

Eigener serieller Anschluß für das Modul

Etwas lästig ist der Umstand, dass man zum Übertragen des Sketches jedes Mal die Datenpins des BT-Moduls entfernen muss. Dem kann abgeholfen werden, indem man eine Funktionsbibliothek verwendet, die es ermöglicht, einen weiteren Seriellen Port mit eigenem Namen zu erstellen.

Am Beginn des Sketches muss dazu folgendes eingefügt werden:

```
//Software Serial Port, um Pin 0 und 1 freilassen zu können
#include <SoftwareSerial.h>

#define rxPin 2 // Neuer receive Pin
#define txPin 3 // Neuer transmit Pin
// Definiere einen seriellen Anschluss mit den zuvor festgelegten Pins
// Der Anschluss hat den Namen BtSerial
SoftwareSerial BtSerial(rxPin, txPin);
```

Im Weiteren müssen alle Bezüge auf die Serielle Schnittstelle, die bislang `Serial.read()` u.ä. geheißen haben ersetzt werden durch `BtSerial.read()` und analog:

```
// Der Anschluss hat den Namen BtSerial
SoftwareSerial BtSerial(rxPin, txPin);

String message; //Zeichenkettenvariable
void setup()
{
  pinMode(13,OUTPUT);
  BtSerial.begin(9600); //set baud rate
}

void loop()
{
  // Solange Eingaben ankommen..
  while(BtSerial.available())
  {
    message+=char(BtSerial.read()); // Die Zeichen an die Zeichenkettenvariable anhaengen
  }
  if(!BtSerial.available())
  {
    if(message!="")
    { //Wenn eine Nachricht angekommen ist
      if (message == "h" ) {
        digitalWrite(13,HIGH);
      }
    }
  }
}
```

Verwende den folgenden Sketch als Basis für weitere Aufgaben.

[softserial.ino](#)

```
//Software Serial Port, um Pin 0 und 1 freilassen zu können
#include <SoftwareSerial.h>
```

```
#define rxPin 2 // Neuer receive Pin
#define txPin 3 // Neuer transmit Pin
// Definiere einen seriellen Anschluss mit den zuvor festgelegten Pins
// Der Anschluss hat den Namen BtSerial
SoftwareSerial BtSerial(rxPin, txPin);

String message; //Zeichenkettenvariable
void setup()
{
  pinMode(13,OUTPUT);
  BtSerial.begin(9600); //set baud rate
}

void loop()
{
  // Solange Eingaben ankommen..
  while(BtSerial.available())
  {
    message+=char(BtSerial.read()); // Die Zeichen an die
    Zeichenkettenvariable anhaengen
  }
  if(!BtSerial.available())
  {
    if(message!="")
    { //Wenn eine Nachricht angekommen ist
      if (message == "h" ) {
        digitalWrite(13,HIGH);
      }
      if (message == "l" ) {
        digitalWrite(13,LOW);
      }
      BtSerial.println(message); //ausgeben!
      message=""; //Zeichenkettenvariable zuruecksetzen
    }
  }
  delay(20); //Alle 5 Sekunden
}
```



(A1)

Übernahme des Sketch, versuche ihn zu verstehen. Beschreibe auf einem Notizzettel, was der Sketch

macht und teste, ob er funktioniert.

Ändere die Anschlüsse für das BT Modul so, dass anstelle von Pin 2 und Pin 3, Pin 7 und Pin 8 für die Kommunikation verwendet werden.



(A2)

Baue eine ferngesteuerte Lichtorgel aus 4 LEDs.

- Bei Eingabe von 1 soll ein Lauflicht erscheinen, das 5 Mal durchläuft
- Bei Eingabe von z sollen die LEDs 10 Sekunden lang zufällig blinken (<https://www.arduino.cc/en/Reference/Random>)
- Bei Eingabe von 0 sollen alle LEDs ausgeschaltet bleiben, bis 1 eingegeben wird.

Erweitere deine Schaltung um einen Taster oder Schalter, der dafür sorgt, dass die Lichtorgel keine Befehle vom Handy entgegen nimmt, solange der Schalter gedrückt ist.

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
https://wiki.qg-moessingen.de/faecher:nwt:bluetooth_steckdosen:bluetooth:serial_bib:start

Last update: **14.02.2023 18:52**

