nos vamos a Herramientas/Placa y seleccionamos el modelo de placa que estemos usando.
  - b. Nos vamos a Herramientas/Puerto y seleccionamos aquel en el que se haya detectado la placa.
4. Pulsamos sobre el botón “Subir” para iniciar la compilación y subida del software al microcontrolador de la placa de desarrollo.

En el repositorio de GitHub de los sensores y balizas del proyecto MOTAM se pueden encontrar los sketches para el uso de la shield NB-IoT: [https://github.com/nicslabdev/MOTAM-Arduino\\_RFduino](https://github.com/nicslabdev/MOTAM-Arduino_RFduino)

#### ATCOMMANDS:

Este sketch es solo compatible con las placas con más de un puerto serie/UART (Arduino Leonardo). Establece un bypass entre el puerto serie al que se conecta el PC –a través del puerto USB- y el puerto serie al que se conecta el NB-IoT. Esto permite comunicarnos directamente con el módem NB-IoT mediante comandos AT.

Para realizar esta conexión es necesario abrir el monitor serie del IDE de Arduino y seleccionar un baudrate de 9600 y configurar la finalización de cada línea enviada con “Retorno de carro”.

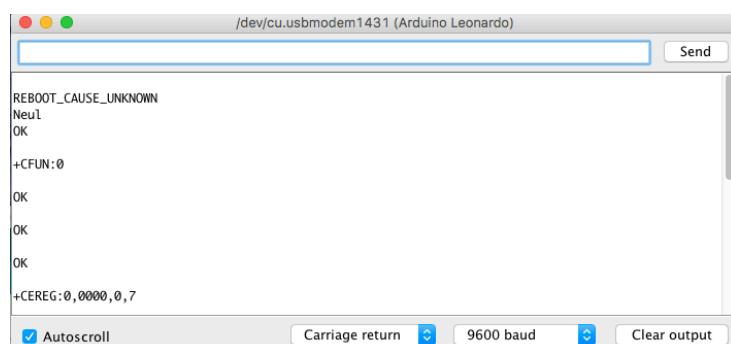


ILUSTRACIÓN 20: CONFIGURACIÓN DEL MONITOR SERIE PARA LA CONEXIÓN MEDIANTE COMANDOS AT

Como ejemplo tenemos la siguiente secuencia de comandos AT:

AT+NRB	→ Reinicia el módulo
AT+CEREG=2	→ Se habilita el registro a la red
AT+CSCON=1	→ Se habilita el CSCON URC
AT+CFUN=1	→ Se establece el modo de máxima funcionalidad
AT+CIMI	→ Devuelve el IMSI de la tarjeta SIM
AT+CGDCONT=0,"IP",""	→ Se definen los parámetros de conexión
AT+COPS=1,2,"21401"	→ Se selecciona el operador Vodafone. Devuelve una serie de parámetros que indican el estado actual de la conexión. Tras un periodo de espera debe devolver algo similar a "+CEREG:5,102F,46CEA3D,7". Si aparece 5 como la primera cifra indica que el registro en la red se ha realizado correctamente. Puede devolver otros parámetros ante de que el registro se complete.
AT+CGDCONT?	→ Devuelve los parámetros de la conexión una vez que se ha realizado el registro.

AT+CSQ	→ Devuelve la calidad de la señal. 99,99 si la conexión no se ha realizado correctamente.
AT+CGATT?	→ Devuelve 1 si el dispositivo se encuentra registrado en la red.
AT+CGPADDR	→ Devuelve la IP asignada por la red al dispositivo.
AT+NUESTATS	→ Devuelve información variada sobre la red y la conexión.