

# Basteln mit dem IoT – Mikrorechnerlösungen ganz einfach mit Linker Kit

Das Internet der Dinge, kurz IoT, ist DAS Thema derzeit. Wie einfach man auch als reiner Anwender bzw. Programmier-Einsteiger zu Lösungen und eigenen Applikationen kommt, soll unser Beitrag zeigen. Mit dem Linker-Kit-System und der grafischen Programmieroberfläche ArduBlock wird dies zum Kinderspiel, ohne dass man auch nur einmal zum LötKolben oder zur Breadboard-Verkabelung greifen muss.

## Mikrocontroller-Applikation ohne Löten

Das Internet der Dinge ist zwar auf den ersten Blick ein abstrakter Begriff, aber er lässt sich heute quasi

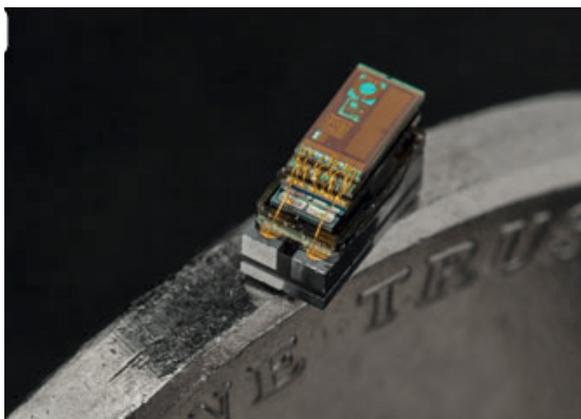


Bild 1: M<sup>3</sup> – der derzeit kleinste eigenständige Mikrocomputer der Welt. Quelle: umich.edu

im Handumdrehen mit Leben füllen. Vereinfacht gesagt: Man nehme ein kleines Mikrocontrollersystem, verbinde es mit einem Sensor und lasse die erfassten Daten an einem gewünschten Ort auswerten, visualisieren oder Reaktionen auslösen. Den Startschuss dafür lieferten bereits vor geraumer Zeit die Erfinder der Einchip-Mikrorechner vom Schläge des AVR, PIC, STM32, Arduino, Raspberry Pi. Derzeit ist hier eine enorme Dynamik zu beobachten – die leistungsfähigen ARM-Plattformen erweitern die Möglichkeiten durch integrierte Betriebssysteme und enorme Schnittstellenvielfalt – fast wöchentlich erscheint eine neue Plattform – und sie werden immer kompakter. Die bisherige Spitze in puncto Miniaturisierung haben sich unlängst Forscher der University of Michigan mit dem M<sup>3</sup> erarbeitet, einem nur einen Kubikmillimeter beanspruchenden kompletten Mikrorechner (Bild 1), der mit verschiedenen Sensoren bestückt werden kann und sogar, etwa zum Zweck der Gesundheitsüberwachung, implantierbar wäre [1].



Die dem normalen Anwender zur Verfügung stehenden Systeme sind freilich noch etwas größer, dafür auch besser handhabbar und universeller. Nun entwickelt sich aber der Kreis von Menschen, die kleine Mikrorechner-Anwendungen selbst entwickeln und bauen wollen, so dass die Schnittmenge zum „gelernten“ Elektroniker immer geringer wird. Für den Programmierer und den Anwender ist es eher interessant, die Kombinationsmöglichkeiten und softwaremäßigen Anwendungen zu nutzen, als sich mit Hardware, Löten von Bauteilen, Platinen und Geräten zu beschäftigen – er betrachtet z. B. eine Sensorplatine als Black Box, allenfalls das Datenblatt des Sensors interessiert noch für die Programmierung. Nur so erreicht das Thema auch eine große Breite, denn vielen des angesprochenen Anwenderkreises ist das Löten ein eher fremdes Handwerk, weshalb das Breadboard hier auch einen festen Stamplatz auf dem Tisch hat. Aber selbst dies ist vielen rein lösungsorientierten Anwendern noch zu viel, so kamen findige Anbieter auf die Idee, es nach dem Baukasten-Prinzip zu realisieren – sie entwickeln Sensor-, Eingabe- und Ausgabebausteine, die über eine einheitliche Schnittstelle und eine einfache Steckverbindung mit dem Mikrorechnersystem verbunden werden. Über diese Schnittstelle, heute fast immer I<sup>2</sup>C, erfolgt dann die gesamte Kommunikation mit den angeschlossenen Bausteinen. Die Verbindung „in die Welt“ wird dann per Netzwerk, Kurzstreckenfunk oder USB hergestellt.

Ein Anbieter eben jener Bausteine ist „LinkSprite“. ELV hat diese Bausteinreihe, genannt „Linker Kit“, in das Shop-Angebot übernommen.

### Es geht auch einfach – Linker Kit

Das System – Bild 2 zeigt einige Bausteine daraus – besteht aus diversen Ein- und Ausgabebausteinen sowie Sensoren, die jeweils per I<sup>2</sup>C mit einem der beliebten Mikrocontrollersysteme „Arduino/pcDuino“, „Raspberry Pi“, „Beaglebone“ und kompatiblen Systemen verbunden werden. Selbst die erforderlichen vierpoligen Steckverbinderleitungen werden einsatzfertig angeboten.

Grundlage des Systems ist eine genial einfache, für viele Peripheriebausteine gleichzeitig einsetzbare Anschlussplattform für das jeweilige Mikrorechnersystem, das „Base Shield“. In Bild 3 ist das Base



Bild 2: Einige Module des Linker-Kit-Systems – allen gemeinsam ist die einheitliche Schnittstelle.

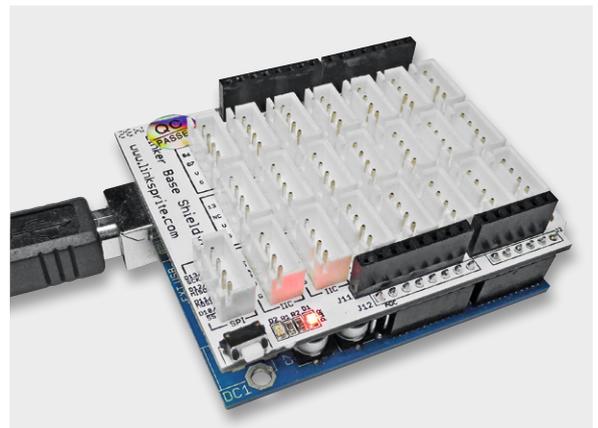


Bild 3: Erweitert den I<sup>2</sup>C-Bus des Arduino um 22 Anschlüsse – Base Shield für Arduino/pcDuino.

Shield für den „Arduino/pcDuino“ zu sehen, es gibt auch eines für Raspberry Pi & Co. Ein Blick auf die Schaltung dieses Shields in Bild 4, das zusätzlich noch einen AD-Wandler beherbergt, lässt erkennen, dass es sich im Prinzip lediglich um eine I<sup>2</sup>C-Bus- und Energieverteilung handelt. Der Einsatz ist denkbar einfach, da diese Base Shields pinkompatibel zu den Expansions- bzw. GPIO-Buchsenleisten der Rechnersysteme sind: aufstecken, fertig!

Ebenso einfach erfolgt der Anschluss der Linker-Kit-Module, sie werden mit dem Base Shield verbunden und somit ist eine Anwendung bereits hardwareseitig fertig (Bild 5).

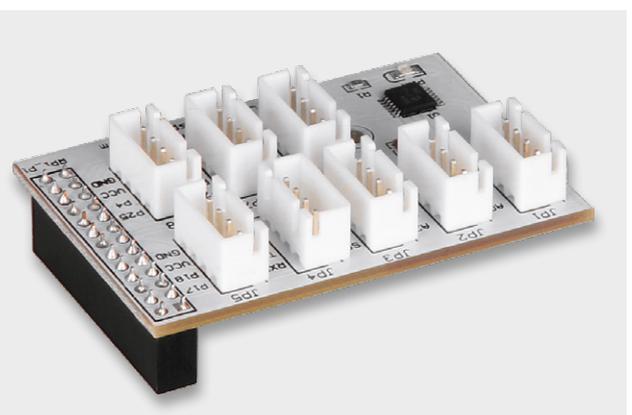
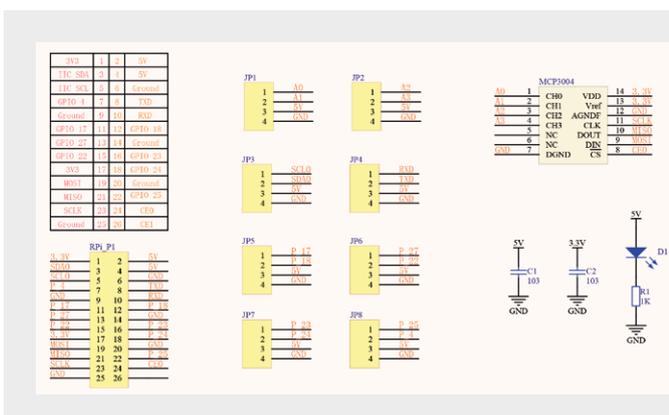


Bild 4: Die Schaltung des Linker-Kit-Base-Shields für den Raspberry Pi – dieses trägt zusätzlich einen 10-Bit-A/D-Wandler MCP3004.

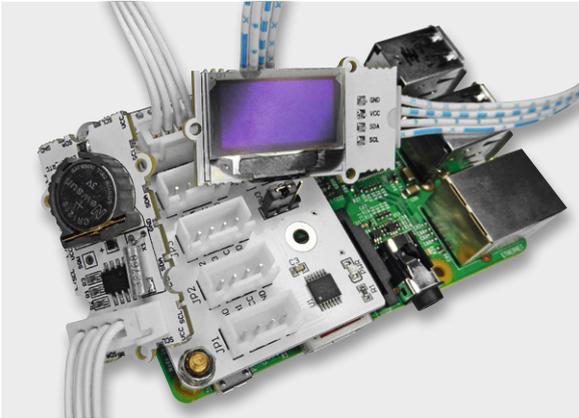


Bild 5: Echtes Plug & Play – Hardwarebaustein anstecken, fertig!

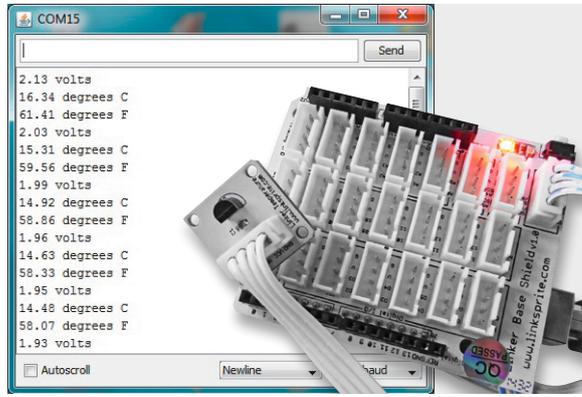


Bild 6: So einfach erfolgt zum Beispiel die Erfassung und Ausgabe der Temperatur im Linker-Kit-System.

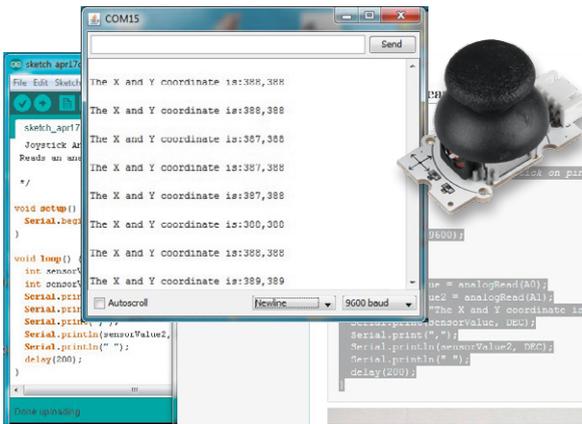


Bild 7: Einfache Ausgabe der Joystickstellung – ebenfalls als Programmbeispiel hinterlegt

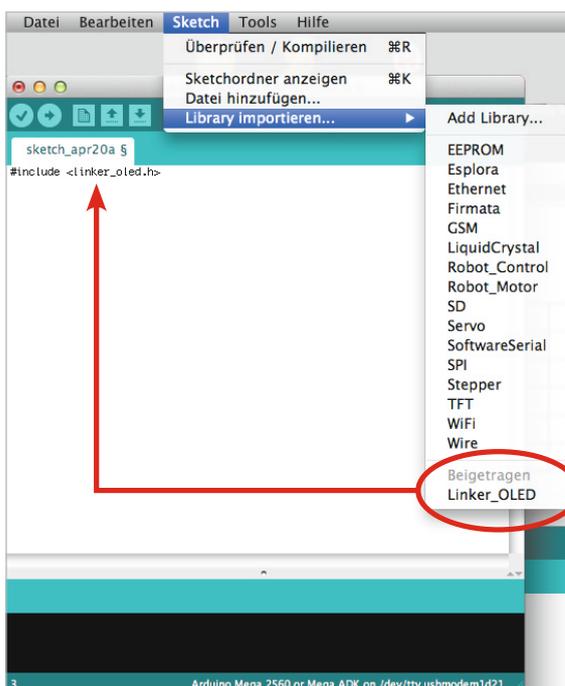


Bild 8: Auch komplette Bibliotheken (Libraries) werden zu den Bausteinen angeboten.

Danach geht es nur noch softwareseitig weiter. Je nach Mikrorechnersystem gibt der Hersteller hier eine sehr umfangreiche Unterstützung in Wiki-Form ([2], hier sind alle Linker-Kit-Komponenten weiter unten mit Direktlinks ins Wiki aufgeführt), hier findet sich alles Wissen zu jedem Modul inklusive Softwarebeispiel, das z. B. direkt in die Arduino-IDE übertragen, kompiliert und zum Arduino geschickt werden kann. So kommt man in Minutenschnelle selbst als totaler Einsteiger zum „Hello World“-Erlebnis. Bild 6 zeigt ein Beispiel dazu mit dem Temperatursensor des Systems. Das Beispielprogramm erzeugt eine serielle Ausgabe der Messspannung sowie der Temperatur in Grad Celsius und Fahrenheit direkt über den seriellen Monitor der Arduino-IDE. Das Originalprogramm verfährt hier allerdings ungewollt nach dem Prinzip „finde den Fehler und habe deinen ersten Programmverständnis-Erfolg!“ – der Umrechnungsfaktor von Spannung auf Temperatur ist falsch. Weitere Applikationen visualisieren die Sensor- oder Bedienelementeingaben ähnlich einfach über den seriellen Monitor, andere geben direkt über die LED- oder Anzeigebausteine, Relais oder Signalgeber des Systems aus. So ist in Bild 7 die Abfrage des Joystick-Eingabebausteins zu sehen.

Mit diesen hervorragend präsentierten Grundkenntnissen versehen, ist der Weg zur eigenen Anwendung, etwa das Aufleuchten einer LED bei Erreichen einer bestimmten Temperatur, nur noch ganz kurz. Danach folgt dann das Einarbeiten in eine Displaysteuerung, und schon haben wir ein Digitalthermometer. Das Einarbeiten geht hier dermaßen spielerisch – für bestimmte Module, etwa die vierstellige LED-Anzeige oder das OLED-Modul (Bild 8), wird eine in die Arduino-IDE einbindbare Library zur Verfügung gestellt –, dass man sich schnell sicher fühlt und an eigene Applikationen gehen kann. Man muss sich zunächst nur an den Programmbeispielen und Libraries „entlanghangeln“ und lernt so schnell grundsätzliche Programmierschritte, quasi das Arduino-modifizierte C „von hinten“. Der gelernte Programmier-Purist schreit an dieser Stelle auf, der Einsteiger freut sich und lernt einfacher, weil nicht „trocken“.

### Spezialfall „pcDuino“

Der pcDuino (Bild 9), auch aus dem Hause „LinkSprite“ stammend, ist ja die eierlegende Wollmilchsau un-

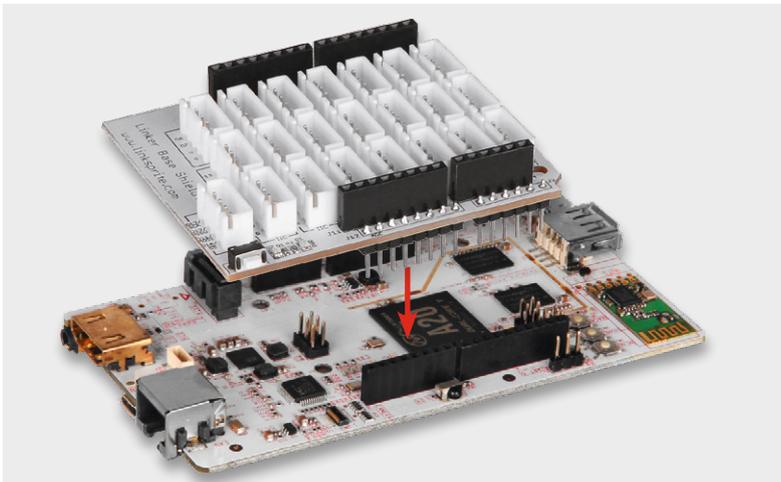


Bild 9: Das Arduino-Base-Shield passt auch auf den pcDuino.

ter den Kleinstrechnern. Denn er verbindet die Welt des ARM-Rechners mit der des AVR-basierten Arduino. In seinem 4-GB-Flash-Speicher ist innerhalb des vorinstallierten Linux-Derivats „Ubuntu“ u. a. auch die Arduino-IDE fest vorinstalliert (Bild 10). Dazu trägt er neben unendlich vielen anderen Schnittstellen und Ports auch einen Arduino-Header, auf den die in enormer Vielzahl, quasi für jeden Anwendungszweck, vorhandenen Arduino-Shields aufgesteckt und eingesetzt werden können.

Da bot es sich natürlich an, auch hier das Arduino-Base-Shield einzusetzen – und schon kann man, ohne einen PC, diesen direkt mit HDMI-Bildschirm und USB-Tastatur als eigenständigen Kleinrechner betreibbaren Linux-Rechner als Rechnersystem für die Linker-Kit-Bausteine einsetzen, und dies quasi neben dem Einsatz etwa als Mini-PC.

### Noch einfacher – Programme zusammenstecken!

Systeme wie Linker Kit, Arduino, Raspberry Pi wenden sich ja ursprünglich an den Programmierer-Nachwuchs und lösungsorientierte Anwender, also Kinder, Jugendliche, Auszubildende, Studenten, Programmier-einsteiger und Leute, bei denen der Weg zu einer Lösung sekundär ist. Herauskommen soll die angestrebte Anwendung, nicht mehr und nicht weniger. Da ist dann womöglich selbst das praktische Handwerkszeug „Arduino-IDE“ eine Hemmschwelle, weil man sich zumindest rudimentär mit C beschäftigen muss. Für diese Fälle gibt es erfolgreiche grafikorientierte Programmieroberflächen, vom Lego-Mindstorms-System bis Scratch für die Linux-Systeme.

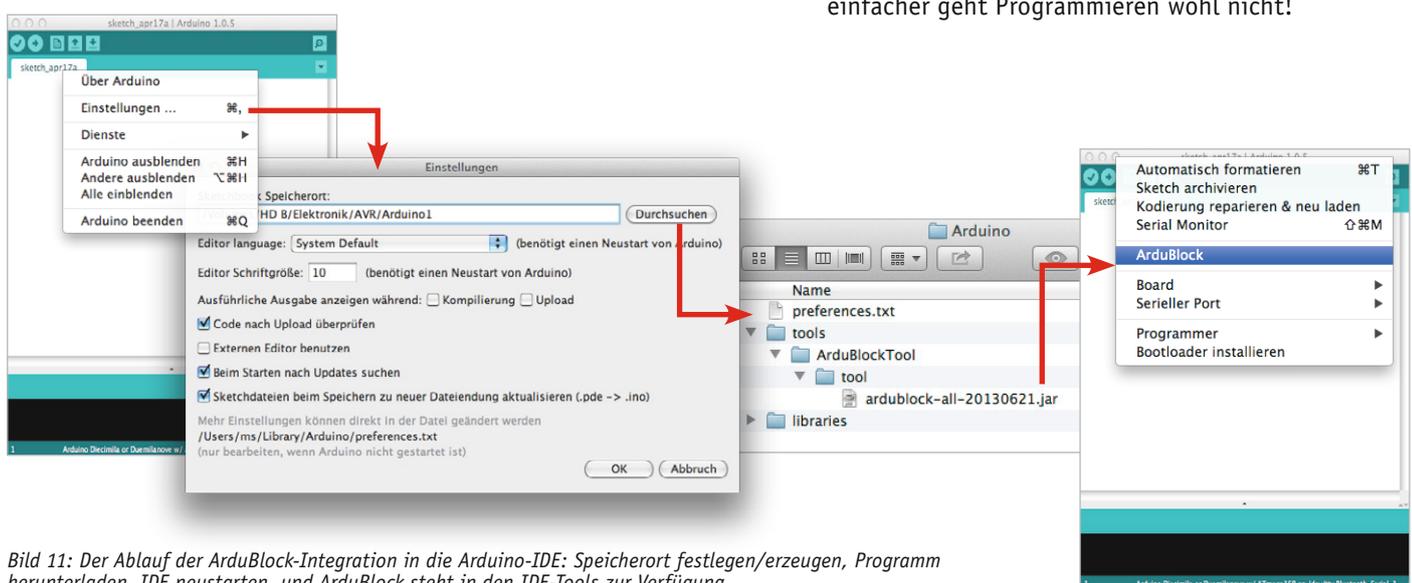


Bild 11: Der Ablauf der ArduBlock-Integration in die Arduino-IDE: Speicherort festlegen/erzeugen, Programm herunterladen, IDE neustarten, und ArduBlock steht in den IDE-Tools zur Verfügung.

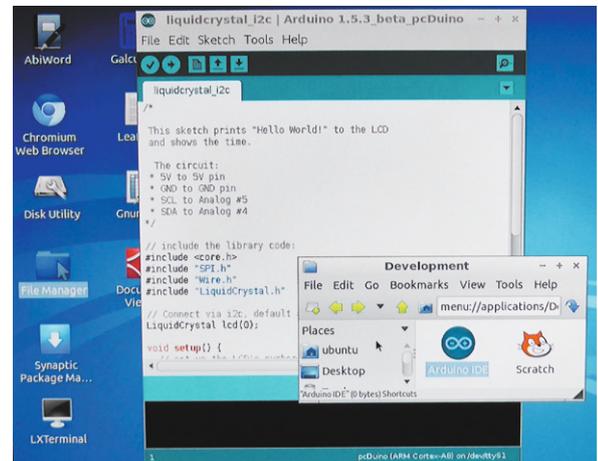


Bild 10: Die Arduino-IDE ist im Flash-Speicher des pcDuino schon fest installiert.

Und genau dies gibt es auch für die Arduino-IDE – die grafische Programmieroberfläche „ArduBlock“ [3]. Diese wird einfach in die Arduino-IDE – egal ob auf dem PC oder dem pcDuino – installiert (Bild 11 zeigt den prinzipiellen Ablauf, der auch auf der ArduBlock-Seite detailliert erläutert ist) und ist dort als Tool zugänglich. Das Schöne ist, man kann sie in deutscher Sprache installieren, und so kann sich jeder, der lesen kann und sich etwas unter „wiederhole fortlaufend“ – in C „loop“ genannt – vorstellen kann, blitzschnell mit Bausteinen ein Programm zusammenbauen.

Öffnet man das Tool, sieht man sofort, dass es neben den Befehlen und Anweisungen auch mehrere Abteilungen zu IoT-Baustein-Systemen, darunter eben auch Linker Kit, gibt. Durch einen Mausklick in die Menüpunkte links öffnet sich jeweils eine Befehls- oder Modulauswahl (Bild 12), aus der man sich den gewünschten Befehl, Operator oder das Modul auf die Programmierfläche zieht.

In Bild 13 ist ein sehr einfaches Beispiel nebst Aufbau zu sehen: 5 Klicks und 2 Einstellungen – fertig ist das kleine Programm!

Sogar solche Vorgänge wie auf den I<sup>2</sup>C-Bus schreiben sind hier mit einem Mausklick erledigt – einfacher geht Programmieren wohl nicht!

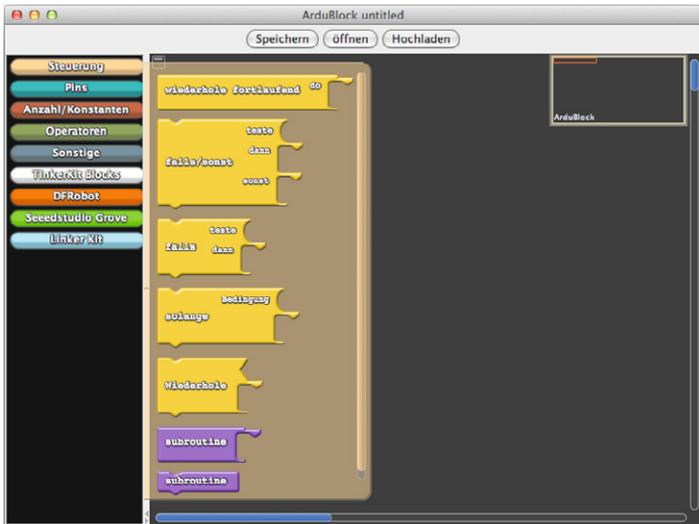


Bild 12: Die einzelnen Befehle, Instruktionen und Programmbausteine werden per Drag & Drop auf die grafische Oberfläche gezogen.

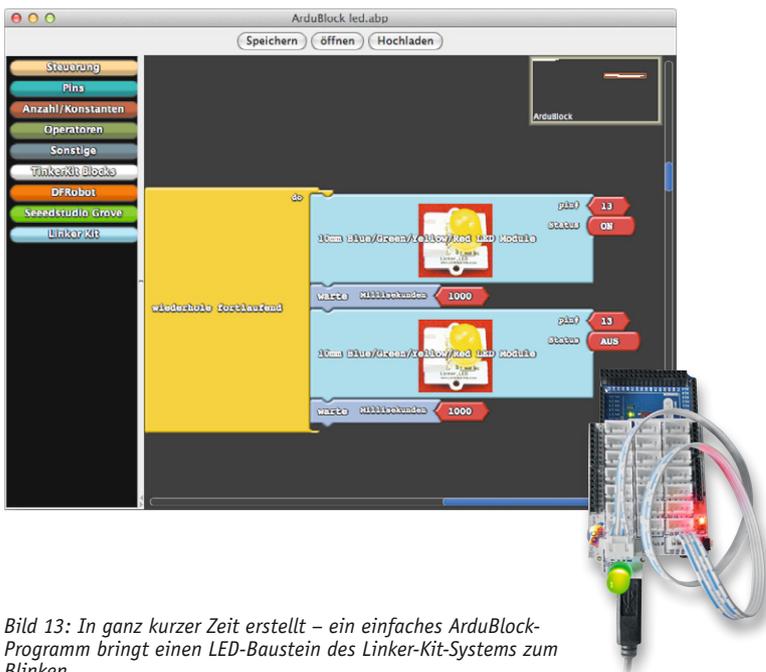


Bild 13: In ganz kurzer Zeit erstellt – ein einfaches ArduBlock-Programm bringt einen LED-Baustein des Linker-Kit-Systems zum Blinken.

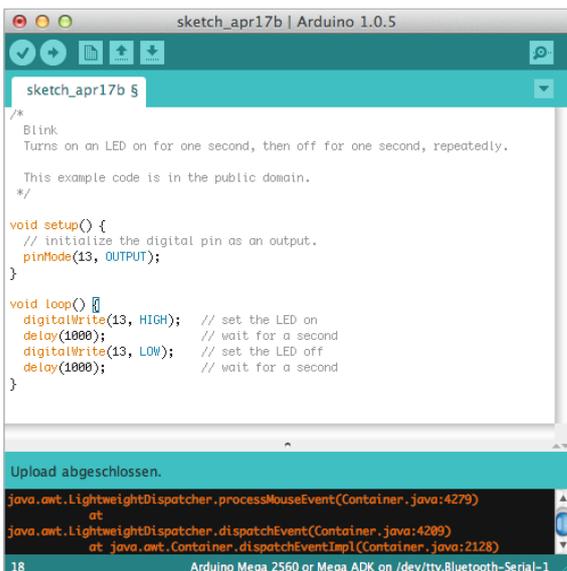


Bild 14: Das mit ArduBlock erzeugte Programm aus Bild 13 in der normalen Arduino-IDE-Ansicht

Hier sind lediglich Lesen, logisches Denken, etwas Abstraktionsvermögen und „nicht das Ziel aus dem Auge verlieren“ gefragt – etwas, das Kinder und Jugendliche besonders gut können. Bild 14 zeigt das so erzeugte C-Programm in der IDE, das bekommt der ArduBlock-Programmierer nicht einmal zu Gesicht, es wird direkt aus dem Tool kompiliert und auf den Arduino/pcDuino geladen.

### Auch für ARM-Rechner geeignet

I<sup>2</sup>C ist I<sup>2</sup>C, ob am AVR oder am ARM-SoC. Ergo ist das Linker-Kit-System auch für letztere Mikrorechner vom Schlage des Raspberry Pi geeignet. In Bild 4 haben wir es ja schon angedeutet gesehen – es gibt auch ein Linker-Kit-Base-Shield für den Raspberry Pi und die GPIO-gleichen Verwandten [4]. Das Shield bietet sogar noch ein zusätzliches Feature, nämlich einen ADC, der dem Raspberry Pi fehlt. Auch hier gibt es ein ausführliches Tutorial zur Einbindung des hier verbauten MCP3004 [5] samt Treiber, die Module des Linker-Kit-Systems sind z. B. entweder über Python oder Scratch ansprechbar, die C-Programmbeispiele für die Arduino-IDE liefern die genaue Vorlage dazu.

So sind also auch diese oft als Datenlogger, Spieleemulatoren usw. eingesetzten Mini-Rechner mit dem Linker-Kit-System einfach einsetzbar.

So steht hier ein geradezu ideales Einsteigersystem zur Verfügung, mit dem sich schnell, ohne Löten, und (zunächst) ohne tiefere Programmierkenntnisse, Aufgaben lösen lassen – gerade in der Ausbildung und für Hobbyzwecke ideal! Erst wenn es dann an „höhere Aufgaben“ geht, muss man tiefer einsteigen. Für die erste Bekanntschaft mit Elektronik und Programmieren ist solch ein System unschlagbar.

Wir danken unserem Distributor „SiMAC Electronics GmbH“ für die hardwaremäßige Unterstützung dieses Beitrags! **ELV**

Alle Linker Kit-Produkte finden Sie im ELV-Web-Shop unter Webcode #1373



### Weitere Infos:

- [1] [www.eecs.umich.edu/eecs/about/articles/2015/Worlds-Smallest-Computer-Michigan-Micro-Mote.html](http://www.eecs.umich.edu/eecs/about/articles/2015/Worlds-Smallest-Computer-Michigan-Micro-Mote.html)
- [2] <http://linksprite.com/wiki/index.php5?title=LinkerKit>
- [3] <http://blog.ardublock.com>
- [4] [http://linksprite.com/wiki/index.php5?title=Linker\\_kit\\_Base\\_Shield\\_for\\_Raspberry\\_Pi\\_with\\_ADC\\_Interface](http://linksprite.com/wiki/index.php5?title=Linker_kit_Base_Shield_for_Raspberry_Pi_with_ADC_Interface)
- [5] <http://learn.linksprite.com/raspberry-pi/shield/how-to-use-linker-kit-base-shield-for-raspberry-pi-with-adc-interface-with-python-code>