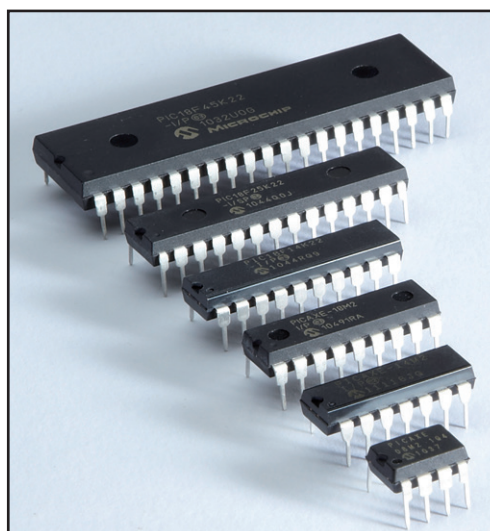


BASIC für PICs

Einführung in PICAXE

Standard-Mikrocontroller eignen sich häufig auch für Spezialanwendungen. Wenn man die entsprechende Firmware programmiert, kann man eigene Projekte mit besonderen Fähigkeiten ausstatten. Mit dem PICAXE-System kann jeder Elektroniker einen Mikrocontroller samt Peripherie mit wenig Aufwand und Kosten programmieren.



Von Wouter Spruit (NL)

Bild 1.
PICAXE-Chip-Größen.

Wer? Wie? Was?

Bei den PICAXE-Mikrocontrollern handelt es sich um PICs von Microchip, die mit einer speziellen Firmware von Revolution Education [1] programmiert sind. Das Konzept des PICAXE-Systems ermöglicht das Programmieren und Implementieren von Mikrocontrollern in Projekte durch ein preiswertes System, das nicht viel Erfahrung voraussetzt, aber dennoch ein komplexes Interfacing erlaubt, wodurch Mikrocontroller auch in größeren Projekten einsetzbar sind. In der PICAXE-Serie findet man Mikrocontroller unterschiedlicher Größen und Formen, vom einfachen Typ mit nur acht Pins bis hin zum komplexeren, 40-poligen Modell 40X2 (**Bild 1**). In der Zeit seit der Einführung von PICAXE vor über 15 Jahren wurden die PIC-Mikrocontrollertypen durch neue Modelle ersetzt, was nicht nur zu mehr Speicher und mehr Rechenleistung führte, sondern in vielen Fällen sogar die parallele Ausführung von Tasks [2] ermöglicht.

Der Webstore von Revolution Education [3] bietet nicht nur PICAXE-Mikrocontroller, sondern auch Zubehör wie Bauteile, Platinen, Kabel, I²C-Peripherie, LCDs und Starter-Kits an. Über diese Webseite kann man also alles beziehen, um direkt mit der PICAXE-Program-

mierung los zu legen. Es gibt sogar Bausätze für Spielzeug wie PICAXE-basierte Roboter. Die angebotene Peripherie ist mit umfangreichen Beschreibungen zum Interfacing mit Mikrocontrollern versehen. Selbst wenn man keine PICAXE-Mikrocontroller einsetzt, ist das Angebot auch für andere Controller-Familien interessant.

Wie schon erwähnt ist die spezielle Firmware das, was PICAXE ausmacht. Ein PICAXE-Chip ist ein „normaler“ PIC, der schon mit der PICAXE-Firmware programmiert wurde und daher einen BASIC-Interpreter und etliche vorgefertigte Funktionen mitbringt. Dank PICAXE kann man jederzeit ein PICAXE-BASIC-Programm über eine serielle Verbindung auf den Controller laden, solange diese Schnittstelle nicht vom Programm selbst verwendet wird. Die Ausführung des neuen Codes beginnt sofort. Da der Code hier interpretiert wird, ist die Ausführungsgeschwindigkeit begrenzt, was code-getriebene Kommunikation mit Peripherie erschwert. Für diese Zwecke gibt es allerdings fertige und schnelle Ein-/Ausgangs-Funktionen, sodass man problemlos via I²C mit Peripherie kommunizieren oder beispielsweise IR-Signale generieren kann. Ein großer Vorteil der Verwendung eines

Interpreters liegt in der Art, wie Code auf den Controller übertragen werden kann. Für die Programmierung eines „normalen“ PICs benötigt man nämlich ein umfangreiches Entwicklungssystem, das im Vergleich zum PICAXE-Konzept deutlich mehr Schwierigkeiten bei Handling und Einsatz macht. PICAXE-Projekte sind üblicherweise mit einer Steckverbindung ausgestattet, die mit den Pins einer seriellen Schnittstelle des Controllers verbunden ist. Auf diese Weise kann das Projekt jederzeit leicht mit neuem Code bestückt werden. Sobald die Elektronik eingeschaltet und ein PC angeschlossen ist, kann ein neues Programm auf den PICAXE-Chip übertragen werden. Auch wenn das Schreiben und Hochladen von PICAXE-BASIC-Programmen sehr schnell von der Hand geht, gibt es doch aufgrund des interpretierten Codes Probleme, wenn eine Anwendung das Ausführen komplexerer mathematischer Funktionen in hoher Geschwindigkeit erfordert. Die Ausführungsgeschwindigkeit ist für Echtzeit-Anwendungen einfach zu gering.

Alternativen zu PICAXE

Ein „normaler“ PIC benötigt einen PIC-Programmer, um neuen Code in den Chip zu bekommen. Auch wenn Projekte wie „el cheapo“ [4] die Programmier-Kosten in Grenzen halten, können die Kosten für den Einstieg mit einem simplen Beispiel wie einer blinkenden LED doch davon abschrecken, so etwas einfach mal auszuprobieren. Ähnliches gilt für die AVR-Controller von Atmel. Auch wenn durch Arduino [5], das aus einem definierten Board mit AVR-Mikrocontroller samt IDE und Dokumentation besteht, der Einstieg in die Welt der Mikrocontroller erleichtert wird, so ist dieses Konzept doch teurer und komplexer als das PICAXE-System.

Support und Dokumentation

Aufgrund des Lern- und Ausbildungscharakters von PICAXE überrascht es nicht, dass hier umfangreiche Dokumentation zur Verfügung steht. Das PICAXE-Manual ist beim „Programming Editor“ (Windows) und beim AXEpad (OS X oder Linux) dabei und zusätzlich auf der PICAXE-Website [6] als PDF verfügbar. Das Handbuch ist dreigeteilt in „Getting Started“, „BASIC Commands“ und „Mikrocontroller Interfacing Circuits“. Im ersten Teil geht es um technische Eigenschaften und

die Pin-Belegung diverser PICAXE-Produkte sowie um Inbetriebnahme und Anschluss plus einige Tutorials zum Einstieg. Im zweiten Teil werden die BASIC-Befehle beschrieben und aufgeführt, gefolgt von einigen Programmier-Beispielen. Der dritte Teil besteht aus einer Sammlung von Peripherie-Demos. Im Handbuch sind nicht nur einfache Dinge beschreiben wie der Anschluss von Schaltern, LEDs oder eines Motors mit Hilfe von Darling-ton-Transistoren - es geht auch um das Interfacing von PICAXE-Chips mit kommerziellen Modulen und Chips, wie etwa die Ansteuerung eines LCDs via I²C oder die serielle Kommunikation PICAXE<->PC. Das Besondere ist, dass sich die Beispiele sogar auf das Interfacing von Peripherie und Modulen mit Nicht-PICAXE-Chips beziehen. Hierbei sollten allerdings zulässige Spannungen und Steuerströme der verwendeten Hardware überprüft werden. Auch wenn PICAXE ursprünglich für Lern- und Ausbildungszwecke gedacht war, so ist das System doch sehr gut in richtigen kleinen Projekten einsetzbar. PICAXE-User stellen ihre Projekte gerne auf der PICAXE-Webseite der Allgemeinheit zur Verfügung. Anwender-Projekte haben sich daher zu einem großen Schaufenster für die Fähigkeiten dieser Plattform entwickelt. Die dort gehosteten Foren stellen eine gute Quelle für Hilfe und Unterstützung dar, und man bekommt hier gute Antworten zu Fragen der Integration von PICAXE-Chips und ihrer Peripherie in eigene Projekte.

Software & Tools

Wie schon erwähnt kann man mit dem PICAXE-System BASIC-Programme vom PC per serieller Verbindung auf einen PICAXE-Chip laden. Für Windows gibt es eine umfangreiche Software namens „Programming Editor“, mit dem man PICAXE-Chips ganz einfach mit Software bestücken und diese debuggen kann. AXEpad ist die Cross-Plattform-Alternative dazu und für OS X und Linux erhältlich. Sie bietet die wichtigsten Fähigkeiten der Windows-Lösung: einen Editor und das Hochladen des Codes auf den Controller sowie ein Terminal und etliche Wizards zur Code-Erzeugung. Im Vergleich zum Programming Editor vermisst man aber einige Fähigkeiten. Das wichtigste Manko ist sicherlich das Fehlen eines On-screen-Tools zum Test eines PICAXE-Programms über eine schritt-



Bild 2.
USB/Seriell-Kabel.

weise Simulation.

Will man den Editor einsetzen, muss man einen PICAXE-Chip direkt seriell oder über einen USB/Seriell-Konverter an einen PC anschließen (siehe **Bild 2**) und den verwendeten seriellen Port beim Programming Editor bzw. bei AXEpad angeben. Außerdem muss man noch festlegen, welche Art von Chip verwendet wird. Die Software enthält sogar eine Funktion, mit der man überprüfen kann, welcher PICAXE-Chip-Typ angeschlossen ist. Sie kann sogar die Firmware-Version des angeschlossenen Chips auslesen. Ein Programm kann nach dem Schreiben direkt aus dem Editor in den Chip übertragen werden. Das neue Programm startet dann sofort.

Blinkende LED

Die Hello-World-Variante für Mikrocontroller besteht bekanntlich darin, eine angeschlossene LED blinken zu lassen. Bei diesem Beispiel wird ein Controller des Typs PICAXE 08M verwendet, was aber problemlos auf jeden anderen PICAXE-Chip übertragbar ist. Anhand der Pin-Belegung [7] kann man schauen, an welchen Pin sich die LED anschließen lässt. Beim PICAXE 08M haben wir hierzu Pin 3 (C.4) ausgewählt – die zugehörige Pin-Belegung des PICAXE 08M2 ist in **Bild 3** zu sehen. Obwohl von Revolution Education Starter-Packs angeboten werden, die alle nötige Interface-Elektronik umfassen, wird in diesem Beispiel demonstriert, wie man einen PICA-

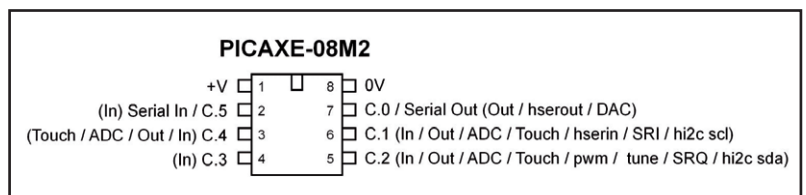


Bild 3.
Pin-Belegung des PICAXE 08M2.

XE-Chip via Steckbrett anschließt. Die hier relevanten Pins sind +V (1), 0V (8), „serial in“ (2), „serial out“ (7) und der Ausgangs-Pin für die LED (3).

Bild 4 zeigt die Schaltung für die blinkende LED. Die Anschlüsse für die Stromversorgung +V und 0V werden mit einem Netzteil verbunden, das eine Spannung zwischen 3 V und 5 V liefert. Man kann die Schaltung auch per Batterien in einem Halter (gibt es auch im Shop von Revolution Education) versorgen, sollte aber im Auge behalten, dass nicht mehr als drei Zellen zu 1,5 V oder vier Akkus mit je

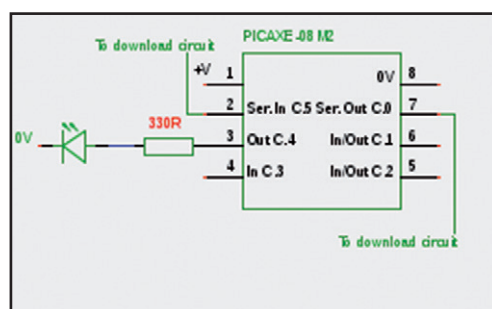


Bild 4.
Blinkende LED.

1,2 V in Serie geschaltet werden. Wie in **Bild 5** gezeigt, wird der Code-Upload über die PICAXE-Pins „serial in“, „serial out“ und „0V“ vorgenommen. Wenn die Upload-Schaltung nicht an den PICAXE angeschlossen ist, sollte „serial in“ Pin (2) über einen 33-kΩ-Pull-down-Widerstand auf Masse gezogen werden. Die LED wird über einen Vorwiderstand von 330 Ω an Pin 3 und Masse angeschlossen.

PICAXE-Programmierung

Zunächst installiert man AXEpad (bei GNU/Linux sowie OS X) oder aber den Programming Editor unter Windows [8]. Bei Verwendung eines USB/Seriell-Konverters benötigen Betriebssysteme in der Regel passende Treiber, bevor man den Konverter nutzen kann [9]. Im Beispiel wird AXEpad unter Linux eingesetzt und ein PICAXE 08M auf einem Steckbrett mit einem seriellen RS232-Kabel über eine 3,5-mm-Buchse an den PC angeschlossen. Einige Sekunden nachdem die Verbindung zum Chip hergestellt und die Stromversorgung eingeschaltet ist, startet man die Software. Bevor man ein Programm hochladen kann, müssen die richtige PICAXE-Chip-Version und der verwendete serielle Port spezifiziert sein. Bei AXEpad selektiert man den seriellen Port unter View -> Options -> Tab:Port. Auch der Port eines USB/Seriell-Konverters wird hier aufgelistet. Um die Verbindung zu prüfen, klickt man den Knopf mit der Firmware im Tab des Dialogs View -> Options.

Nachdem alles richtig eingestellt ist, wird auf die Betätigung des Firmware-Knopfs hin der Typ des gerade angeschlossenen Chips ausgelesen. Wenn ein anderer als der ausgelesene Typ eingestellt ist, sollte man dies korrigieren. Ist alles in Ordnung, kann man jetzt sein erstes PICAXE-Programm schreiben. Hierzu gibt man den Code von **Listing 1** im Editor ein. Noch ein Wort zu den Pin-Bezeichnungen: Die internen Namen der Pins sind etwas verwirrend. Einige Chips kann haben mehrere interne IO-Blöcke, bei denen ein Präfix anzeigt, welcher Block gemeint ist. In diesem Beispiel wird der ältere Chip 08M verwendet, bei dem keine Präfixe notwendig sind. Wenn beim Code von Listing 1 Probleme mit Ihrer Version des PICAXE-Chips auftauchen, versuchen Sie einfach mal, das passende Präfix zu verwenden. Beispielsweise für Pin 3, der Ausgang 4 des Blocks „C“ entspricht: „high C.4“. Nun wird das Programm auf den Chip

Listing 1: Ein erstes PICAXE-Programm

```
do ;repeat forever
  high 4 ;set output 4 high
  pause 1000 ;wait 1 second
  low 4 ;set output 4 low
  pause 1000 ;wait 1 second
loop
```

Listing 2: LED und Taster

```
do ;repeat forever
  if pin3=0 then ;if the button is pressed
    high 4 ;light LED
  else ;if the button is NOT pressed
    low 4 ;turn off LED
  endif ;closes if statement
loop
```

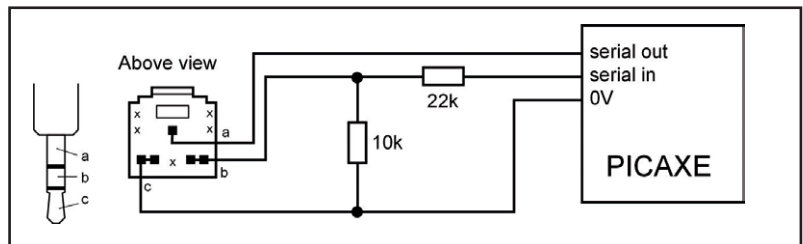


Bild 5. Anschluss an den PC.

transferiert, indem der Upload-Knopf betätigt, die Option PICAXE -> Programm gewählt oder F5 gedrückt wird. Der Fortschritt des Uploads wird angezeigt und der verfügbare Speicher des Chips angegeben. Nach dem Upload erscheint „programming successful“. Das neue Programm startet sofort, und die LED sollte jetzt blinken.

Noch ein Taster

Das nächste Beispiel demonstriert, wie man eine LED dann leuchten lassen kann, wenn

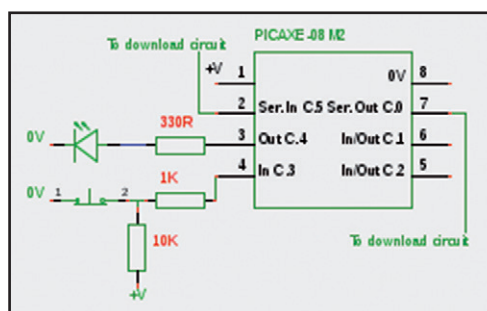


Bild 6. LED und Taster.

ein Taster betätigt wird. Hierzu schließt man den Taster wie in **Bild 6** gezeigt an Pin 4 an. Anschließend wird das Programm von **Listing 2** auf den PICAXE-Chip geladen. Nach dem Upload sollte die LED nur noch leuchten, wenn der Taster gedrückt ist. Die Bedingung „if“ im Code prüft, ob der Pegel von Pin 4 durch den Taster auf „low“ gezogen wird. Wenn ja, dann leuchtet die LED – andernfalls nicht. **Bild 7** zeigt die Steckbrett-Version der Schaltung, bei der ein PICAXE 08M mit dem Code von Listing 2 bestückt ist. Die gezeigte Schaltung ist hier nicht mit dem PC verbunden.

Die beiden Beispiele demonstrieren, wie einfach solche Sachen mit PICAXE sein können. Die benötigten Bauteile findet man leicht im Shop von Revolution Education und die zugehörige Dokumentation ebenfalls. Gerade Einsteiger können so sehr leicht PICAXE-Mikrocontroller für eigene Projekte nutzen.

(130137)

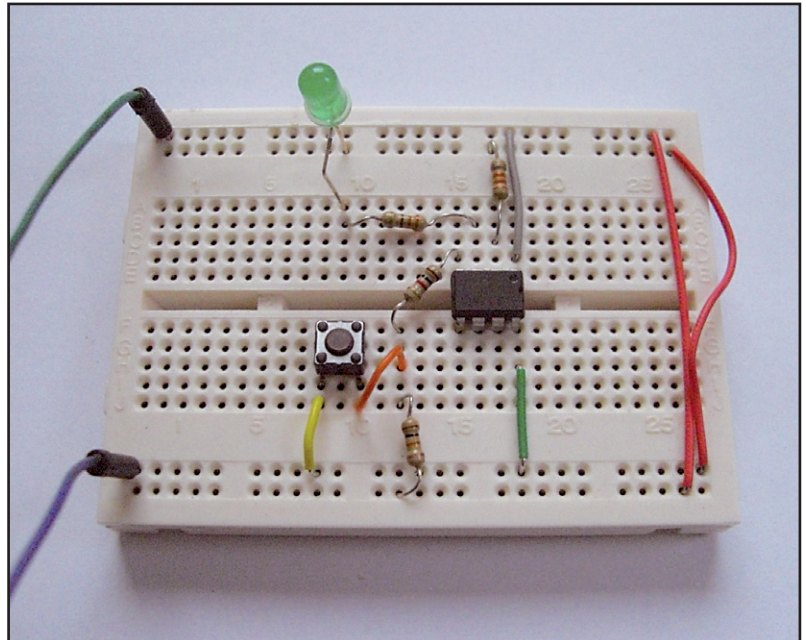


Bild 7. So sieht die Schaltung mit LED und Taster samt Chip auf einem Steckbrett aus.

Weblinks

- [1] www.picaxe.com
- [2] www.picaxe.com/What-is-PICAXE/PICAXE-Chip-Sizes
- [3] www.techsupplies.co.uk/PICAXE
- [4] www.rentron.com/myke4.htm
- [5] www.arduino.cc
- [6] www.picaxe.com/Getting-Started/
- [PICAXE-Manuals](#)
- [7] www.picaxe.com/What-is-PICAXE/PICAXE-Pinouts
- [8] www.picaxe.com/Software
- [9] www.picaxe.com/Getting-Started/Driver-Installation