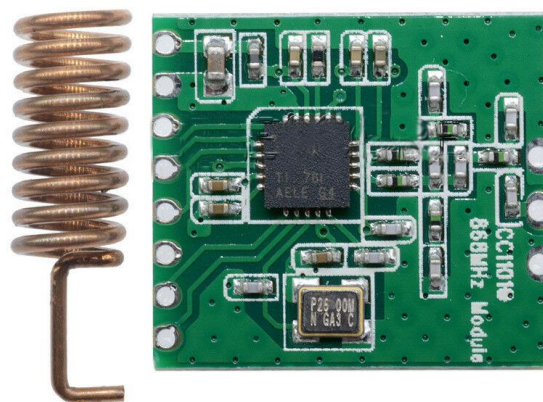


RBS16398 CC1101 Funkmodul Transceiver 868Mhz



Sicherheitshinweise	1
Technische Daten	2
Pinbelegung	2
Anschlussplan	2
Arduino Beispiel mit RadioLib	3

Sicherheitshinweise

Lesen Sie diese Hinweise sorgfältig durch, bevor Sie das Funkmodul verwenden oder installieren. Eine unsachgemäße Verwendung kann zu Schäden am Modul, an angeschlossenen Geräten oder zu Funktionsstörungen führen.

Verwenden Sie das Modul ausschließlich innerhalb der angegebenen Betriebsspannung von 1,8 V bis 3,6 V. Eine höhere Spannung kann das Modul dauerhaft beschädigen. Schließen Sie das Modul niemals direkt an 5 V an.

Achten Sie auf die korrekte Verdrahtung der Pins. Falsch angeschlossene Leitungen können zu Kurzschlüssen oder zur Beschädigung des Moduls und des Mikrocontrollers führen.

Betreiben Sie das Modul nur mit geeigneten Mikrocontrollern oder Entwicklungsboards, die kompatible Logikpegel verwenden oder über entsprechende Pegelwandler verfügen.

Vermeiden Sie elektrostatische Entladungen. Elektronische Bauteile können durch statische Elektrizität beschädigt werden. Berühren Sie vor dem Umgang mit dem Modul möglichst eine geerdete Oberfläche.

Betreiben Sie das Modul nur in trockener Umgebung. Feuchtigkeit, Kondenswasser oder Flüssigkeiten können Kurzschlüsse und Schäden verursachen.

Verwenden Sie eine geeignete Antenne für das 868-MHz-Frequenzband. Der Betrieb ohne Antenne kann die Funkleistung stark reduzieren und im ungünstigen Fall das Modul beschädigen.

Achten Sie darauf, dass das Modul ausreichend belüftet ist und nicht in der Nähe starker Wärmequellen betrieben wird.

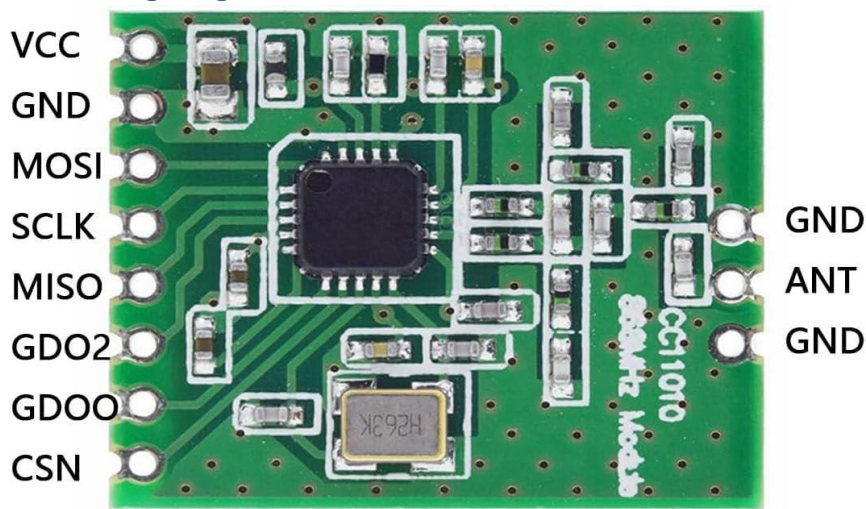
Verwenden Sie das Modul nur für Entwicklungs-, Lern- und Hobbyprojekte oder in dafür geeigneten elektronischen Anwendungen.

Beachten Sie bei der Nutzung von Funktechnik die in Ihrem Land geltenden gesetzlichen Bestimmungen für das 868-MHz-Frequenzband.

Technische Daten

- Mehrkanal-Transceiver @ 868 MHz
- Unterstützt FSK, GFSK, ASK / OOK und MSK
- Konfigurierbare Datenrate: 1,2 kbps bis 500 kbps
- Unabhängige 64-Byte-RX- und TX-Daten-FIFOs
- Unterstützt Radio Wake
- Unterstützt automatische Clear Channel Assessment
- Effiziente SPI-Schnittstelle
- Niedrige Stromversorgung (1,8V-3,6 V)

Pinbelegung



Anschlussplan

CC1101 Modul Pin	Funktion	Arduino Uno / Nano Pin
3.3V	Versorgungsspannung	3.3V
GND	Masse	GND
SI	SPI MOSI (Master Out Slave In)	D11
SO	SPI MISO (Master In Slave Out)	D12
SCLK	SPI Clock	D13
CSN	Chip Select	D10
GDO0	Interrupt / Status Pin	D2
GDO2	optionaler Status Pin	D3 (optional)
ANT	Antennenanschluss	868 MHz Antenne
GND	Masse für Antenne	GND

Achtung für den Anschluss an ein 5V Arduino Board wird ein Pegelwandler benötigt!

Arduino Beispiel mit RadioLib

Benötigte Komponenten

Du brauchst idealerweise **zwei gleiche Module**:

- Arduino 1 + CC1101 = Sender
- Arduino 2 + CC1101 = Empfänger

Wichtig:

- beide auf **868.0 MHz**
- gleiche **Bitrate**
- gleicher **Sync Word**
- **Antenne** anschließen

Dann solltest du im seriellen Monitor des Empfängers die Nachricht sehen.

Bibliothek installieren

In der Arduino IDE:

Bibliothek verwalten → nach RadioLib suchen → installieren

Sender-Beispiel

```
#include <RadioLib.h>

// CC1101: CS = 10, GD00 = 2, RST = nicht benutzt, GD02 = 3
CC1101 radio = new Module(10, 2, RADIOLIB_NC, 3);

void setup() {
  Serial.begin(9600);
  delay(1000);

  Serial.println("CC1101 Sender startet...");

  int state = radio.begin(868.0); // Frequenz 868 MHz

  if (state != RADIOLIB_ERR_NONE) {
    Serial.print("Fehler beim Starten, Code: ");
    Serial.println(state);
    while (true);
  }

  // optionale Einstellungen
  radio.setBitRate(4.8);
  radio.setFrequencyDeviation(20.0);
  radio.setOutputPower(10); // je nach Modul/Bibliothek evtl. -30 bis +10
dBm
  radio.setSyncWord(0x12, 0xAD);

  Serial.println("Sender bereit.");
}

void loop() {
  String text = "Hallo vom CC1101";

  Serial.print("Sende: ");
  Serial.println(text);

  int state = radio.transmit(text);

  if (state == RADIOLIB_ERR_NONE) {
    Serial.println("Senden erfolgreich");
  } else {
    Serial.print("Sende-Fehler, Code: ");
    Serial.println(state);
  }

  delay(2000);
}
```

Empfänger-Beispiel

```
#include <RadioLib.h>

// CC1101: CS = 10, GD00 = 2, RST = nicht benutzt, GD02 = 3
CC1101 radio = new Module(10, 2, RADIOLIB_NC, 3);

void setup() {
  Serial.begin(9600);
  delay(1000);

  Serial.println("CC1101 Empfänger startet...");

  int state = radio.begin(868.0); // Frequenz 868 MHz

  if (state != RADIOLIB_ERR_NONE) {
    Serial.print("Fehler beim Starten, Code: ");
    Serial.println(state);
    while (true);
  }

  // gleiche Einstellungen wie Sender!
  radio.setBitRate(4.8);
  radio.setFrequencyDeviation(20.0);
  radio.setSyncWord(0x12, 0xAD);

  Serial.println("Empfänger bereit.");
}

void loop() {
  String str;
  int state = radio.receive(str);

  if (state == RADIOLIB_ERR_NONE) {
    Serial.print("Empfangen: ");
    Serial.println(str);

    Serial.print("RSSI: ");
    Serial.print(radio.getRSSI());
    Serial.println(" dBm");

    Serial.print("LQI: ");
    Serial.println(radio.getLQI());
  } else if (state != RADIOLIB_ERR_RX_TIMEOUT) {
    Serial.print("Empfangsfehler, Code: ");
    Serial.println(state);
  }
}
```