Search
DFRobot Products (DFR) Index
Sensor (SEN) Index
Telecom Module (TEL) Index
Motor Driver (DRI) Index
Fitting (FIT) Index
Robot (ROB) Index
micro:bit (MBT) Index
Kit (KIT) Index
Toy (TOY) Index
Servos (SER) Index
What is XX series
How to select series
How it works series
PinPong Library

Introduction
Features
Specification
Board Overview
Scan with mobile app
Scan with ESP32
FAQ
More Documents

Self-powered Wireless/Radio Switch Module

SKU:TEL0173

Introduction

In life, remote controllers can be seen everywhere, and most of them are inseparable from the battery.However, we often encounter problems when using the battery, for instance, it is troublesome to replace the battery, batteries are easy to corrode in a humid environment, or the waste batteries may cause serious environmental pollution.Do you know that wireless remote control can actually work without batteries!

DFRobot has developed a self-powered wireless switch that can transmit wireless signals without the need for batteries. The switch consists of a micro generator and a BLE (Bluetooth Low Energy) RF circuit. The micro generator uses electromagnetic induction to convert the kinetic energy from pressing into electrical energy. The RF circuit rectifies and stores this electrical energy and ultimately broadcasts the BLE signal wirelessly. Any nearby device capable of scanning for BLE signals can receive the message. This switch is ideal as a maintenance-free solution for long-term monitoring of press, collision, and other state changes, making it suitable for applications such as wireless doorbells, call buttons, order buttons, and wall switches.

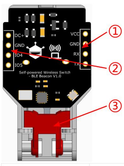
Features

- No battery needed, maintenance-free
- Wireless signal, wiring-free
- 6-byte unique address code
- Advertising BLE signals
- 100,000 press lifespan

Specification

- Frequency: 2.4GHz
- Advertising channels: 37 + 38 + 39 (2402MHz, 2426MHz, 2480MHz)
- Advertising data format:Supplier-customized Beacon
- Generated energy: 250μJ
- Actuation force: typical value 1300g
- Module weight: 14g ± 2.5g (excluding antenna)
- Module dimensions: 44 x 22 x 11mm
- Press frequency: up to 2 presses per second (pressing too quickly may trigger a protection state, requiring approximately 10 seconds to reset)
- Communication distance: >40m (with a smartphone as the receiver, in open outdoor conditions)

Board Overview

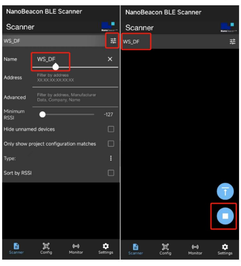


Name	Function
1. Programming port	Reserved for custom functions
2. Expansion port	Reserved for custom functions
3. Power generation button	Generates energy when pressed

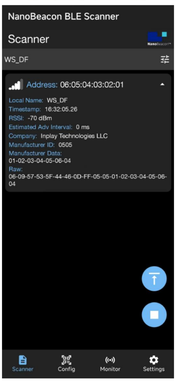
- Note: For custom advertising formats, please contact us. Supported advertising formats and related functionality can be tested using TEL0168-Fermion: BLE Sensor Beacon

Scan with mobile app

- 1.Install and openNanoBeacon™ BLE Scanner
- 2.Click on the top-left filter button to open the filter, then filter by name "WS_DF". Once done, click the filter button again to close the filter. Note that if there is a triangle symbol at the bottom-right, it means scanning is paused. You can resume scanning by clicking the triangle symbol or scrolling down the page.



- 3.Press the power generation button to see the app detect the button press.

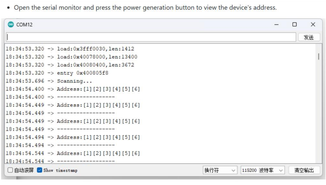


Scan with ESP32

- Ready for Arduino IDE & ESP32 Environment: DFR0654 FireBeetle_ESP32-E_first use
- Burn the following program for the ESP32

```
/*
   Based on Neil Kolban example for IDF: https://github.com/nkolban/esp32-snippets/blob/master/cpp_utils/tests/BLE%20Tests/SampleScan.cpp
   Ported to Arduino ESP32 by Evandro Copercini
   Changed to a beacon scanner to report iBeacon, EddystoneURL and EddystoneTLM beacons by beegee-tokyo
*/

#include <Arduino.h>
#include <BLEDevice.h>
#include <BLEUtils.h>
#include <BLEScan.h>
#include <BLEAdvertisedDevice.h>
#include <BLEEddystoneURL.h>
#include <BLEEddystoneTLM.h>
#include <BLEBeacon.h>
#define ENDIAN_CHANGE_U16(x) ((((x)&0xFF00) >> 8) + (((x)&0xFF) << 8))

float TemperatureData,HumidityData;
float Temperature,Humidity;

int scanTime = 3; //In seconds
BLEScan *pBLEScan;

class MyAdvertisedDeviceCallbacks : public BLEAdvertisedDeviceCallbacks
{
    void onResult(BLEAdvertisedDevice advertisedDevice)
    {
      if (advertisedDevice.haveName())
      {
        if(String(advertisedDevice.getName().c_str()) == "WS_DF")
        {
          std::string strManufacturerData = advertisedDevice.getManufacturerData();
          uint8_t cManufacturerData[100];
          strManufacturerData.copy((char *)cManufacturerData, strManufacturerData.length(), 0);
          Serial.printf("Address:");

          for (int i = 2; i < 8; i++)
          {
            Serial.printf("[%X]", cManufacturerData[i]);
          }
          Serial.println();
          Serial.println("-----------------");
        }
      }
    }
};
void setup()
{
  Serial.begin(115200);
  Serial.println("Scanning...");

  BLEDevice::init("");
  pBLEScan = BLEDevice::getScan(); //create new scan
  pBLEScan->setAdvertisedDeviceCallbacks(new MyAdvertisedDeviceCallbacks());
  pBLEScan->setActiveScan(true); //active scan uses more power, but get results faster
  pBLEScan->setInterval(100);
  pBLEScan->setWindow(99); // less or equal setInterval value
}
void loop()
{
  // put your main code here, to run repeatedly:
  BLEScanResults foundDevices = pBLEScan->start(scanTime, false);
  pBLEScan->clearResults(); // delete results fromBLEScan buffer to release memory
  delay(2000);
}
```

- Open the serial monitor and press the power generation button to view the device's address.



FAQ

- 1: After pressing the button, sometimes the mobile app cannot scan BLE information?
  - Please close other background processes on your phone, disconnect from WiFi and mobile networks, and only enable Bluetooth. When your phone is busy processing tasks from other apps, it can cause the NanoBeacon BLE Scanner app to miss some BLE broadcasts during scanning.
  - When there are many obstacles between the receiver and transmitter, it significantly reduces the communication distance.
  - Pressing the button too quickly can lead to excessive power generation, causing the module to enter a protection state. Stop pressing the button and wait for about 10 seconds for it to recover on its own.
  - When the mobile app hasn't received BLE advertisements for a long time, it automatically enters sleep mode and stops scanning. Click the start button in the bottom right corner or pull down the interface to refresh.
- 2: After pressing the button, sometimes the ESP32 cannot scan BLE information?
  - Please avoid testing in environments with too many electronic devices. In surroundings where there are numerous BLE devices, the ESP32's processing speed may be insufficient, leading to missed BLE advertisement scans.

More questions and cool idea, please visit DFRobot Forum

More Documents

- Schematic Diagram
- Dimensions & Component Layout
- Generator Dimension Drawing