Inklusive CD

Ulli Sommer

Arduino™

Mikrocontroller-Programmierung mit Arduino™/Freeduino

- Über 70 Quellcodes zu den Experimenten
- Open-Source-VB.NET-Programme zum Messen und Steuern

FRANZIS

Ulli Sommer

Arduino™

Mikrocontroller-Programmierung mit Arduino™/Freeduino

Ulli Sommer

Arduino™

Mikrocontroller-Programmierung mit Arduino™/Freeduino

- Über 70 Quellcodes zu den Experimenten
- Open-Source-VB.NET-Programme zum Messen und Steuern

Bibliografische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte Daten sind im Internet über http://dnb.ddb.de abrufbar.

Satz: DTP-Satz A. Kugge, München
art & design: www.ideehoch2.de
Druck: C.H. Beck, Nördlingen
Printed in Germany

ISBN 978-3-645-65147-9

Vorwort

Vielen fällt der Einstieg in die Mikrocontroller-Programmierung und die dazugehörige Elektronik schwer. Bei den meisten Mikrocontroller-Systemen muss man sich erst durch unzählige und für den Anfänger nur schwer verständliche Datenblätter wühlen, bevor man überhaupt mit der eigentlichen Programmierung beginnen kann. Und auch bei der Programmierung ist man stets auf das Datenblatt angewiesen. Die Programmieroberflächen sind zudem meist sehr kompliziert und für den Profi-Programmierer ausgelegt, der bereits Erfahrung mit Mikrocontrollern besitzt. Somit bleibt manchem, der an Mikrocontrollern und dem Programmieren interessiert ist, der Zugang in diese Welt verwehrt.

Arduino ist eine leicht verständliche und schnell zu erlernende Open-Source-Plattform, basierend auf einem Mikrocontrollerboard mit einem Atmel-AVR-Controller und einer einfach gehaltenen Programmierumgebung. Für die Interaktion zwischen Mensch und Mikrocontroller können diverse analoge und digitale Sensoren angeschlossen werden, die die Umwelt erfassen und die Daten an den Mikrocontroller weitergeben. Der Mikrocontroller verarbeitet die eingehenden Daten, und durch das Programm entstehen neue Ausgabedaten in analoger oder digitaler Form. Hierbei sind der Kreativität des Entwicklers keine Grenzen gesetzt. Egal, ob Sie eine Haussteuerung bauen möchten oder eine LED-Leuchte mit Farbwechsel basteln – mit Arduino ist es auch für den Quereinsteiger leicht, die ersten funktionsfähigen Programme zu schreiben und so seine Ideen zu verwirklichen.

Das einfache Zusammenspiel aus Hard- und Software bildet die Basis für »Physical Computing«, der Verbindung der realen Welt mit der des Mikrocontrollers, die aus Bits und Bytes besteht.

Dieses Buch ist eine erweiterte Auskoppelung aus dem Franzis Arduino-Lernpaket. Es vermittelt die Grundlagen der Arduino-Programmierung und einige Kniffe, die Sie bei Ihren eigenen Projekten verwenden können. Ein paar Grundkenntnisse in der Elektronik sollten Sie jedoch bereits mitbringen, da in diesem Buch nicht auf jedes Detail eingegangen werden kann. Es ist zudem ein Vorteil, wenn Sie sich bereits mit einer Programmiersprache beschäftigt haben, weil Sie dann die Zusammenhänge leichter verstehen. Aber auch der blutige Anfänger wird mit etwas Eigeninitiative schnell Erfolge erzielen.

CD zum Buch

Diesem Buch liegt eine CD bei, die verschiedene Programme, Tools, Datenblätter und Beispiele enthält. Die CD erleichtert das Arbeiten mit diesem Buch. Die hier abgedruckten Beispiele sind auf der CD enthalten.

Inhalt der CD

- Arduino-Entwicklungsumgebung (IDE)
- Beispiel-Programmcode

- diverse Tools
- Datenblätter
- Schaltpläne

GPL (General Public License)

Sie können Ihre eigenen Programme mit anderen Anwendern über das Internet austauschen. Die Beispielprogramme stehen unter der Open-Source-Lizenz GPL (General Public License). Daher sind Sie berechtigt, die Programme unter den Bedingungen der GPL zu modifizieren, zu veröffentlichen und anderen Anwendern zur Verfügung zu stellen, sofern Sie Ihre Programme dann ebenfalls unter die GPL-Lizenz stellen.

Systemvoraussetzungen:

Windows XP/Vista/7/8 32 Bit und 64 Bit, Linux 32 Bit und 64 Bit, Mac OS X, CD-Laufwerk, Java

Näheres finden Sie auf den Internetseiten zu den verwendeten Programmen:

www.arduino.cc

www.fritzing.org

www.processing.org

Updates und Support:

Arduino wird ständig weiterentwickelt. Updates können kostenlos von der Website *http://arduino.cc* heruntergeladen werden.

Sicherheitshinweise

Die Arduino-Platine ist weitgehend gegen Fehler abgesichert, sodass es kaum möglich ist, den PC zu beschädigen. Die Anschlüsse der USB-Buchse sind auf der Platinenunterseite nicht isoliert. Wenn Sie die Platine auf einen metallischen Leiter stellen, kann es daher zu einem höheren Strom kommen, was den PC und die Platine beschädigen könnte.

Beachten Sie bitte die folgenden Sicherheitsregeln:

- Vermeiden Sie metallische Gegenstände unter der Platine oder isolieren Sie die gesamte Unterseite mit einer nicht leitenden Schutzplatte oder Isolierband.
- Halten Sie Netzteile, andere Spannungsquellen oder führende Leiter mit mehr als 5 V von der Experimentierplatine fern.
- Schließen Sie die Platine nach Möglichkeit nicht direkt an den PC an, sondern über einen Hub. Dieser enthält meist eine zusätzliche wirksame Schutzschaltung. Wenn dennoch etwas passiert, wird im Normalfall nur der Hub und nicht der PC beschädigt.

Inhaltsverzeichnis

1 Mikrocontroller-Grundlagen

Mikrocontroller werden vor allem im Bereich der Automatisierungs-, Mess-, Steuer- und Regeltechnik eingesetzt. Der Vorteil eines Mikrocontroller-Systems ist, dass es auf kleinstem Raum energie- und kosteneffizient physikalische Größen misst und interpretiert, darauf aufbauend Entscheidungen trifft und Aktionen durchführt. Im Grunde ist jede Aufgabe, die man mit Arduino lösen möchte, eine MSR(Messen, Steuern, Regeln)-Aufgabe.

1.1 Messen

Unter Messen versteht man im Allgemeinen, physikalische Eingangswerte von z. B. Schaltern, Lichtsensoren, Drucksensoren, Bewegungsmeldern, Lichtschranken, Spannungsteilern (Potenziometern) u. v. m. zu erfassen. Diese werden dem Controller über die digitalen oder analogen Eingänge zugeführt. Arduino versteht Signale mit einem Spannungspegel von 0 Volt (V) oder 5 V bei den digitalen Eingängen. Die analogen Eingänge hingegen können 0 V bis 5 V mit einer Auflösung von 10 Bit auswerten. Dabei entsprechen 0 V 0 und 5 V 1.023. Diese Werte können in der Software entsprechend interpretiert werden, um z. B. eine Mignonzelle auf ihre Spannung zu überprüfen. Für größere Spannungen an den Eingängen, egal ob digital oder analog, benötigen wir einen Spannungsteiler, der die Eingangsspannung auf die maximale Eingangsspannung der Arduino-Ports herunterteilt. Dazu später mehr.

1.2 Steuern

Unter Steuern versteht man, auf einen Eingangswert zu reagieren. Ein einfaches Beispiel ist ein Bügeleisen. Es steuert die Temperatur über einen Bimetallschalter. Ist die Bügelfläche kälter als am Bügeleisen eingestellt, wird so lange geheizt, bis die gewünschte Temperatur erreicht ist. Fällt die Temperatur unter einen bestimmten Wert, wird nachgeheizt, bis die Temperatur wieder stimmt. Zwischen den beiden Punkten »Heizung ein« und »Heizung aus« befindet sich ein kleines Fenster, eine sogenannte *Hysterese.* Sie sorgt dafür, dass nicht zu oft ein- bzw. ausgeschaltet wird, dadurch die mechanischen Kontakte nicht zu schnell verschleißen und die Steuerung nicht zu »nervös« reagiert. Bei einem Mikrocontroller könnte man das Bimetall durch einen Temperatursensor ersetzen. Der Mikrocontroller würde den Wert analog oder digital über einen passenden Temperatursensor erfassen, das Programm würde die Schwellenwerte für »ein« und »aus« vergleichen und einen digitalen Ausgang schalten, der wiederum ein Relais oder einen Transistor ansteuert, um die Heizung zu schalten.

1.3 Regeln

Der Unterschied zwischen Steuern und Regeln besteht darin, dass eine Steuerung nur bestimmte Ein/Aus-Zustände kennt. Eine Regelung hingegen ist stufenlos. Ein Tempomat im Auto ist z. B. eine Regelung, die versucht, immer die gespeicherte Geschwindigkeit zu halten. Wäre hier nur eine Steuerung verbaut, wäre die Fahrt mit Tempomat sehr unangenehm, da dieser nur Gasgeben, Nichtstun und Bremsen kennen würde.

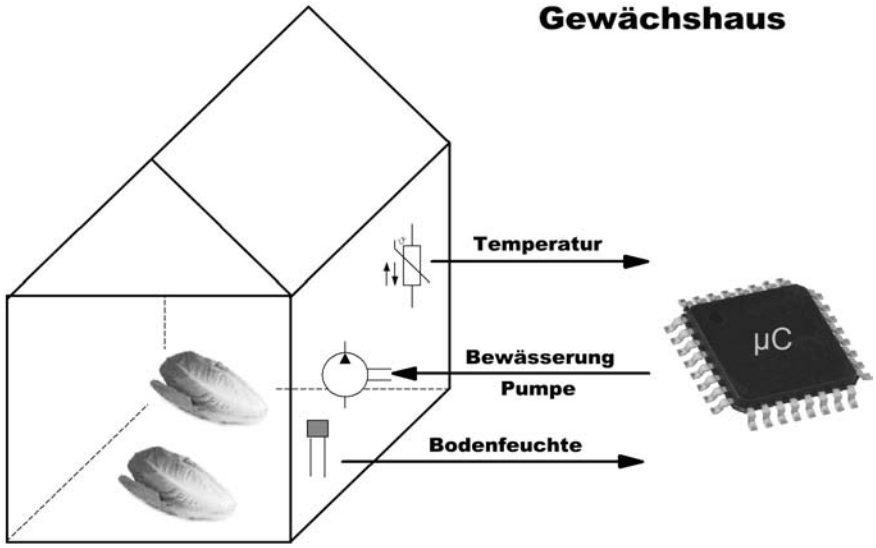

Bild 1.1: Beispiel einer Ein- und Ausgabeverarbeitung anhand eines Gewächshauses

Das Spektrum möglicher Anwendungen von Mikrocontrollern reicht vom privaten (z. B. der Steuerung eines Gewächshauses, des Aquariums oder der Hausbeleuchtung) bis zum industriellen Bereich, in dem komplette Produktionsanlagen mit Mikrocontroller-Systemen gesteuert, gewartet und betrieben werden können. Die Abbildung zeigt eine typische Datenverarbeitung zur Steuerung der Bewässerungsanlage eines Gewächshauses. Der Mikrocontroller nimmt dabei über Sensoren die Messwerte der Umgebungstemperatur und der Bodenfeuchte auf. Die Messwerte werden durch eine digitale Logik in Form eines Programms im Mikrocontroller (kurz: *µC* oder *MC*) interpretiert und es wird bestimmt, ob die Pumpe nun gießen soll oder nicht.

1.4 Aufbau und Funktionsweise

Als vollwertiger Computer im Kleinstformat weist jeder Mikrocontroller – ähnlich einem PC – grundlegende elektronische Bausteine auf. Grundbausteine eines jeden Mikrocontrollers sind die CPU, der Arbeitsspeicher (RAM) sowie der Programmspeicher (Flash) und die Peripherie.

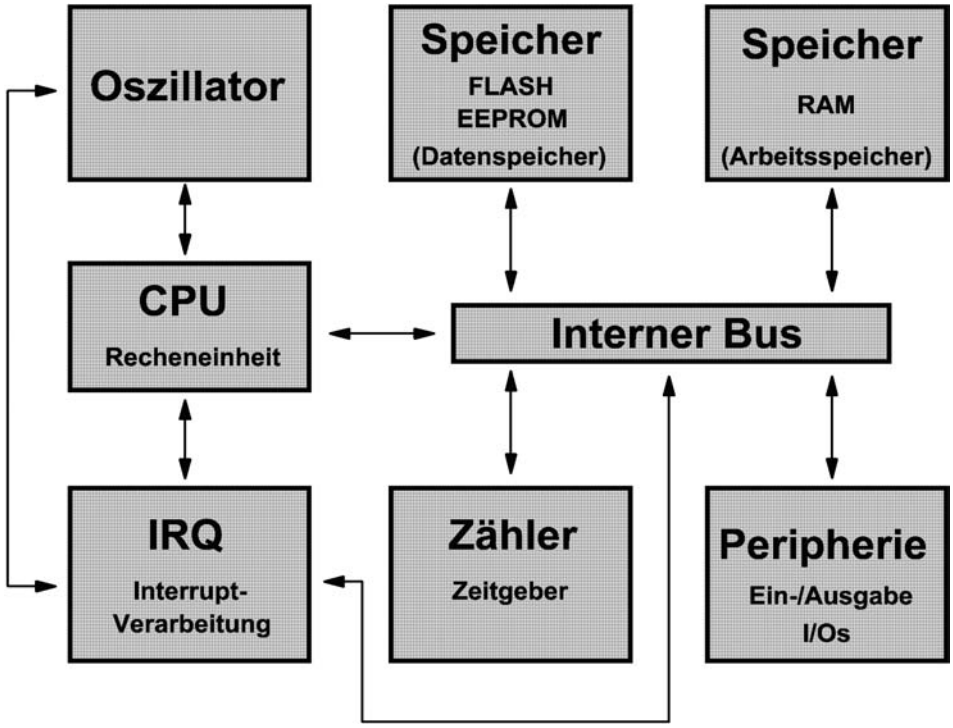

Bild 1.2: Prinzipieller Aufbau eines Mikrocontrollers

1.4.1 CPU

Die wichtigste Funktionseinheit ist die zentrale Recheneinheit, die CPU (engl.: Central Processing Unit). Sie kann als das »Gehirn« des Mikrocontrollers verstanden werden. Dort werden die Befehle und arithmetische Operationen abgearbeitet.

1.4.2 Arbeits- und Programmspeicher

Arbeits- und Programmspeicher sind logisch getrennt. Das Benutzerprogramm wird dabei in einem nichtflüchtigen Flash-Speicher abgelegt, dem Programmspeicher. Je nach Controllersystem kann man auf (implementierten) Programmspeicher von mehreren Kilo- bis Megabyte zurückgreifen. Bei einigen Systemen ist es darüber hinaus möglich, den Programmspeicher durch externe Flash-Komponenten aufzustocken. Der Arbeitsspeicher (RAM) dient zur temporären Ablage von Rechen-, Mess- und Steuergrößen. Ziel ist, möglichst schnell auf eine begrenzte Anzahl von Daten zugreifen zu können. Der RAM(Random Access Memory)-Speicher ist in der Regel deutlich kleiner als der Flash-Speicher, dafür aber um ein Vielfaches schneller. Die Werte des RAM werden zur Laufzeit erzeugt und sind, anders als beim Flash-Speicher, flüchtig. D. h., dass nach einem Neustart des Controllers im RAM keine Werte gespeichert sind.

1.4.3 Peripherie

Als *Peripherie* bezeichnet man jene Komponenten eines Mikrocontrollers, die nicht durch CPU und Speicherbausteine abgedeckt werden, insbesondere Komponenten, die eine Schnittstelle zur Außenwelt darstellen. Digitale Ein- und Ausgänge (kurz: I/O für Input/Output) werden z. B. zu den Peripheriebausteinen gezählt. Die meisten Mikrocontrollerboards, auch unser Arduino, bieten eine Vielzahl von digitalen und analogen Ein- und Ausgängen mit verschiedenen Funktionen.

1.5 Programmierung der Mikrocontroller

Ein Programm ist die Beschreibung eines Informationsverarbeitungsprozesses. Im Lauf eines solchen Prozesses wird aus einer Menge von variablen oder konstanten Eingangswerten eine Menge von Ausgangswerten berechnet. Die Ausgangswerte sind entweder selbst Ziel der Informationsgewinnung oder dienen mittelbar zur Reaktion auf die Eingangswerte. Neben den eigentlichen Berechnungen kann ein Programm Anweisungen zum Zugriff auf die Hardware des Computers oder zur Steuerung des Programmflusses enthalten. Ein Programm besteht aus mehreren Zeilen sogenannten Quelltextes. Dabei enthält jede Zeile eine oder mehrere Rechen- oder Steueranweisungen. Neben diesen Anweisungen selbst bestimmt ihre Reihenfolge wesentlich die eingangs beschriebene Informationsverarbeitung. Die Ausführung der den Anweisungen entsprechenden Operationen durch den Steuercomputer erfolgt sequenziell, also der Reihe nach. Eine Folge von Programmanweisungen mit einem bestimmten Ziel nennt man auch *Algorithmus*.

1.5.1 Arduino versteht C

C oder auch ANSI-C ist eine einfach zu erlernende Programmiersprache. C ist eine imperative Programmiersprache, die der Informatiker Dennis Ritchie in den frühen 70er-Jahren an den Bell Laboratories für das Betriebssystem Unix entwickelt hat. Seitdem ist sie weltweit stark verbreitet. Die Anwendungsbereiche von C sind sehr verschieden. Es wird z. B. zur System- und Anwendungsprogrammierung eingesetzt. Die grundlegenden Programme aller Unix-Systeme und die Systemkerne vieler Betriebssysteme sind in C programmiert. Zahlreiche Sprachen wie C++, Objective-C, C#, Java, PHP oder Perl orientieren sich an der Syntax und anderen Eigenschaften von C. Es ist also mehr als lohnend, sich mit dieser Programmiersprache zu beschäftigen, da man später auch leicht auf andere Mikrocontrollersysteme umsteigen kann. Für fast alle Mikrocontroller existiert ein freier C-Compiler, den die Hersteller zum Download anbieten. Das C von Arduino ist jedoch um einiges einfacher gehalten als die professionellen C-Compiler und nimmt dem Anwender dadurch sehr viel Arbeit ab. Vor allem um die komplizierten Hardware-Routinen muss man sich bei Arduino nicht kümmern, da sie bereits als feste Befehle in der Entwicklungsumgebung integriert sind. Zudem gibt es mittlerweile für fast jede Hardware eine Arduino-Bibliothek, die einfach eingebunden wird – und schon kann man mit der neuen Hardware, z. B. einem digitalen Drucksensor, kommunizieren.

2 Übersicht über die Arduino-Boards

Die Arduino-Hardware verwendet ausschließlich gängige, allgemein verfügbare Bauteile. Daher ist es leicht, die Funktionsweise zu verstehen und die Schaltung an eigene Wünsche anzupassen oder Erweiterungen vorzunehmen. Den Kern bildet ein ATmega-Controller aus Atmels weitverbreiteter 8-Bit-AVR-Familie. Hinzu kommen Schaltungsteile zur Stromversorgung und eine serielle Schnittstelle. Letztere ist bei den neueren Arduino-Versionen als USB-Interface ausgelegt. Über diesen Anschluss erfolgt der Download unserer Programme und bei Bedarf auch die Kommunikation zwischen PC und Arduino während der Programmausführung, um z. B. der Hardware Kommandos zu erteilen oder Messwerte von Arduino zu lesen.

Weil Arduino-Boards so einfach und universell ausgelegt sind, werden sie häufig auch schlicht als *I/O-Board* bezeichnet. Das Arduino-UNO-Board stellt dem Anwender 14 digitale Ein-/Ausgänge (I/Os) zur Verfügung. Davon sind sechs als Analogausgang (8 Bit PWM) zu verwenden. Weitere sechs Eingänge können analoge Signale erfassen (10 Bit ADC). Bei Bedarf stehen SPI und I^2C als weitere digitale Schnittstellen zur Kommunikation mit anderen Baugruppen wie Sensoren bereit.

Arduino-Boards gibt es in mehreren Varianten. Seit Kurzem gibt es neben den klassischen 8-Bit-Arduinos auch eine 32-Bit-Variante namens *Arduino DUE* und einen Atmel SAM3X8E ARM Cortex-M3 mit viel Rechenleistung für komplexeste Aufgaben. Die Original-Arduino-Platinen stammen vom Hersteller Smart Projects aus Italien und sind sehr günstig über diverse Internetshops zu beziehen. Es gibt auch zahllose Klone und Nachbauten von anderen Anbietern, schließlich handelt es sich um Open Hardware. Ein wichtiger Unterstützer des Arduino-Projekts ist SparkFun aus Boulder, Colorado. Die Kooperation mit dem US-Partner hat eine Reihe optimierter Arduino-Boards und Unmengen von Sensoren und Aktoren hervorgebracht. Außerdem ist mit LilyPad ein wichtiger Ableger entstanden, der das Thema Wearable Computing aufgreift.

Die meisten Anwender setzen auf das von Smart Projects gefertigte Board Arduino UNO. Es dient auch im vorliegenden Buch als Grundlage für die Experimente.

Weitere Varianten sind die Arduino-Mega-Boards mit einem leistungsstärkeren Mikrocontroller (Atmega1280), der mehr Speicher, I/O-Pins und Funktionen bietet.

Wesentlich kleiner ist Arduino Mini, ein Board im DIP24-Format. Das ganze Modul lässt sich auf einen 24-poligen DIL-Sockel stecken. Die Version Arduino Pro Mini von SparkFun ist nahezu identisch damit, wird aber ohne »Beinchen« (seitliche Stifte)

geliefert. Diese Module erfordern zum Programmieren einen USB-Adapter, der an der Schmalseite der Module angesteckt werden kann.

Das LilyPad-Board von Leah Buechley (in Zusammenarbeit mit Sparkfun) ist ebenfalls Arduino-kompatibel und verfolgt einen ganz eigenen Zweck. LilyPad und Zubehör sind dafür ausgelegt, in Kleidung eingenäht zu werden, um dort eine möglichst enge Symbiose von Technik und Künstler zu realisieren. Die charakteristische runde Form des LilyPad-Arduinos erregt ebenso Aufmerksamkeit wie die Farbgebung und die kreisförmige Anordnung der Kontakte. Zahlreiche kleine Peripherieplatinen (Sensoren, LEDs, Taster ...) ergänzen LilyPad zu einem ganzen System unter dem Motto »Elektronik mit der Nähmaschine«.

Tipp: Über weitere Board-Versionen und Zubehörteile informieren Sie die Arduino-Projektseite und die Produktseiten von SparkFun Electronics unter *http://www.sparkfun.com/categories/103*.

2.1 Arduino Mega

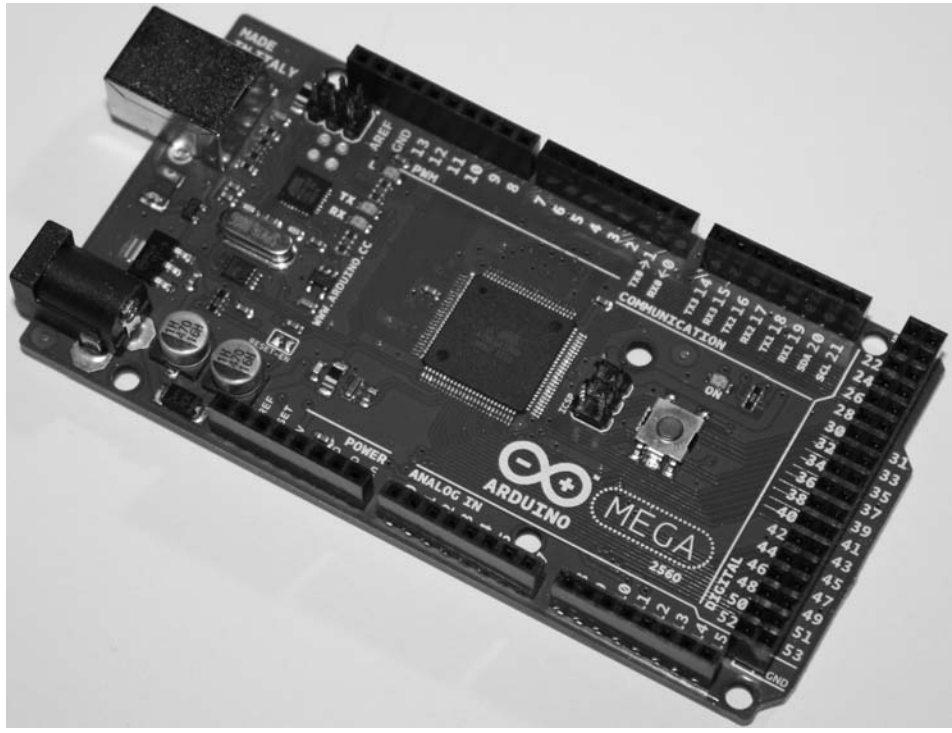

Bild 2.1: Arduino-Board MEGA

Technische Daten

- ATmega2560 Mikrocontroller
- 16-MHz-Takt
- 256 KB Flash (4 KB davon benötigt der Bootloader)
- 8 KB SRAM, 4 KB EEPROM
- 54 digitale I/O-Pins, davon 15 als PWM nutzbar
- Hardware-UARTs
- I^2C-Interface, SPI
- 16 analoge Eingänge (10 Bit)
- USB-Interface, Spannungsversorgung, Bootloader etc. wie beim Arduino Duemilanove
- Abmessungen ca. 101 x 53 x 15 mm

2.2 Arduino UNO

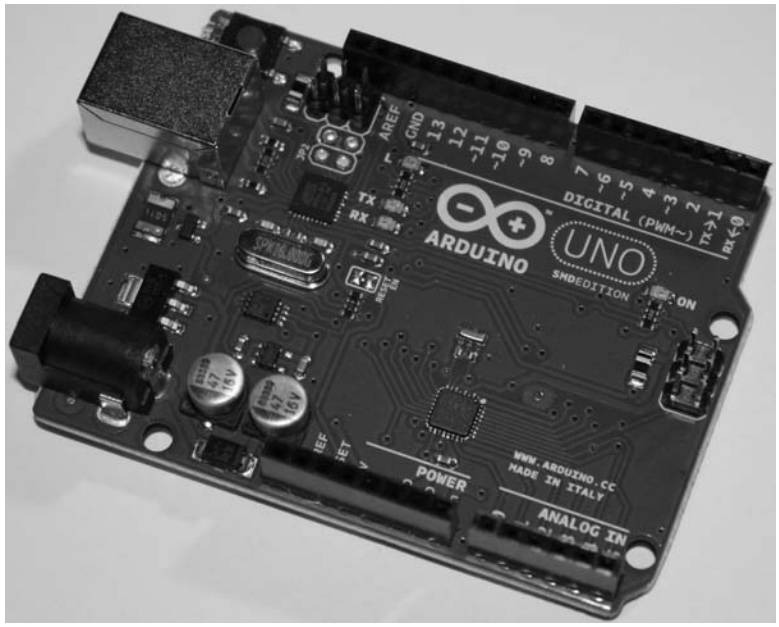

Bild 2.2: Arduino UNO (SMD-Edition)

Dieses Board wird in den folgenden Experimenten verwendet und ist zudem das Arduino-Standard-Board, basierend auf dem Atmel Atmega328P Mikrocontroller. Dieses günstige und leistungsfähige Board dürfte das meistverkaufte Mikrocontroller-Board weltweit sein. Es besitzt 14 digitale I/Os, wovon sechs als PWM-Ausgänge verwendet werden können. Ferner verfügt es über sechs analoge Eingänge, einen 16-MHZ-Oszillator als Taktgeber, eine USB-B-Buchse zur Programmierung und Datenausgabe, einen Reset-Taster, eine Stromversorgungsbuchse und einen ISP-Stecker zur Programmierung über einen Atmel Programmer. Es verträgt als Stromversorgung Gleichspannung. Diese Version ist der direkte Nachfolger der ersten Arduino-Boards und behält die Arduino-Standards, die zu den Anfangszeiten gesetzt wurden, zu 100 % bei. Die großen Unterschiede sind: Platine in SMD-Bestückung, kein FTDI-USB-zu-UART-Brücken-Chip mehr, sondern Atmega16U2, und ein leistungsfähigerer Mikrocontroller.

Technische Daten

- ATmega328P Mikrocontroller
- 16-MHz-Takt
- 32 KB Flash (davon 0,5 KB für Bootloader reserviert)
- 2 KB SRAM, 1 KB EEPROM
- 14 digitale I/O-Pins, davon sechs als PWM nutzbar
- sechs analoge Eingänge (10 Bit)
- On-Board-USB-Schnittstelle mit Mega16U2 von Atmel
- 5 V Betriebsspannung, Speisung über USB oder über Spannungsregler (7 V bis 12 V Eingangsspannung)
- Abmessungen ca. 69 x 53 x 15 mm

2.3 Arduino Leonardo

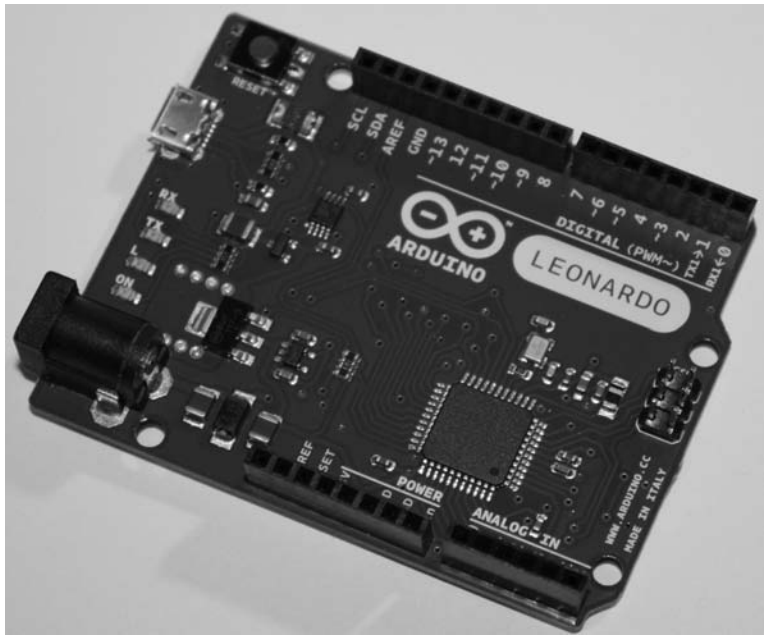

Bild 2.3: Arduino Leonardo

Das Leonardo-Board ist ein auf dem Atmega32U4 basierendes Mikrocontrollerboard. Es besitzt 20 digitale I/Os, wobei sieben davon als PWM-Ausgänge genutzt werden können. Programmiert wird das Board wie üblich über USB und den auf dem ATmega befindlichen Bootloader. Der Unterschied zu den anderen Boards ist, dass hier, statt der üblichen Standard-USB-B-Buchse, eine Mikro-USB-Buchse vorhanden ist. Zudem befindet sich auf diesem Board kein zusätzlicher USB-Chip, der die Kommunikation zwischen PC und Mikrocontroller übernähme. Dieser USB-Chip ist bereits im Mikrocontroller enthalten. Das ermöglicht zudem die Kommunikation zum PC als HID (Human Interface Device, also Maus- und Tastatursimulation).

Technische Daten

- ATmega32U4
- 16-MHz-Takt
- Programmierung über USB
- 5-V-Technik
- 20 digitale I/Os, sieben davon können zur PWM-Erzeugung genutzt werden
- 12 analoge 10-Bit-Eingänge

- 32 KB Flash (4 KB benötigt der Bootloader)
- 1 KB SRAM
- 1 KB EEPROM
- Ausgangsstrom pro I/O max. 40 mA
- Versorgungsspannung 7 V bis 12 V
- Abmessungen ca. 69 x 53 x 15 mm

2.4 Arduino Ethernet

Bild 2.4: Arduino Ethernet Board

Das Arduino Ethernet Board basiert auf einem Atmel-Atmega328-Mikrocontroller. Es besitzt 14 I/Os und sechs analoge Eingänge, einen 16-MHz-Oszillator, einen RJ45-Anschluss, eine Stromversorgungsbuchse, einen ICSP(ISP)-Stecker und einen Reset-Taster. Zudem kann eine Power-over-Ethernet-Platine optional eingelötet werden. Der große Unterschied zwischen diesem und anderen Arduino-Boards ist, dass hier anstatt der USB-Buchse eine Ethernet-Buchse zur Programmierung vorhanden ist. Die Ethernet-Kommunikation übernimmt der auf dem Board befindliche Wiznet-Ethernet-Controller, der an den Atmega328 angeschlossen ist. Der microSD-Card-Leser kann zum Speichern/Lesen von Daten und Internetseiten verwendet werden.

Technische Daten

- ATmega328
- 16-MHz-Takt
- Programmierung über Ethernet RJ45
- 5-V-Technik
- 14 digitale I/Os, vier davon können zur PWM-Erzeugung genutzt werden
- 6 analoge 10-Bit-Eingänge
- 32 KB Flash (0,5 KB benötigt der Bootloader)
- 2 KB SRAM
- 1 KB EEPROM
- Ausgangsstrom pro Port +/–40 mA
- Versorgungsspannung 7 V bis 12 V
- Abmessungen ca. 70 x 53 x 15 mm

2.5 ArduPilot

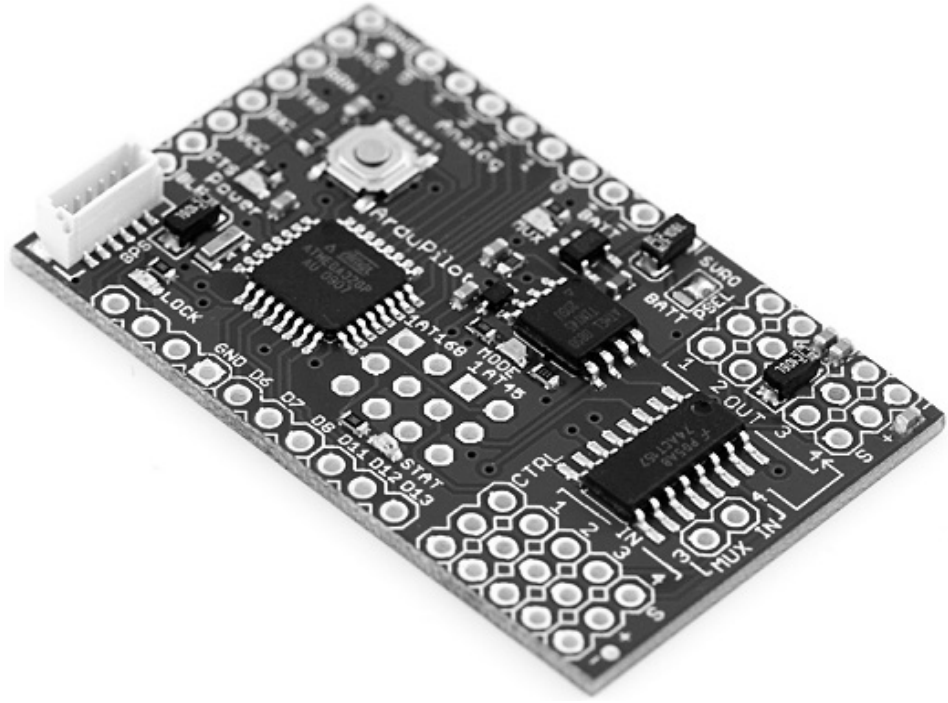

Bild 2.5: ArduPilot – Arduino-kompatibler UAV-Controller ATmega328 (Quelle: SparkFun)

Für die Modellflieger ist der ArduPilot ein äußerst interessantes Spielzeug. Er ermöglicht das autonome Fliegen eines Modellflugzeugs.

Tipp: Mehr dazu finden Sie unter *http://diydrones.com*.

2.6 LilyPad

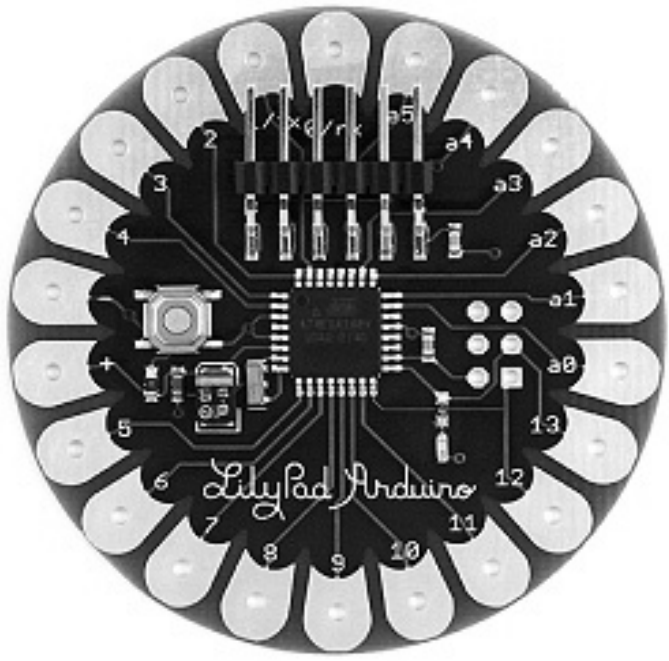

Bild 2.6: LilyPad Arduino (Quelle: Elmicro)

Das LilyPad ist für elektronische Kleidung entworfen worden. Es kann direkt in ein Textilstück eingenäht werden. Die Verbindung zu Sensoren und Aktuatoren kann über leitfähige Fäden hergestellt und die ganze Schaltung unsichtbar verstaut werden. Entworfen wurde das LilyPad von Leah Buechley und SparkFun Electronics.

Technische Daten

- ATmega328V und ältere ATmega168V mit 16-MHz-Quarztakt
- Programmierung über USB-Adapter (Arduino/USB)
- Spannungsversorgung 2,7 V bis 5,5 V
- 14 digitale I/O-Pins (sechs davon als PWM nutzbar)
- 6 analoge 10-Bit-Eingänge

- Ausgangsstrom pro Digital-Port 40 mA
- 32 KB oder 16 KB (ATmega168) Flash (2 KB benötigt der Bootloader für sich)
- 1 KB (ATmega168) oder 2 KB (ATmega328) SRAM
- 512- (ATmega168) oder 1-KB-EEPROM

2.7 USB-Adapter

Bild 2.7: USB-Adapter mit FTDI-Chip (Quelle: Elmicro)

Diesen Programmieradapter gibt es in 3,3-V- und in 5-V-Ausführung. Der Adapter wird zum Programmieren der Arduino-Boards ohne USB-Anschluss, wie dem Arduino Mini, benötigt. Die Pin-Belegung entspricht den originalen Arduino-Spezifikationen. Er kann auch zur Kommunikation (virtuelle serielle Schnittstelle) verwendet werden. Dieses Feature muss man für eigene Entwicklungen einfach haben. Es ermöglicht, einen Sketch auf das Board zu laden, ohne die Reset-Taste zu drücken.

Bezugsquellen

Conrad Electronic SE
Klaus-Conrad-Straße 1
92240 Hirschau
www.conrad.de

Elektronikladen Zentrale
Hohe Straße 9-13
04107 Leipzig

Elektronikladen Vertrieb
Bielefelder Straße 561
32758 Detmold
www.elmicro.com

Roboter-Teile
EDV-Beratung & Robotertechnik Jörg Pohl
Baluschekstraße 9
01159 Dresden
www.roboter-teile.de

Electronic Assembly GmbH
Zeppelinstraße 19
82205 Gilching bei München
http://www.lcd-module.de

Sommer-Robotics
Ulli Sommer
Bahnhofstraße 8
92726 Waidhaus
www.sommer-robotics.de

Ulli Sommer

Arduino™

Mikrocontroller-Programmierung mit Arduino™/Freeduino

Das kleine Controllerboard mit den vielen Möglichkeiten.

Mikrocontroller werden vor allem im Bereich der Automatisierungs-, Mess-, Steuer- und Regeltechnik eingesetzt. Der Vorteil eines Mikrocontroller-Systems ist, dass es auf kleinstem Raum energie- und kosteneffizient physikalische Größen misst und interpretiert, um darauf aufbauend Entscheidungen zu treffen und Aktionen durchzuführen. Im Grunde ist jede Aufgabe, die man mit Arduino™ lösen möchte, eine Aufgabe aus dem Bereich MSR (Messen, Steuern, Regeln).

Damit Ihr Einstieg in die Welt der Mikrocontroller schnell und effizient gelingt, finden Sie in diesem Buch alles, was Sie über die Arduino™-Programmierung wissen müssen. Neben Grundlagenwissen bietet es die besten Tricks und Kniffe, die Sie bei Ihren eigenen Projekten einsetzen können.

Verleihen Sie Ihren Ideen Flügel: Egal ob Sie eine Haussteuerung oder eine LED-Leuchte mit Farbwechsel bauen möchten – mit Arduino™ ist es auch für Quereinsteiger leicht, funktionsfähige Programme zu schreiben und Projekte zu verwirklichen, die den Alltag erleichtern und einfach Spaß machen.

Vom Dämmerungsschalter über Alarmanlagen und Codeschlösser bis zum romantischen Mikrocontroller-Kerzenlicht: Entdecken Sie die Möglichkeiten, die die Arduino™-Programmierung bietet!

Aus dem Inhalt:

Über den Autor:

Ulli Sommer beschäftigt sich seit Jahren professionell mit Mikrocontrollern. Als Autor zahlreicher Bücher und Lernpakete des Franzis Verlags hat er sich auf die Themen Mikrocontroller, Robotik und Elektronik spezialisiert. Er entwirft und fertigt auf Mikrocontrollern basierende Steuerungen, entwickelt Robotersysteme zur Überwachung, Inspektion, Naturbeobachtung und Ausbildung sowie Software im Bereich Automatisierung und Prozessüberwachung.

9 783645 651479

30,– EUR [D] **/ 30,90 EUR** [A]
ISBN 978-3-645-65147-9

Besuchen Sie unsere Website
www.franzis.de

FRANZIS