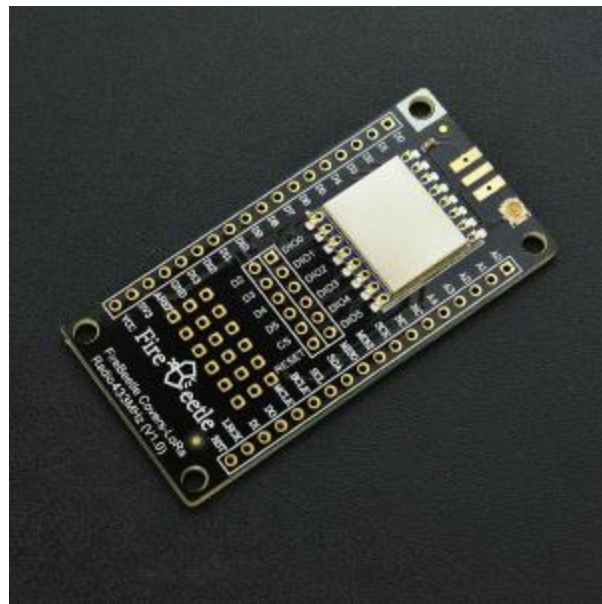


FireBeetle Covers-LoRa Radio 433MHz

SKU:TEL0121

Introduction



FireBeetle Covers-LoRa Radio 433MHz SKU: TEL0121

DFRobot FireBeetle IOT series are low power consumption controllers designed for Internet of Things (IoT) development. The FireBeetle series are aiming straight at the point fast and convenient low-power IoT building. There are three categories of FireBeetle series, containing Boards (main control), Covers (expansion boards) and related Accessories.

FireBeetle LoRa 433MHz wireless transmission module adopts a high-performance chip SX127x LoRa 433MHz and with built-in PA (power amplifier) gain, compatible with Arduino. It supports long-range monitor, FSK modulation and the maximum transmission rate is 300K[null bps]. It also supports LoRaTM, the maximum transmission rate is 37.5Kbps. The circuit current is even lower than 0.2uA in the low power mode. It can be widely used in remote home automation detection, health monitoring and wearable device and so on.

NOTE: All parameters related to wireless signals in the same network should maintain consistency to avoid communication problems.

Specifications

- Operating Current: 3.3V
- Operating Frequency: 433MHz
- Modulation Type: LoRaTM
- Output Power: $\leq 20\text{dBm}$
- Receiving Sensitivity: -139dBm
- Emersion Current: $\leq 120\text{mA}$
- Receiving Current: $\leq 15\text{mA}$
- Sleep Current: $\leq 1\mu\text{A}$
- Transmission Power Range: $-1\text{dBm} \sim 20\text{dBm}$ (Maximum)
- Transmission Mode: FIFO/ DMA
- Operating Frequency: $403\text{MHz} \sim 463\text{MHz}$ (typical value: 433MHz)
- Transmission Range: 1 Km
- Modulation Type: LoRaTM /FSK /GFSK /OOK
- Built-in aerials offer good anti-jamming capacity
- The Maximum RSSI is 127dB
- FIFO Volume: 256bytes
- Support CRC Frequency-hopping
- SPI Interface
- GPIO Interface: 5 (can be used as interruption)
- Operating Temperature: $-20^{\circ}\text{C} \sim +70^{\circ}\text{C}$
- Dimension: 58 x 29 mm/ 2.28 x 1.14 inches

Board Overview

FireBeetle Covers-LoRa Radio 433MHz is main controlled via SPI interface (MISO, MOSI, SCK, CS, RESET). Connect these pins to MISO, MOSI, SCK on FireBeetle board, connect CS (chip-select) and RESET to D2, D3, D4, D5 with Dupont wires.

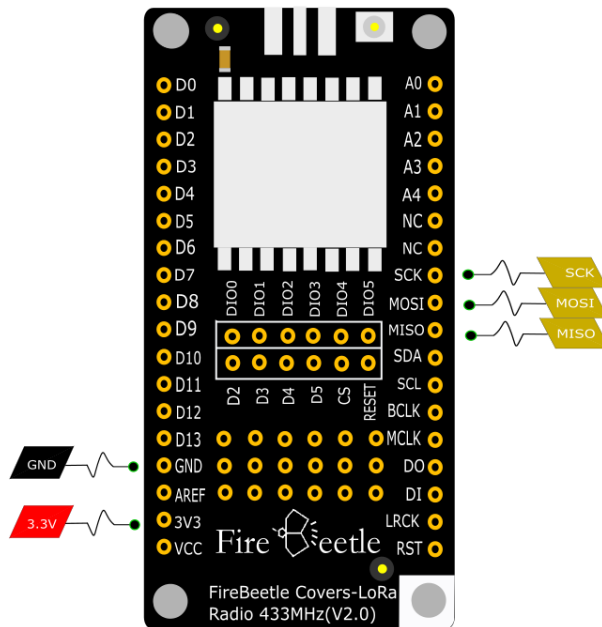



Fig1: FireBeetle Covers-LoRa Radio 433MHz data control pins

 NOTE: When disconnect NC, VCC is the output voltage of the power supply(USB power supply: 5V, lithium battery power supply: 3.7V)

Tutorial

Requirements

- Hardware
 - 2 x ESP32 Board
 - 1 x FireBeetle Covers-LoRa Radio 433MHz
 - Some wire

The default connection setting is to connect CS to D4, RESET to D2, you can also modify programs to connect pins to other IO interfaces (with init pass parameters: `init(uint8_t NSS_PIn = NSS_PIN, uint8_t NRESET_PIn = RESET_PIN)`).

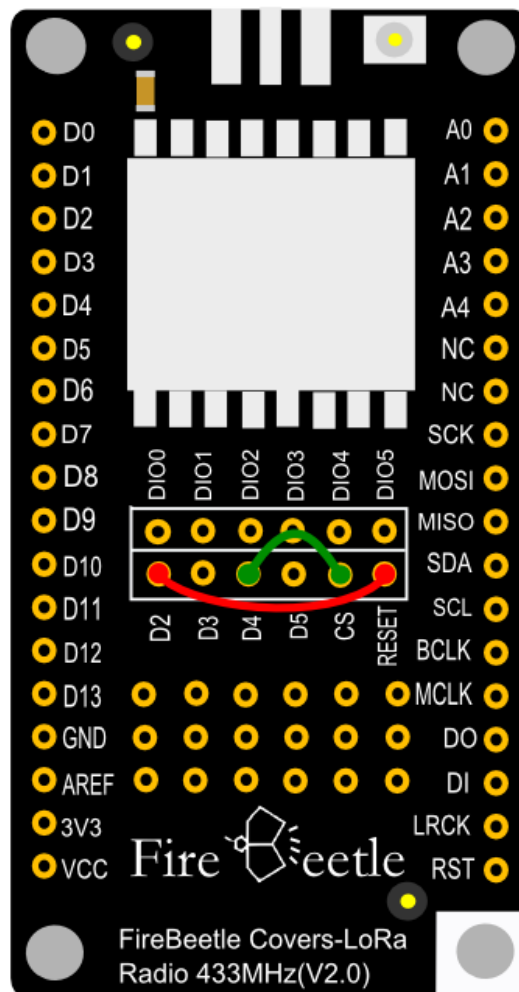


Fig2: Firebeetle_Covers-433MHz-Connect_CS&RESET

- Software
Arduino IDE [Click to Download Arduino IDE from Arduino®](#)

Connection

- Joint ESP32 board and LoRa module together
- Connect ESP32 board to PC via USB line

Sample Code

[Click to download FireBeetle Covers-LoRa Radio 433MHz library first.](#)

- Download the master program and the slave program to two ESP32 board respectively.
- The Slave program code

```
/*!
 * @file receiverTest.ino
 * @brief DFRobot's received data
 * @n [Get the module here]
 * @n This example is receive.
 * @n [Connection and Diagram]
 *
 * @copyright [DFRobot] (http://www.dfrobot.com), 2016
 * @copyright GNU Lesser General Public License
 *
 * @author [yangyang]
 * @version V1.0
 * @date 2017-04-10
 */

#include <DFRobot_LoRa.h>

DFRobot_LoRa lora;

uint8_t len;
uint8_t rxBuf[32];

void setup()
```

```
{  
  
  Serial.begin(115200);  
  
  pinMode(LED_BUILTIN, OUTPUT);  
  Serial.println("Receiver Test");  
  
  while(!lora.init()) {  
    Serial.println("Starting LoRa failed!");  
    delay(100);  
  }  
  
  lora.rxInit();  
}  
void loop()  
{  
  
  if(lora.waitIrq()) { // wait for RXDONE interrupt  
    lora.clearIRQFlags();  
    len = lora.receivePackage(rxBuf); // receive data  
    Serial.write(rxBuf, len);  
    Serial.println();  
    lora.rxInit(); // wait for packet from master  
  
    // print RSSI of packet  
    Serial.print("with RSSI ");  
    Serial.println(lora.readRSSI());  
  
    static uint8_t i;  
    i = ~i;  
    digitalWrite(LED_BUILTIN, i);  
  }  
}
```

- The Master program code

```
/*!
 * @file sendTest.ino
 * @brief DFRobot's send data
 * @n [Get the module here]
 * @n This example is send.
 * @n [Connection and Diagram]
 *
 * @copyright [DFRobot] (http://www.dfrobot.com), 2016
 * @copyright GNU Lesser General Public License
 *
 * @author [yangyang]
 * @version V1.0
 * @date 2017-04-10
 */

#include <DFRobot_LoRa.h>

DFRobot_LoRa lora;

uint8_t counter = 0;
uint8_t sendBuf[] = "HelloWorld!";

/* The default pin:
 *
 * SS:D4
 *
 * RST:D2 (If you are using the FireBeetle Board-ESP8266 motherbo
ard controller, the RST defaults to D3)
 */

void setup()
{
    Serial.begin(115200);

    while(!lora.init()) {
```

```

        Serial.println("Starting LoRa failed!");
        delay(100);
    }
}
void loop()
{
    Serial.print("Sending packet: ");
    Serial.println(counter);

    // send packet
    lora.sendPackage(sendBuf, 11); // sending data
    lora.idle(); // turn to standby mode

    counter++;

    #if 0
    if(counter%10 == 0) {
        lora.sleep();
        delay (5000); // sleep 5 seconds
    }
    #endif

    delay(500);
}

```

Result

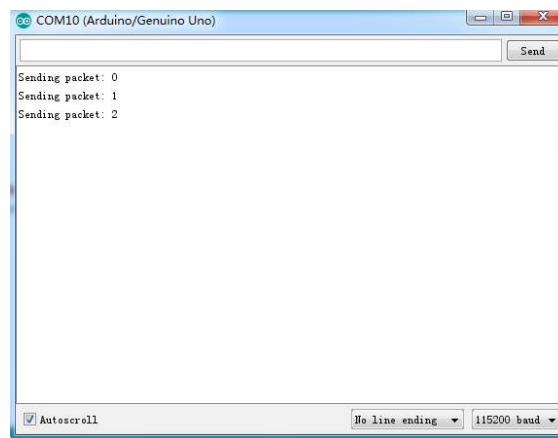


Fig3: The master sends data

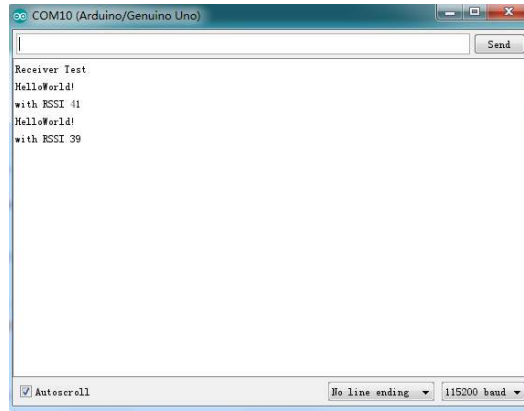


Fig3: The slave receives data

FAQ

NOTE: For any questions, advice or cool ideas to share, please visit the [DFRobot Forum](#).

More Documents

- Schematic
- Dimension
- Datasheet (Chinese)
- CE Certification