

EUR 79,95  
Ausdrucksförderung  
Preiswahl: Experimentier-Platine im Wert von ca. EUR 75,00

Leichter Einstieg in die Mikrocontroller-Programmierung  
mit Arduino/Freeduino

# Lernpaket Arduino

## Mikrocontroller-Programmierung

- Vollständige Hardware und Software zum Programmieren und Experimentieren
- Platine, Steckboard, Bauteile und universell einsetzbarer Mikrocontroller ATmega168
- Grundlagenkurs zur Arduino-C-Programmierung
- Eigene Applikationen selbst entwickeln
- Über 70 Experimente mit ausführlicher Anleitung

Mit Platine und  
Mikrocontroller  
ATmega168

### Über 70 Software-Experimente:

- LED-Dimmer
- Berührungssensor
- Reaktionstester
- Kapazitätsmessung
- Licht- und Temperaturmessung
- Dämmerungsschalter
- Alarmanlage
- Codeschloss
- Lüftersteuerung
- Datenübertragung und Telegrafie
- Speicheroszilloskop
- Messen und Steuern mit VB.NET und viele weitere mehr

Ohne Lötcolben  
experimentieren!  
Bauteile einstecken  
und starten!

Inklusive: 200-seitiges Handbuch, Experimentierplatine, Steckboard, 20 Bauteile + Software auf CD-ROM

FRANZIS

## Das Arduino-Projekt:

## Mikrocontroller-Programmierung ganz einfach

Denkt man an Mikrocontroller-Programmierung, so winken selbst viele Elektroniker ab – zu aufwändig zu lernende Programmiersprachen, komplizierte Rechenalgorithmen. So macht man dann entweder einen Bogen um den kleinen Controller oder man sucht sich jemanden, der programmieren kann. Das muss man mit Arduino nicht mehr! Das aus einer einfachen und universellen Hardware und einer zugehörigen Programmierumgebung bestehende Projekt macht das Realisieren eigener Mikrocontrollerprojekte einfacher. Wir stellen Arduino anhand des speziell zusammengestellten Arduino-Lernpakets von FRANZIS vor.

### Mikrocontroller und Designer

Künstler und Designer sind eine Gruppe, die davon lebt, ständig neue, teilweise zuvor unrealistisch erscheinende Ideen zu entwickeln und zu realisieren. Eine solche kam den Studenten und Lehrenden des Interaction Design Institute in Ivrea (Italien) vor einigen Jahren: warum nicht auf eine andere Weise mit Computern kommunizieren als mit den bekannten Werkzeugen? Warum nicht Aktionen auf dem PC-Bildschirm durch Gesten und Bewegungen steuern? Und auf welche unaufwändige Weise bekommt man einen PC dazu, Ereignisse und Einflüsse aus der Umgebung aufzunehmen und zu verarbeiten?

Was bei den Überlegungen herauskam, war eine sogenannte Physical-Computing-Plattform. Die macht genau das, was wir eben erläutert haben, d. h. per Mikrocontroller und Sensoren bzw. Aktoren auf Ereignisse reagieren bzw. einwirken. Neben dem ursprünglich

angedachten Einsatz in der Künstler- und Designerwelt haben sich inzwischen vor allem die Hobbyelektroniker diese Plattform angeeignet, um recht schnell und mit wenig Hardware- und Programmieraufwand kleine Elektronikprojekte realisieren zu können. Für den Techniker übersetzt: Die Italiener haben nichts anderes gemacht als eine Physical-Computing-Plattform (oder für Techniker verständlicher: embedded Platform) entwickelt, die aus einem kleinen AVR-Board und einer sehr einfach beherrschbaren Programmiersoftware – Arduino – besteht. Seitdem hat sich dieses System rasant verbreitet, vor allem in der Ausbildung, an Design- und Kunsthochschulen, aber, wie gesagt, immer häufiger auch unter Hobbyelektronikern.

Ein riesiger Vorteil des Systems, das auch die Verbreitung sehr gefördert hat: Sowohl Technik als auch Software inklusive der damit erarbeiteten Programme sind Open Source!

Auf der Arduino-Webseite [1] ist das Original zu Hause.

### Die Arduino-Hardware

Basis der inzwischen in großer Vielfalt, siehe [2], verfügbaren Arduino-Boards ist ein AVR-Controller der ATmega-Reihe, meist kommt ein ATmega 168 oder 328 zum Einsatz. Das aktuellste Standard-Board ist das „UNO“ von arduino.cc (Bild 1), dessen Schaltung in Bild 2 abgebildet ist. Bisher hieß das Standard-Board „Duemilanove“.

Hervorstechend ist immer, dass alle verfügbaren I/O-Ports und die Betriebsspannung übersichtlich auf Buchsenleisten herausgeführt sind, und dies für eine möglichst große Anzahl von Board-Designs einheitlich. Warum, werden wir noch sehen. Dies und der angestrebte Verwendungszweck bedingen auch die Bezeichnung des Boards als I/O-Board. Neben den Standard-Boards gibt es noch besonders kompakte Versionen, solche mit integriertem Bluetooth oder Kurzstreckenfunk oder so spezielle wie das Lily-Pad, das speziell für den Einsatz in E-Textilien (Wearables) designed ist.

Da sowohl Schaltung als auch das Referenz-Design Open Source (Creative Common License) sind, gibt es inzwischen eine große Anzahl von Derivaten dieser Boards, so ist besonders die Freeware-Version Freeduino [3] von NKC bekannt, das zum sehr verbreiteten „Duemilanove“-Board kompatibel ist. Auf der Freeduino-

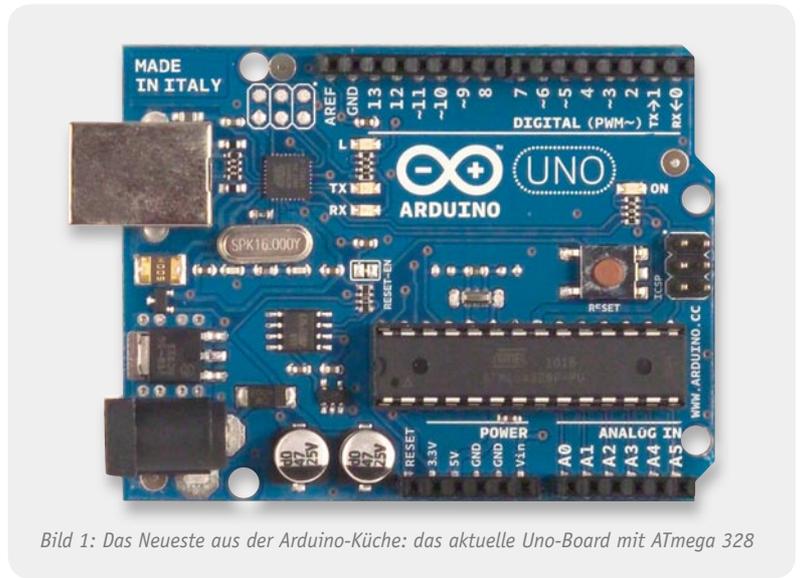


Bild 1: Das Neueste aus der Arduino-Küche: das aktuelle Uno-Board mit ATmega 328

Seite findet man dann auch eine lange Liste von weiteren Derivaten und auch die dazugehörigen Projekte.

Die ersten Board-Generationen trugen übrigens noch eine serielle Schnittstelle (RS232), heute ist ein USB-Port die Norm.

Auch das Board, das die Grundlage für das hier vorzustellende „Lernpaket Arduino“ aus dem Franzis-Verlag bildet, ist ein solches Freeduino-Derivat.

Wollen wir uns dieses Board und seine Hardware-Ausstattung einmal näher betrachten. Bild 3 zeigt

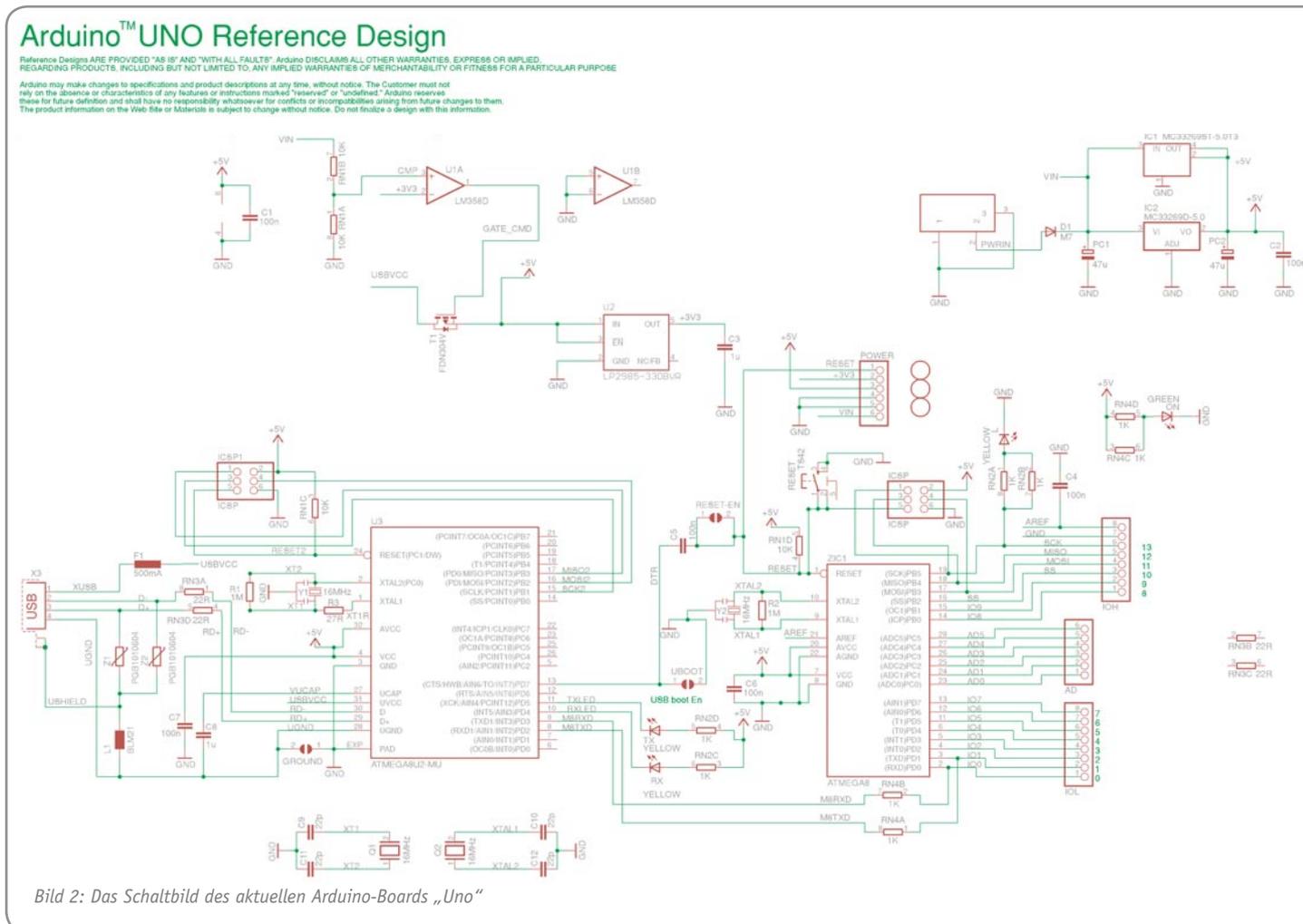


Bild 2: Das Schaltbild des aktuellen Arduino-Boards „Uno“

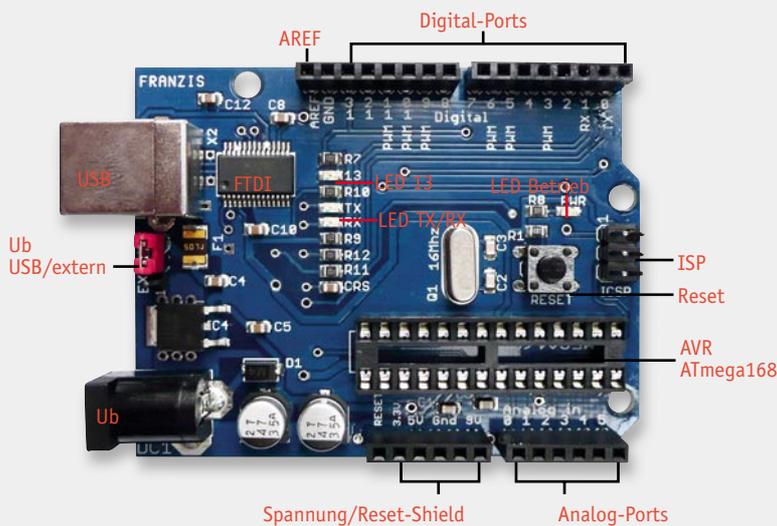


Bild 3: Das Freeduino-Derivat des Lernpakets ist kompatibel zum Original „Duemilanove“.

das (einseitig bestückte) Board, die „Experimentierplatine“ des Lernpakets, mit einigen Erläuterungen. Kernstück ist ein AVR des Typs ATmega 168. Dieser ist mit 16 MHz per Quarz getaktet und besitzt 16 Kbyte-Flash-Speicher für Programme, 1 Kbyte SRAM und einen 512-Byte-EEPROM. Insgesamt sind 20 I/O-Pins verfügbar: 14 digitale (davon 6 als PWM-Ausgänge nutzbar) und 6 analoge Eingänge mit 10 Bit Auflösung. Natürlich sind die im Datenblatt des ATmega 168 aufgezählten Schnittstellen SPI und TWI sowie die externen Interrupts verfügbar. Insgesamt ist der Controller als ganz normaler Mikrocontroller nach der Atmel-Spezifikation nutzbar, siehe auch Anhang A.1 im Franzis-Begleitbuch.

Das Board kann sowohl per USB als auch über eine externe Spannungsquelle (6 bis 20 Vdc, empfohlener

Bereich: 7 bis 12 V) betrieben werden. An der Buchsenleiste „Power“ stehen für den Betrieb externer Baugruppen die Eingangsspannung, die auf dem Board erzeugten 5 V und mit bis zu 50 mA belastbare 3,3 V, erzeugt vom USB-FTDI-Controller des Boards, auf dem Board.

Neben der USB-Schnittstelle stehen eine ISP-Schnittstelle sowie eine UART-Schnittstelle zur Verfügung. Komplettiert wird die Ausstattung durch einen Reset-Taster sowie LEDs für die Betriebsanzeige, den Datenverkehr auf der UART-Schnittstelle und eine fest an den I/O-Pin 13 geschaltete LED, die das allererste Experiment mit dem Board sogar ganz ohne periphere Bauteile möglich macht.

Schließlich ist über die Buchsenleisten auch der Anschluss einer externen Referenzspannung für die Analog-Eingänge möglich. So kann deren Standard-Spannungsbereich von 5 V verändert werden.

Was bleibt noch? Der AVR ist ab Werk mit einem Bootloader (STK500-Protokoll) versehen, der das sofortige Laden von Programmen erlaubt, ohne einen sonst erforderlichen Programmer einsetzen zu müssen. Alternativ ist der AVR aber auch per ISP am Bootloader vorbei erreichbar.

## Das Franzis-Lernpaket

Bevor wir zur Software kommen, wollen wir das „Lernpaket Arduino“ (Bild 4) näher betrachten. Es ist traditionell, wie von Franzis gewohnt, komplett bestückt mit allem, was man für die im Begleit- und Lernheft (210 Seiten) ausführlich beschriebenen über 70 Experimente benötigt: Transistoren, LEDs, Widerstände, Taster, Poti, Lichtsensor und Piezo-Signalgeber. Gute Tradition auch: Man benötigt, außer vielleicht einem Netzteil für autarken Betrieb von Programmen, kein Werkzeug, also auch keinen Lötkolben. Ein mitgeliefertes, kleines Steckbrett (Breadboard) und einige Drahtstücke ermöglichen eine lötfreie Verdrahtung der Experimente.

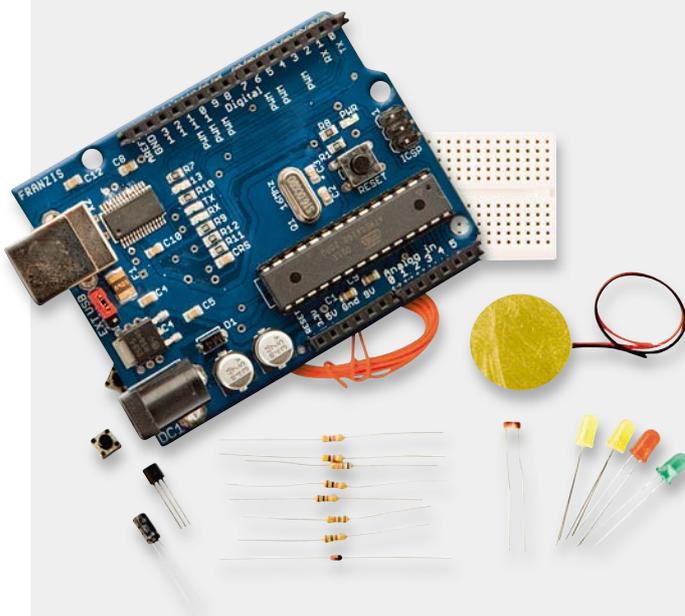


Bild 4: Das Lernpaket Arduino von Franzis



Komplettiert wird das Paket durch eine CD-ROM, auf der sich nicht nur alle zum Betrieb des Boards nötigen Softwareprogramme inklusive der noch zu besprechenden Entwicklungsumgebung befinden, sondern auch insgesamt 76 Programmlistings, sogenannte Sketches. Auch die komplette Dokumentation des Boards sowie den Bootloader findet man auf der CD-ROM sowie eine Datenblattsammlung mit diversen Zusatzschaltungen, u. a. von sogenannten Shields, den Hardware-Zusätzen, die dem Board im praktischen Einsatz den Kontakt zur Umwelt und zum Benutzer ermöglichen.

Damit erhält man ein komplettes Paket, dessen Board auch nach Abschluss der Lern- und Experimentierphase als vollwertiges Arduino-Board einsetzbar ist. In dieser Zusammenstellung ist das Lernpaket auch eine sehr geeignete Ausrüstung für die Ausbildung.

Betrachten wir den zweiten Bestandteil des Arduino-Projekts, die Software.

### Die Arduino-Software

Neben dem einheitlichen Hardware-Design ist die Entwicklungsumgebung und die Programmiersprache von Arduino das eigentliche Highlight des Systems. Denn das Ganze ist so einfach gehalten, dass auch jemand, der bisher nichts oder kaum etwas von Programmierung hielt, Anwendungen programmieren kann.

Die Programmiersprache basiert auf einem stark vereinfachten C-Dialekt (Wiring) und die Entwicklungsumgebung (IDE) auf der IDE von Processing [4], einer Open-Source-Programmiersprache und -Umgebung, die vor allem auf den Einsatz im künstlerischen Bereich wie Grafik, Animation und Interaktion sowie auf den Ausbildungs- und Hobby-Bereich spezialisiert ist. Sie ist plattformunabhängig, basiert auf Java, und es gibt unendlich viele fertige Anwendungs-Bibliotheken (Libraries), die den Nutzer von lästiger Standard-Programmierarbeit befreien.

Sieht man sich die Arduino-IDE an (Bild 5), so fällt

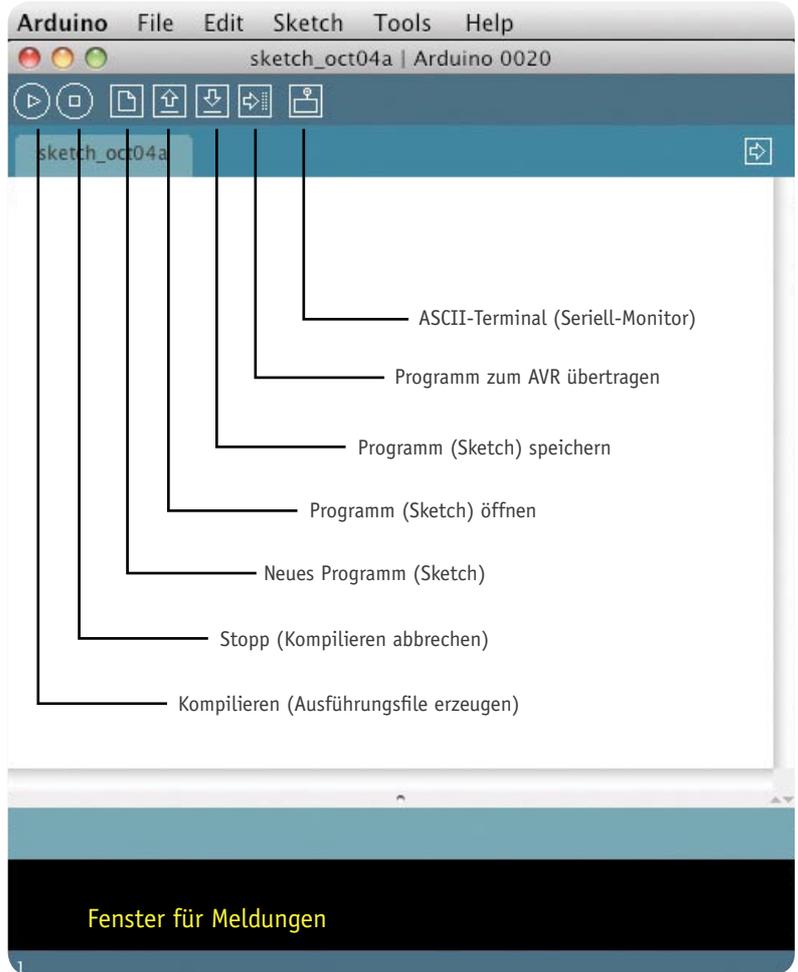


Bild 5: Übersichtlich – die Arduino-IDE

erst einmal eines auf: Sie ist sehr übersichtlich. Nur zwei Fenster, nur wenige Buttons und Menüpunkte. Genau das ist die Stärke von Arduino – es ist einfach. Oben ein Programmfenster, unten je nach Situation Meldungen oder Terminalbetrieb, fertig!

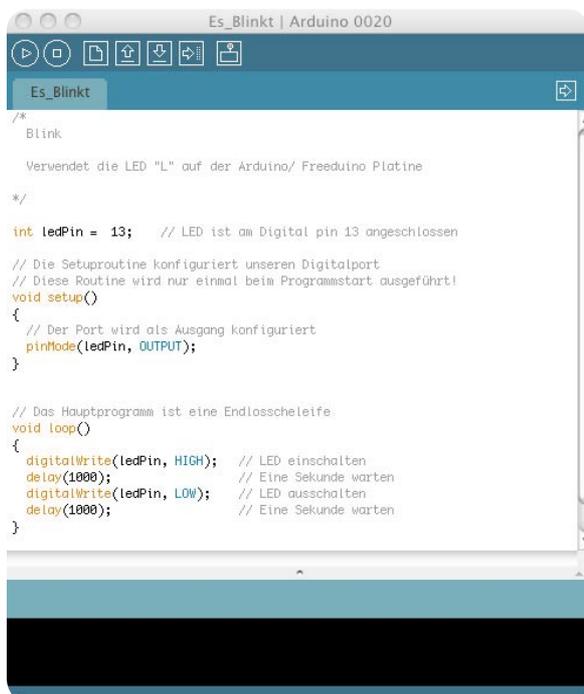


Bild 6: Hallo Welt mit Arduino – das erste Programm lässt eine LED blinken.

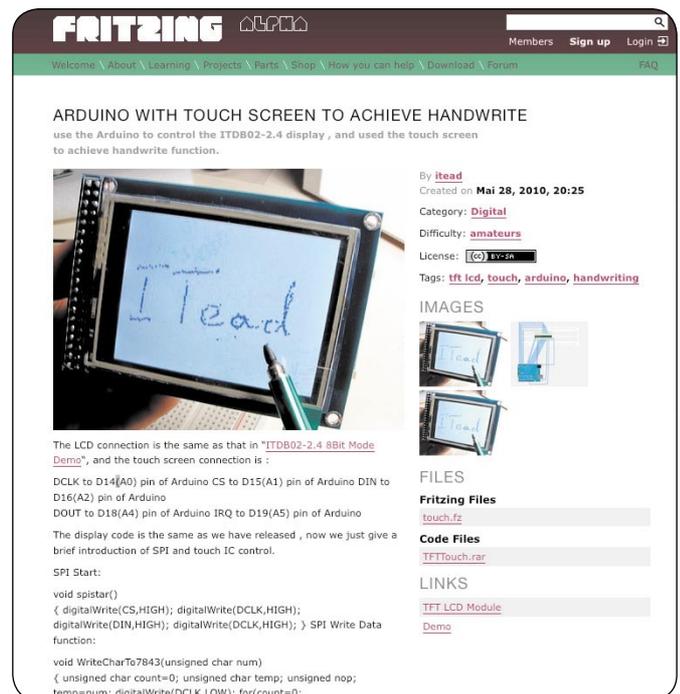


Bild 7: Tolle Projektseite – „Fritzing“

Genauso einfach ist der Programmaufbau eines Arduino-Programms:

- Infotexte und Programmbeschreibung
- Header-Files und Libraries (einfach per Menü ladbar) einblenden
- Globale Variablen anlegen
- Setup-Routine „Void setup()“ (Ports usw. konfigurieren)
- Hauptschleife „void loop()“
- Eigene Prozeduren

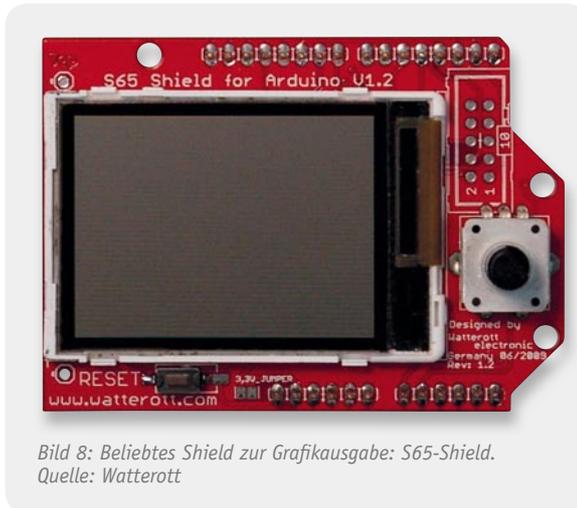


Bild 8: Beliebtes Shield zur Grafikausgabe: S65-Shield.  
Quelle: Watterott

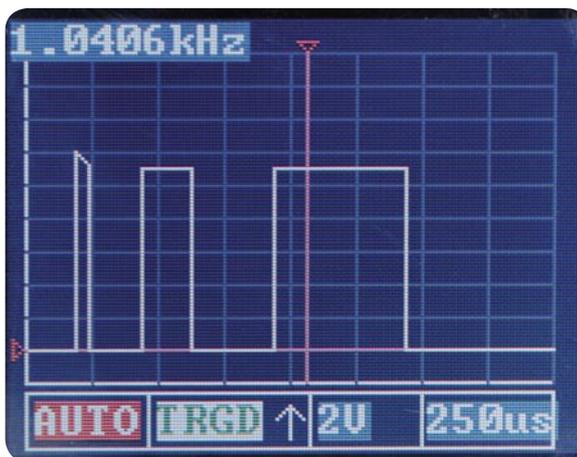


Bild 9: Das S65-Shield in Aktion – hier eine Oszilloskop-Applikation, entstanden in wenigen Stunden aus ein paar Libraries, einem Open Source-Programm und einer individuellen Oberfläche.

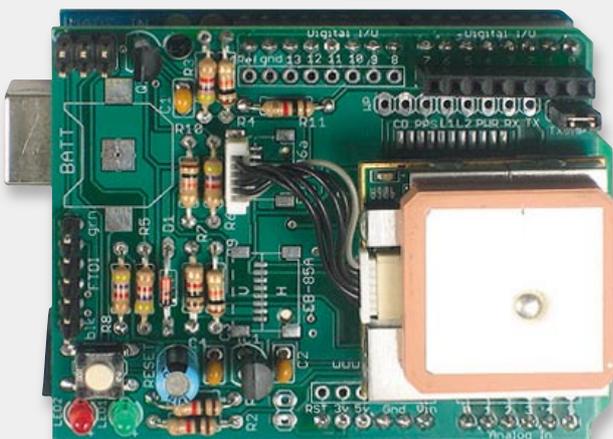


Bild 10: GPS-Datenlogger als Arduino-Shield. Quelle: Watterott

In Bild 6 kann man diesen Aufbau an einem einfachen „Hallo Welt für LEDs“-Programm deutlich sehen. C/C++-Kenner werden es sofort sehen, es ist ein einfacher C-Dialekt, der mit ganz wenigen Grundroutinen auskommt. Natürlich ist es eine komplette Programmiersprache, die hier eingesetzt wird, man kommt also dennoch nicht um einen Kurs in C-Arduino herum, will man nicht beim ausschließlichen Verwenden bereits vorhandener Bibliotheken und Programme stehen bleiben, sondern selbst Anwendungen schreiben.

Hier setzt die Stärke des Begleitbuches des Franzis-Lernpakets an. Es beschreibt natürlich alle Prozeduren der Inbetriebnahme des Boards bis hin zur Einrichtung des USB-Ports, ab Kapitel 9.5 wird es aber interessant. Hier findet der Anwender (und vor allem der Einsteiger) einen kompletten Kurs in Arduino-C, immer untermauert von Beispielen, die dann auch von der CD-ROM abrufbar sind.

Diverse Befehle und Routinen werden dann auch mit den ersten Experimenten begleitet, so dass man das Erlernte unmittelbar in die Praxis umsetzen kann. Hier erkennt man: Ein nacktes Arduino-Board ist bei den einschlägigen Distributoren oder sogar im Eigenbau preiswerter zu haben als das Franzis-Paket, der Wert des Lernpakets offenbart sich in dem ausführlichen und einsteigergerechten Lernmaterial. So komplex und in Deutsch findet man es kaum im Internet, hier herrschen englische Kurse vor, und die deutsche gedruckte Literatur beschränkt sich noch auf nur wenige Bücher. Die Stärken der einschlägigen Internetforen: Hier gibt es Brandaktuelles und dazu wird jede weitere Frage beantwortet.

Um zum Franzis-Lernpaket zurückzukommen – nach dem C-Kurs kommt dann die Praxis. In ca. 30 Experimenten bis hin zur Messtechnik wird immer tiefer in Routineaufgaben und spezielle Aufgaben eingeführt, z. B. PWM oder ADC-Nutzung. Am Schluss treffen wir alte Bekannte aus dem AVR-BASCOM-Lernpaket [5] wieder: die Visual-Basic-Anwendung „6-Kanal-Voltmeter“, den VB.NET-Spannungsplotter, das VB.NET-Speicheroszilloskop sowie den Datenlogger StampPlot. Auch hierfür wird die Programmierung mit Arduino anhand schon recht komplexer Aufgaben gezeigt.

## Lösungen aus dem Netz

Hat man dies alles durchgearbeitet, ist man gerüstet für die Praxis. Und die hat den Vorteil einer riesigen Gemeinde mit ständig aktuellen Foren, Unmengen komplett veröffentlichter Projekte und allem Wissen (sowie Bezugsquellen), das man benötigt, um aus der Arduino-Platine ein funktionierendes Gerät für das eigene Projekt zu machen. Beispiele hierfür finden sich u. a. unter [6], [7], [8], [9] und [10]. In Deutschland besonders aktiv ist – neben „Watterott“ – „Fritzing“, eine Initiative des Interaction Design Lab der Fachhochschule Potsdam. Die Liste der Projekte und das qualifizierte Forum sind eine unerschöpfliche Fundgrube für jeden Arduino-Fan. In der Projektliste findet sich alles – vom einfachsten Einsteigerprojekt bis zur Handschriften-Erfassung per Touchscreen (Bild 7), wie üblich mit kompletter Hardware- und Projektbeschreibung sowie der Software dazu. Ähnliche Projektsammlungen „hängen“ an jeder Arduino-/Freeduino-Seite.

Und da man eben für eigene Anwendungen noch mehr braucht als das Steckbrett, bieten sich neben der völlig allein entwickelten und gebauten Hardware-Ergänzung die „Shields“ an.

## Arduino und die Shields

Shields sind Hardware-Zusätze, die mechanisch und mit ihren Steckverbindern genau auf die Arduino-Boards passen, Bild 8 zeigt eines der derzeit populärsten, bei [7] erhältlich, ein Aufsatz, der das preiswert erhältliche (oder zusammen mit einem Bausatz bzw. Fertiggerät lieferbare) Farbdisplay des Siemens-Handys S65 als grafisches Ausgabegerät nutzt und praktischerweise gleich noch einen Drehgeber und auf der Rückseite einen microSD-Sockel mitbringt. Daraus haben sich unzählige Elektroniker inzwischen die tollsten Geräte gebaut, vom Haustechnik-Display über Mini-Oszilloskope (Bild 9) bis hin zum Videospiel. Und überall gibt es die fertige Open-Source-Software gleich mit dazu, die man sich natürlich nach seinen eigenen Wünschen anpassen und wieder veröffentlichen kann – vielleicht sucht ja schon jemand genau die Library, die Sie gerade „gebaut“ haben!

Die Bilder 10 und 11 zeigen zwei weitere populäre Shields, eines für den Aufbau eines GPS-Loggers und eines für die Anbindung an ein Ethernet-Netzwerk. Mit Letzterem lässt sich der Arduino z. B. zum Standalone-Webserver machen, oder man erfasst Messwerte über das Netzwerk. Auch die Funkanbindung per Xbee/ZigBee-Modul lässt sich einfach lösen.

Die Shields sind meist sowohl als Bausatz (Bild 12) als auch als Fertiggerät zu moderaten Preisen erhältlich. Natürlich bleibt es jedem unbenommen, auch seine eigenen Shields oder Zusatzgeräte zu entwerfen und zu bauen.

Eines der weltweit bekanntesten Projekte soll den Abschluss unserer Vorstellung bilden. Es ist das „ArduPilot“-Projekt von DIY Drones (Bild 13, [11]). Es basiert auf dem speicher- und leistungsstärkeren ArduMega-Board und steuert bis hin zur völligen Autonomie Flugmodelle wie Luftschiffmodelle oder Quadrocopter. Gerade dieses Beispiel beweist die Leistungsfähigkeit des Arduino-Konzepts, das sicher weitere viele Freunde finden wird. **ELV**

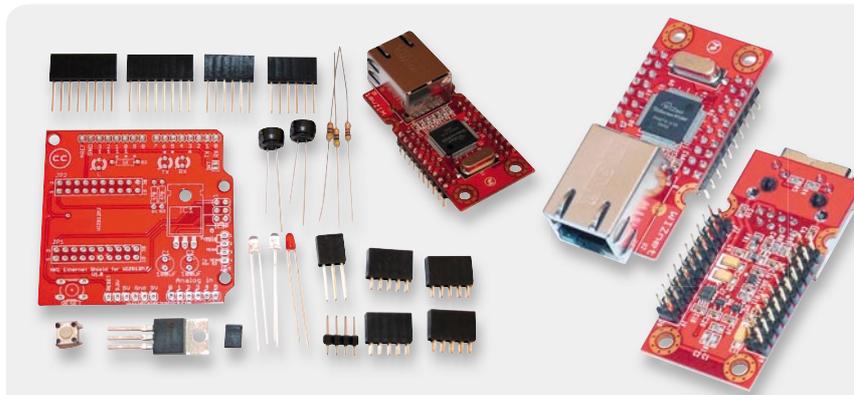


Bild 11: Ethernet-Shield von NKC. Quelle: Watterott

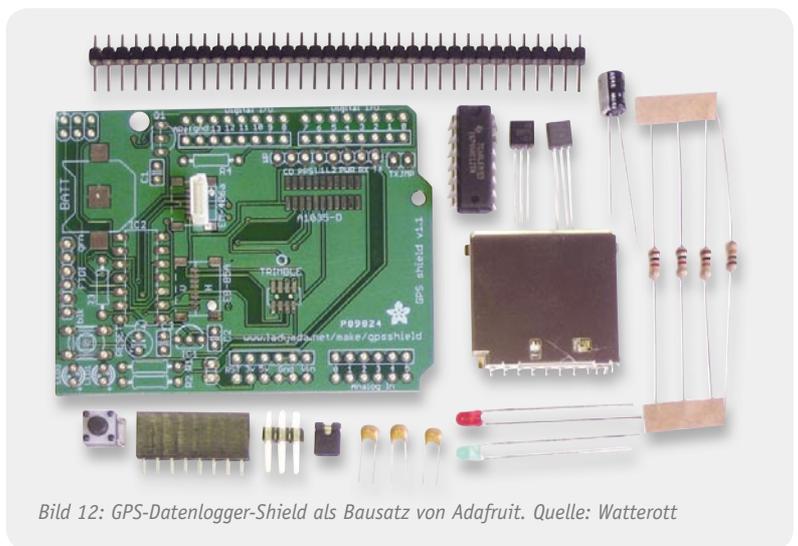


Bild 12: GPS-Datenlogger-Shield als Bausatz von Adafruit. Quelle: Watterott

### arducopter

Arduino-based autopilot for multicopter craft, from quadcopters to traditional helis

Project Home
Downloads
Wiki
Issues
Source

Summary
Updates
People

## ArduCopter Quad - The Full Featured UAV Multicopter!

They are coming...

DIY DRONES  
ArduCopter

AC Quad Beta

AeroQuad has teamed up with [DIY Drones](#) to develop a very full featured multicopter UAV!

It is based on the ArduPilot Pro Mega (APM) and the APM sensor board IMU 1.4 nicknamed the "APM Shield/Oil Pan". Together with a GPS module, it is a full UAV, with scripted waypoints, Ground Station and more.

ArduCopter Quad is just the first member of a complete Arduino compatible UAV platform for rotary wing craft. Stay tuned!

Here's an initial ArduCopter Platform feature list and software road map.

Please chime in! Your input is valuable! We'll keep updating our Wiki frequently.

Bild 13: Spektakuläre Arduino-Anwendung: das ArduPilot-Projekt von DIY Drones



### Weitere Infos:

- [1] [www.arduino.cc](http://www.arduino.cc)
- [2] [www.arduino.cc/en/Main/Hardware](http://www.arduino.cc/en/Main/Hardware)
- [3] [www.freeduino.org](http://www.freeduino.org)
- [4] [www.processing.org](http://www.processing.org)
- [5] ELVjournal 3/2010, S. 54
- [6] [www.freeduino.de](http://www.freeduino.de)
- [7] [www.watterott.com](http://www.watterott.com)
- [8] [www.elmicro.com](http://www.elmicro.com)
- [9] <http://fritzing.org>
- [10] [www.arduino.org](http://www.arduino.org)
- [11] <http://diydrone.com>