

26112025---100---Freenove.Schulungsbaukasten.Arduino

Laufendes-2025-645-501.html

siehe auch Laufendes-2025-645.html

Laufendes-2025-645-501.pdf

Elf Seiten Hinweise zum Freenove-Baukasten FNK0059.

Diese Beobachtungen habe ich gemacht, als ich die Übungen ausgeführt habe.

Diese Hinweise sind eine Ergänzung zu der sehr guten Freenove-Bedienanleitung.

Die Arbeitsbefehle habe ich aus der Freenovebedienanleitung herauskopiert,

dann in das Arduino-IDE-Werkzeug eingesetzt, dann auf den Kleinrechner Arduino geschoben.

Dieser liegt dem Kasten bei und er wird auf die Grundplatte aufgesteckt.

Formal und inhaltlich braucht man sich nicht mit den Arbeitsaufträgen befassen, das kann man später erlernen. Für den ersten Durchgang reicht es, wenn man alle Übungen einmal gemacht hat und es gut geht.

Vorbemerkungen

Vorabhinweise

---> die Gleitlager des kleinen Motors habe ich mit Nähmaschinenöl geschmiert.

--->eine 9-Volt-Block-Batterie habe ich gesondert beschafft

-->alle anderen Dinge sind im Schulungskasten von Freenove enthalten

--> für das Durcharbeiten habe ich zwei Tage gebraucht, gewisse Vorkenntnisse waren schon vorhanden

-->bis auf den Infrarot-Bewegungserkennung haben alle Versuche einen vollen Erfolg gebracht.

-->die Arbeitsauftragstexte aus der Freenove-Bedienanleitung kopiert und in das Arduino-IDE-Werkzeug eingefügt und die Texte dann in den Arduino hochgeladen, das ging alles sehr gut

--->nachfolgend eine Beschreibung des Schulungskastens FNK 0059 von Freenove

--->die Bedienanleitung von Freenove ist sehr gut, hier unten weitere Ergänzungen dazu, was mit aufgefallen ist, wie ich es durchgearbeitet habe. Es ist als Ergänzung der sehr guten Anleitung von Freenove zu sehen.

--->die Grundplatte hat drei Mehrfach-Zwergschalter. Deren Hebel habe ich mit einem Kugelschreiber bewegt, vorher die Mine auf „nicht-schreiben“ gestellt, die rohrartige Öffnung dann benutzt.

Dieses Schreiben umfaßt 11 DinA4-Seiten

Einleitung

Freenove, dort 106 Schulungsbücher und auch viele Schulungskästen.

<https://freenove.com/tutorial>

Einen dieser Schulungskästen haben wir bei Amazon bestellt --> FNK 0059

Freenove Projects Kit with Control Board V4 (Compatible with Arduino IDE), 238-Page Detailed Tutorials, 46 Projects, No Soldering, Simple Wiring --> **59,95Euro**

Sonntag bestellt, Mi 26112025 geliefert

Bestellt bei Amazon--->

2 Pack GY-NEO6MV2 NEO-6M GPS-Modulplatine-Flight Controller Board 3V-5V mit superstarken Keramikantenne, die mit Arduino EEPROM APM 2,5 kompatibel ist

zwei Pack für zusammen 10,99Euro

dazu an derer Stelle später mehr

Beschaffung der Bedienanleitung aus dem Weltnetz

Suchwort in Google --> Freenove FNK 0059 ---><https://freenove.com/tutorial>

-->dort-->zu FNK 0059 gehen-->dort download anklicken-->

es erscheint eine neue Download-Datei-->diese abspeichern-->

bei uns unter „Rob---26112025---Freenove.FNK0059“

--->die Datei aufmachen-->die pdf-Datei „Tutorial“ anklicken ggfs. Doppelklick

Dort erscheint die über 200seitige Bedienanleitung mit vielen Bildern und vielen Lehrhinweisen zum Tätigwerden.

Beschreibung des Schulungskastens

Der Schulungskasten enthält eine Grundplatte, auf der sehr viele Kleingeräte fest eingebaut und verdrahtet sind.

Dazu gehören ein Halter für den Kleinrechner, der einem Arduino ähnlich ist, Rechnereingangsgeber (Temperaturfühler, Spannungsfühler), Rechnerausgangsverwerter (Lampe, Lautsprecher, Bildschirm) sonstiges Zubehör (Schalter)

Es sind auch drei Reihen mit Vielfachschaltern vorhanden.

Die Vielfachschalter enthalten je 8 Einzelschalter.

Die Vielfachschalter sind sehr platzsparend.

Mit denen kann man unterschiedliche Geräte mit unterschiedlichen Beinen des Rechners verbinden, einfach durch die Bewegung von einigen dieser Schalter.

Zusätzlich zur Grundplatte sind im Schulungskasten auch einige Einzelgeräte vorhanden, ein Motor, ein Servo und anderes. Diese können an die Grundplatte mit beigefügten Kabeln angesteckt werden.

Seite 2 von 208 der Bedienanleitung

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic

circuits so that they may transform their creative ideas into prototypes and new and innovative products. To

this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi® and micro:bit®

- Electronic Component Assortments, Electronic Modules and Specialized Tools

- Product Development and Customization Services

You can find more about Freenove and get our latest news and updates through our website:

<http://www.freenove.com>

Seite 3 von 208

Inhaltsverzeichnis

Seite 5 von 208

Der mit dem Arduino austauschbare Kleinrechner und seine Einzelteile.

Seite 7 von 208

Die Beschreibung der Grundplatte mit den vielen eingebauten Einzelteilen.

Seite 8 von 208

Das Verschrauben von Bodenplatte und Grundplatte.

Die Beschaffung derjenigen elektronischen Werkzeuge mit denen wir Arbeitsaufträge in den Kleinrechner hineinbringen werden.

Wir haben einen Bürorechner mit Weltnetzzugang und wir gehen zur Arduinoseite

First, install Arduino Software (IDE): visit <https://www.arduino.cc/en/software/>

Dort:

Arduino IDE 2.3.6 Release notes

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger. For more details, check the Arduino IDE 2.0 documentation. Download

Dann bei uns abspeichern unter --> 26112025---Freenove.Schulungsbaukasten.Arduino

Dann installieren in --> C:\Program Files\Arduino IDE

Dann Arduino IDE, so heißt obengenanntes Werkzeug, an „Start anheften“.

Beschreibung des Arduino-IDE-Kleinrechnerbefüllwerkzeugs.

Erste Schritte mit dem Arduino-IDE-Kleinrechnerbefüllwerkzeug.

Arduino-Ide öffnen-->

Tools-->Board-->Arduino Uno

Tools-->Port-->Com3

Port ist die Buchse, an der das Überspielkabel in den Bürorechner eingesteckt wird.

Man steckt das Kabel ein, geht auf „Tools-->Port“, dort werden dann mehrere Ports (=Eingänge am Bürorechner) angezeigt. Dann zieht man das Kabel wieder heraus und jetzt wird ein Port weniger angezeigt. Der fehlende Port ist derjenige, den wir danach benutzen. Wir stecken also das Kabel wieder ein, dann „Tools-Port“ und wählen jetzt aus, denn jetzt wissen wir, was wir brauchen. Hinweis: Die Namen jedes Ports können sich ändern, man muß die bei Bedarf auf die obige Art ermitteln.

File-->Basics-->Blink

In dem obenstehenden grünen Band den „rechtszeigenden Pfeil im Kreis“ anklicken.

Jetzt gibt das Werkzeug den Arbeitsauftrag vom Bürorechner über das Vrbindungskabel in den Kleinrechner hinein.

Eine in den Kleinrechner eingebaute kleine Lampe blinkt im Sekundentakt.

Die vier farbigen Kappen auf die Taster aufstecken. Das geht schwer und das knackt.

Chapter 1 LED Blink

Earlier we tried to make the LED marked "L" blink. Now let us use Freenove Projects Board to reproduce this phenomenon and try to understand its principle.

Chapter 2 Flowing LED We have learned previously how to control 1 LED through Sketch on the control board and learned some basic knowledge of programming. Now let us try to control 14 LEDs and learn how to simplify the code.

Die kleinen Lämpche D0 bis D13 auf der Grundplatte im „I/O Indicator“

werden nacheinander eingeschaltet von D0 beginnend bis D13, dann rückwärts bis D0, dann immer so weiter.

Die Arbeitsauftragstexte stehen auf Seite 30. Diese kopieren und im Arduino-IDE-Werkzeug einfügen.

Die Bedienanleitung von Freenove ist eine pdf. Man muß diese pdf-Datei mit einem Werkzeug öffnen, welches sowohl die Datei öffnet als auch das Herauskopieren von Text ermöglicht.

Kapitel 3.1

Seite 32 bis 35 von 208

Der viereckige blaue Taster schaltet die blaue LED=Bauteil Nummer 6 beim Tasterdrücken ein und **beim Tasterloslassen aus.**

Kapitel 3.2

Seite 36 bis 37 von 208

Man drückt den blauen Taster kurz und die blaue Lampe wechselt den Zustand zwischen ein und aus und behält den neuen Zustand bei, bis der Taster das nächstmal gedrückt und wieder losgelassen wird.

Zugleich, und auch notwendigerweise, macht man das mechanische Tasterprellen auf elektronische Art unschädlich, durch vorgeschriebene Wartezeiten im Millisekundenbereich.

Hinweis 1.

Anstelle von boolean schreibt man bei Arduino besser bool.

Arduino kann auch mit boolean umgehen.

Aber in der Textdarstellung im Arduino-IDE-Werkzeug erscheint boolean in schwarz wie normaler Text. Dagegen bool in blauer Schrift, um anzuzeigen, dass es eine Zeichenfolge aus dem Arduino-Arbeitsaufträge-Baukasten ist.

Hinweis 2.

Der Arbeitsauftrag an den Arduino ist vierteilig.

Anfangszeilen, dann `void setup()` , dann `void loop()` , dann `void reverseLED()`

Der Arbeitsauftrag wird in den Arduino-Kleinrechner hineingetragen.

Danach arbeitet der Arduino die `Anfangszeilen` und `void setup()` ein einziges Mal ab. Damit speichert er sich ein gewisses Grundwissen ein, auf welches er später zurückgreifen wird. Danach arbeitet er immer wieder die Zeichenkette im Bereich `void loop()` durch. Das ist eine Kreislauf-Tätigkeit.

Wenn nun innerhalb dieses Arbeitsauftrag-Bereichs einmal die Zeichenfolge `reverseLED()` auftaucht, dann hält der Arduino an und sucht im gesamten Arbeitsauftrag nach der Zeichenfolge `void reverseLED()`

Warum hält der Arduino an und sucht? Die runden Klammern in `reverseLED()` , eine auf, eine zu, veranlassen den Arduino zur Suche.

Wenn er dann `void reverseLED()` gefunden hat, dann sagt genau diese Zeichenfolge: „arbeite die folgenden Arbeitsaufträge durch“

Und dann arbeitet er die dahinterstehenden Aufgaben durch und danach macht er in `void loop()` wieder weiter.

Hinweis 3.

Beim Arduino ist die Zeichenfolge `void ()` oder `void vom Benutzer wählbarer Name ()` ein Hinweis darauf, dass es um einen abgegrenzten Teil des Gesamt-Arbeitsauftrags handelt.

Hinweis 4.

`void setup()` , und `void loop()` sind in der Arduinosprache feststehende Begriffe.

Diese beiden Arbeitsbereiche kommen in fast jedem Arbeitsauftrag des Arduino vor.

Hinweis 5.

Ähnlich aussehende Zeichenfolgen können beim Arduino unterschiedliche Dinge bewirken.

Gleich aussehende Zeichenfolgen können beim Arduino unterschiedliche Dinge bewirken, je nachdem an welcher Stelle im Gesamtarbeitsauftrag sie stehen.

Project 4.1 Send Data through Serial

We will use the serial port on control board to send data to computer.

Wir benutzen den Bildschirm des Bürorechners, um Dinge anzuzeigen, welche der Kleinrechner Arduino ausgibt und mit dem USB-Kabel an den Bürorechner sendet.

In diesem Fall schickt der Rechner eine vom Arduino selbsterzeugte Zeitzählreihe an den Bürorechner-Bildschirm

Lief am 27112025 beim ersten Versuch gut.

Project 4.2 Receive Data through Serial Port

In the previous section, we used Serial port on control board to send data to a computer, now we will use it to receive data from computer

Advantages of interrupt here: Processor won't need to check whether the event has happened every now and then, but when an event occurs, it informs the controller immediately. **When an interrupt occurs**, the processor will jump to the interrupt function to handle interrupt events, then return to where the interrupt occurs after finishing it and go on this program.

Auf Seite 44--→der Arbeitsauftrag zum Anzeigen von Zahlen und Wörtern auf dem Bildschirm des Bürorechners.

Lief am 27112025 beim ersten Versuch gut, sowohl mit Kaesekuchen als auch mit Käsekuchen, es geht also auch mit Umlauten. Der Bildschirm des Bürorechners zeigte beides richtig an.

Auf Seite 45--→der Arbeitsauftrag für zwei Arbeitsreihen, **zwischen denen der Arduino selbständig umschaltet**.

Arbeitsreihe A= ein Zeitzahlenzählen und Anzeigen auf dem Bildschirm des Bürorechners.

Arbeitsreihe B=Aufnahme von Zeichen aus der Tastatur des Bürorechners und Bearbeiten derselben, hier indem sie auf dem Bürorechnerbildschirm angezeigt werden.

Der Arduino macht zunächst Arbeitsreihe A.

Im Hintergrund lauert der Auftrag **void serialEvent()** in genau dieser Zeichenfolge.

Auslöser für dessen Tätigwerden ist es, wenn man in der Sendezeile des Arduinowerkzeugs ein Absendebefehl=Hakentaste gedrückt wurde, nachdem dort vorher Text eingegeben wurde..

Dann springt der Arduino auf Arbeitsreihe B, führt diese aus, macht danach mit Arbeitsreihe A weiter.

Hinweis 1

void serialEvent() ist ein Wächter, der Hintergrund wacht, ob in den Arduino Textzeichen hineingegeben werden sollen. Dieser Wächter ist unabhängig von den Tätigkeiten im Bereich void loop()

Hinweis 2

Das Arduino-IDE-Werkzeug arbeitet mit dem Bürorechner zusammen **und es benutzt den Bildschirm des Bürorechners gleichzeitig für drei Aufgaben**.

Eingabe der Arbeitsaufträge für den Arduino.

Eingabe von Zeichen, die der Arduino zur Anzeige soll.

Anzeige von Zeichen, die der Arduino ausgibt.

Das Arduino-IDE-Werkzeug macht es so, dass es den Bildschirm des Bürorechners in drei Bereich aufteilt.

Oben die allgemeinen Arbeitsaufträge

darunter eine Zeile zur zusätzlichen Zeicheneingabe

und noch darunter ein Feld für anzuzeigende Dinge, die der Arduino ausgibt.

Lief am 27112025 beim ersten Versuch gut.

Kapitel 4.3

Seite 47 bis 50 von 208

Einstellung der Helligkeit einer Lampe durch Eingabe einer Zahl.

Project 4.3 Application of Serial

We will use the serial port on control board to control one LED.

Hinweis für weiteres Lernen, unabhängig von dieser Freenove-Schulung hier

Using Serial.parseInt() with Arduino 15 Minuten Film

Programming Electronics Academy

<https://www.youtube.com/watch?v=lcjLUdDB1Us>

Ich habe kurz vor dem Ende von `void loop()` noch folgende Zeile zusätzlich in das Arduino-IDE-Werkzeug eingegeben --→ `delay(5000);`

Mit dieser Ergänzung lief diese Übung am 27112025 gut ab.

Zeitbedarf eine Stunde zum Finden dieser Lösung

Kapitel 5.1

Seite 51 bis 55 von 208

Die Umwandlung von stetigen Werten in stufenförmige Werte --→

Analog-Digital-Wandlung

Mit einem Drehwiderstand wird die Spannung durch Drehen stetig zwischen Null und 5 Volt verändert. Das Gerät empfängt die Spannung, rechnet den Wert auf eine Stufe von 0 bis 1023 um und zeigt das an.

Zugleich verteilt der die 1024 möglichen Werte auf den Bereich null bis 5 Volt und zeigt auch diese Spannung an.

Diese Übung lief am am 27112025 gut ab.

Kapitel 5.2

Seite 56 bis 58 von 208

Einstellung der Helligkeit einer Lampe durch das Drehen an einem Drehwiderstand.

Diese Übung lief am am 27112025 gut ab.

Kapitel 5.3

Seite 59 bis 62 von 208

Einstellung der Helligkeit einer Lampe durch das Beschatten eines lichtabhängigen Widerstands.

Diese Übung lief am am 27112025 gut ab.

Kapitel 6.1

Seite 63 bis 66 von 208

Einstellung der Farbe und Helligkeit einer Lampe durch Drehen an drei Drehwiderständen.

Diese Übung lief am am 27112025 gut ab.

Kapitel 6.2

Seite 67 bis 69 von 208

Einstellung der Farbe und Helligkeit einer Lampe durch Zufallswerte, die der Arduino ausgibt.

Diese Übung lief am am 27112025 gut ab.

Kapitel 7.1

Seite 70 bis 76 von 208

Einstellung der Farbe an einem Ring aus 8 Lämpchen. In jedem Lämpchen sind einige LEDs. Rot, Grün, Blau, vielleicht auch noch weiß. Das alles wird gesteuert durch Zufallswerte aus dem Arduino.

Diese Übung lief am am 27112025 großteils gut ab.

Es wa

r jedoch eine blaue LED ohne gute Wirkung, die übrigen LEDs waren gut.

Somit ist der gesamte Lampenring nur bedingt einsetzbar. Zum Lernen ist er gut genug.

Unterschiedliche Farben innerhalb eines Achtlampenrings, veränderbar mit einem Drehwiderstand

Diese Übung lief am am 27112025 gut ab in dem Sinne--→Arbeitsauftrag von der Freenove-Bedienanleitung kopiert und in das Arduino-IDE-Werkzeug hineinkopiert, dann in den Arduino hineingeschoben, vorher den Farblampenring an die Grundplatte angeschlossen und dann lief alles beim ersten Versuch.

Die bebilderte Bedienanleitung von Freenove ist sehr ausführlich.

Einschalten und Ausschalten eines „**aktiven**“ **Summers** mit dem Arduino

Diese Übung lief am am 27112025 gut.

Einschalten und Ausschalten eines „passiven“ Summers mit dem Arduino

Diese Übung lief am am 27112025 gut.

In der **Freenove-Bedienanleitung** steht in der ersten Zeile

`int buzzerPin = 6;` --→aber auf der Grundplatte bei den Zwergschaltern steht p-Buzzer geht zu D7--→also habe ich den Arbeitsauftragstext geändert auf

`int buzzerPin = 6;`

Die Fehlersuche hat etwa 30 Minuten gedauert.

Der Arduino **schaltet ein Relais mit Hilfe eines Tasters**.

Mit elektronischer Unschädlichmachung des mechanischen Tasterprellens.

Diese Übung lief am am 27112025 gut in dem Sinne: alles einstellen, Befehle aus der Freenove-Bedienanleitung kopieren und in das Arduino-IDE-Werkzeug hinschieben. Dann den blauen Taster drücken, das Relais schaltet.

Mit den Einzelheiten der Elektronik und der Befehlszeilen kann man sich später noch einmal befassen.

Der Arduino **schaltet einen Motor**. Ein, aus, Laufrichtung und Drehzahl, alles mit einem einzigen Drehwiderstand.

Der Motor braucht eine eigene Stromversorgung, dafür reicht der Arduino nicht.

Wir haben aber in unserem Elegoo-Kasten ein 9-Volt-Netzteil mit Verbindungskabel und das benutzen wir für diese Lernarbeit.

Abwechselnd dazu benutzen wir auch eine 9-Volt-Batterie aus unserem Vorrat, das Verbindungskabel liegt dem Freenove-Kasten bei.

Diese Übung lief am am 27112025 mittelmäßig.

Der Motor lief nicht an, man muß an der Welle anfangs mit Drehen von Hand nachhelfen. Er startete auch dann nur bei höchster Einstellung, also am Drehwiderstand ganz links oder ganz rechts gedreht.

Dann den Motor mehrmals mit WD40 eingesprüht. Danach hing er gut am Drehwiderstand, man konnte Drehrichtung, Drehzahl, ein und aus allein mit dem Drehwiderstand einstellen.

Am Folgetag will ich die Motorgleitlager mit **Nähmaschinenöl** vesehen. Das habe ich gemacht, und das hat sehr viel gebracht. Der Motor läßt sich jetzt mit dem Drehwiderstand leicht einstellen. WD40 dient mehr zum Schraubenlösen, weniger zur Schmierung. Aber es ist besser als nichts.

Der Motor im Freenove-Lernkasten ist **ein ganz kleiner und sehr billiger Spielzeugmotor**. Damit will Freenove nur zeigen, daß es überhaupt geht und wie der Motor angesteuert wird. Der gesamte Kasten kostet 60Euro, und da können meistens nur sehr billige Bauteile verwendet werden.

Beim Hochladen vom Arduino-IDE-Werkzeug in den Arduino hinein gab es Schwierigkeiten. Das USB-Kabel aus dem Bürorechner gezogen, nochmal eingesteckt, nochmal hochgeladen, dann ging das Hochladen gut. Dieser Fehler ist mehrmals aufgetreten, war aber auf obige Art immer sehr einfach zu beheben.

Kapitel 11.1

Seite 102 bis 106 von 208

Ansteuern eines Servos mit dem Arduino

Diese Übung lief am am 27112025 innerhalb weniger Minuten sehr gut.
Einen der beigelegten Servoarme an den Servo angebaut.
Beim Hochladen gab es Schwierigkeiten, das war leicht zu beheben, siehe Kapitel10.1

Kapitel 11.2

Seite 107 bis 109 von 208

Ansteuern eines Servos mit dem Arduino **und einem Drehwiderstand**.

Diese Übung lief am am 27112025 innerhalb von Minuten sehr gut.
Die Drehbewegung am Drehwiderstand überträgt sich in eine 180 Grad Drehbewegung am Servo.
Diese Übung lief am am 27112025 innerhalb von Minuten sehr gut.

Kapitel 12.1

Seite 110 bis 113 von 208

Ein hitzeveränderlicher Widerstand steuert den Arduino.

Diese Übung lief am am 27112025 innerhalb von Minuten sehr gut.
Der Bürorechnerbildschirm zeigt 19Grad Celcius an, und wenn ich den hitzeveränderlichen Widerstand mit Daumen und Zeigefinger berühre, dann zeigt er 30 Grad Celcius an.
Die **Anzeigegenauigkeit** ist zwei Stellen hinter dem Komma, sodaß man Veränderungen sofort sieht. Die **Meßgenauigkeit** ist wahrscheinlich viel geringer, deren Wert kenne ich nicht.

Kapitel 13.1

Seite 114 bis 118 von 208

Ein Kreuz-Seitenschieb-Geber mit Zusatzschalter steuert den Arduino.

Diese Übung lief am am 27112025 innerhalb von Minuten sehr gut.

Die X-Achse verläuft von links nach rechts, deren Werte sind null bis 1023, bedient durch den Kreuzhebel. Nach links geschoben=1023, nach rechts=null

Die Y-Achse verläuft von unten nach oben, deren Werte sind null bis 1023 ebenfalls bedient durch den Kreuzhebel. Hochgeschoben=1023 unten=null.

Die Z-Achse wird durch einen Schalter betätigt, nimmt den Wert null oder 1 an, wird durch senkrechtes Drücken des Kreuzhebels betätigt. Gedrückt=null unbetätigt=eins.

Hinweis

Set the port to INPUT_PULLUP mode, which is equivalent to configuring the port to INPUT mode, then connect a resistor with high resistance value to VCC behind the port.

pinMode(A3, INPUT);

Buchse A3 wird als eine zweiwertige Eingangsbuchse bestimmt. Sie kann nur die Werte null und eins annehmen. Zugleich hängt am Arduino innen noch ein nicht zu großer Widerstand

zwischen plus 5 Volt und der Buchse A3. Dadurch hat die Buchse A3 immer dann 5 Volt am Eingang, wenn nicht ein deutliches Null-Signal anliegt. Das alles wird mit der obigen gelb hinterlegten Zeile ausgesagt.

Zu den Buchsen A1 und A2 wird an dieser Stelle nichts gesagt, sie behalten ihre Grundeigenschaften bei, sie sind und bleiben 1024-wertige Eingangsbuchsen. Sie geben eingehende Spannungen von null bis 5 Volt als Werte von null bis 1023 in den Arduino hinein.

Kapitel 14.1

Seite 119 bis 125 von 208

Ein **Beschleunigungsmeßgerät und Drehzahlmeßgerät** steuert den Arduino.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut.

Die Dinge werden am Bürorechnerbildschirm angezeigt.

Was die einzelnen Werte genau bedeuten, ist unklar.

Aber das Gerät MPU6050 arbeitet und man kann viele Lehrfilme bei Youtube anschauen, wenn man sich näher damit befassen will.

Kapitel 15.1

Seite 126 bis 131 von 208

Der Arduino **steuert einen Lichtbalken** aus zehn nebeneinanderliegenden LEDs.

Es wird immer eine Lampe nach der anderen angemacht und das wirkt so, als würde das Licht hin- und herwandern.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut.

Kapitel 16.1

Seite 132 bis 136 von 208

Der Arduino **steuert eine siebenteilige LED-Anzeige für Ziffern und Buchstaben** und zeigt dabei die Zeichenfolge 0123 an.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut

Kapitel 16.2

Seite 137 bis 140 von 208

Der Arduino **steuert eine siebenteilige LED-Anzeige für Ziffern und Buchstaben** und zählt dabei im Sekundentakt von 0001 an aufwärts und zeigt das an.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut

Kapitel 17.1

Seite 141 bis 148 von 208

Der Arduino steuert **ein schachbrettartiges Viellämpchenfeld** an, eine sogenannte LED-Matrix.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut

Kapitel 18.1

Seite 149 bis 153 von 208

Der Arduino steuert ein 6cm breites und 2cm hohes **Bildschirmgerät** und zeigt Zeichen an, die ich vorher am Bürorechner eingegeben habe

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut

Kapitel 18.2

Seite 154 bis 162 von 208

Der Arduino steuert ein 6cm breites und 2cm hohes **Bildschirmgerät** und er selbst wird von einem **Hitzefühler** gesteuert.

Diese Übung lief am 27.11.2025 innerhalb von Minuten gut

Kapitel 19.1

Seite 163 bis 168 von 208

Der Arduino steuert einen **Schrittmotor** an. Damit kann die Winkelstellung der Abtriebswelle des Motors genau eingestellt werden.

Für den Steppermotor reicht die Stromversorgung durch das USB-Kabel an den Arduino-ähnlichen Kleinrechner offensichtlich aus.

Diese Übung lief am am 28112025 innerhalb von Minuten gut.

Kapitel 20.1

Seite 169 bis 173 von 208

Der Arduino wird **von einem schachbrettartigen Tastenfeld** gesteuert, von einem sogenannten Keypad

Das Tastenfeld hat 16 Taster und es wird mit 8 Kabeln mit dem Arduino verbunden.

Diese Übung lief am am 28112025 innerhalb von Minuten gut.

Kapitel 21.1

Seite 174 bis 178 von 208

Eine **Fernbedienung mit vielen Tasten** sendet Infrarotzeichen an ein **Infrarotempfänger**, dieses ist mit dem Arduino verbunden. Man drückt eine Taste am Sendegerät, und auf dem arduino gesteuerten Bildschirm des Bürorechners erscheint der Name der Taste als mehrstellige Buchstaben-Zahlen-Folge.

Zum Beispiel die Taste „2“ hat die zugeordnete Zeichenfolge „FF18E7“

die Taste „Menu“ hat die zugeordnete Zeichenfolge „FFE21D“

Diese Übung lief am am 28112025 innerhalb von Minuten gut.

Kapitel 21.2

Seite 179 bis 182 von 208

Infrarot-Fernbedienung--→**Infrarotempfänger**--→Arduino--→**Summer und Lämpchen**.

Mit dem Sender kann ich verschiedene Lichtstärken und ein-aus einstellen. Während des Sendevorgangs ertönt ein Summer. Das Lämpchen ist eine LED.

Diese Übung lief am am 28112025 innerhalb von Minuten gut.

Kapitel 22.1

Seite 183 bis 187 von 208

Infrarotgerät **zur Entdeckung von Bewegungen** und Weitermeldung an den Arduino.

Wenn es soweit alles eingestellt ist, **muß man noch eine Minute warten**, dann erst ist das Gerät einsatzbereit.

Diese Übung ist am am 28112025 nicht gelungen. Aus Zeitgründen dann auch nicht weiter versucht.

Kapitel 23.1

Seite 188 bis 194 von 208

Abstandsmessung mit einem Ultraschallgerät.

Diese Übung lief am am 28112025 innerhalb von Minuten gut.

Zu den Arbeitsaufträgen für den Arduino.

Es gibt **Erklärtexte**, sogenannte Kommentare, die man in den Arbeitsauftrag für den Arduino einfügen kann. Der Mensch liest die und versteht dann den Gesamttext besser.

Der Arduino darf es aber nicht lesen, weil sonst das Hochladen gestoppt wird.

Daher gibt es folgende „Arduino-Lesestopp-Sonderzeichen“

// Einzeiliger Erklärtext

und

/*Einzeiliger oder mehrzeiliger Erklärtext*/

Wenn man nun aus der Freenove-Bedienanleitung die Arbeitsaufträge kopiert und dann in das Arduino-IDE-Werkzeug einfügt, dann kann es sein, daß ein Erklärtext mit „//“ beginnt, aber länger als eine Zeile ist. Das verhindert dann das Hochladen in den Arduino hinein.

Daher dann im Arduino-IDE-Werkzeug entweder am ersten Zeilenende den Zeilenumbruch ausschalten, durch Löschen dieser Stelle, oder aber den Erklärtext vorn und hinten mit den Sonderzeichen wie oben gezeigt versehen.

Die Freenove-Bedienanleitung bietet zwei unterschiedliche Formen des Arbeitsauftrags an.
Beide liefen gut.

Kapitel 24.1

Seite 195 bis 202 von 208

Ein **Kurzstreckenfunkgerät mit Kennnummer und Zusatzinhalt** wirkt auf den Arduino.
Hier geht es darum, die Nummer des Gerätes zu erkennen und auf dem Bildschirm
anzuzeigen. Das Gerät sieht aus wie ein Einkaufswagenchip.
Diese Übung lief am 28.11.2025 innerhalb von Minuten gut.

Kapitel 24.2

Seite 203 bis 206 von 208

Wir schreiben eigenen Text auf obiges einkaufswagenchipartigaussehendes
Kurzstreckenfunkgerät und damit speichern wir den Text auch.
Damit haben wir ein Gerät, das unseren eigenen Text enthält, eine eigene Kennnummer hat
und von sehr vielen Lesegeräten erkannt werden kann.
Das Gerät ist also Ausweis, Funkgerät und Textspeicher in einem.

Zeile 81 des Arbeitsauftrags in der Freenove-Bedienanleitung, dort auf Seite 205 von 208
`Serial.println("Write card OK!");`
Stattdessen habe ich geschrieben
`Serial.println("Heute wollen wir den Hof kehren");`

Dann habe ich das Gerät von der Leseplatte weggenommen und wieder aufgelegt und der Text
erschien wieder auf dem Bürorechnerbildschirm, angesteuert vom Arduino, in dem Arduino-

IDE-Werkzeug, zu welchem ja auch die Mitbenutzung des Bürorechner-Bildschirms gehört.
Diese Übung lief am 28.11.2025 innerhalb von etwa einer Stunde gut.

Weitere Hinweise von Freenove

Seite 207 von 208

Die ASCII -Tabelle

Seite 208 von 208

26.11.2025---100---Freenove.Schulungsbaukasten.Arduino
Laufendes-2025-645-501.html siehe auch Laufendes-2025-645.html
Laufendes-2025-645-501.pdf

---Ende---