

Análisis del módulo de comunicaciones FiPy

Nicolás Quiroz-Hernández, José J. Medina-García,
Selene E. Maya-Rueda, Aideé Montiel-Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Electrónica,
México

{nicolas.quirozh, selene.maya, aidee.montiel}@correo.buap.mx,
jose.medinag@alumno.buap.mx

Resumen. El presente trabajo busca facilitar el uso e implementación del módulo de comunicaciones FiPy que cuenta con 5 tipos de tecnologías de comunicación diferentes, diseñado principalmente para IoT. El internet de las cosas (IoT) es una red colectiva de dispositivos conectados y a las tecnologías de comunicación a estos dispositivos entre sí y la nube. Las redes LPWAN permiten transmitir datos mediante comunicación inalámbrica entre dispositivos separados por kilómetros. Se realizó un análisis de algunos dispositivos que integran estas diferentes tecnologías de comunicación. Este dispositivo está dirigido a proyectos en los que la facilidad de intercambiar sistemas de comunicación se vital.

Palabras clave: Pycom, Redes IoT, LoRaWAN, Sigfox, LTE-M, WiFi, Bluetooth.

Analysis of the FiPy Communications Module

Abstract. Deaf community uses sign language as its main form of communication; however, most of the speaking community does not know how to understand that language, therefore the sign language recognition through technological developments has been an area of great interest for years. In this work, a proposal for this problem is presented, where regions of interest detection, manual and non-manual features extraction are carried out and for the recognition some BiLSTM networks with different architectures are used. The results obtained are an 73.99% accuracy, which are promising for the upcoming experiments. Finally, various actions are presented with the aim of improving the results as future work.

Keywords: Pycom, IoT networks, LoRaWAN, Sigfox, LTE-M, WiFi, Bluetooth.

1. Introducción

El IoT o internet de las cosas, se refiere a una red colectiva de dispositivos conectados y a las tecnologías que comunican a estos dispositivos entre sí y la nube [1]. En los últimos años, el internet de las cosas se ha convertido en una de las tecnologías

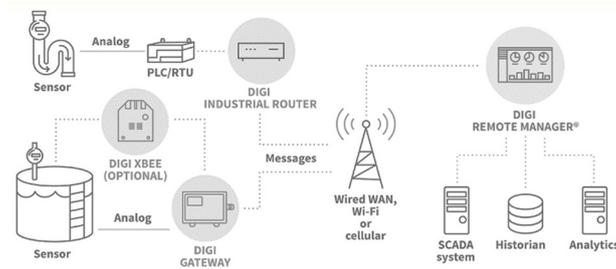


Fig. 1. Etapas de un sistema IoT [4].

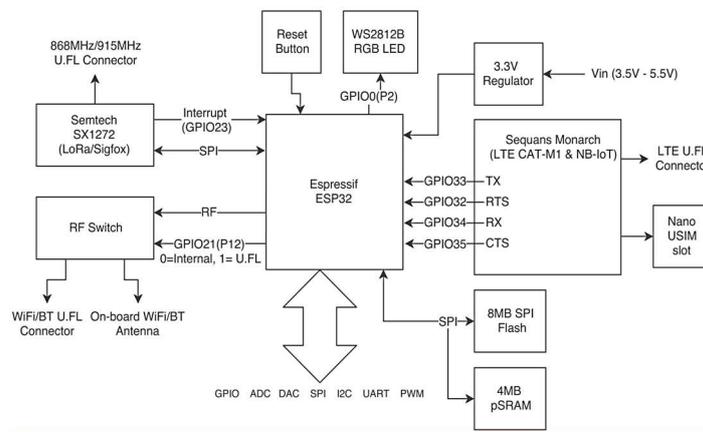


Fig. 2. Diagrama de bloques del módulo FiPy.

```

from network import WLAN
wlan = WLAN()

#Configuración de SSID y contraseña
wlan.init(ssid="FiPy", auth=(WLAN.WPA2, "Pycom_FCE"))
#id=1 indica que el modulo trabaja en modo WiFi AP
print(wlan.ifconfig(id=1))
    
```

Fig. 3. Código para crear un punto de acceso WiFi con SSID y contraseña propios.

más importantes en la actualidad, ya que podemos conectar objetos cotidianos como electrodomésticos, automóviles, termostatos, monitores para bebés, cámaras, etcétera, a internet a través de dispositivos integrados [2].

Internet se está desarrollando rápidamente y como resultado, Internet de las cosas se ha convertido en una realidad. La gloria de esta tecnología reside principalmente en todas las aplicaciones y oportunidades que ofrece para mejorar el día a día de las personas y los entornos empresariales. En la figura 1 se muestran los 5 componentes básicos de un sistema IoT, los cuales son sensores y actuadores, controladores, sistemas de comunicación, la nube y el monitoreo.

Por otro lado, para buscar dispositivos IoT se debe tener en consideración aspectos como bajo consumo energético y que sean pequeños, de ahí que los SoCs sean una parte importante de estos dispositivos [3].

```

from network import WLAN
import machine
wlan = WLAN(mode = WLAN.STA)

wlan.connect(ssid='Xiaomi', auth=(WLAN.WPA2, '123456'))
while not wlan.isdisconnected():
    machine.idle()
print("WiFi CConexión exitosa")
print(wlan.ifconfig())

```

Fig. 4. Código de conexión a red WiFi existente.

```

from network import Bluetooth
from machine import Timer

battery = 100
update = False
def conn_cb(chr):
    events = chr.events()
    if events & Bluetooth.CLIENT_CONNECTED:
        print('Cliente conectado')
    elif events & Bluetooth.CLIENT_DISCONNECTED:
        print('Cleinte desconectado')
        update = False
def chr1_handler(chr, data):
    global battery
    global update
    events = chr.events()
    print("eventos: ", events)
    if events & (Bluetooth.CHAR_READ_EVENT | Bluetooth.CHAR_SUBSCRIBE_EVENT):
        chr.value(battery)
        print("Transmitiendo :", battery)
        if (events & Bluetooth.CHAR_SUBSCRIBE_EVENT):
            update = True
bluetooth = Bluetooth()
bluetooth.set_advertisement(name='FiPy', manufacturer_data="Pycom", service_uuid=0xec00)
bluetooth.callback(trigger=Bluetooth.CLIENT_CONNECTED | Bluetooth.CLIENT_DISCONNECTED, handler=conn_cb)
bluetooth.advertise(True)
srv1 = bluetooth.service(uuid=0xec00, isprimary=True, nbr_chars=1)
chr1 = srv1.characteristic(uuid=0xec0e, value='leer_desde_aqui')
chr1.callback(trigger=(Bluetooth.CHAR_READ_EVENT | Bluetooth.CHAR_SUBSCRIBE_EVENT), handler=chr1_handler)
print('Iniciar servicio BLE')
def update_handler(update_alarm):
    global battery
    global update
    battery-=1
    if battery == 1:
        battery = 100
    if update:
        chr1.value(str(battery))
update_alarm = Timer.Alarm(update_handler, 1, periodic=True)

```

Fig. 5. Código para servidor BLE, que envía niveles de batería cada segundo.

Las redes LPWAN (Redes de Área Amplia de Baja Potencia) permiten transmitir datos mediante comunicación inalámbrica entre un dispositivo y una estación base o gateway separados por kilómetros con un bajo consumo de energía [5], existen 4 tipos de redes líderes con esta tecnología LoRaWAN, Sigfox, LET CAT-M1 y NB-IoT; LoRaWAN que es gratuito y es apoyado por la comunidad, lo que hace que esta red solo pueda implementarse si algún usuario a instalado un gateway de forma fija y con acceso libre, los dispositivos de comunicación de esta red son fáciles de obtener y de bajo costo rondando entre 10 y 25 dólares [6].

La red de Sigfox no es gratuita tiene un costo de 16 dólares anuales, pero no hace falta implementar un gateway ya que ellos proveen de esta infraestructura y existe una amplia cobertura de esta red, los dispositivos para poder acceder a la red de Sigfox son

```
from network import Bluetooth
from machine import Timer

def char_notify_callback(char, arg):
    char_value = (char.value())
    print("Nuevo valor: {}".format(char_value))
bt = Bluetooth()
print("Iniciando escaneo de servicios BLE")
bt.start_scan(-1)
adv = None
while(True):
    adv = bt.get_adv()
    if adv:
        try:
            if bt.resolve_adv_data(adv.data, Bluetooth.ADV_NAME_CMPL)="FiPy":
                conn = bt.connect(adv.mac)
                print("Connected to FiPy")
                try:
                    services = conn.services()
                    for service in services:
                        chars = service.characteristics()
                        for char in chars:
                            c_uuid = char.uuid()
                            if c_uuid == 0xec0e:
                                if (char.properties() & Bluetooth.PROP_NOTIFY):
                                    char.callback(trigger=Bluetooth.CHAR_NOTIFY_EVENT, handler=char_notify_callback)
                                    print(c_uuid)
                                    break
                except:
                    continue
            except:
                continue
    bt.stop_scan()
    bt.disconnect_client()
```

Fig. 6. Código para cliente bluetooth.

de costos más elevados entre 40 y 200 dólares, sin embargo, la mayoría de estos dispositivos cuentan con una suscripción gratuita de un año a esta red [7].

Las redes LTE CAT-M1 y NB-IoT son redes dedicadas para dispositivos que requieren datos durante largos periodos de tiempo en lugares de difícil acceso, su costo es más elevado ya que funciona de la misma forma que una red LTE con una cuota para mantener la red activa [8].

Pycom es una empresa de origen europeo que crea dispositivos que contienen controladores y sistemas de comunicación, proporciona un entorno completo de dispositivo a nube y herramientas de administración segura y confiable para implementar IoT [9].

FiPy es un módulo programable en lenguaje MicroPython que integra WiFi, Bluetooth de baja energía, LoRa, Sigfox y LTE-M dual, que es de gran apoyo cuando se quieren integrar proyectos que tienen la necesidad de ser versátiles en la comunicación de datos a una base de datos [9].

2. Desarrollo

Para programar el módulo FiPy se utiliza Visual Studio Code, este puede tener la función de ser controlador y sistema de comunicación al mismo tiempo, ya que integra un SoC de Espressif ESP32, en la figura 2 se muestra un diagrama de bloques de lo que integra el módulo FiPy, este cuenta con dispositivos de comunicación LPWAN, BLE y WiFi, tiene conectores para antenas de los diferentes sistemas de comunicación y adicionalmente cuenta con pines dedicados para el usuario.

```

from network import LoRa
import socket
import ubinascii
import struct
#Inicializamos el modo LoRaWAN para México 915 MHz
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.US915)
#Parametros de autenticación ABPn
dev_addr = struct.unpack(">I", ubinascii.unhexlify('00000005'))[0]
nwk_swkey = ubinascii.unhexlify('2B7E151628AED2A6ABF7158809CF4F3C')
app_swkey = ubinascii.unhexlify('2B7E151628AED2A6ABF7158809CF4F3C')
for i in range(0,8):
    lora.remove_channel(i)
for i in range(16,65):
    lora.remove_channel(i)
for i in range(66,72):
    lora.remove_channel(i)
lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
s.setblocking(True)
#Algunos datos a enviar
s.send(bytes([0x01, 0x02, 0x03]))
s.setblocking(False)
#Espera de datos Downlink
data = s.recv(64)
print(data)

```

Fig. 7. Programa de comunicación LoRaWAN en modo ABP.

2.1. WiFi

Los módulos de desarrollo de Pycom tienen una antena integrada, por lo que no es necesario utilizar una antena externa, el módulo puede trabajar de dos formas como punto de acceso donde el dispositivo crea una red local o como cliente donde el dispositivo se conecta a una red existente. El consumo energético de WiFi en modo punto de acceso es de 126 mA mientras que en modo cliente es de 137 mA, estas mediciones son proporcionadas por el fabricante.

2.1.1. Punto de acceso WiFi

Para configurar un punto de acceso básico se debe utilizar el código mostrado en la figura 3, este establecerá de manera predeterminada la dirección IP 192.168.4.1.

2.1.2. Cliente WiFi

Para conectar el módulo a una red existente, se debe configurar el modo WLAN.STA, en la figura 4 se muestra el código de programa básico para conectarse a una red existente.

2.2. Bluetooth

El módulo FiPy cuenta con Bluetooth 4.2 de baja energía y el consumo de este durante su uso es de 121 mA, esta medición es proporcionada por el fabricante. Se pueden escanear los dispositivos BLE cercanos y funcionar como cliente o servidor bluetooth.

```
from network import LoRa
import socket
import time
import ubinascii
#Inicializar modo LoRaWAN para México 915 MHz
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.US915)
#Crear parametros de autenticación OTAA
#Estos datos deben cambiar de acuerdo al dispositivo FiPy
app_eui = ubinascii.unhexlify('ADA4DAE3AC12676B')
app_key = ubinascii.unhexlify('11B0282A189B75B0B4D2D8C7FA38548B')
for i in range(0,8):
    lora.remove_channel(i)
for i in range(16,65):
    lora.remove_channel(i)
for i in range(66,72):
    lora.remove_channel(i)
lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0)
while not lora.has_joined():
    time.sleep(2.5)
    print('No conectado...')
print('Conectado')
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)
s.setblocking(True)
#Envio de datos
s.send(bytes([0x01, 0x02, 0x03]))
s.setblocking(False)
#Espera de datos Downlink
data = s.recv(64)
print(data)
```

Fig. 8. Programa de comunicación LoRaWAN en modo OTAA.

```
from network import Sigfox
import socket
#Iniciar Sigfox para México zona RCZ2
sigfox = Sigfox(mode=Sigfox.SIGFOX, rcz=Sigfox.RCZ2)
s = socket.socket(socket.AF_SIGFOX, socket.SOCK_RAW)
s.setblocking(True)
s.setsockopt(socket.SOL_SIGFOX, socket.SO_RX, False)
#Envio de datos
s.send(bytes([1, 2, 3, 4]))
#Espera de datod Downlink
r = s.recv(32)
print(ubinascii.hexlify(r))
```

Fig. 9. Programa para enviar datos y esperar mensaje de bajada de Sigfox.

2.2.1. Servidor BLE

En la figura 5 se muestra el código para poner el módulo como servidor BLE, este programa espera la conexión de un cliente BLE para que pueda enviar el nivel de batería cada segundo disminuyendo constantemente.

Tabla 1. Comparación entre dispositivos que integran diferentes sistemas de comunicación.

Tecnología de comunicación	Dispositivo				
	FiPy	TTGO LoRa32	SIM7600	MKRFOX1200	SFM11R2D
Sigfox	x			x	x
LoRaWAN	x	x			
LTE			x		
LTE-M	x				
BLE	x	x			
WiFi	x	x			
Procesador	ESP32	ESP32		SAMD21 Cortex-M0	
Costo (MXN)	\$1,985	\$1500	\$1900	\$1300	\$890

2.2.2. Cliente BLE

En la figura 6 se muestra un ejemplo de cliente bluetooth, que busca un servidor bluetooth llamado FiPy y se conecta a este, para obtener los datos del ejemplo anterior e imprimirlos en la consola de VS Code.

2.3. LoRa

Para poder conectarse a la red LoRaWAN es necesario contar con una puerta de enlace cercana a el nodo final en este caso al módulo FiPy, la gran ventaja de esta red es la cantidad de datos por mensaje que se pueden enviar.

Para utilizar el protocolo LoRa es necesario conectar la antena adecuada en el conector Ipx Ufl, existen dos formas básicas de acceder a la red de LoRaWAN ABP y OTAA.

En el modo ABP las claves de cifrado se configuran manualmente en el dispositivo y puede enviar tramas sin necesidad de un intercambio de claves, en la figura 7 se muestra un ejemplo de programa para poder realizar el envío de datos utilizando este método.

En el modo OTAA el módulo envía una solicitud de unión a la puerta de enlace LoRaWAN utilizando `app_eui` y `app_key`, si estas claves son correctas la puerta de enlace aceptara la unión, en la figura 8 se muestra un ejemplo de implementación de este modo.

2.4. Sigfox

La ventaja de la red Sigfox es su cobertura ya que no es necesario implementar una puerta de enlace, además de que el dispositivo FiPy cuenta con 1 año de suscripción a esta red, la cantidad máxima de datos que pueden ser enviados son 12 bytes por mensaje con un máximo de 140 mensajes cada 24 horas.

El módulo se puede configurar para funcionar en diferentes países en función de las zonas RCZ específicas, en la figura 9 se muestra un ejemplo de programa para poder enviar y esperar datos de bajada a través de Sigfox.

Tabla 2. Comparación entre dispositivos que integran el sistema de comunicación Sigfox.

Especificaciones	Dispositivo		
	FiPy	MKRFOX1200	SFM11R2D
Máximo poder Tx	+20dBm	+14.5dBm	+22.5dBm
Rango al nodo	>50 Km	>20 Km	>50 Km
Consumo	192 mA	130mA	170mA
Frecuencia	RCZ1-4	RCZ1	RCZ1-2

Tabla 3. Comparación entre dispositivos que integran el sistema de comunicación LoRaWAN.

Especificaciones	Dispositivo	
	FiPy	TTGO LoRa32
Máximo poder Tx	57 dBm	20 dBm
Rango al nodo	>40 Km	<30 Km
Consumo	156 mA	130 mA
Frecuencia	860-1020 MHz	868-915 MHz

Tabla 4. Comparación entre dispositivos que integran el sistema de comunicación LTE y LTE- M.

Especificaciones	Dispositivo	
	FiPy	SIM7600
Velocidad de datos	40-375 kbps	50 Mbps
Versión de LTE	CAT-M1, NB-IoT	CAT4
Consumo	420mA	530 mA
Frecuencia	699-2690 MHz	824-1910 MHz

Tabla 5. Comparación entre dispositivos que integran el sistema de comunicación BLE.

Especificaciones	Dispositivo	
	FiPy	TTGO LoRa32
Versión BLE	4.2	4.2
Máximo poder de Tx	+12 dBm	+9 dBm
Consumo	121 mA	130 mA
Frecuencia	30 MHz-12.5 GHz	30 MHz-12.5 GHz

Tabla 6. Comparación entre dispositivos que integran el sistema de comunicación WiFi.

Especificaciones	Dispositivos	
	FiPy	TTGO LoRa32
Estándar WiFi	802.11 b/g/n/e/i	802.11 b/g/n
Protección de acceso	WPA/WPA2/WPA2-Enterprise/WPS	WPA/WPA2/WPS
Frecuencia	2412-2484 MHz	2412-2048 MHz
Máximo poder de Tx	15 dBm	14 dBm
Consumo	137 mA	240 mA

2.5. LTE-M

FiPy es compatible con LTE CAT-M1 y NB-IoT, estos son los protocolos celulares más nuevos, de baja potencia y largo alcance. Al momento de escribir este trabajo estas conectividades no están ampliamente disponibles, sin embargo, el dispositivo tiene la posibilidad de poder implementarla ya que cuenta con el hardware para esta red.

3. Resultados

Para poder implementar el programa de LoRaWAN fue necesario crear un gateway con un SoC ESP32, ya que en la zona no existía alguno, y como ya se mencionó para las redes LTE-M, en México apenas se empieza con estas redes por lo que no existe cobertura tan amplia de estas.

En la tabla 1 se muestran algunos dispositivos que integran 1 o más de estas redes, comparando diferentes módulos existentes para las 5 redes integradas en este dispositivo.

En las tablas 2 a 6 se muestra una comparación de los diferentes dispositivos enlistados en la tabla 1, clasificados de acuerdo con la tecnología de comunicación que integran.

4. Conclusión

La implementación de este dispositivo es más eficiente y barato a la hora de integrar varias redes en un solo proyecto y resulta en un consumo energético más bajo que integrar varios módulos para diferentes redes. este módulo está dirigido a proyectos en los que la versatilidad de la comunicación es un factor importante, ya que puede cambiar de red en cualquier instante, sin embargo, si el proyecto solo requiere de una red para envío de datos o control basta con adquirir cualquiera de los otros dispositivos enlistados anteriormente.

Referencias

1. AWS Amazon: ¿Qué es IoT? (2022) aws.amazon.com/es/what-is/iot
2. ORACLE: ¿Qué es el IoT? (2022) www.oracle.com/mx/internet-of-things/what-is-iot
3. Gracia, M: Deloitte, IoT - Internet of Things (2022) www2.deloitte.com/es/es/pages/technology/articles/IoT-internet-of-things.html
4. DIGI (2022) es.digi.com/blog/post/the-4-stages-of-iot-architecture
5. Becolve Digital: LPWAN: qué son y para qué se utilizan (2022) www.m2mlogitek.com/lpwan-que-son-y-para-que-se-utilizan
6. LoRaWAN (2022) lorawan.es
7. Sigfox OG Technology: Sigfox oG technology by unabiz (2022) www.sigfox.com/en
8. NC Tech. The New TechCompany: ¿Qué es IoT de banda estrecha (NB-IoT)? (2022) nctech.com.mx/blog/iot-industrial/nb-iot/
9. pycom: Next Generation Internet of Things Platform (2022) pycom.io