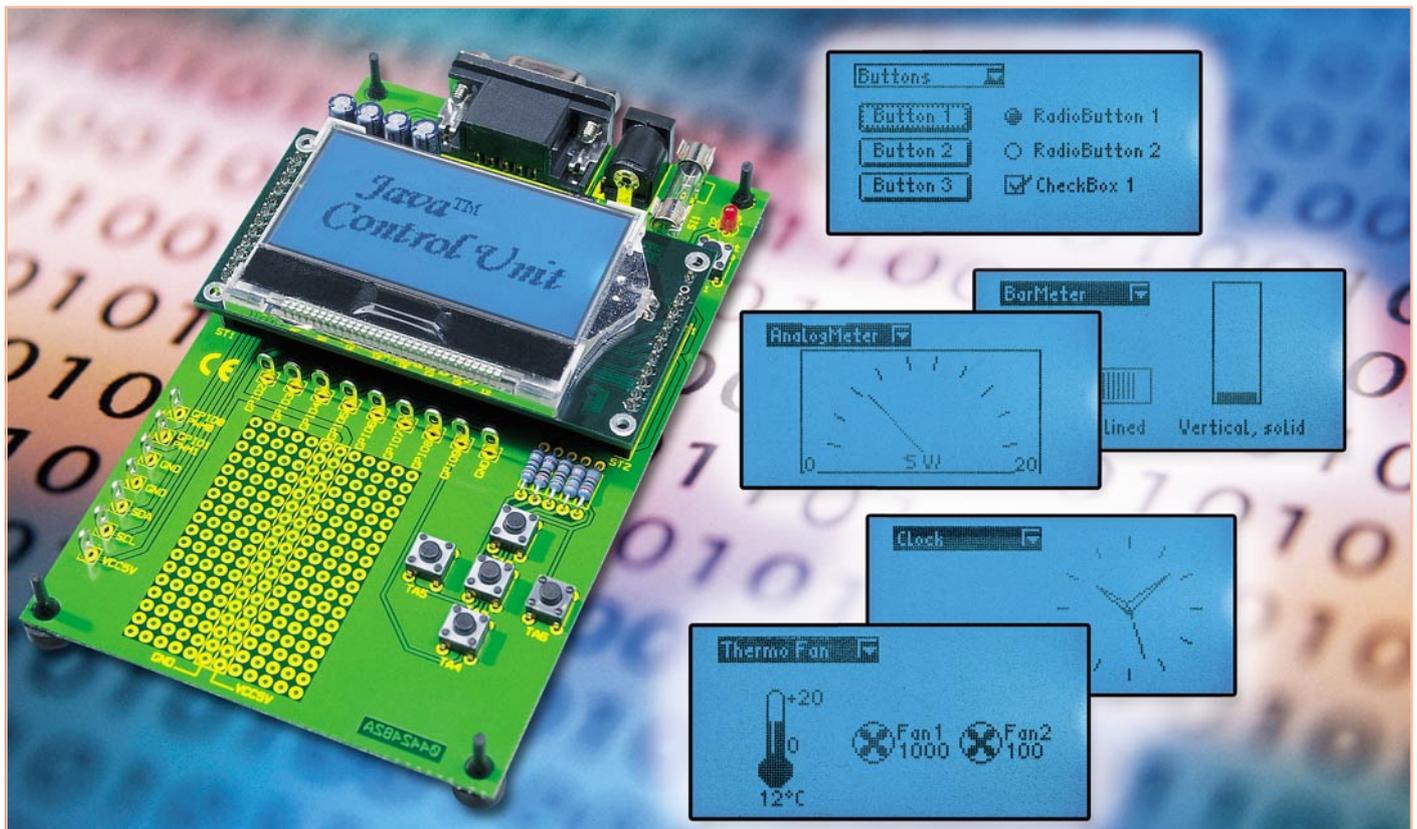


Steuern und visualisieren –



Java™-Control-Unit JCU 10 mit JControl-Technologie Teil 2

Die einfache Ansteuerung eines grafischen Displays mit der Programmiersprache Java macht die Java™-Control-Unit universell einsetzbar. Nachdem wir im ersten Teil dieser Artikelserie die Vorteile der Programmiersprache Java™ vorgestellt und einige Möglichkeiten des grafischen Displays gezeigt haben, werden wir uns in dieser Ausgabe mit den Schaltungen und dem Nachbau der Java™-Control-Unit und des Evaluation Boards befassen, bevor es in der nächsten Folge mit dem Einstieg in die Programmierung der Java™-Control-Unit weiter in Richtung eigene Applikation geht.

Schaltung

Java™-Control-Unit

Abbildung 9 zeigt den Schaltplan der Java™-Control-Unit. Das Herzstück der Schaltung ist der Mikrocontroller IC 3. Hierbei handelt es sich um einen 8-Bit-Mikrocontroller, der bereits programmiert ist und die JControl Virtual Machine (JCVM 8) bildet. Der Controller arbeitet mit einer Taktfrequenz von 16 MHz, die der Keramikresonator Q 1 stabilisiert.

Die vom Anwender in Java™ selbst erstellten Applikationen werden im 64 KByte großen Flash-Speicher IC 2 abgelegt und nach

der Initialisierung des Controllers von dort aus abgearbeitet.

Das LC-Display LCD 1 wird über sein paralleles Interface vom Mikrocontroller angesteuert. Hierbei handelt es sich um ein grafisches Display mit 128 x 64 Pixel mit einem Punktabstand von 0,33 mm auf einer sichtbaren Fläche von 46 x 23 Millimetern. Durch die FSTN-Technologie (Film Compensated Super Twisted Nematic) hat das Display einen hohen Kontrast und einen weiten Betrachtungswinkel.

Das Displaymodul ist eine Ausführung in der COG (Chip On Glass)-Technologie, d. h., dass der Displaytreiber sich mit auf dem Modul befindet und damit diese Auf-

gabe nicht vom Mikrocontroller übernommen werden muss. Praktisch ist ebenfalls, dass die höhere Spannung, die das Display zum Betrieb benötigt, vom Displaymodul selbst und den externen Kondensatoren C 2 bis C 9 heraufgesetzt wird.

Das Displaymodul ist in einen Kunststoffrahmen montiert, in dem eine blaue LED für die Hintergrundbeleuchtung des Displays sorgt.

Der Kontrast des Displays und die Helligkeit der Hintergrundbeleuchtung lassen sich per Software einstellen. Die Displayhelligkeit wird hierbei über das parallele Interface des Displaytreibers eingestellt, während für die Hintergrundbeleuchtung

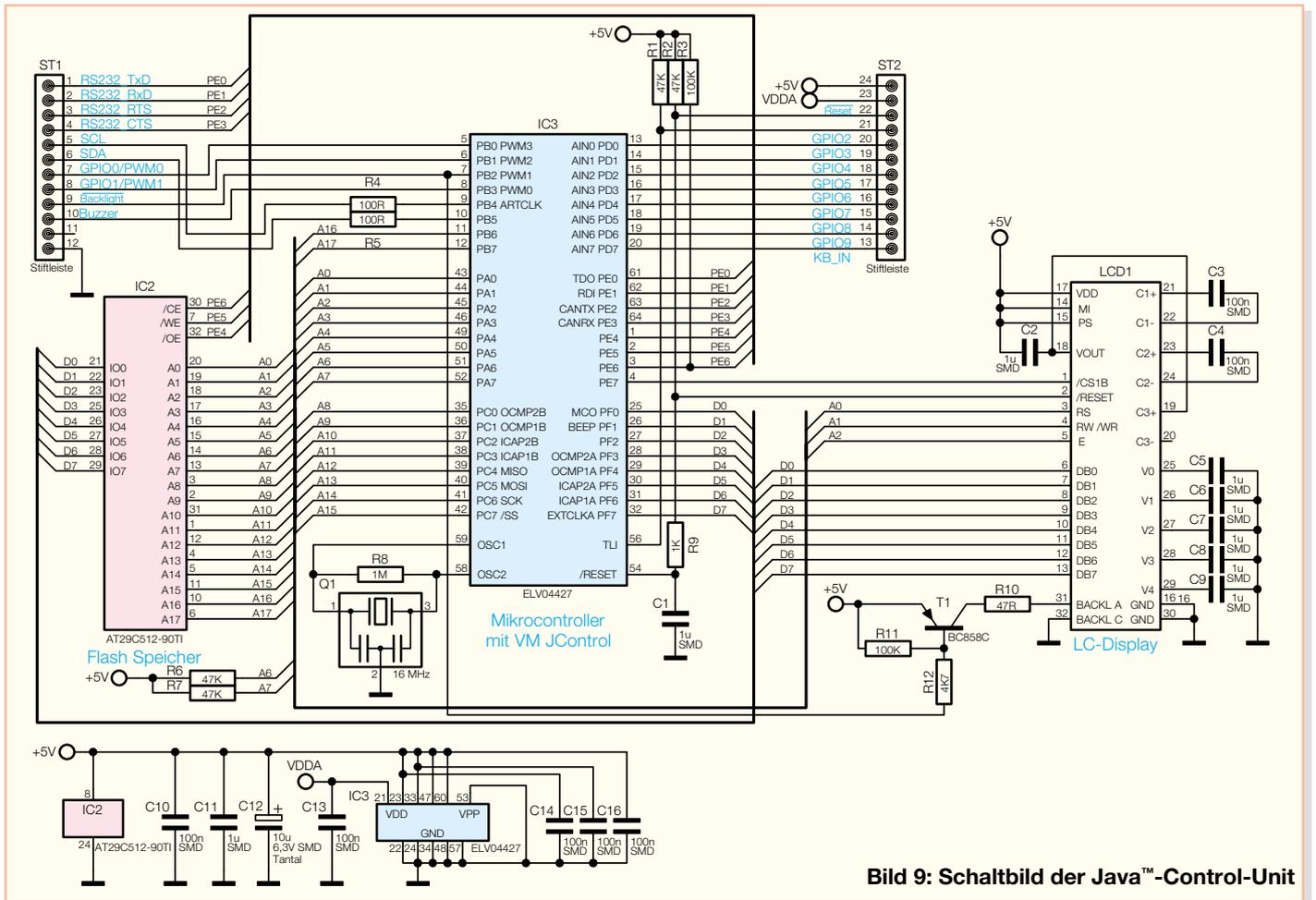


Bild 9: Schaltbild der Java™-Control-Unit

der Mikrocontroller mit einem PWM-Signal über den Transistor T 1 die Leuchtdiode im Displayrahmen ansteuert.

Der Reset-Pin des Mikrocontrollers liegt im Normalbetrieb über R 2 und R 9 auf High-Potenzial, über Pin 22 der Stiftleiste ST 2 kann der Reset-Pin auf Low-Potenzial gelegt und so ein Reset des Controllers und des Displaymoduls ausgelöst werden. Der Controller verfügt über einen Reset-Sequenz-Manager, so dass auch der Controller selbst einen Reset auslösen und den Reset-Anschluss dabei auf Low-Potenzial legen kann.

Beim Anlegen der Betriebsspannung wird der Reset-Pin bis zum Überschreiten einer Spannung von 4,2 V ebenfalls auf Low-Potenzial gehalten. Dies führt zu einem Reset des Displaymoduls und evtl. weiterer an Pin 22 der Stiftleiste ST 2 angeschlossenen Komponenten.

Auf die beiden Stiftleisten des Moduls sind weiterhin die Anschlüsse für die Spannungsversorgung, die serielle Schnittstelle, der I²C-Bus, die Displaybeleuchtung, die Ansteuerung für den Soundtransducer (Buzzer), die Resetschaltung, die Referenzspannung des A/D-Wandlers (VDDA) und die 10 Ein-/Ausgangspins geführt.

Auf die speziellen Funktionen der IO-Pins werden wir später noch genauer eingehen.

Evaluation Board

Das Evaluation Board ist eine Entwicklungs- und Experimentierplatine für die Java™-Control-Unit. Abbildung 10 zeigt das Schaltbild. Über die Hohlsteckerbuchse BU 1 wird die Schaltung mit einer Gleichspannung von 9 bis 12 V versorgt. Die Diode D 1 schützt hierbei die Schaltung vor einer falsch gepolten Versorgungsspannung. Der Spannungsregler IC 1 erzeugt eine stabilisierte Gleichspannung von 5 V, die auch als Betriebsspannung an den Buchsenleisten zur Verfügung steht, in die die Java™-Control-Unit eingesetzt wird.

Über diese Buchsenleisten werden auch alle für den Betrieb benötigten Anschlüsse der Java™-Control-Unit an Komponenten des Evaluation Boards oder an die Lötstifte am Lochrasterfeld, das für kleine Testschaltungen gedacht ist, geführt.

Die Anschlüsse der seriellen Schnittstelle (1 bis 4 an der Stiftleiste ST 1) der Java™-Control-Unit werden über den Pegelwandler IC 2 auf die 9-polige Sub-D-Buchse geführt.

Diese Schnittstelle dient der Übertragung von mit der JControl/IDE selbst geschriebenen Anwendungen über die serielle Schnittstelle des PCs in den Flash-Speicher der Java™-Control-Unit. Natürlich ist die Schnittstelle auch in eigenen Anwen-

dungen zur seriellen Kommunikation verwendbar.

An den Buzzer-Ausgang (Pin 10) ist ein Soundtransducer angeschlossen, über den Töne und Melodien ausgegeben werden können.

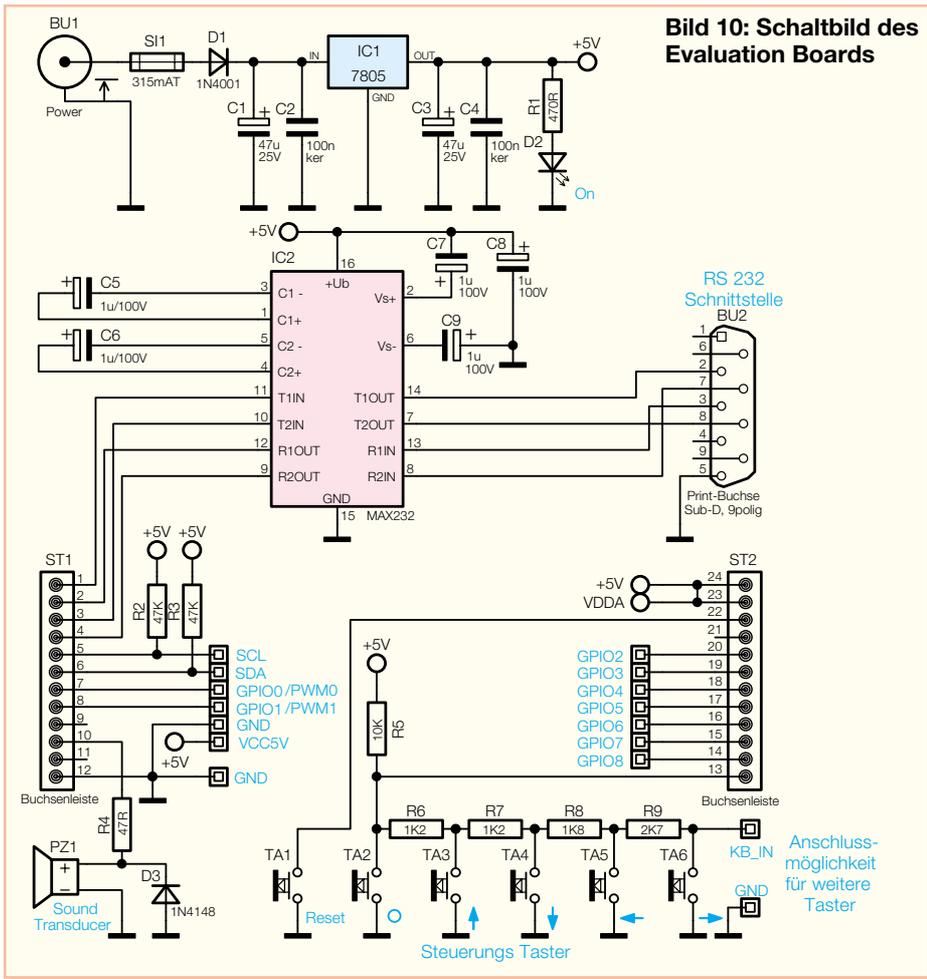
Der Reset-Anschluss (Pin 22) ist auf den Reset-Taster TA 1 geführt. Durch Drücken des Tasters wird der Pin auf Low-Potenzial gelegt und so ein Reset der Java™-Control-Unit ausgelöst.

Pin 23 der Stiftleiste ist der Anschluss für die Referenzspannung des A/D-Wandlers, dieser wird auf dem Evaluation Board mit der 5-V-Betriebsspannung verbunden.

An GPIO 9 (Pin 13 an der Stiftleiste), der hier als analoger Tastatur-Decoder verwendet wird, sind die Taster TA 2 bis TA 6 angeschlossen, mit denen die Funktionen up, down, left, right and select realisiert sind.

Am Lötstift KBIN kann man noch bis zu 5 weitere Taster mit entsprechenden Widerständen anschließen. Die Funktion des Tastatur-Decoders wird im Zusammenhang mit der Besprechung der IO-Pins noch genauer erläutert.

Die übrigen IO-Anschlüsse sowie die Anschlüsse des I²C-Busses stehen an den Lötstiften, die um das Lochrasterfeld platziert sind, zur Verfügung.



Stückliste:
Java™-Control-Unit JCU 10

Widerstände:

47 Ω/SMD	R10
100 Ω/SMD	R4, R5
1 kΩ/SMD	R9
4,7 kΩ/SMD	R12
47 kΩ/SMD	R1, R2, R6, R7
100 kΩ/SMD	R3, R11
1 MΩ/SMD	R8

Kondensatoren:

100 nF/SMD . C3, C4, C10, C13–C16	
1 µF/SMD/Bauform 1206	C1, C2, C5–C9, C11
10 µF/6,3 V/tantal/SMD	C12

Halbleiter:

AT29C512-90TI/SMD	IC2
ELV04427/SMD	IC3
BC858C	T1
LC-Display mit Stiftleiste	LCD1

Sonstiges:

Keramikschwinger, 16 MHz, SMD ..	Q1
LCD-Rahmen mit Hintergrundbeleuchtung, Blau	LCD1
Stiftleisten, 1 x 12-polig, gerade, print	ST1, ST2
1 CD Software JCU10	

grammiert als komplizierte SMD-Schaltungen zu bestücken.

Nachbau

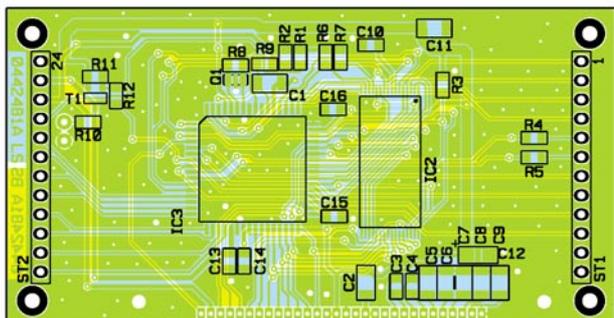
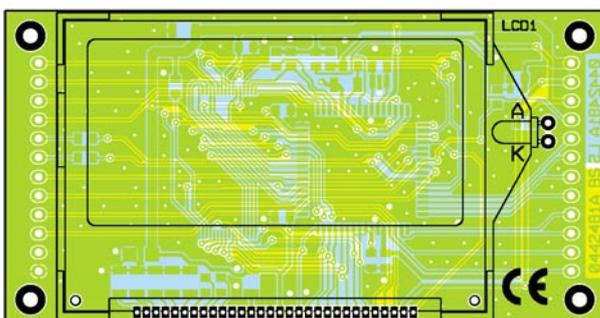
Java™-Control-Unit

Da die Platine ausschließlich mit SMD-Bauteilen bestückt ist und das Verlöten der Anschlüsse des Mikrocontrollers, des Dis-

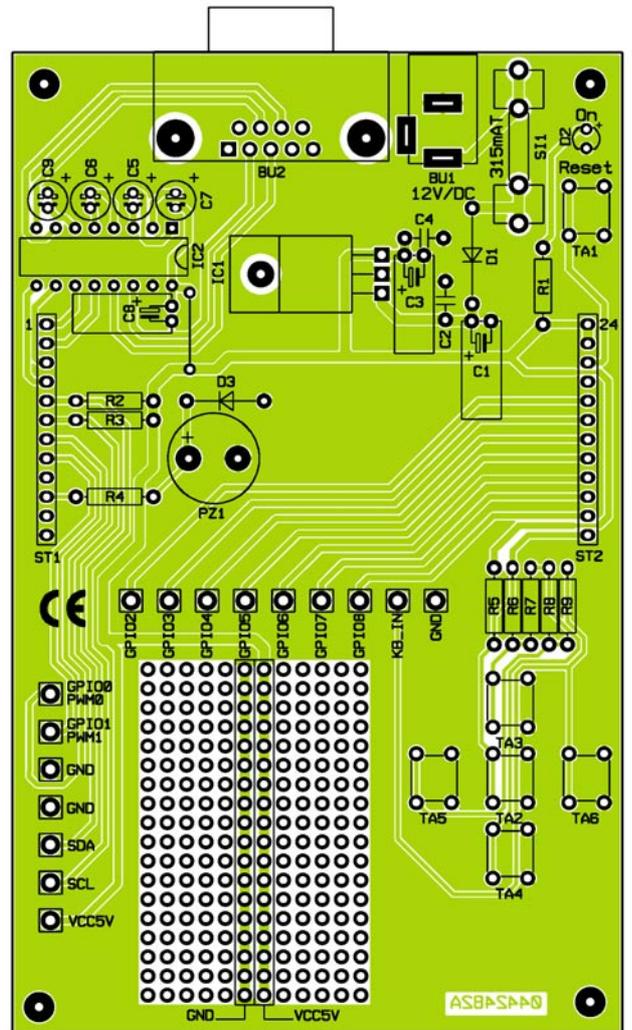
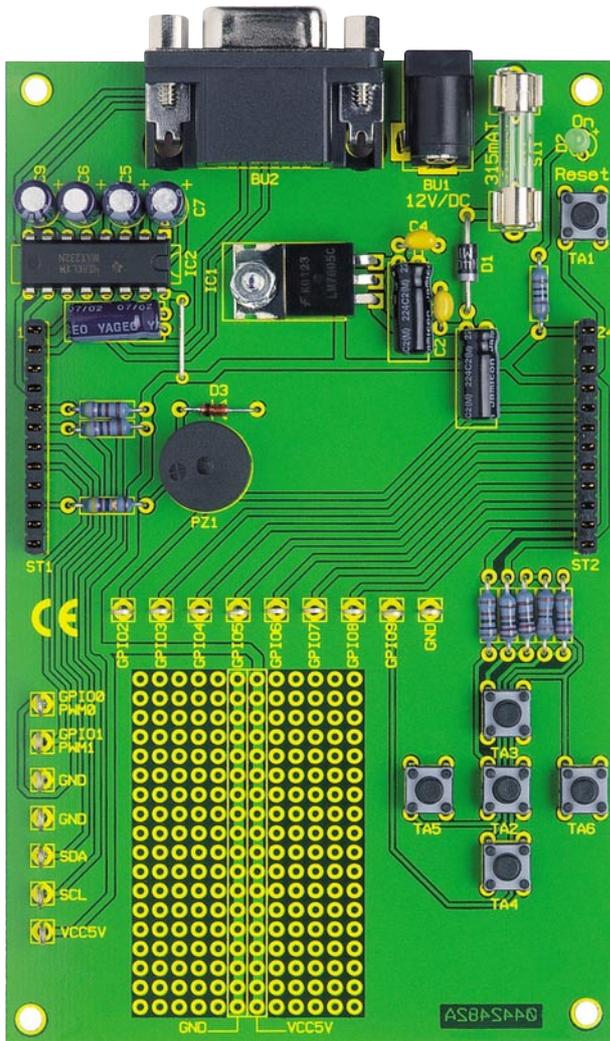
playmoduls und insbesondere des Flash-Speichers mit der dem Hobbyelektroniker zur Verfügung stehenden Ausrüstung kaum zu bewerkstelligen wäre, haben wir uns entschlossen, diese Einheit als betriebsfähiges Fertigergerät zu liefern. Dies erleichtert auch dem den Einstieg, der lieber pro-

Evaluation Board

Der Aufbau des Evaluation Boards ist hingegen sehr einfach und auch für Elektronik-Neulinge ohne Probleme durchführbar. Der Aufbau mit konventionell bedrahteten Bauelementen gewährleistet dabei



Ansicht der fertig bestückten Java™-Control-Platine mit zugehörigem Bestückungsplan, links von der Displayseite, rechts von der Lötseite



Ansicht des fertig bestückten Evaluation Boards mit zugehörigem Bestückungsplan

eine hohe Nachbausicherheit. Als Werkzeuge benötigt man lediglich „normales“ Elektronik-Werkzeug wie beispielsweise einen Elektroniklötkolben, Lötzinn und einen Elektronik-Seitenschneider.

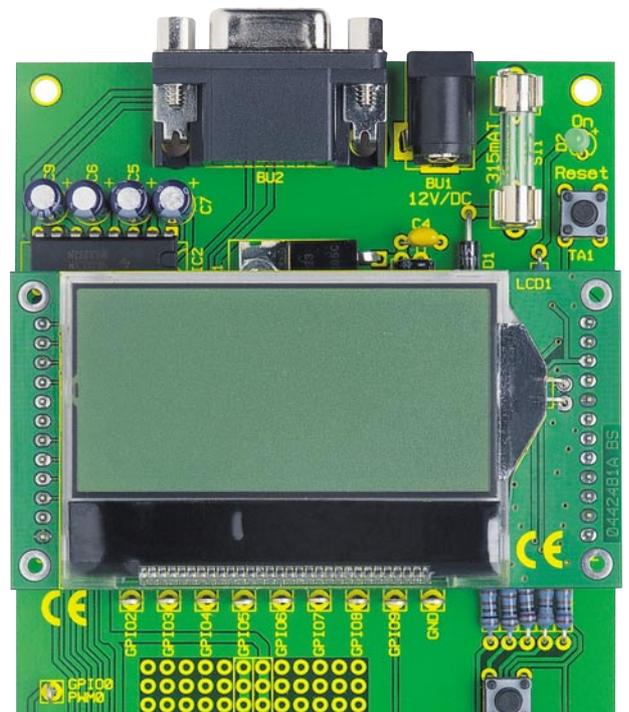
Der Aufbau der Platine erfolgt anhand der Stückliste, des Bestückungsdruckes und des Platinenfotos.

Die Bestückung beginnt traditionell mit den flachsten Bauteilen, in diesem Fall mit der Silberdrahtbrücke, den Widerständen und den Dioden. Die Bauteile sind auf das richtige Rastermaß abzuwinkeln, von der Bestückungsseite einzusetzen und auf der Lötseite vorsichtig zu verlöten. Bei den Dioden D 1 und D 3 ist die Polung zu beachten, sie sind an der Katode mit einem Ring gekennzeichnet und entsprechend dem Schaltzeichen im Bestückungsdruck einzusetzen.

Als Nächstes wird der Spannungsregler IC 1 bestückt. Dazu wird er nach dem Abwinkeln seiner Anschlussbeine um 90 Grad nach hinten, im Abstand von 2,5 mm zum Gehäuse, nach dem Einsetzen der Anschlüsse mit der Schraube M3 x 8 mm, Fächerscheibe und Mutter zuerst auf die Platine geschraubt. Erst dann sind seine Anschlüsse zu verlöten.

Danach wird IC 2 eingelötet, hierbei ist darauf zu achten, dass die Pin-1-Markierung am IC mit der im Bestückungsdruck übereinstimmt.

Nun folgen die Taster TA 1 bis TA 6 (bis zum Anschlag einsetzen, auf geraden Sitz achten), die Keramik-Kondensatoren C 2 und C 4 sowie die liegend zu bestückenden



Hier ist die Java™-Control-Unit bereits auf dem Evaluation Board montiert.

Stückliste: Evaluation Board

Widerstände:

47 Ω	R4
470 Ω	R1
1,2 kΩ	R6, R7
1,8 kΩ	R8
2,7 kΩ	R9
10 kΩ	R5
47 kΩ	R2, R3

Kondensatoren:

100 nF/ker	C2, C4
1 µF/100 V	C5–C9
47 µF/25 V	C1, C3

Halbleiter:

7805	IC1
MAX232	IC2
1N4001	D1
1N4148	D3
LED, 3mm, Grün	D2

Sonstiges:

Hohlsteckerbuchse, 2,1 mm, print	BU1
Sub-D-Buchsenleiste, 9-polig, print	BU2
Mini-Drucktaster, 1 x ein, 1 mm Tastknopflänge	TA1–TA6
Soundtransducer, 3V, print	PZ1
Buchsenleisten, 1 x 12-polig, print, gerade	ST1, ST2
Sicherung, 0,315 A, träge	SI1
8 Lötstifte mit Lötöse	
1 Platinensicherungshalter (2 Hälften)	
1 Zylinderkopfschraube, M3 x 8 mm	
1 Mutter, M3	
1 Fächerscheibe, M3	
4 Gehäuse-Gummifüße, zylindrisch, Schwarz	
3 cm Schaltdraht, blank, versilbert	

Elektrolyt-Kondensatoren C 1, C 3 und C 8. Hier ist wieder die richtige Polung zu beachten (Minuspol ist am Elko markiert). Entsprechend sind die Anschlüsse abzuwickeln, und der jeweilige Kondensator wird gemäß des Bestückungsdrucks eingelötet.

Auch beim danach zu bestückenden Soundtransducer PZ 1 ist die Polung zu beachten, hier muss die Plus-Markierung am Gehäuse mit der im Bestückungsdruck übereinstimmen.

Schließlich wird die Bestückung abgeschlossen mit dem Einsetzen der Buchsenleisten ST 1 und ST 2 (Buchsenkörper plan auf die Platine aufsetzen), der Sicherungshalter für die Sicherung SI 1 (die Winkel, die das Herausrutschen der Sicherung verhindern sollen, müssen jeweils nach außen weisen), der Hohlsteckerbuchse BU 1

(Buchsenkörper plan auf die Platine setzen), der stehend eingebauten Elektrolyt-Kondensatoren C 5, C 6, C 7 und C 9 (Polung beachten, Minuspol ist am Elko gekennzeichnet), der Sub-D-Buchse BU 2 (auch hier Buchsenkörper plan auf die Platine setzen), der Lötstifte und der LED D 2. Bei Letzterer ist wieder der polrichtige Einbau zu beachten. Der längere Anschluss ist die Anode, er gehört an die Plus-Markierung im Bestückungsdruck. Die Einbauhöhe der Leuchtdiode kann nach eigenem Wunsch gewählt werden.

Nun sind nur noch die Gummifüße durch die entsprechenden Bohrungen an den Ecken der Platine sowie die Sicherung SI 1 in den Sicherungshalter einzusetzen.

Vor der Inbetriebnahme erfolgt noch eine sorgfältige Kontrolle auf Bestückungsfehler sowie Lötbrücken.

Funktion und Bedienung

In den folgenden Abschnitten wollen wir näher auf einzelne Komponenten der Java™-Control-Unit eingehen. Im Zusammenhang mit der Programmierung werden hierbei zum Teil Packages, Klassen und Methoden (kursiv gedruckt) genannt, mit denen Einstellungen vorgenommen, Eingaben abgefragt und Ausgaben gemacht werden können. Ausführliche Beschreibungen aller Packages mit den darin enthaltenen Klassen sind in der API-Dokumentation und der der JCU 10 beiliegenden Dokumentation zu finden.

Als ersten Einstieg in die Praxis beschäftigen wir uns außerdem mit den Systemeinstellungen und dem zur Kommunikation wichtigen „Download-Mode“.

Display

Der Kontrast des Displays kann über die Systemeinstellungen mit dem Programm PropertyEdit aus der JControl/IDE eingestellt werden, alternativ kann man auch aus der Anwendungssoftware heraus den Displaykontrast mit der Methode

```
setProperty(String key, String wert)
```

aus der Klasse

```
jcontrol.system.Management
```

ändern (siehe auch Abschnitt „Systemeinstellungen“).

Die Hintergrundbeleuchtung der Java™-Control-Unit ist in der Anwendungssoftware einstellbar. Die Einstellung wird mit der Methode

```
setBrightness(int value)
```

aus der Klasse

```
jcontrol.io.Backlight
```

vorgenommen, in der die Beleuchtung in 256 Schritten von 0 (Aus) bis 255 (max. Helligkeit) einstellbar ist.

Die Klasse

```
jcontrol.io.Display
```

enthält neben den Methoden zum Ansteuern einzelner Pixel und dem Darstellen von einfachen geometrischen Figuren auch die Methode

```
setDisplayMode(int mode)
```

mit der sich die Darstellung des Displays invertieren lässt.

Für die Programmierung des Displays ist noch zu erwähnen, dass die Nummerierung der Zeilen und Spalten in der linken oberen Ecke mit 0 beginnt und dann für die Zeilen unten mit 63 und für die Spalten rechts mit 127 endet.

IO-PINS – GPIO

Die Java™-Control-Unit verfügt über 10 konfigurierbare IO-Anschlüsse, von denen jeder als GPIO (General Purpose Input Output) einsetzbar ist. Die Klasse

```
jcontrol.io.GPIO
```

steht zur Verfügung, um diese IOs zu steuern. Es gibt vier verschiedene Konfigurationsmöglichkeiten:

- FLOATING: digitaler Eingang
- PULLUP: digitaler Eingang mit internem Pull-up-Widerstand
- PUSH/PULL: digitaler Ausgang
- OPENDRAIN: digitaler Ausgang (high impedance bei High-Signal)

PWM

Zwei der IO-Pins (GPIO 0/PWM 0 und GPIO 1/PWM 1) sind alternativ als PWM-Ausgänge nutzbar. Die Methoden zur Aktivierung und Konfiguration befinden sich in der Klasse:

```
jcontrol.io.PWM
```

Aktiviert und deaktiviert wird die Funktion der PWM-Ausgänge mit der Methode

```
setActive(int channel, boolean active)
```

mit der für den entsprechenden Kanal der Aktiv-Status gesetzt oder zurückgesetzt werden kann.

Das Tastverhältnis lässt sich für jeden der Kanäle mit der Methode

```
setDuty(int channel, int duty)
```

in 256 Stufen einstellen.

Da der Mikrocontroller nur einen Fre-

Tabelle 1: Widerstands-/Spannungswerte für den analogen Tastaturdecoder

Taste	zugeordneter Wert	Widerstand	Widerstandswert in Ohm	Spannung	Ergebnis der A/D-Wandlung
TA 2	S (select)		0	0	0
TA 3	U (up)	R6	1100	0,5	25
TA 4	D (down)	R7	1300	1,0	50
TA 5	L (left)	R8	1800	1,5	76
TA 6	R (right)	R9	2400	2,0	102
Eigene 1*	1	R10*	3300	2,5	127
Eigene 2*	2	R11*	5100	3,0	153
Eigene 3*	3	R12*	8200	3,5	179
Eigene 4*	4	R13*	16.000	4,0	204
Eigene 5*	5	R14*	51.000	4,5	230

* Diese Bauteile sind nicht auf dem Evaluation Board vorhanden

quenzgenerator für alle PWM-Kanäle verwendet, ist die Frequenz mit der Methode

```
setFrequency(int freq)
```

für alle Kanäle gemeinsam im Bereich von 244 Hz bis 32,7 kHz in 1-Hz-Schritten einzustellen.

A/D-Wandler

Acht der zehn IO-Pins (GPIO 2/ADC 0 bis GPIO 9/ADC 7) können alternativ als 8-Bit-Analog-Digital-Wandler benutzt werden. Die Methode

```
int getValue(int channel)
```

zur Abfrage der einzelnen Kanäle befindet sich in der Klasse:

```
jcontrol.io.ADC
```

Wenn ein IO-Pin als A/D-Wandler benutzt werden soll, so ist er als FLOATING zu konfigurieren. Als Referenzspannung für die A/D-Wandler wird die Spannung VDDA (Pin 23 an der Stiftleiste ST 2) verwendet. Beim Evaluation Board ist diese Spannung identisch mit der Betriebsspannung; bei Anwendungen, bei denen es erforderlich ist, mit einer stabileren bzw. anderen Referenzspannung zu arbeiten, kann hier auch eine andere Spannung an-

gelegt werden, die aber nicht größer als die Betriebsspannung (5 V) sein darf.

Analoger Tastaturdecoder

Der analoge Tastaturdecoder für bis zu 10 Tasten arbeitet intern mit einem Analog-Digital-Wandler und ordnet verschiedenen Spannungswerten bestimmte Tasten zu.

Über den Widerstand R 5 liegt der Eingang des A/D-Wandlers auf „high“, solange keine Taste gedrückt ist. Beim Betätigen der einzelnen Tasten werden jeweils verschiedene Widerstände gegen Masse geschaltet. So entsteht ein Spannungsteiler, der am Eingang des A/D-Wandlers für jede Taste eine bestimmte Spannung erzeugt. Diese Spannungen werden vom A/D-Wandler alle 16 ms gemessen und ausgewertet. In Tabelle 1 ist zu sehen, welchen Spannungswerten die Tasten zugeordnet werden, diese Werte gelten für eine Referenzspannung der A/D-Wandler (VDDA) von 5 V. Wie in Abbildung 11 dargestellt, können bis zu fünf weitere Tasten – „Eigene 1“ bis „Eigene 5“ – mit den zugehörigen Widerständen R 10 bis R 14 zusätzlich zu den auf dem Evaluation Board vorhandenen Tastern am Lötstift KBIN angeschaltet werden.

Der Mikrocontroller arbeitet beim Auswerten der Spannungen mit Toleranz, so dass jedem Spannungswert eine Taste zugeordnet wird, auch wenn dieser nicht ge-

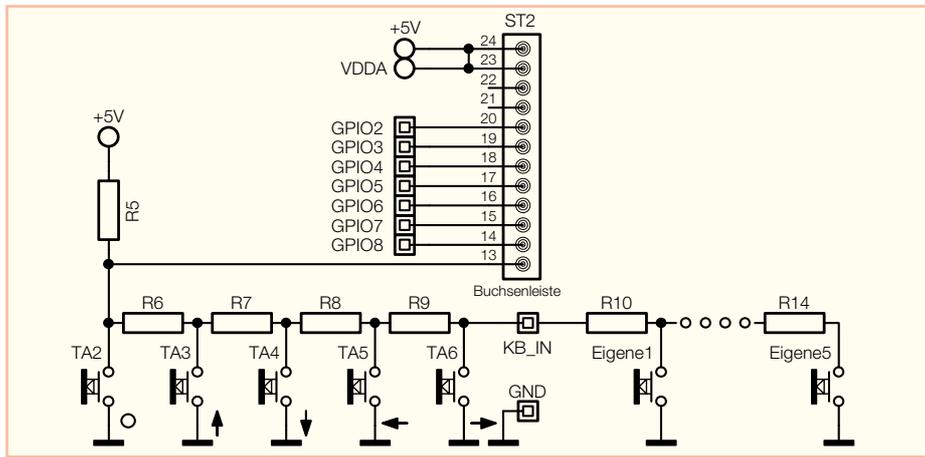


Bild 11: Anschluss weiterer Taster

nau mit dem theoretischen Wert übereinstimmt.

Die Methoden zum Abfragen der Tasten werden in der Klasse

```
jcontrol.io.Keyboard
```

zur Verfügung gestellt.

Buzzer

Beim Buzzer-Ausgang (Pin 10 der Stiftleiste) handelt es sich um einen PWM-Ausgang, der zur Ausgabe von akustischen Signalen von der System- und der Anwendersoftware verwendet werden kann. Von der Systemsoftware aus kann der Buzzer z. B. bei jedem Tastendruck ein Signal ausgeben, und in der Anwendersoftware ist der Buzzer-Ausgang mit den Methoden der Klassen

```
jcontrol.io.Buzzer  
und jcontrol.toolkit.iMelody
```

ansteuerbar. Mit diesen Methoden kann der Ausgang für eine bestimmte Zeit mit einer bestimmten Frequenz aktiviert werden oder auch ganze Melodien im iMelody-Format abspielen.

RS-232-Schnittstelle

Die Java™-Control-Unit bietet eine serielle Schnittstelle mit TTL-Pegeln. Die Anschlüsse TXD, RXD, RTS und CTS finden sich an den Pins 1 bis 4 der Stiftleiste ST 1.

Beim Evaluation Board werden diese Anschlüsse über einen Pegelwandler an eine Sub-D-Buchse geführt. Die Klasse

```
jcontrol.comm.RS232
```

stellt hierbei die Methoden zum Senden, Empfangen und zur Konfiguration der Schnittstelle zur Verfügung.

Die Schnittstelle unterstützt die in Tabelle 2 angegebenen Baudraten, die mit der Methode

```
setBaudrate(int baudrate)
```

einstellbar sind. Wird von der Applikation her keine Baudrate eingestellt, kommt der Wert zum Einsatz, der in den Systemeinstellungen vorgegeben ist. Versucht eine Applikation, eine ungültige Baudrate einzustellen, wird immer der Standardwert 19.200 benutzt.

Die Übertragung erfolgt mit 8 Datenbit und einem Stopbit. Als Parameter der Datenübertragung lassen sich mit der Methode

```
setParams(int params)
```

die Parität, die Fluss-Steuerung und das Echo der empfangenen Zeichen einstellen.

**Tabelle 2:
Unterstützte Baudraten**

Baudrate	Parameter für <i>setBaudrate(int baudrate)</i>
300	300
600	600
1200	1200
4800	4800
9600	9600
19.200	19200
31.250	31250
62.500	62
125.000	125
250.000	250

Tabelle 3 zeigt, welche Bits in *params* für die jeweiligen Einstellungen gesetzt sein müssen und welche nicht.

I²C-Bus

Der I²C-Bus ist der Industriestandard für die Onboard-Kommunikation zwischen integrierten Bauelementen. Der SM-Bus (system management) ist ein auf dem gleichen Prinzip basierender Bus, der auf einigen PC-Motherboards zur Temperaturüberwachung und Lüftersteuerung verwendet wird.

An der Stiftleiste finden sich die Busleitungen an Pin 5 und 6, wobei an Pin 5 die Taktleitung (SCL bzw. SMBCLK) und an Pin 6 die Datenleitung (SDA bzw. SMBDAT) angeschlossen ist.

Die Klasse

```
jcontrol.comm.I2C
```

stellt die Methoden zur Verfügung, mit denen die Java™-Control-Unit als Bus-Master betrieben werden kann, um Daten und Datenanforderungen zu senden sowie angeforderte Daten zu empfangen.

Echtzeit-Uhr

Die Java™-Control-Unit stellt eine per Software nachgebildete, so genannte Real Time Clock (RTC) zur Verfügung. Die Zeit wird mit Jahr, Datum, Stunden, Minuten und Sekunden angegeben. Neben der aktuellen Zeit ist auch noch eine Alarmzeit programmierbar, die ein Alarm-Flag

Tabelle 3: Parameter der seriellen Schnittstelle

Parität	Echo	Fluss- Steuerung	Wert für params in <i>setParams (int params)</i>					
			Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
keine even odd	on off	keine Software (XON/XOFF) Hardware (RTS/CTS)	0	0			0	0
			0	1			0	1
			0	0	0		1	0
			0	1				
			1	0				

setzt, wenn die aktuelle Zeit mit der Alarmzeit übereinstimmt. Die Alarmzeit kann auf täglichen Alarm oder auch mit Datum programmiert werden.

Die Klasse

```
jcontrol.system.RTC
```

enthält Methoden, mit denen man die Systemzeit und die Alarmzeit setzen und auslesen lassen kann. Die Zeiten werden in Instanzen der Klasse

```
jcontrol.system.Time
```

gespeichert, die die Felder Jahr, Monat, Tag, Wochentag, Stunde, Minute und Sekunde in einer Struktur zusammenfasst.

Systemeinstellungen

Die Systemeinstellungen der Java™-Control-Unit lassen sich im Download-Mode mit dem Programm „PropertyEdit“ ansehen und auch einstellen, falls es sich um eine veränderbare Einstellung wie z. B. Keyboard Beep, System Beep, Display Contrast oder Baudrate handelt.

Alternativ sind diese Einstellungen auch, wie schon bei der Einstellung des Displaykontrasts beschrieben, aus der Anwendungssoftware heraus änderbar.

Dies ist mit der Methode

```
setProperty(String key, String value)
```

aus der Klasse

```
jcontrol.system.Management
```

möglich. Die Parameter sind hier beide als Strings einzugeben, die „keys“ für die entsprechenden Einstellungen sind in der API-Dokumentation aufgelistet, und der Wert ist als Zahl in Anführungszeichen anzugeben, z. B.:

```
setProperty("display.contrast", "90")
```

Sollen die in der Anwendungssoftware geänderten Einstellungen auch nach einem Reset erhalten bleiben, so ist nach der Änderung die Methode

```
saveProperties()
```

aufzurufen.

Download-Mode

Der Download-Mode wird benutzt, um mit der Entwicklungsumgebung „JControl/IDE“ Anwendungsprogramme in den Flash-Speicher der Java™-Control-Unit zu schreiben oder um den Flash-Speicher auszullesen. Befindet sich die Java™-Control-Unit im Download-Mode, kann das Modul auch von der „JControl/IDE“ mit der Autodetect-Funktion erkannt werden, und mit dem Tool „PropertyEdit“ sind die Systemeinstellungen des Moduls, wie bereits besprochen, veränderbar.

In den Download-Mode kann man auf verschiedene Arten gelangen:

- Wird die Java™-Control-Unit eingeschaltet und es befindet sich keine Anwendungssoftware im Flash-Speicher, wird nach der Initialisierung automatisch in den Download-Mode gewechselt.
- Wenn ein Reset ausgeführt wird, d. h. Pin 22 der Stiftleiste ST 2 auf Low-Potenzial liegt, während GPIO 9/KBIN auch auf Low-Potenzial liegt. Befindet sich die Java™-Control-Unit im Evaluation Board, kann sie also durch Drücken der Tasten „Select“ und „Reset“ in den Download-Mode gebracht werden.
- Wenn in einer Anwendung die Methode

```
run()
```

aus der Klasse

```
jcontrol.system.Download
```

aufgerufen wird.

Wird der Download-Mode nicht aus einer Anwendung heraus gestartet, also z. B. durch Drücken der Tasten „Select“ und „Reset“, so erscheint auf dem Display eine Informationsseite, in der der Name des Moduls, das Build-Datum der JCVM 8 und die Parameter der seriellen Schnittstelle (19.200 Baud, 8 Datenbit, keine Parität, 1 Stoppbit), die sich für den Download-Mode nicht ändern lassen, zu sehen ist.

Beim Wechsel in den Download-Mode aus einer Anwendung heraus wird keine Informationsseite angezeigt, und für die serielle Schnittstelle werden die Parameter aus den Systemeinstellungen verwendet.

In der nächsten Ausgabe beschäftigen wir uns mit der Programmierung der Java™-Control-Unit und werden anhand eines einfachen Beispielprogramms bzw. Projektes zeigen, wie man von der Erstellung des Quelltextes bis zur laufenden Anwendung kommt.