

Bluetooth
(„Blauzahn“)



Bluetooth ist ein Industriestandard für die Datenübertragung zwischen Geräten auf kurze Distanz. Bluetooth bildet eine Schnittstelle, über die sowohl mobile Kleingeräte wie z.B. Mobiltelefone als auch Computer und Peripheriegeräte miteinander kommunizieren können. Hauptzweck von Bluetooth ist das Ersetzen von Kabelverbindungen zwischen Geräten. Es ist leistungsfähiger als rfid-Verbindungen, denn im Gegensatz zur rfid-Kommunikation können Daten nicht nur vom Sender zum Empfänger, sondern auch umgekehrt übertragen werden.

Daten

Rfid **Sender** **Empfänger**

Bluetooth **Sender** **Empfänger**

Auch die Reichweite ist größer, denn auch die Sender besitzen eine Energieversorgung. Bluetooth-Geräte senden in einem lizenzenfreien Band zwischen 2,402 GHz und 2,480 GHz. Sie dürfen weltweit zulassungsfrei betrieben werden, können aber z.B. durch WLANs oder auch Mikrowellenherde gestört werden. Die Reichweite beträgt im Allgemeinen mehrere Meter.

Jedes Bluetooth-Gerät besitzt eine individuelle und unverwechselbare 48 bit (also 6 Byte) lange MAC-Adresse (MediaAccessControl), mit der die einzelnen Geräte identifiziert werden können.

Beispiel: Um ein Smartphone mit einem Bluetooth-Gerät zu verbinden, scannt man die Bluetooth-Geräte in der Umgebung (etwa unter *Einstellungen->Bluetooth*) und koppelt dann ggf das Gerät mit dem Smartphone. Der Verbindungsauflauf erfolgt dann immer wieder automatisch, sobald Smartphone und Gerät in Reichweite zueinander sind.

Aufgaben:

1. Gib Beispiele für Bluetooth-Verbindungen an.
2. Was bedeutet „lizenzenfreies Band“ bzw. „zulassungsfrei“?

Wir wollen eine Bluetooth-Verbindung zwischen einem Arduino-Board und einem Smartphone aufbauen, um dann mit Hilfe einer APP das Arduino-Board zu steuern. Weil wir die APP mit Hilfe des AppInventors2 (AI2) von Google programmieren wollen, benötigen wir ein Google-Konto und ein Android-Smartphone, auf dem dann die APP installiert wird.

Die Arbeitsgruppen sollten sich also so zusammenfinden, dass in jeder Arbeitsgruppe mindestens ein Android-Smartphone zur Verfügung steht.

Die APP wird zweckmäßigerweise am Computer entwickelt. Dazu muss der AppInventor2 auf dem Rechner installiert sein. Die Programmierung erfolgt dann im Browser Fenster.

Die Übertragung des Programms (APP) auf das Smartphone erfolgt am einfachsten durch einen QR-Code-Reader (ansonsten durch Synchronisierung des Google-Kontos). Deshalb sollte auf dem Smartphone eine derartige APP installiert sein.

Aufgabe: Zuerst soll eine LED per APP an- und ausgeschaltet werden:

Neben dem Bluetooth-Modul wird wie üblich eine LED angeschlossen.

Zum Bluetooth-Modul (hier: HC-05-Modul) des Arduino-Boards:

Verkabelung:

VCC	→ 5 V
GND	→ GND
TX	→ RX // also über Kreuz anschließen!
RX	→ TX // rx: receive (empfangen) ; tx: transmit (senden)

Die Anschlüsse KEY und STATE werden nicht verwendet.

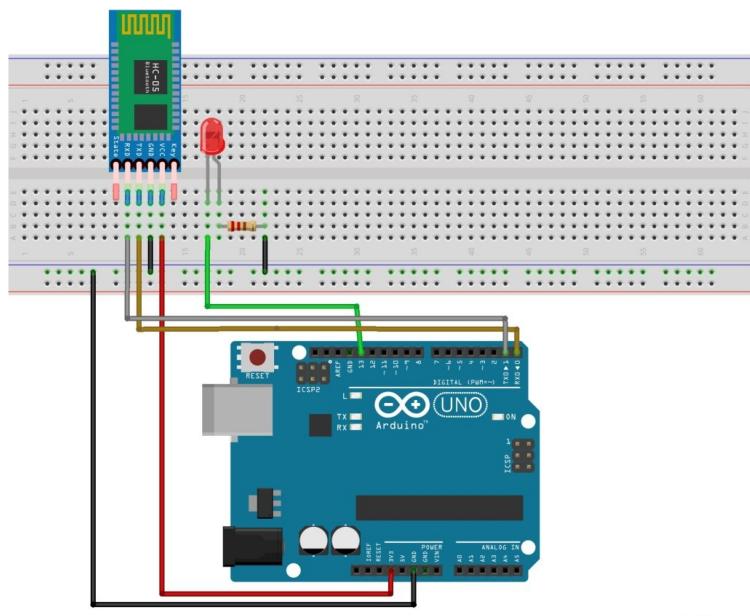
Es wird die Bibliothek *SoftwareSerial.h* eingebunden.

```
#include <SoftwareSerial.h>
```

```
SoftwareSerial BTSerial(0,1); // RX.TX Festlegung der Kanäle
```

```
int ledPin = 9;
int status=0 ;

void setup() {
  pinMode(ledPin, OUTPUT);
  BTSerial.begin(9600);
}
void loop() {
  if (BTSerial.available() > 0) {
    status = BTSerial.read();
    Serial.println(status);
  }
  analogWrite(ledPin, status);
  if (status == 0) {
    digitalWrite(ledPin, LOW);
  }
  if (status == 255) {
    digitalWrite(ledPin, HIGH);
  }
  delay(10);
}
```



Die kursiv gedruckten Zeilen sind nicht unbedingt erforderlich. Fügt man diese jedoch ein, können mit Hilfe des Sliders beliebige Werte zwischen 0 und 255 übertragen werden (Slider

entsprechend in der APP einstellen). Mit dem Anweisung `<analogWrite(ledPin, status);>` kann man dadurch die Helligkeit der LED steuern.

Achtung ! Beim Hochladen auf den Mikrocontroller muss das Bluetooth Modul herausgenommen werden. Sonst erscheint die Fehlermeldung, dass der Code nicht hochgeladen werden kann. Nach dem Hochladen kann man das Modul wieder einsetzen.

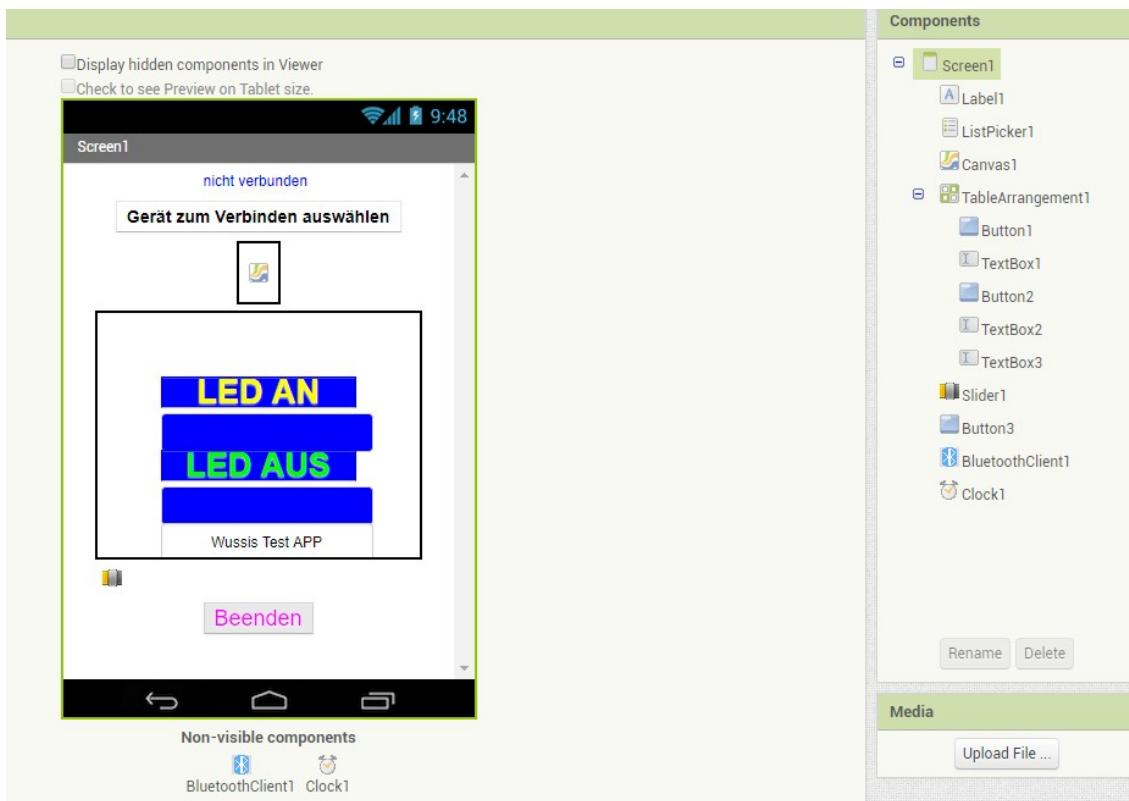
Am Bluetooth Modul HC-05 sollte eine rote LED, nachdem es wieder eingesteckt wurde, schnell blinken. Das bedeutet, dass es bereit ist sich mit einem anderen Gerät zu verbinden. Also kann nun das Bluetooth Modul mit dem Smartphone verbunden werden. Dazu sucht man in den Bluetooth Einstellungen des Smartphones nach Geräten. Das Bluetooth Modul sollte unter dem Namen HC-05 gefunden werden. Dann kann man sich mit dem Gerät verbinden. Dazu wird nach einem PIN gefragt. Dieser ist meistens als 1234 voreingestellt.

Achtung ! Beim Verbinden darauf achten, dass jeweils nur eine Gruppe einen Verbindungsauflauf herstellt, weil sonst am Smartphone mehrere Module gleichzeitig angezeigt werden (mit verschiedenen MAC-Nummern).

Nun zur Programmierung der APP.

Im APP-Inventor beginnt man ein neues Projekt.
Folgende Komponenten können hinzugefügt werden:

Mit
der



Komponente ListPicker werden auf der Canvas (Leinwand) die möglichen Bluetooth-Geräte angezeigt. Man kann dann ein Gerät selektieren – natürlich in unserem Fall HC-05.

Nach der Selektion sind das Smartphone und das HC05-Modul gekoppelt. Es können Daten übertragen werden. In unserem Fall entweder 0 oder 255. Das Arduino-Board reagiert dann auf diese Eingaben.

Nachfolgend erkennt man die Programmierung.

```

when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames

when ListPicker1 .AfterPicking
do if call BluetoothClient1 .Connect
  address ListPicker1 . Selection
then set Label1 . TextColor to green
  set Label1 . Text to "Verbunden"
else set Label1 . TextColor to red
  set Label1 . Text to "Nicht verbunden"

when Button1 .Click
do call BluetoothClient1 .Send1ByteNumber
  number 255
  set Slider1 . ThumbPosition to 255

when Button2 .Click
do call BluetoothClient1 .Send1ByteNumber
  number 0
  set Slider1 . ThumbPosition to 0

when Slider1 .PositionChanged
  thumbPosition
do call BluetoothClient1 .Send1ByteNumber
  number round Slider1 . ThumbPosition

when Button3 .Click
do close application

```

Weitere Projekte:

1. Bedarfsampel: Wenn in der APP auf einen Knopf gedrückt wird, schaltet die Ampel auf Grün. Anderer Knopf: Blinklicht erzeugen.
2. Garagentor (über Servo) mittels Knopfdruck betätigen (öffnen oder schließen).
3. Daten übertragen, z.B. Entfernungsmessung auf dem Smartphone anzeigen.