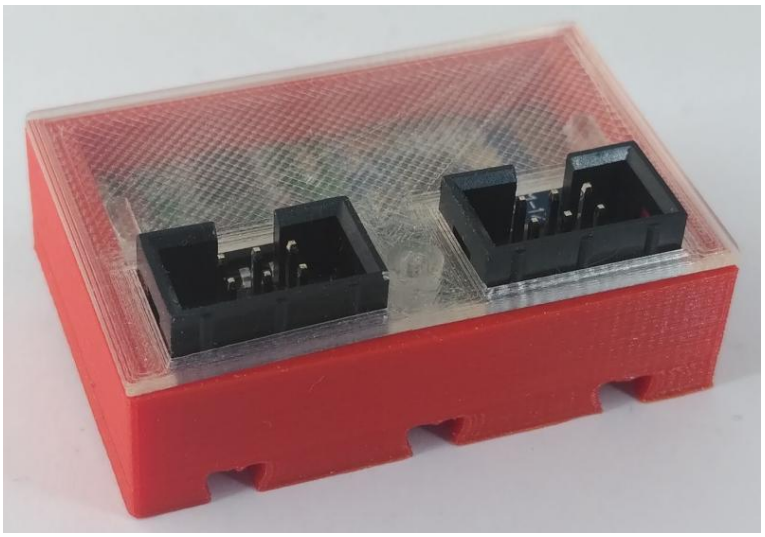
ftDuino-Bluetooth-Adapter

Bluetooth-Low-Energy-Adapter für den ftDuino



Anleitung

Dr.-Ing. Till Harbaum

8. Januar 2021

Projekt-Homepage: http://ftduino.de/blue
Forum: https://forum.ftcommunity.de/

Inhaltsverzeichnis

Kapitel 1

Einleitung

Der ftDuino ist eine Bluetooth-Erweiterung für den ftDuino. Der ftDuino (siehe http://ftduino.de) ist ein Arduino-kompatibler Controller für das fischertechnik-Konstruktionsbaukastensystem.

Der ftDuino verfügt aber Werk über eine USB-Schnittstelle zur kabelgebundenen Verbindung zu vorrangig einem PC. Der ftDuino-Bluetooth-Adapter erweitert den ftDuino um eine drahtlose Bluetooth-Schnittstelle, über die der ftDuino kabellos vor allem mit Smartdevices gekoppelt werden kann. Typische Anwendung ist die drahtlose Fernsteuerung eines fischertechnik-Modells mit Hilfe einer App auf Apple-IOS- oder Google-Android-basierten Smartdevices.

Der ftDuino-Bluetooth-Adapter basiert auf Bluetooth-Low-Energy (BLE) und ist damit primär mit Mobilgeräten kompatibel. Diverse Apps aus den gängigen App-Stores für Mobilgeräte (Apple-Store, Google-Play-Store) sind mit dem ftDuino-Bluetooth-Adapter kompatibel.

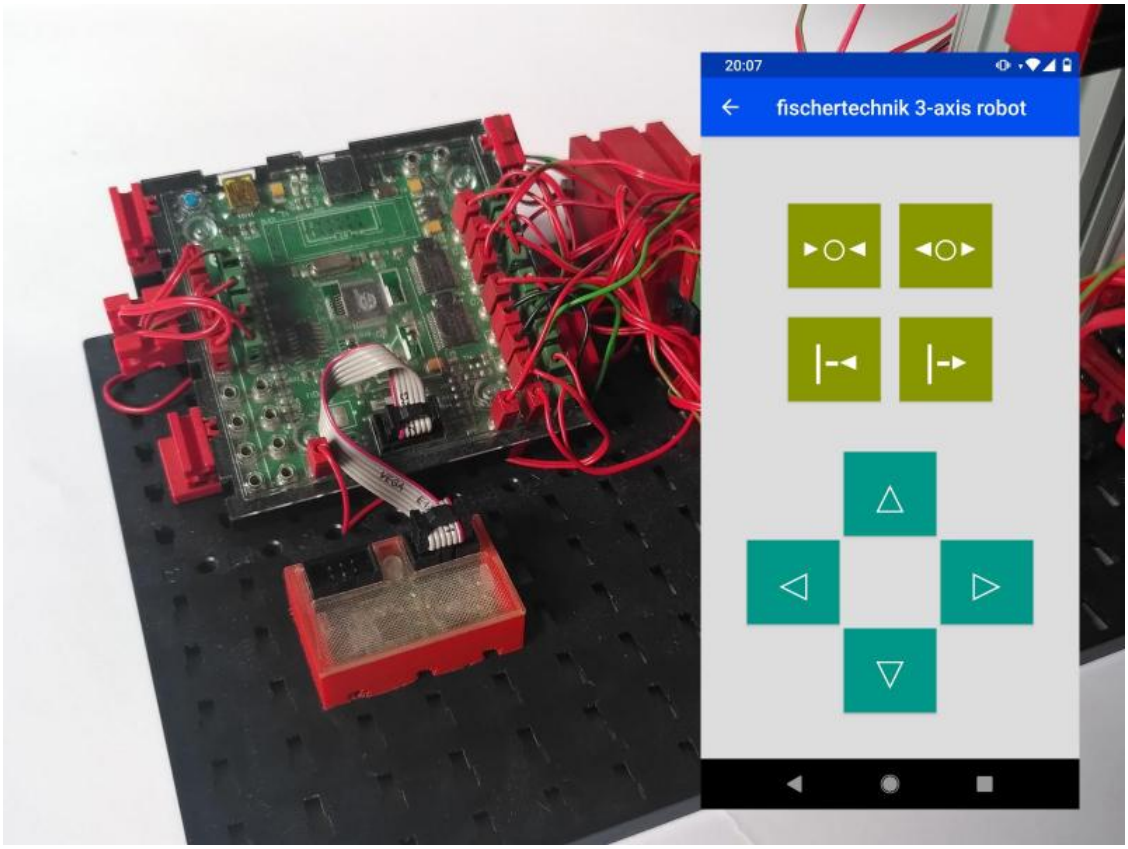


Abbildung 1.1: ftDuino-Bluetooth-Adapter und ftDuino im Robotermodell

Kapitel 2

Der Adapter

Der ftDuino-Bluetooth-Adapter wird als Fertiggerät geliefert. Er wird an den I^2C-Erweiterungsanschluss des ftDuino angeschlossen und stellt seinerseits einen weiteren I^2C-Anschluss für die Verbindung weiterer I^2C-Erweiterungen zur Verfügung. Beide I^2C-Anschlüsse des ftDuino-Bluetooth-Adapters sind gleichberechtigt und können beliebig zum Anschluss an den ftDuino oder andere Erweiterungen verwendet werden.

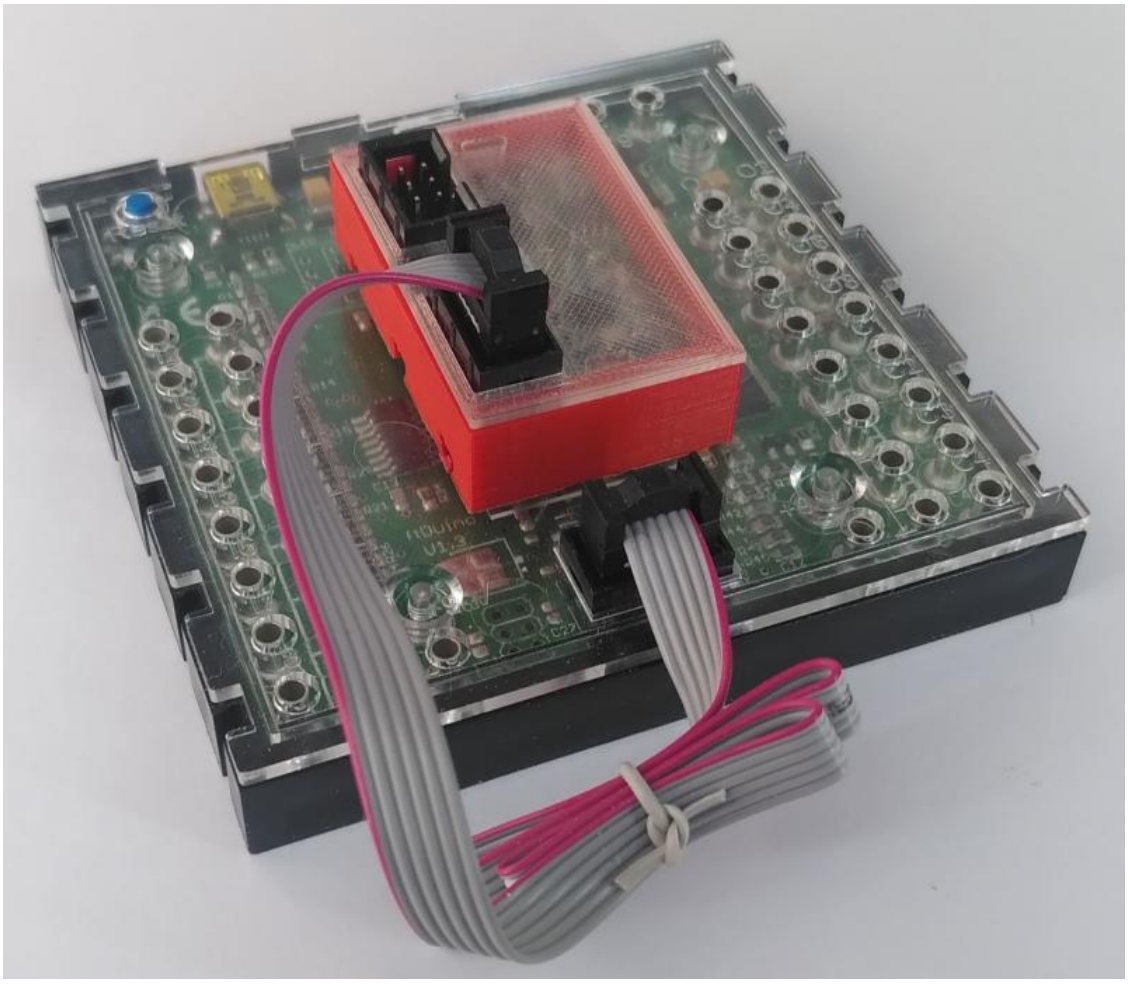


Abbildung 2.1: ftDuino-Bluetooth-Adapter am ftDuino

Der ftDuino-Bluetooth-Adapter wird inklusive fischertechnik-kompatiblem Gehäuse und passendem Anschlusskabel für den ftDuino ausgeliefert. Der Adapter kann mit Hilfe zweier fischertechnik-Federnocken sowohl direkt auf dem ftDuino befestigt werden. Er kann aber auch mit Hilfe des 25cm langen Kabels an beliebiger anderer Stelle im Modell montiert werden.

2.1 Bluetooth-Chip

Der ftDuino-Bluetooth-Adapter basiert auf dem HM-11-Bluetooth-Modul. Das HM-11-Modul ist wiederum eine mechanisch kleinere Variante des bekannten HM-10-Moduls.

Das HM-10 ist in der Arduino-Community weit verbreitet und wird von zahlreichen Projekten verwendet. Passende Software für gängige Smart-Device-Betriebssysteme sind in den App-Stores unter dem Stichwort "HM-10" zu finden.

Die meiste für das HM-10 geschriebene Software lässt sich auch mit dem HM-11 und damit mit dem ftDuino-Bluetooth-Adapter nutzen (siehe Kapitel 3). Die meisten für den Arduino und das HM-10-Modul geschriebenen Sketches lassen sich leicht an den ftDuino und den ftDuino-Bluetooth-Adapter anpassen (siehe Kapitel 4).

Für einige Smartdevice-Apps liefert die ftDuino-Einrichtung in der Arduino-IDE unter `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth` bereits einige Beispiele mit.

2.2 Leuchtdiode

Der Adapter selbst verfügt über keine Bedienelemente, sondern lediglich über eine rote Leuchtdiode, über die der Betriebszustand angezeigt wird. Sie blinkt einmal pro Sekunde, wenn das Bluetooth-Modul empfangsbereit ist und sie leuchtet durchgängig, wenn eine Bluetooth-Verbindung zu einem anderen Gerät besteht.

Kapitel 3

Software

Eine Nutzung des ftDuino-Bluetooth-Adapters am ftDuino erfordert einen passenden Sketch und eine dazugehörige Smartdevice-App. Beides kann man selbst entwickeln oder aber auf vorgefertigte Lösungen zurückgreifen. Dieses Kapitel stellt einige solche Lösungen vor. Weitere und ähnliche Projekte finden sich im Internet und sind in der Regel leicht an den ftDuino-Bluetooth-Adapter anpassbar.

Die hier dargestellten Beispiele nutzen alle die gleichen Art der softwareseitigen Einbindung des ftDuino-Bluetooth-Adapters in den jeweiligen Beispielsketch. Diese Einbindung überprüft bei Sketchstart die Verbindung zum Bluetooth-Modul und zeigt ein Problem durch schnelles Blinken der ftDuino-internen Leuchtdiode an. Die Leuchtdiode im ftDuino-Bluetooth-Adapter zeigt dagegen durch langsames Blinken die Verbindungsbereitschaft an.

Von den im Folgenden dargestellten Lösungen bietet ArduinoBlue den einfachsten Einstieg und eignet sich für die ersten Schritte mit dem ftDuino-Bluetooth-Adapter.

3.1 ArduinoBlue

ArduinoBlue (https://sites.google.com/stonybrook.edu/arduinoble/home) ist eine kostenlose und werbefreie Smartdevice-App, die für Apple-IOS und Android verfügbar ist sowie einer Arduino-Bibliothek, die über den Bibliotheks-Verwalter in der Arduino-IDE installiert werden kann.

ArduinoBlue ist für die Verwendung mit dem HM-10-Modul entwickelt worden und funktioniert damit auch uneingeschränkt mit dem ftDuino-Bluetooth-Adapter verwendbar.

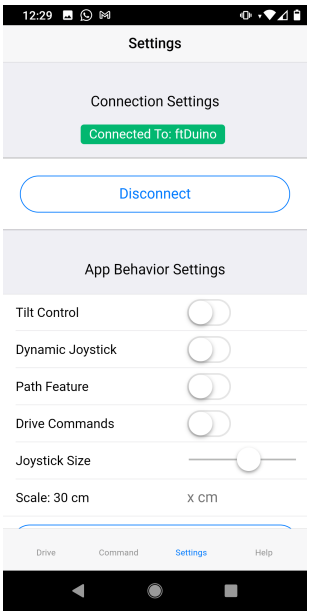
Unter `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ ArduinoBlue` findet sich in der Arduino-IDE ein fertig an den ftDuino-Bluetooth-Adapter angepasstes Beispiel. Es kann unverändert auf den ftDuino geladen werden. Das Beispiel bildet die zwei Achsen des On-Screen-Joysticks in ArduinoBlue auf zwei Motoren an den ftDuino-Ausgängen `M1` und `M2` ab.

ArduinoBlue eignet sich mit dem On-Screen-Joystick vor allem zur Steuerung von Fahrzeugen. Es können aber auch benutzerdefinierte Schaltflächen ("Buttons") und Schieberegler ("Slider") eingerichtet werden, über die eine universellere Nutzung möglich ist.

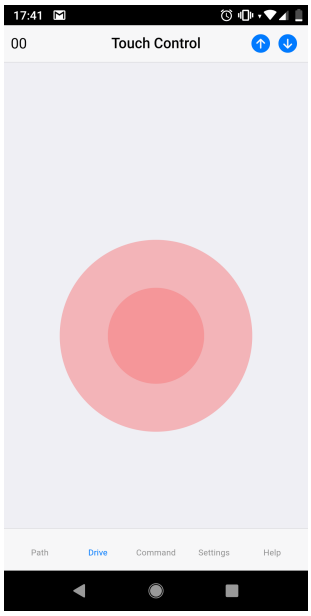
3.2 1Sheeld

OneSheeld (https://www.1sheeld.com/) besteht wie ArduinoBlue aus einer für IOS und Android kostenlos verfügbaren App sowie einer dazugehörigen Arduino-Bibliothek. OneSheeld bietet dem Arduino "virtuelle Shields" in Form von Tastenfeldern, Anzeigen etc. auf dem Smartdevice-Bildschirm. Diese sollen reale Tastenfelder, Anzeigen u.ä. ersetzen.

OneSheeld vertreibt dazu ein spezielles Sheeld, das auf den Arduino aufgesteckt wird und einen Kommunikationsprozessor sowie den Bluetooth-Chip enthält.



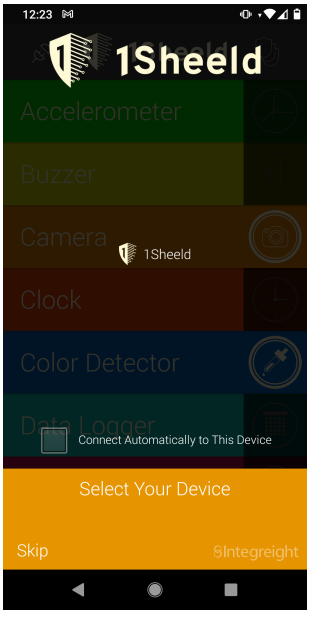
(a) Verbindung



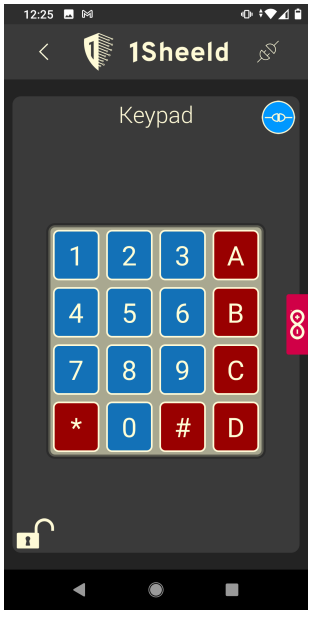
(b) On-Screen-Joystick

Abbildung 3.1: ArduinoBlue

Im ftDuino-Beispielsketch zum OneSheeld unter `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ 1Sheeld` übernimmt eine Arduino-Bibliothek die Rolle des Kommunikationsprozessors und der ftDuino-Bluetooth-Adapter die Rolle des Bluetooth-Moduls. Damit die OneSheeld-App den ftDuino-Bluetooth-Adapter erkennt, muss der Gerätename mit der Zeichenfolge "1Sheeld" beginnen. Der Name lässt sich z.B. mit Hilfe des Bluetooth-Config-Sketches (siehe 3.5) entsprechend einstellen.



(a) Verbindung



(b) Virtuelles Keypad

Abbildung 3.2: OneSheeld

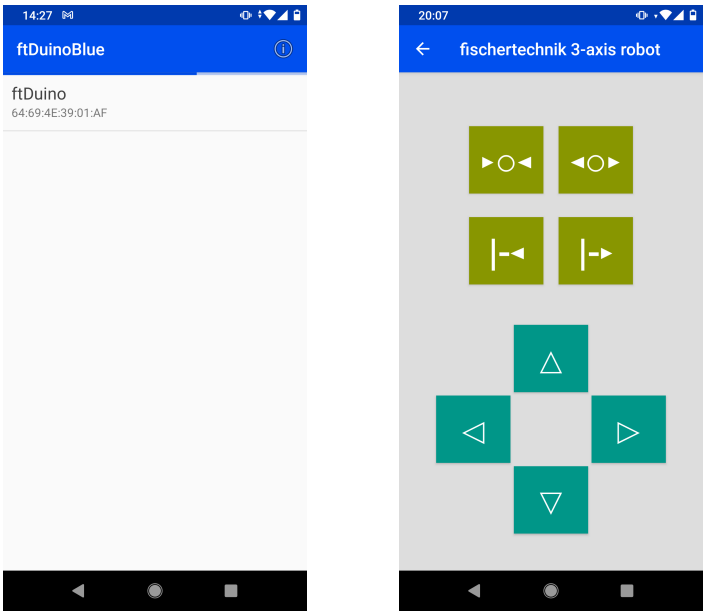
Der Beispiel-Sketch verwendet des virtuelle "Keypad"-Shield der OneSheeld-App, um die Ausgänge O1 bis O8 des ftDuino anzusteuern. Ein Druck auf die Tastenfelder 1 bis 8 des Keypads schaltet die entsprechenden Ausgänge des ftDuino ein bzw. wieder aus.

3.3 ftDuinoBlue

ArduinoBlue und OneSheeld ist gemeinsam, dass man passende Einstellungen in der Smartdevice-App und im ftDuino-Sketch vornehmen muss. Das hat u.a. den Nachteil, dass man nicht einfach zwischen mehreren Modellen wechseln kann, sondern bei jedem Wechsel in der App die entsprechenden Anpassungen vornehmen muss. Außerdem ist man in der Regel auf einige wenige von der App vorgegebene Steuerelemente beschränkt.

ftDuinoBlue (http://ftduino.de/blue) ist der Versuch, diese Beschränkungen zu umgehen. Dazu findet in der bisher nur für Android verfügbaren App keinerlei Einstellungen statt. Stattdessen gibt es im Sketch erweiterte Einstellungen, die neben der Funktion des ftDuino auch die gewünschte Benutzeroberfläche auf dem Handy beschreiben. Sobald sich die App dann mit dem Modell verbindet liest es die gewünschte Benutzeroberfläche aus und stellt sie entsprechend dar.

Die Vorteile dieser Lösung sind, dass man beliebig zwischen Modellen wechseln kann und sich die Benutzeroberfläche auf dem Smartdevice immer entsprechend anpasst. Außerdem lassen sich die Benutzeroberflächen sehr flexibel aus einer Vielzahl von Bedienelementen (Schalter, Knöpfe, Schieber, Texte, ...) zusammenstellen.



(a) Verbindung (b) 3-Achs-Roboter

Abbildung 3.3: ftDuinoBlue

Der Nachteil ist die etwas aufwändigere Erstellung des Sketches, in dem man neben der eigentlichen Funktion des Modells auch die Benutzeroberfläche hinterlegen muss. Mehr Erklärungen dazu finden sich unter http://ftduino.de/blue.

Ein einfacher Beispielsketch findet sich unter `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ ftduinoblue_demo`. Ein komplexes Beispiel zur Steuerung des 3-Achs-Roboters aus dem fischertechnik Automation-Robots-Baukasten findet sich unter `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ txt_automation_3axis`.

3.4 Serial Bluetooth Terminal

Neben den spezialisierten Apps gibt es auch allgemeine Kommunikationsprogramme für Smartdevices. Die Android-App "Serial Bluetooth Terminal" ist ein Beispiel dafür. Vergleichbare Programme gibt es auch für Apple-IOS.

Diese sogenannten Terminalprogramme stellen wenig eigene Funktionalität bereit, stattdessen stellen sie alle via Bluetooth empfangenen Zeichen direkt auf dem Bildschirm des Smartdevice dar. In Gegenrichtung werden alle am Smartdevice gemachten (Onscreen-)Tastatureingaben über Bluetooth versendet. Vor allem für einfache Tests und Experimente ist diese direkte Kommunikation gut geeignet.

Nutzt man im Sketch eine einfache Text-Kommando-orientierte Steuerung, so lassen sich auch mit einem Terminalprogramme eigene Modelle steuern.

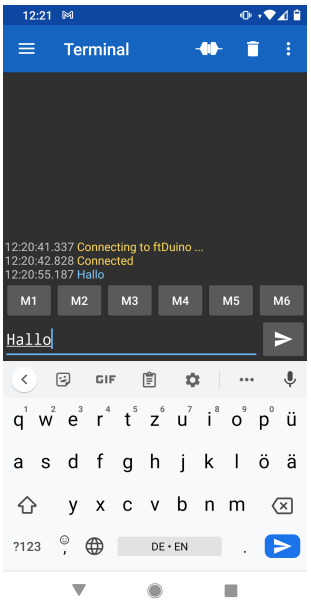


Abbildung 3.4: Serial Terminal unter Android

Als Gegenseite bietet sich der Sketch `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ Bridge` an. Er stellt ebenfalls kaum eigene Funktion bereit, sondern veranlasst den ftDuino alle via Bluetooth empfangenen Zeichen über USB an den angeschlossenen PC zu senden. Um Gegenzug sendet dieser Sketch alle vom PC über USB empfangen Daten über Bluetooth.

Nutzt man PC-seitig z.B. den "Seriellen Monitor" der Arduino-IDE, so werden alle Texteingaben dort zunächst per USB zum ftDuino und dann über Bluetooth zur Terminal-App auf dem Smartdevice übertragen. Am PC eingegeben Texte erscheinen also auf dem Bildschirm des Smartdevices. In der Gegenrichtung funktioniert es ähnlich und alle am Smartdevice eingegebenen Zeichen erscheinen als Ausgabe im Textfenster des Seriellen Monitors.

3.5 Bluetooth-Konfigurations-Werkzeug

Im Inneren des ftDuino-Bluetooth-Adapter steckt ein HM-11-Bluetooth-Modul. Dieses Modul hat weitreichende Einstellungsmöglichkeiten. Die meisten davon sind für die Verwendung am ftDuino von geringer Bedeutung. Eine interessante Möglichkeit ist es aber, den Gerätenamen ändern zu können. Vor allem wenn man über mehrere ftDuinos und passende ftDuino-Bluetooth-Adapter verfügt kann man die Module durch Umbenennung unterscheidbar machen.

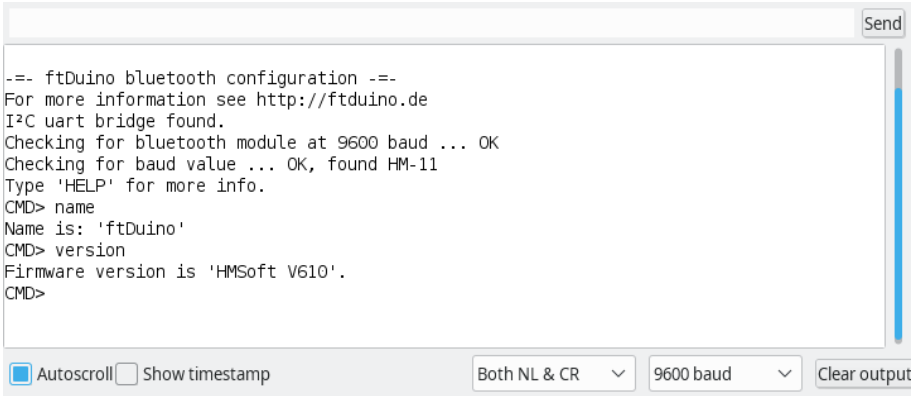


Abbildung 3.5: Bluetooth-Konfigurations-Werkzeug

Es ist aber auch möglich, das Modul durch das manuelle Senden von ungeschickt gewählten Befehlen z.B. bei Einsatz des Bridge-Sketches umzukonfigurieren, dass das Modul nicht mehr wie gewünscht kommuniziert.

In diesen Fällen kommt das Bluetooth-Konfigurations-Werkzeug `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ bluetooth_config` zum Einsatz. Seine wichtigste Funktion ist es, eine umkonfigurierte Kommunikationsschnittstelle wieder in den Ausgangszustand zu versetzen. Dies passiert automatisch beim Start des Sketches.

Zusätzlich bietet dieser Sketch die Möglichkeit, den ftDuino-Bluetooth-Adapter umzubenennen.

Kapitel 4

Programmierung

Der ftDuino-Bluetooth-Adapter folgt der fischertechnik-Philosophie eines flexiblen Baukastensystems. Er stellt keine "Black-Box" dar, sondern kann bei Interesse selbst programmiert werden. Dieses Kapitel liefert das dafür Notwendige Know-How.

4.1 UART oder I²C

Zur Verbindung verschiedener in Computern zum Einsatz kommenden Bauelemente gibt es zahlreiche sogenannte Schnittstellen. Eine Schnittstelle beschreibt z.B. wie viele elektrische Verbindungen zur Kommunikation benötigt werden und in welcher zeitlichen Abfolge und in welche Richtung Signale übertragen werden. Der ftDuino bietet auf seinem 6-poligen Erweiterungsstecker eine sogenannte I²C-Schnittstelle. Diese Schnittstelle ist weit verbreitet wenn es um den einfachen Anschluss einfacher Sensorkomponenten geht. Über die I²C-Schnittstelle würde ursprünglich zum Einsatz in Fernsehern entwickelt und über sie lassen sich mit nur zwei elektrischen Signalen gleich eine Vielzahl von Sensoren ansprechen. Ebenfalls recht verbreitet ist die UART-Schnittstelle. Sie hat ihre Wurzeln u.a. in der seriellen Schnittstelle früher PCs. Auch sie kommt mit zwei elektrischen Verbindungen aus, eignet sich aber nur zum Anschluss eines weiteren Gerätes. Viele gängige Bluetooth-Chips und -module verwenden einen UART-Anschluss. Der ftDuino-Bluetooth-Adapter enthält daher neben dem Bluetooth-Modul einen Brückenbaustein, der zwischen der UART-Schnittstelle des Bluetooth-Moduls und der I²C-Schnittstelle übersetzt.

4.2 I²C-Schnittstelle

Jedes I²C-Gerät verfügt über eine sogenannte Adresse, über das es angesprochen wird und das es von anderen Geräten an der gleichen I²C-Schnittstelle unterscheidbar macht. Der mit der ftDuino-Installation ausgelieferte Sketch `Datei ▷ Beispiele ▷ FtduinoSimple ▷ I2C ▷ I2cScanner` kann die I²C-Schnittstelle des ftDuino nach angeschlossenen I²C-Geräten absuchen.

Abbildung 4.1: I2cScanner-Ausgabe bei angeschlossenem ftDuino-Bluetooth-Adapter

Der ftDuino-Bluetooth-Adapter reagiert auf Adresse 5. Diese Adresse ist fest im Gerät eingespeichert und kann nicht verändert werden.

Der I²C-nach-UART-Umsetzer im ftDuino-Bluetooth-Adapter unterstützt alle möglichen Datenraten des HM-11-Moduls zwischen 1200 bit/s und 230400 bit/s. Damit ist im Gegensatz zu vielen einfacheren Arduino-Lösungen sichergestellt, dass das HM-11-Modul immer erreichbar bleibt, auch wenn die ab Werk eingestellte Datenrate vom Anwender verstellt wird. Zusätzlich kann auf diesem Weg das `KEY`-Signal am HM-11 betätigt werden, mit dessen Hilfe sich Teile des Moduls bei einer Fehlkonfiguration zurücksetzen lassen.

All diese Funktionen werden in den im vorigen Kapitel erwähnten Beispielen durch die in jedem Sketch hinterlegten Dateien `I2cSerialBt.cpp` und `I2cSerialBt.h` umgesetzt. Der in diesen beiden Dateien enthaltene Code kontrolliert die I²C-Schnittstelle des ftDuino und bietet dem Sketch seinerseits eine Arduino-übliche sogenannte Stream-Schnittstelle. Die Stream-Schnittstelle lässt sich in Sketches überall dort einsetzen, wo sonst eine serielle UART-Schnittstelle (Serial, SoftSerial, ...) zum Einsatz kommt. Auf diese Weise lassen sich fast alle bestehenden Sketches für den Arduino und das dort verbreitete HM-10-Modul an den ftDuino und das ftb anpasst. Die Code-Beispiele aus Kapitel 3 lassen sich dafür als Vorlage nutzen.

Das ebenfalls dort näher erklärte Beispiel `Datei ▷ Beispiele ▷ Ftduino ▷ Bluetooth ▷ ftDuino-Bluetooth ▷ bluetooth_config` macht maximalen Gebrauch der `I2cSerialBt`-Schnittstelle und kann damit automatisiert auch einen umkonfigurierten ftDuino-Bluetooth-Adapter wieder in einen definierten Zustand versetzen.

Weitere Details zur Programmierung des HM-11-Moduls finden sich in den Datenblättern unter http://www.jnhuamao.cn. In den allermeisten Fällen ist aber keine Umkonfiguration durch den Benutzer nötig. Die Werkseinstellungen erlauben direkt die Kopplung mit gängigen Smartdevices von Apple und Android.