Benutzerhandbuch:

Ebyte E32 LoRa-Modul Bibliothek für Raspberry Pi

Übersicht

Diese Bibliothek ermöglicht die Konfiguration und Steuerung von Ebyte E32 LoRa-Modulen über einen Raspberry Pi. Sie bietet Funktionen zum Lesen und Schreiben von Modulparametern, zur Steuerung der Betriebsmodi und zur Interaktion mit der Hardware über GPIO-Pins.

Voraussetzungen

- Raspberry Pi mit installiertem Betriebssystem (z.B. Raspberry Pi OS)
- Python 3 und die **serial**-Bibliothek installiert
- Ebyte E32 LoRa-Modul physisch an den Raspberry Pi angeschlossen

Installation

- Stellen Sie sicher, dass Python 3 installiert ist.
- Installieren Sie die **serial**-Bibliothek mit **pip3 install pyserial**.

Konfiguration der GPIO-Pins

- Verbinden Sie die Pins M0, M1 und AUX des Ebyte E32 Moduls mit den entsprechenden GPIO-Pins des Raspberry Pi.
- Notieren Sie sich die GPIO-Pin-Nummern, die Sie für M0, M1 und AUX verwenden.

Grundlegende Benutzung

- **Initialisierung**:
  - Importieren Sie die Bibliothek in Ihr Python-Skript.
  - Initialisieren Sie ein **EbyteRaspberryPi**-Objekt mit den seriellen und GPIO-Pin-Informationen.
- **Lesen von Parametern**:
  - Verwenden Sie die **read_parameters()**-Methode, um die aktuellen Einstellungen des Moduls auszulesen.
- **Schreiben von Parametern**:
  - Ändern Sie die gewünschten Parameter über die entsprechenden Attribute des **Ebyte**-Objekts.
  - Verwenden Sie die **write_parameters()**-Methode, um die geänderten Einstellungen auf das Modul zu übertragen.
- **Betriebsmodi steuern**:
  - Verwenden Sie die **set_mode()**-Methode, um zwischen verschiedenen Betriebsmodi wie Normal, Wake-up, Powersave und Sleep umzuschalten.

Erweiterte Funktionen

- **Parameter setzen**: Setzen Sie spezifische Parameter wie **uart_parity**, **uart_baud**, **air_data_rate** usw. über die entsprechenden Attribute des **Ebyte**-Objekts.
- **Werkseinstellungen zurücksetzen**: Verwenden Sie die **reset**-Befehlszeilenoption, um das Modul auf die Werkseinstellungen zurückzusetzen.

Fehlerbehandlung

- Überprüfen Sie die Verbindungen und Pin-Belegungen, wenn das Modul nicht reagiert.
- Stellen Sie sicher, dass die serielle Kommunikation korrekt konfiguriert ist.

Sicherheitshinweise

- Beachten Sie beim Umgang mit elektronischen Komponenten die entsprechenden Sicherheitsvorkehrungen.
- Stellen Sie sicher, dass Ihr Raspberry Pi ausgeschaltet ist, wenn Sie das LoRa-Modul anschließen oder entfernen.

Beispielcode

python

Copy code

```python
import serial
from ebyte_library import EbyteRaspberryPi

# Ersetzen Sie diese Werte durch Ihre tatsächlichen GPIO-Pin-Nummern
pin_m0 = 22
pin_m1 = 27
pin_aux = 17

# Erstellen Sie ein EbyteRaspberryPi-Objekt
ser = serial.Serial('/dev/serial0')
ebyte = EbyteRaspberryPi(ser, pin_m0, pin_m1, pin_aux)

# Lesen Sie die aktuellen Parameter
print(ebyte.read_parameters())

# Ändern Sie Parameter und schreiben Sie sie zurück auf das Modul
ebyte.uart_baud = 3  # Setzen Sie die Baudrate auf 9600
ebyte.write_parameters(permanent=True)
```

Hinweis: Dieses Handbuch bietet eine Übersicht über die grundlegenden Funktionen und die Verwendung der Ebyte E32 LoRa-Modul-Bibliothek auf dem Raspberry Pi. Für spezifischere Anwendungen oder erweiterte Funktionen lesen Sie bitte die vollständige Dokumentation der Bibliothek.

Aufrufparameter des Ebyte E32 LoRa-Modul Skripts

Übersicht

Das Ebyte E32 LoRa-Modul Skript für Raspberry Pi bietet verschiedene Befehlszeilenoptionen, um die Konfiguration und den Betrieb des Moduls zu steuern. Diese Optionen ermöglichen es Ihnen, Parameter zu lesen, zu schreiben und das Modul auf Werkseinstellungen zurückzusetzen.

Aufrufparameter

- **Allgemeiner Aufbau**:
  - Das Skript wird über die Befehlszeile mit spezifischen Optionen aufgerufen.
  - Beispiel: **python ebyte_script.py [Befehl] [Optionen]**
- **Verfügbare Befehle**:
  - **read**: Liest die aktuellen Modulparameter aus.
  - **write**: Schreibt neue Parameter auf das Modul.
  - **reset**: Setzt das Modul auf Werkseinstellungen zurück.
- **Optionen für jeden Befehl**:
  - **serial**: Pfad zum seriellen Portgerät (z.B. **/dev/serial0**).
  - **pin_m0**: GPIO-Pin-Nummer für den M0-Pin des Moduls.
  - **pin_m1**: GPIO-Pin-Nummer für den M1-Pin des Moduls.
  - **pin_aux**: GPIO-Pin-Nummer für den AUX-Pin des Moduls (optional).
- **Zusätzliche Optionen für den write-Befehl**:
  - **--permanent**: Gibt an, ob die Parameter dauerhaft geschrieben werden sollen (true/false).
  - **--address**: Setzt die Moduladresse (Bereich: 0x0000 bis 0xFFFF).
  - Weitere Parameteroptionen entsprechend der **BIT_LAYOUT**-Konfiguration, wie **--uart_parity**, **--uart_baud**, **--air_data_rate** usw.

- **Beispielaufrufe**:
  - Parameter lesen: **python ebyte_script.py read /dev/serial0 22 27**
  - Parameter schreiben: **python ebyte_script.py write /dev/serial0 22 27 --uart_baud 3 --permanent true**
  - Modul zurücksetzen: **python ebyte_script.py reset /dev/serial0 22 27**

Hinweise zur Verwendung

- Stellen Sie sicher, dass Sie die korrekten GPIO-Pin-Nummern für M0, M1 und AUX verwenden.
- Beim Schreiben von Parametern achten Sie darauf, gültige Werte anzugeben.
- Die **--permanent** Option im **write**-Befehl bestimmt, ob die Änderungen auch nach einem Neustart des Moduls erhalten bleiben.

Fehlerbehandlung

- Wenn das Skript nicht wie erwartet funktioniert, überprüfen Sie die Verbindung zum Modul und die korrekte Konfiguration der seriellen Schnittstelle.
- Stellen Sie sicher, dass die angegebenen Pin-Nummern und Parameterwerte korrekt sind.

Wichtiger Hinweis: Dieses Handbuch bietet eine grundlegende Anleitung zur Verwendung der Befehlszeilenoptionen des Ebyte E32 LoRa-Modul Skripts. Für detailliertere Informationen zu den einzelnen Parametern und erweiterten Funktionen, beziehen Sie sich bitte auf die vollständige Dokumentation der Bibliothek.