

Praktikum Analog- und Digitaltechnik

Versuch D3

Bluetooth-steuerung mit Arduino

Inhalt dieses Versuches:

- Installation und Konfiguration von Software Arduino IDE
- Ansteuerung einer LED über Bluetooth

Vorarbeiten:

Bevor Sie zum Praktikum kommen, müssen Sie Allgemeinwissen zur Mikrocontrollern haben. Dafür haben wir eine Begriffsliste vorbereitet, die Sie im Internet oder Bücher nachschlagen sollen.

- Mikrocontroller
- I/O-Ports
- SPI
- I2C
- asynchrone serielle Schnittstelle
- Flash-Speicher
- Arduino IDE
- Lua
- Python
- COM-Port
- Firmware
- Sketch
- Library
- Internet der Dinge

1. Einleitung

Ein Arduino ist ein Microcontroller, der in dem Praktikum Digitaltechnik vorgestellt werden soll. Man kann viel zu den Arduino schreiben. In dem Digitaltechnik Praktikum wollen wir uns nur auf das Wesentliche beschränken und lieber ein praktisches

Anwendungsbeispiel zeigen. Zudem gibt es schon viele Internet-Seiten als auch Bücher die sehr viele Infos zu den Arduinos liefern und zum zweiten macht das Lernen einfach mehr Spaß wenn man schnell zu praktischen Ergebnissen kommt. Daher beschränken wir uns hier auf die Schritte und das Wissen, das Ihr benötigt um ein erstes erfolgreiches Projekt umzusetzen.

Zuerst einmal ein paar Worte zu den beliebten und am weitesten verbreiteten Arduino Boards (siehe Abb.2). Arduinos richten sich an Mikrocontroller-Einsteiger. Es sind fertig aufgebaute Boards mit zahlreichen Anschlüssen und aufgelöteten Buchsenleisten.

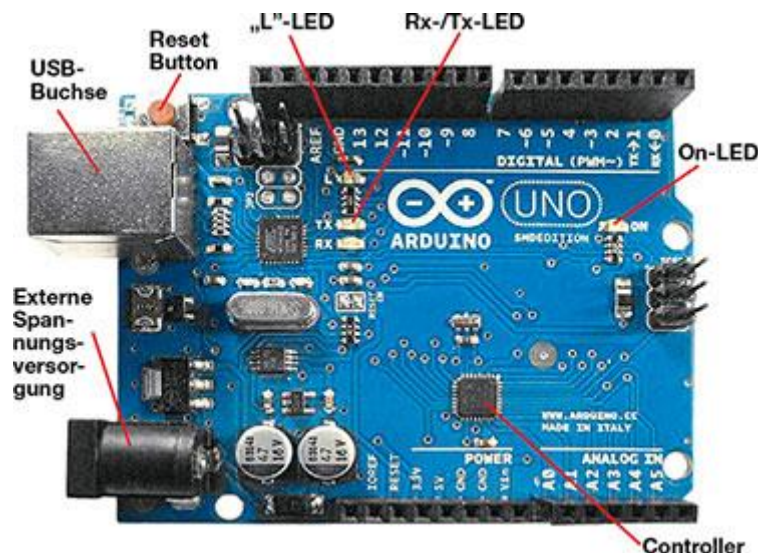


Abb. 1 Beispiel eines Arduino Mikrocontroller

Der Arduino Uno R3 ist der Standard-Arduino. In der aktuellen Version 3 besitzt der Uno 14 Digitale I/O Pins sowie 6 analoge Eingänge. Er besitzt eine Spannungsregelung und darf mit 12V über einen Hohlstecker versorgt werden, kann aber auch an USB angeschlossen werden und besitzt folgende Speicherkapazitäten:

- 32K Flash(0,5 werden vom Bootloader verwendet.)
- 2K SRAM
- 1K EEPROM
- 16 MHz (Prozessortakt)

Neben dem USB-B Anschluss verfügt der Uno noch über folgende Schnittstellen:

- SPI
- I²C
- ICSP

Die Programmierung des Boards ist sehr einfach, aber im Internet existieren einige Tutorials die etwas kompliziert wirken, weil dort gleichzeitig die Programmierung der Module per **Arduino IDE**, **Lua** oder per **Phyton** erklärt werden. Das kann für Einsteiger sehr schwer zu unterscheiden sein.

Möglich ist die Board per Lua oder sogar Phyton zu programmieren, aber dazu muss erst eine Firmware mit einem Compiler in die Boards geladen werden. Das kostet sehr viel Speicherplatz, der Platz für eigene Anwendungen wird dadurch erheblich reduziert. Zudem ist die Lua-Programmierung nicht sonderlich einfach.

Die Programme die man selber schreibt werden Sketch genannt und können per Arduino IDE (Software) kompiliert und hochgeladen werden (siehe Abb.2). Dabei wird das schon vorhanden Sketch auf dem Board jeweils überschrieben. Das sogenannte Flashen (überschreiben) wird durch USB Kabel realisiert.

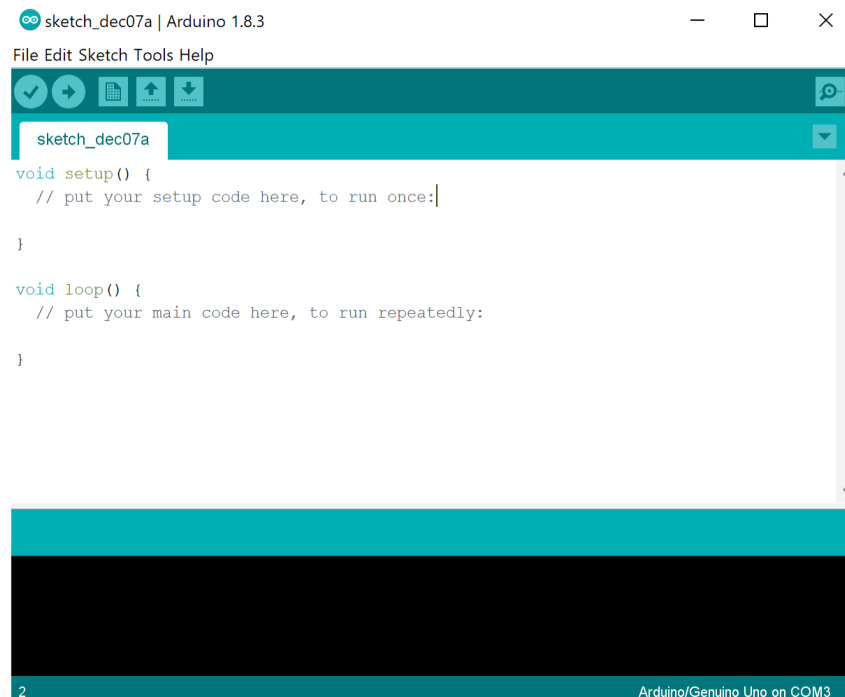


Abb. 2 Oberfläche des Softwares Arduino IDE.

2. Ziel des Praktikums

In dem Praktikum sollen Hardware miteinander verbunden werden, um eine LED über Bluetooth zu steuern. Dazu soll ein Mikrocontoller Arduino-Board, Bluetooth Modul und eine Android Anwendung verwendet werden. Das Arduino-Board wird mit dem Bluetooth Modul, Stromversorgung und einer LED verkabelt und konfiguriert. Für die Programmierung wird ein Arduino IDE-Software verwendet. Das Prinzip der Ansteuerung wird in Abbildung 1 dargestellt. Hier arbeitet der HC 05/06 Bluetooth Modul mit serieller Kommunikation. Die Android App sendet serielle Daten per Bluetooth an das Arduino. Das auf dem Arduino hochgeladene Programm überprüft die ganze Zeit, ob das Bluetooth Modul daten versendet hat. Wenn die empfangenen Daten 1 (ON) sind, leuchtet die LED auf. Die LED erlischt, wenn die empfangenen Daten 0 (OFF) sind.

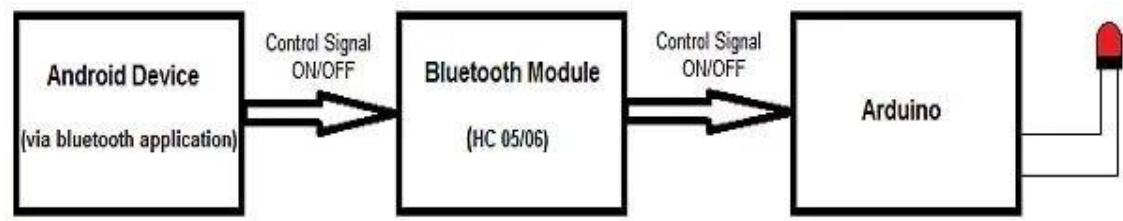


Abb. 1 Prinzip der Ansteuerung

2 Versuchsdurchführung

In diesem Praktikum werden wir die Konfiguration und installation unter Windows vornehmen. Zudem verwenden wir zur Programmierung ausschließlich C, also die Arduino IDE sprache.

2.1 Die Arduino IDE Software installieren und konfigurieren

Die IDE kann man kostenlos über folgende Seite herunterladen:

<https://www.arduino.cc/en/main/software>

In der Regel sollte man immer die neueste Version verwenden. Man kann die Arduino Software wahlweise als ZIP-Datei herunterladen oder eine installierbare Version wählen. Falls Ihr die ZIP-Datei wählt, dann reicht es wenn man diese in ein beliebiges Unterverzeichnis entpackt. Ihr könnt danach die Arduino IDE Software gleich starten, es muss dann nichts mehr installiert werden.

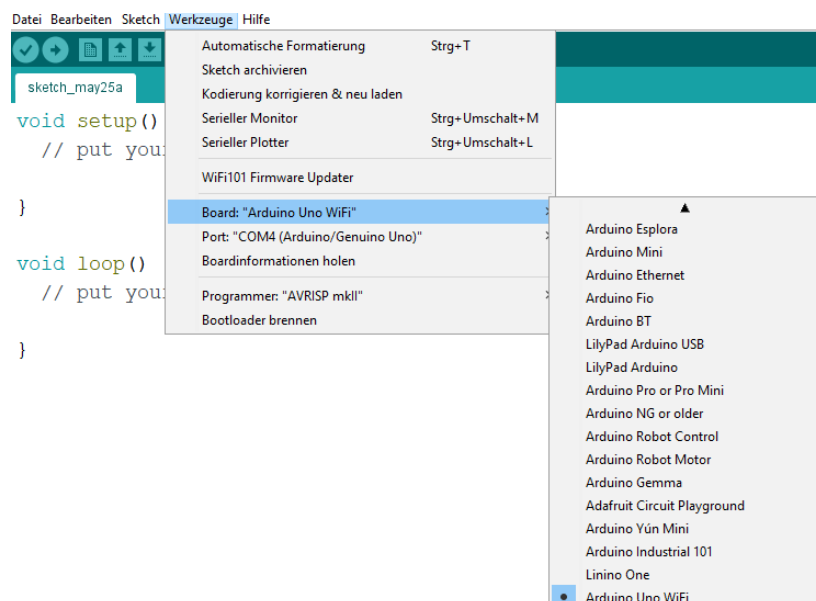


Abb. 3 Board-Typ in Arduino IDE auswählen

Die Arduino IDE unterstützt nach der Installation eine ganze Reihe von verschiedenen Arduino-Boards. Da wir aber Arduino Board programmieren möchten, müssen wir keine Boardpakete mehr dazu hinzufügen.

Um mit dem Arduino Uno zu kommunizieren, muss man zuerst den richtigen Board Typ auswählen. Das könnt Ihr im Menü **Werkzeuge** den Menüpunkt **Board....** machen. Hier müsst ich nach „Arduino Uno“ suchen.

Jetzt muss man der IDE nur noch mitteilen an welchem **COM-Port** das Modul angeschlossen ist, denn der USB-Treiber gibt die Daten über einen virtuellen COM-Port an die IDE weiter. Man könnte jetzt im Gerätemanager von Windows nachschauen welche COM-Ports dort bereitgestellt werden. Es kann auch dort je nach Bedarf der COM-Port verändert werden. Gewöhnlich reicht es aber wenn Ihr in der Arduino IDE über **Werkzeuge** auf den Menüpunkt **Port** geht. Gewöhnlich wird dort angezeigt welcher COM-Port jetzt bereit steht. Wenn es nur einer ist, dann ist die Auswahl ja sehr einfach.



Abb. 4 In Geräten Manager installierte Board mit zugewiesenen COM Port. Hier COM4

4. Erstes Sketch Hochladen

Jetzt sind wir soweit und können unser erstes Sketch für den Arduino ausprobieren. Dazu kann das Sketch aus den Beispielen unter **Datei** aus dem Menüpunkt **Beispiele** gewählt werden. Wir fangen mal mit einer blinkenden im Sekundentakt LED an. Öffnet den Sketch 01.Basic/Blink (siehe Abb. 5) und überträgt ihn per Klick auf den Pfeil (siehe Abb. 6) auf das Arduino Board.

Nach dem erfolgreichen Hochladen sollte eure Board im Takt von eine Sekunde blinken.

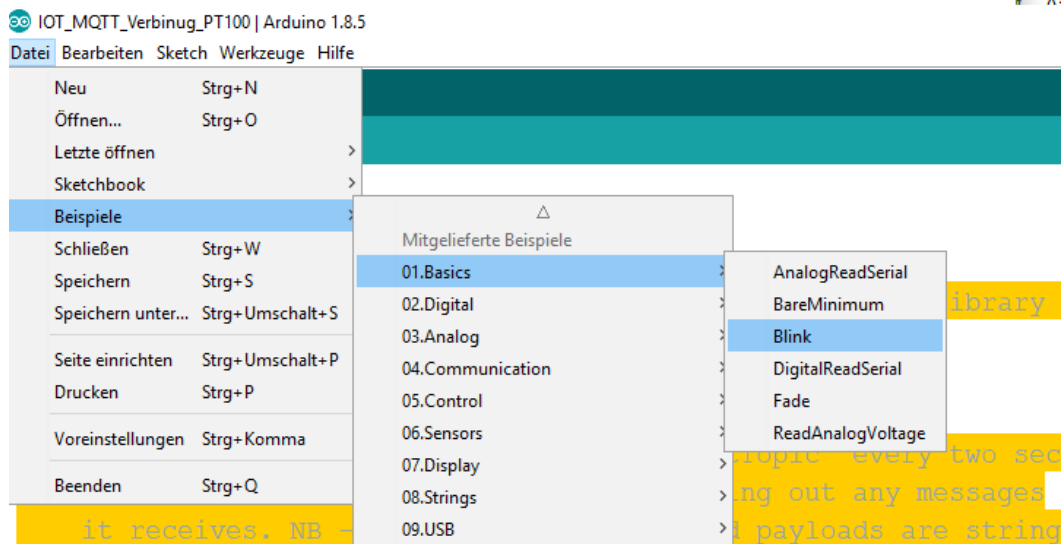


Abb. 5 Wahl eines Beispiels um die Boards-LED blinken zu lassen.

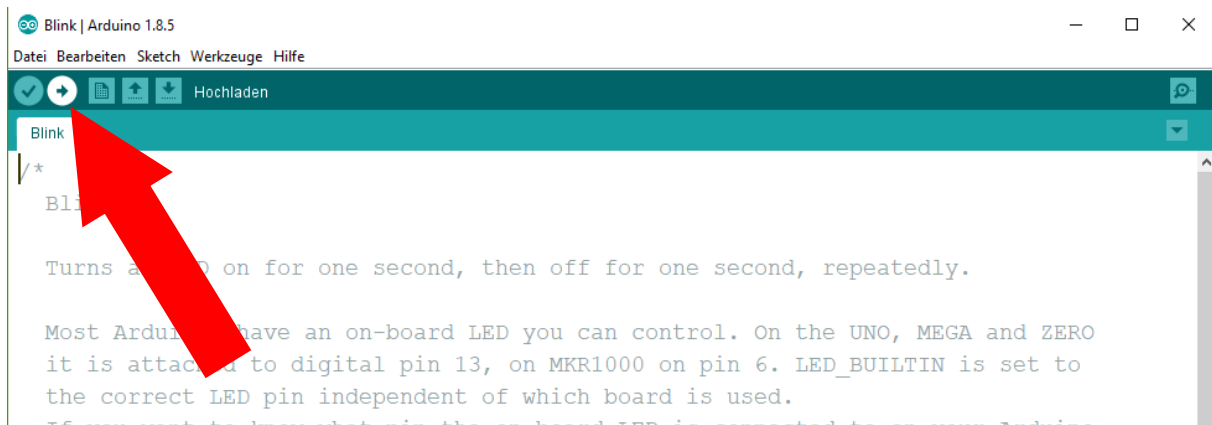
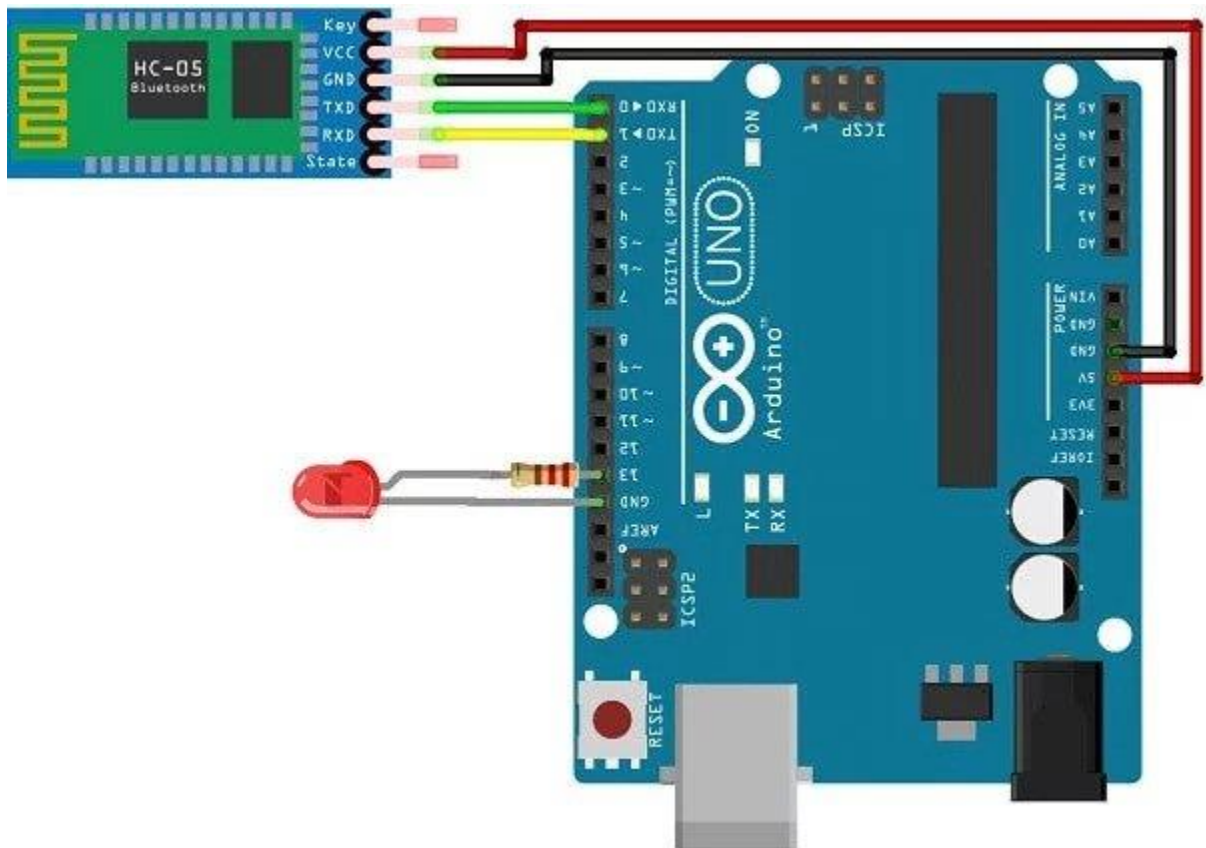


Abb. 6 Program (Sketch) compilieren und an das Board hochladen.

5 Anschließen der Arduino Bluetooth Hardware

Jetzt muss das Arduino Board mit dem Bluetooth Modul und eine LED verbunden werden. Verbinden Sie die Bauteile nach Abbildung 7. Achten Sie darauf, dass eine LED nicht ohne einen Widerstand (220Ω – $1k\Omega$) betrieben werden darf.



Arduino Pins | Bluetooth Pins

RX (Pin 0) ----> TX

TX (Pin 1) ----> RX

5V ----> VCC

GND ----> GND

Abb. 7 Verbindung des Arduino Boards mit einem Bluetooth Modul und eine LED

6 Sketch Hochladen

Nun soll ein neues Sketch erstellt und hochgeladen werden. Der Quellcode ist in Abbildung 8 gegeben und kann übernommen werden. Anschließend wird das neue Sketch über die Arduino IDE Software auf den Arduino geladen.

```
char data = 0;           //Variable for storing received data
void setup()
{
  Serial.begin(9600);     //Sets the data rate in bits per second (baud) for serial d
  pinMode(13, OUTPUT);    //Sets digital pin 13 as output pin
}
void loop()
{
  if(Serial.available() > 0) // Send data only when you receive data:
  {
    data = Serial.read();   //Read the incoming data and store it into variable data
    Serial.print(data);     //Print Value inside data in Serial monitor
    Serial.print("\n");     //New line
    if(data == '1')         //Checks whether value of data is equal to 1
      digitalWrite(13, HIGH); //If value is 1 then LED turns ON
    else if(data == '0')    //Checks whether value of data is equal to 0
      digitalWrite(13, LOW); //If value is 0 then LED turns OFF
  }
}
```

Abb. 8 Quellcode für den neuen Sketch

7 App Instalation

In diesem Praktikum soll nicht auf die programmierung einer Android App eingegangen werden. Die App gibt es zum Download auf der Laborseite.

Nach dem Download und der Installation der App, muss das Smartphone mit dem HC 05/06 Bluetooth Modul synchronisiert werden:

1. Das HC 05/06 Bluetooth einschalten (Stromversorgung über den Arduino).
2. Mit dem Smartphone nach neuen Bluetooth Geräten suchen und mit dem HC 05/06 verbinden (Möglicherweise Passwort wie 1234 oder 0000 eingeben).
3. Die LED App auf dem Smartphone installieren und öffnen (siehe Abb. 9).



Abb. 9 LED App synchronisieren

4. Die Geräte mit dem Button „paired devices“ bestätigen.
5. Das Bluetooth Module aus der Liste auswählen (HC-05/06)
6. Nach dem die Geräte erfolgreich verbunden wurden, kann die LED über die Button „ON“ und „OFF“ des Smartphonedisplays bedient werden. (siehe Abb. 10)



Abb. 10 Smartphone-Ansicht nach verbinden mit LED

Wenn die Schritte 1-6 erfolgreich durchgeführt wurden, ist die Smartphone-Steuerung fertig und kann getestet werden.

Natürlich lassen sich mit der verwendeten Technik noch zahlreiche, weit aus komplexere Versuche realisieren. Dieser Versuch soll jedoch jediglich einen Einblick in die Möglichkeiten der Mikrokontroller bieten. Welche andere Anwendungen für die

benutzte Technik fallen Ihnen noch ein? Recherchieren sie im Netz weitere Anwendungsmöglichkeiten.