

# RXB6 RF RX/TX Pair- 433Mhz



## RXB6 RF RX/TX Pair - 433Mhz

The RXB6 is based on a Superheterodyne design with PLL and Automatic Gain Control. The new design is less immune to noise than our standard TX/RX pair. The RXB6 pair is a great choice for controlling devices remotely.

### Quick Spec

- Frequency Range: 300 ~ 450MHz
- Receiver Sensitivity: -114 ~ -110dBm
- Data Rate: 0.058 ~ 10KBaud
- Supply Voltage: 3.0 ~ 5.5VDC
- Current: 5.7 ~ 7.3mA
- Operating Temperature: -40 ~ +85°C

## Pinout

### RXB6 Receiver



- 1 - Antenna
- 2 - GND
- 3 - GND
- 4 - VCC
- 5 - VCC
- 6 - DER
- 7 - DATA
- 8 - GND

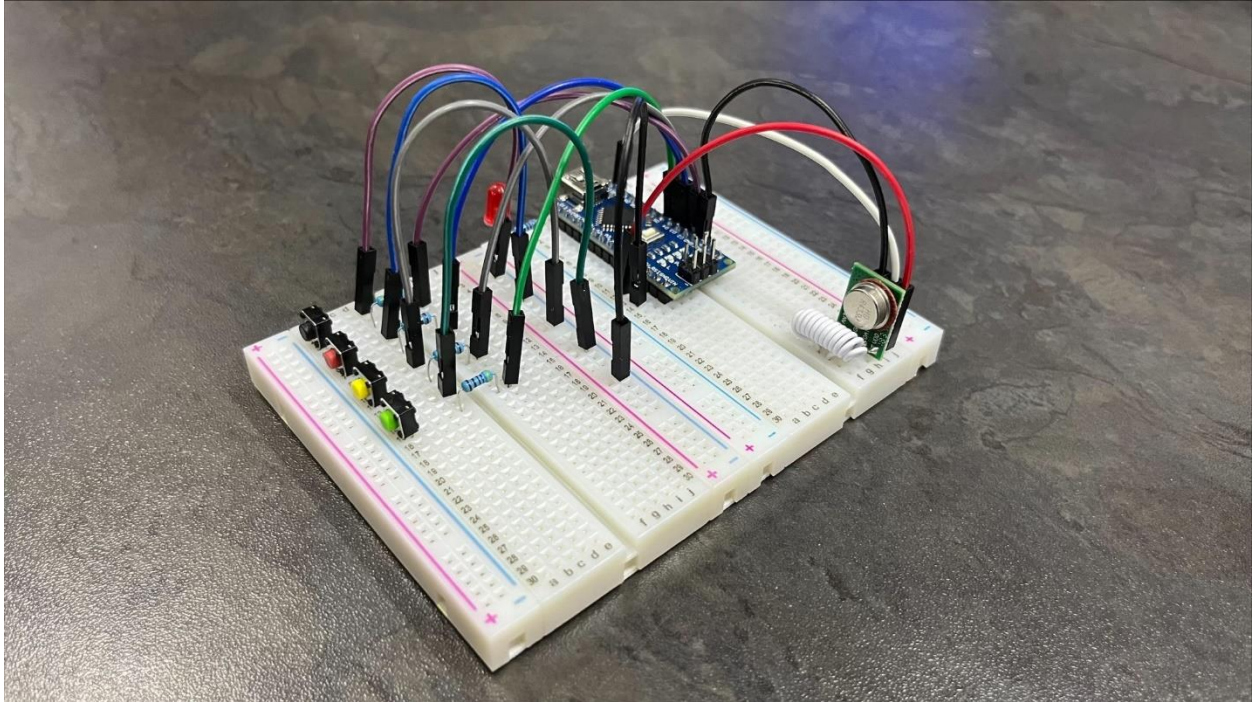
### Transmitter



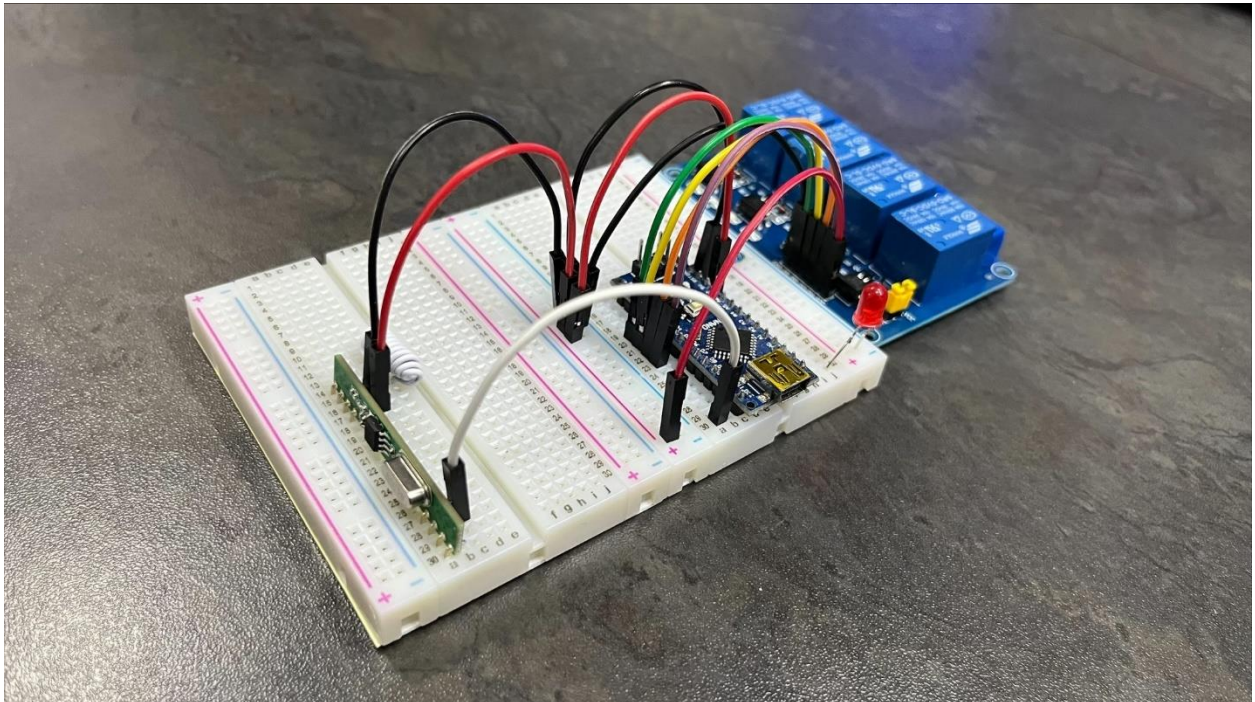
- 1 - GND
- 2 - DATA IN
- 3 - VCC
- 4 - Antenna

# Project 4 Channel Relay

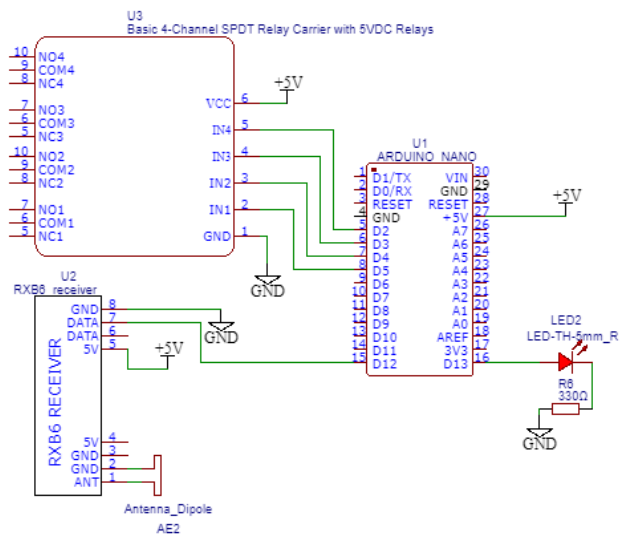
Transmitter:



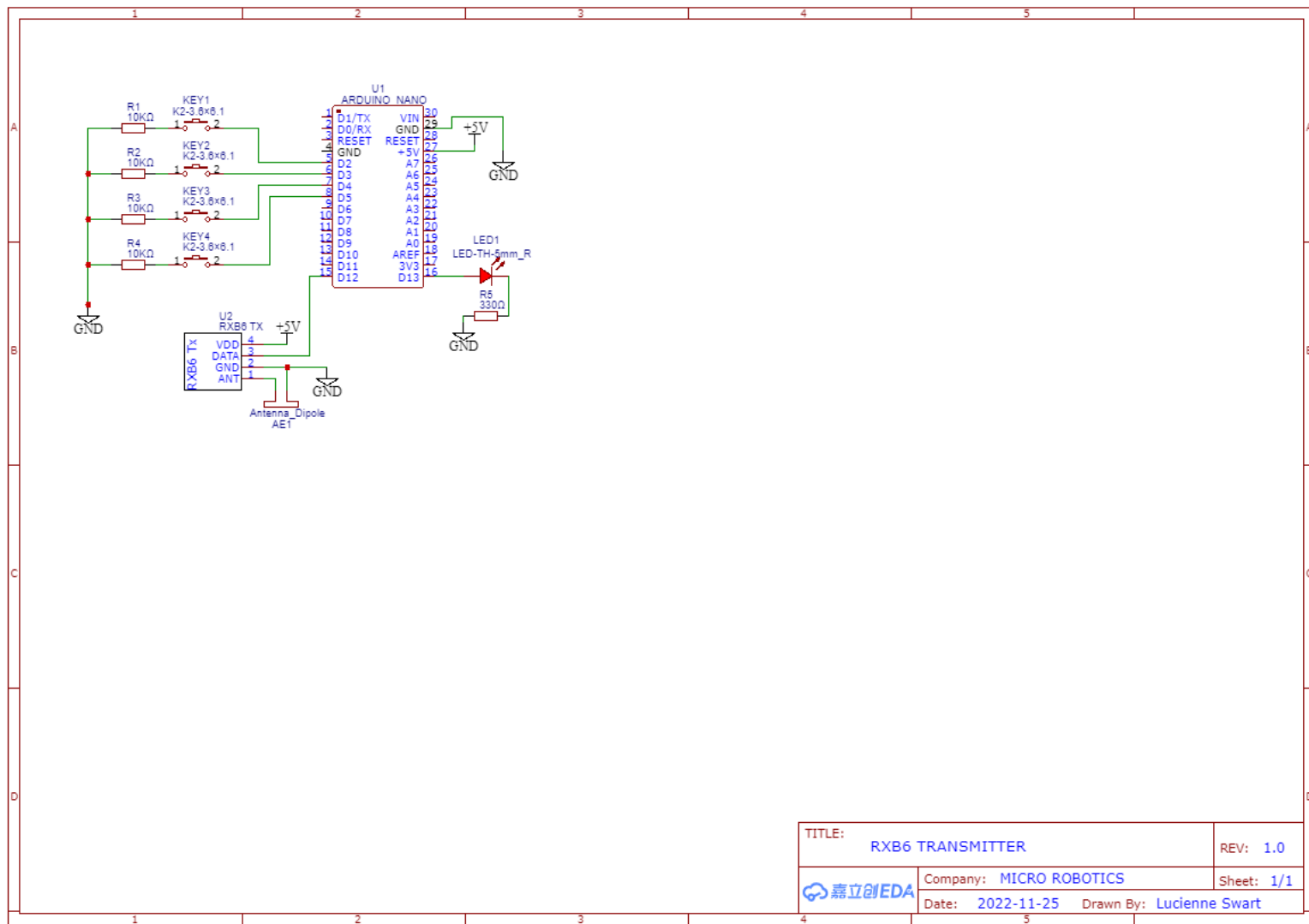
Receiver:



# Schematics



TITLE: RXB6 RECEIVER		REV: 1.0
嘉立创EDA	Company: MICRO ROBOTICS	Sheet: 1/1
	Date: 2022-11-25	Drawn By: LUCIENNE SWART



# Receiver Sketch

```
//Receiver Sketch

#include <VirtualWire.h>
int count;
void setup() {
    Serial.begin(9600);           // Debugging only
    Serial.println("setup");      //Prints "Setup" to the serial monitor
    vw_set_rx_pin(12);            //Sets pin D12 as the RX Pin
    vw_set_ptt_inverted(true);    // Required for DR3100
    vw_setup(4000);               // Bits per sec
    vw_rx_start();                // Start the receiver PLL running

    pinMode(2, OUTPUT);           //Relay one
    pinMode(3, OUTPUT);           //Relay two
    pinMode(4, OUTPUT);           //Relay three
    pinMode(5, OUTPUT);           //Relay four

    digitalWrite(2, LOW);
    digitalWrite(3, LOW);
    digitalWrite(4, LOW);
    digitalWrite(5, LOW);
}

void loop() {
    uint8_t buf[VW_MAX_MESSAGE_LEN];
    uint8_t buflen = VW_MAX_MESSAGE_LEN;
    if (vw_get_message(buf, &buflen)) // Non-blocking
    {
        int i;
        digitalWrite(13, true); // Flash a light to show received good message
        // Message with a good checksum received, dump it.
        Serial.print("Got: ");

        for (i = 0; i < buflen; i++) {
            int c = (buf[i]);
            Serial.print(c);
            Serial.print(" ");
            if (c == 65) {
                digitalWrite(2, !digitalRead(2));
            }
            if (c == 66) {
                digitalWrite(3, !digitalRead(3));
            }
        }
    }
}
```

```
    }

    if (c == 67) {
        digitalWrite(4, !digitalRead(4));
    }

    if (c == 68) {
        digitalWrite(5, !digitalRead(5));
    }
}
count++;
// Serial.print(count);
Serial.println("");
digitalWrite(13, false);
}
}
```



# Transmitter Sketch

```
//Transmitter Sketch

#include <VirtualWire.h>
const int buttonPin1 = 2;
const int buttonPin2 = 3;
const int buttonPin3 = 4;
const int buttonPin4 = 5;

void setup() {
  Serial.begin(9600);          // Debugging only
  Serial.println("setup");    // Prints "Setup to the serial monitor"

  pinMode(buttonPin1, INPUT_PULLUP);
  pinMode(buttonPin2, INPUT_PULLUP);
  pinMode(buttonPin3, INPUT_PULLUP);
  pinMode(buttonPin4, INPUT_PULLUP);

  vw_set_tx_pin(12);          // Sets pin D12 as the TX pin
  vw_set_ptt_inverted(true);  // Required for DR3100
  vw_setup(4000);             // Bits per sec
}

void loop() {
  int buttonState1 = 1;
  buttonState1 = digitalRead(buttonPin1);
  if (buttonState1 == LOW) {
    const char *msg = "A";          // Message to be sent
    digitalWrite(13, true);         // Flash a light to show transmitting
    vw_send((uint8_t *)msg, strlen(msg)); //Sending the message
    vw_wait_tx();                   // Wait until the whole message is
    gone
    Serial.println(*msg);
    digitalWrite(13, false); // Turn the LED off.
    delay(1000);              // A short gap.    }
  }
  {
    int buttonState2 = 1;
    buttonState2 = digitalRead(buttonPin2);
    if (buttonState2 == LOW) {
```



```

    const char *msg = "B";                // Message to be sent
    digitalWrite(13, true);               // Flash a light to show
transmitting
    vw_send((uint8_t *)msg, strlen(msg)); //Sending the message
    vw_wait_tx();                         // Wait until the whole message is
gone
    Serial.println(*msg);
    digitalWrite(13, false); // Turn the LED off.
    delay(1000);                 // A short gap.    }
}
{
    int buttonState3 = 1;
    buttonState3 = digitalRead(buttonPin3);
    if (buttonState3 == LOW) {
        const char *msg = "C";                // Message to be sent
        digitalWrite(13, true);               // Flash a light to show
transmitting
        vw_send((uint8_t *)msg, strlen(msg)); //Sending the message
        vw_wait_tx();                         // Wait until the whole message is
gone
        Serial.println(*msg);
        digitalWrite(13, false); // Turn the LED off.
        delay(1000);                 // A short gap.    }
    }
    {
        int buttonState4 = 1;
        buttonState4 = digitalRead(buttonPin4);
        if (buttonState4 == LOW) {
            const char *msg = "D";                // Message to be sent
            digitalWrite(13, true);               // Flash a light to show
transmitting
            vw_send((uint8_t *)msg, strlen(msg)); //Sending the message
            vw_wait_tx();                         // Wait until the whole message
is gone
            Serial.println(*msg);
            digitalWrite(13, false); // Turn the LED off.
            delay(1000);                 // A short gap.    }
        }
    }
}
}
delay(100);
}

```