USB-Stick mit ARM

Gigabyte-Flashdrive am Mikrocontroller

Von Ursula Engelmann-Schrader und Jürgen Engelmann

Mit diesem Projekt geht ein lang gehegter Wunsch vieler µC-Entwickler in Erfüllung: Ein Gigabyte-Flashspeicher, der sich sowohl an den Mikrocontroller als auch an den PC anschließen lässt, einfachen Datentransfer ermöglicht und auch noch einen LCD-Anschluss und eine Datenlog-Funktion bietet. In Bayern nennt man so etwas eine eierlegende Wollmilchsau…

Nehmen wir einmal an, Sie hätten die Aufgabe, bei einem Windkraft- oder Solar-Projekt die anfallenden großen Datenmengen mit einem Mikrocontrollersystem dauerhaft aufzuzeichnen und später am PC auszuwerten. Wäre es da nicht schön, einen Speicher zu haben, bei dem es auf ein paar Megabyte mehr oder weniger nicht ankommt? Und ganz schön praktisch, wenn man den Speicher nicht nur an den Mikrocontroller, sondern auch direkt an den PC anschließen könnte? Die ideale Lösung wäre so etwas wie ein USB-Stick, der auch über eine RS232-Schnittstelle verfügt. Genau das erhalten Sie mit diesem Projekt!

Janus-Speicher

Intelligenterweise ist dieser „USB-Stick für Mikrocontroller" nicht wirklich ein mit Speicherchips voll gestopfter USB-Stick, sondern eine kleine Platine mit einem Slot für eine MMCard oder SD-Card (FAT16 formatiert). Somit kann man die Kapazität bis in den Gigabyte-Bereich flexibel an die Aufgabe anpassen, vom weiteren Preisverfall dieser Speichermedien profitieren und hat außerdem noch die Wahl zwischen verschiedenen Karten-Qualitäten und Herstellern.
Wie der römische Gott Janus hat auch dieser USB-Flashdrive zwei Gesichter:
Auf der USB-Seite funktioniert er wie ein handelsüblicher USB-Stick und ist genauso Windows- und Linux-PC-kompatibel. Nach dem Einstecken in den PC werden die in der Flashcard gespeicherten Dateien auf dem Bildschirm angezeigt und können wie gewohnt weiterverarbeitet werden. So lassen sich die Daten aus dem zuvor (oder noch) angeschlossenen Mikrocontrollersystem super einfach auf dem PC auswerten.
Was ein normaler USB-Stick aber nicht kann, ist der umgekehrte Weg: Daten, die vom PC in die Flashcard geschrieben wurden, können auch vom Mikrocontrollersystem ausgelesen werden. Dafür sorgt der ARM-Controller mit seiner Firmware, der dem Mikrocontroller auf der RS232-Seite der Platine einen einfachen Schreib-Lese-Zugriff auf den Flash-Speicher ermöglicht. Auf dem Foto oben ist der USB-Flashdrive als Beispiel an das ATmega-Board aus Heft Mai 2006 angeschlossen (Bausatz EPS 050176-71, siehe ELEKTOR-Shop-Anzeige am Heftende).
Der Zugriff vom Mikrocontrollersystem auf die Speicherplatine erfolgt über einen Treiber, der über einen Satz von vordefinierten Befehlen verfügt. Es handelt sich dabei um einfache Befehle wie FileOpen, FileRead, FileWrite, FileClose und so weiter. Der Treiber wird einfach in das vorhandene Programm eingebunden und ermöglicht nicht nur den Zugriff auf das Dateisystem der MMC/SD-Card, sondern auch auf die LCD-Schnittstelle, für die spezielle Befehle enthalten sind.
Der C-Quelltext des Treibers und einige Beispiele für 8051-kompatible Controller stehen auf den Webseiten [1] und [2] zum Download zur Verfügung, ebenso auch einige Pascal-Beispieldateien, die aus einer früheren Anwendung der Platine stammen. Die Speicherplatine wurde dabei als externes Laufwerk an der seriellen Schnittstelle eines DOS-Systems betrieben. Ein weiterer Bestandteil des kostenlosen Software-Downloads ist ein Windowsprogramm zum Testen der Platine („Testsuite").
Besonders einfach ist die Anwendung des USB-Flashdrives im Datenlogger-Modus, der im Textkasten „Datenlogger" beschrieben wird.

ARM7-Controller

Kern- und Herzstück der Schaltung in **Bild 1** ist der ARM7-Controller AT91SAM7S64 (IC1). Es handelt sich dabei um einen 32-bit-Controller mit einer RISC-CPU. Der Befehlssatz der CPU lässt sich je nach Anforderung zwischen 32-bit- und 16-

und RS232

bit-Befehlen umschalten. Dem Programmierer werden vom Controller beachtliche 64 KB Flashspeicher und 16 KB RAM zur Verfügung gestellt. Eine PLL setzt die Taktfrequenz von 12 MHz (X1) intern auf 48 MHz hoch.
Der Controller ist mit einer ganzen Reihe von Zusatzfunktionen ausgestattet. Unter anderem hat der ARM7-Controller die vollständige USB-Hardware bereits an Board, sodass lediglich die vom USB-Stecker kommenden Signale D- und D+ angeschlossen werden. Die beiden Widerstände R9 und R10 bilden einen Spannungsteiler, über den der Controller feststellt, ob die Schaltung an einen USB-Bus angeschlossen ist. Eine kleine Übersicht über die Eigenschaften des ARM7-Controllers ist im Textkasten zu finden.
Wer sich mit dem ARM7-Controller näher beschäftigen möchte, der kann neben kommerziellen Compilern auch einen GNU-Compiler nutzen.

Kurzdaten

Speichermedium:	MMC oder SD-Card bis 2 GB
Dateisystem:	FAT16 (max. vier Dateien gleichzeitig geöffnet)
Schnittstellen:	1 x USB (2.0 und 1.1), 2 x RS232
Datenrate USB:	12 Mbit/s
Datenrate RS232:	9600 bit/s bis 230 kbit/s
Spannungsversorgung:	über USB oder externe 5-V-Spannung
Stromaufnahme:	ca. 50 mA
Optionen:	LCD-Anschluss, serielle Schnittstelle mit TTL-Pegel
Abmessungen:	ca. 41 mm x 77 mm x 18 mm (inkl. Stecker und Karte)

Schnittstellen

Die USB-Schnittstelle (Anschluss ST2) wird mit 12 Mbit/s (brutto!) betrieben und kann an PCs mit USB 1.1- oder USB-2.0-Port angeschlossen werden. Dem PC wird die Geschwindigkeit (Highspeed) über den Widerstand R11 an D+ signalisiert.

Die RS232-Schnittstelle ist konventionell mit MAX232 (IC11) ausgelegt, der mit einer 9-poligen SUB-D-Buchse (CON2) verbunden ist. Eine zweite serielle Schnittstelle ist an JP2 herausgeführt. Der MAX232 sorgt für die Pegelwandlung zwischen nominal ±12 V auf der RS232-Seite und TTL-Pegel auf der Seite des ARM-Controllers IC1. Die RS232-Schnittstelle kann im Bereich von 9600 bit/s bis
230 kbit/s betrieben werden, wobei der ARM-Controller nach einem Reset die 9600 bit/s als Voreinstellung (Default) auswählt. Optional ist es auch möglich, die serielle Schnittstelle an Pin 2 und 3 von JP1 mit TTL-Pegeln zu betreiben. Dazu wird der Löt-Jumper SJ1 geöffnet, um Konflikte mit den Pegeln der Schnittstelle (CON2) zu vermeiden.
Die LCD-Schnittstelle an der 21-poligen Stiftleiste JP3 erlaubt den Anschluss von gängigen Displays mit HD44780 oder ähnlichem Controller. Unterstützt werden ein-, zwei- und vierzeilige Displays mit 16 und 20 Zeichen pro Zeile. Die Spannungsversorgung wird dabei dem USB-Bus entnommen. Bei Anschluss eines LCDs muss auch das optionale SMD-Trimmpotentio-

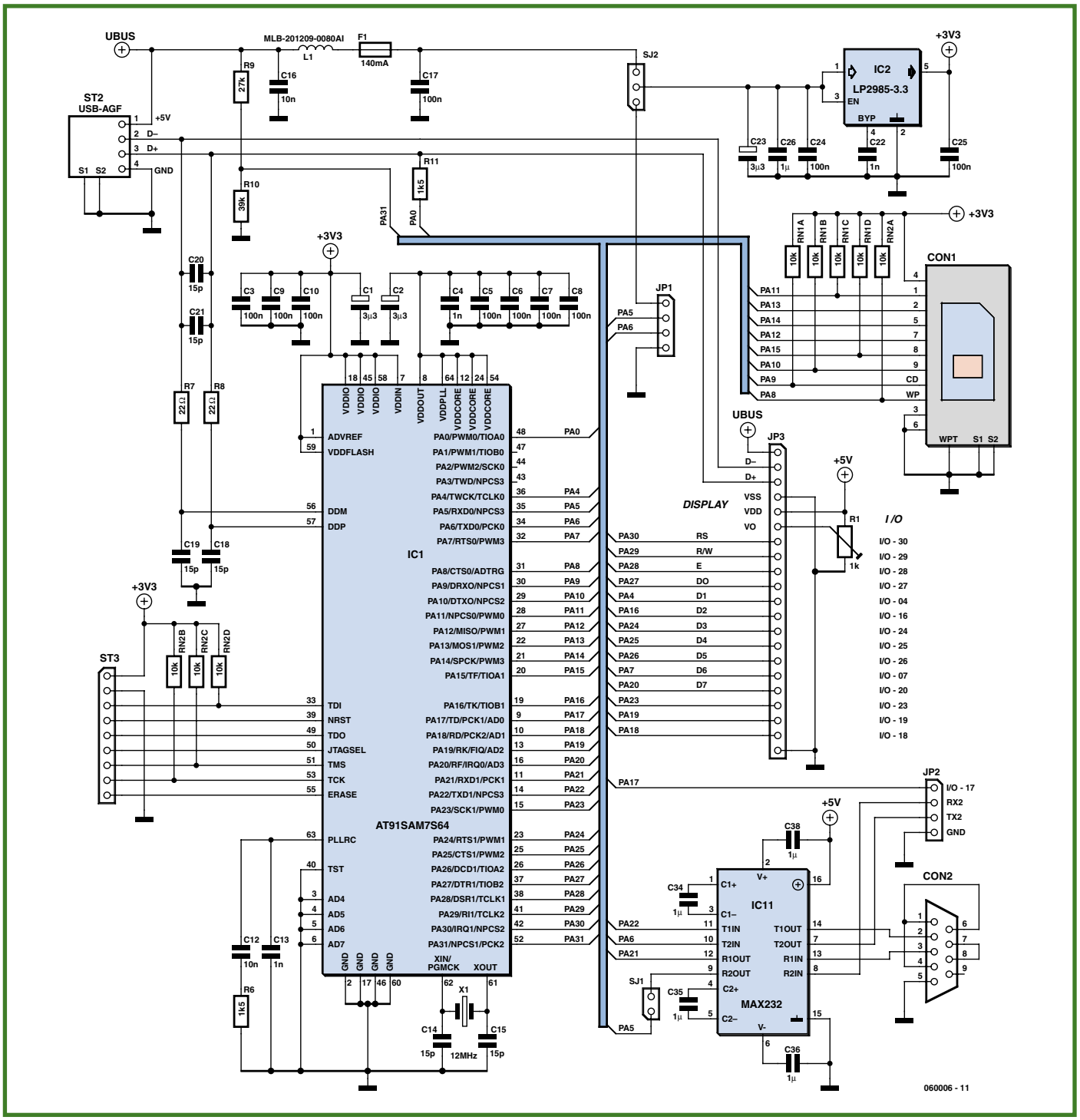


Bild 1. Kern und Herzstück der Schaltung ist ein ARM7-Controller, der die USB-Schnittstelle bereitstellt. Seine Firmware ermöglicht einem über RS232 angeschlossenen Mikrocontrollersystem den einfachen Schreib-Lese-Zugriff auf den Flash-Speicher.

meter (R1) zur Kontrasteinstellung bestückt werden. Das Display belegt elf Pins von JP3. Acht sind für die Daten und je ein Pin ist für RS (Register Select), R/W (Read/Write) und E (Enable) vorgesehen. Ist kein Display angeschlossen, stehen PA7 bis PA20 als allgemeine I/O-Pins zur freien Verfügung.
Der ARM-Controller verfügt auch über eine SPI-Schnittstelle, die hier zur Ansteuerung der Flash-Karten verwendet und im folgenden Abschnitt beschrieben wird.

Speicherkarten

Der Kartenhalter (CON1) auf der Platine nimmt MMC- und SD-Cards auf, wobei er die unterschiedlichen Dicken der beiden Kartentypen ausgleicht. Die MMC- bzw. SD-Cards sind preiswert, gut erhältlich und bieten hohe Speicherkapazitäten. Die Schaltung unterstützt Karten bis zu 2 GB.
Flash-Speicher haben eine durch die Zahl der Löschzyklen (und damit auch der Schreibzyklen) begrenzte Lebensdauer. Diese „Endurance" hängt maßgeblich von der verwendeten Technologie (NOR- oder NAND-Flash) ab. MMC- und SD-Cards gibt es in der Standardausführung (NOR-Flash) mit etwa 100.000 Schreibzyklen und in Industriequalität mit zum Beispiel 400.000 Schreibzyklen und einem erweiterten Temperaturbereich.
Steckt man eine SD-Card in den Kartenhalter, so wird der Schreibschutzschalter

ARM7-Controller

Kurzcharakteristik des AT91SAM7S64

- 32-bit-RISC Architektur
- 64 KB Flash, 16 KB SRAM
- Reset-Controller, Brownout-Detector
- 32 I/O-Pins (Pullups)
- 2 UART, SPI (8- bis 16-bit), TWI
- USB 2.0 Full Speed (12 Mbit/s), DMA-Controller
- Mehrere Timer, z. B. 16-bit-Timer/Counter, PWM-Controller
- 8-Kanal-10-bit-A/D Wandler
- JTAG, umfangreiche Emulations- und Debugmöglichkeiten
- 5-V-tolerante I/Os, 4 I/O-Pins mit bis zu 16 mA
- 64-Pin-LQFP-Gehäuse

dieses Kartentyps ausgewertet. Die Karten sind in Sektoren von 512 Byte aufgeteilt. Dies ist auch die Größe, die in einem Schritt gelesen und geschrieben wird.
Als Dateisystem wird standardmäßig das FAT16-Format unterstützt. So können die Karten auch entnommen und in anderen Geräten (zum Beispiel Kartenlesegeräten) weiter bearbeitet werden.
Beide Kartenvarianten besitzen eine SPI-Schnittstelle. Damit verfügen sie neben dem MMC- bzw. SD-Modus über einen SPI-Modus, den der ARM7-Controller mit seinem SPI-Interface in der Schaltung nutzt. Das SPI-Interface besteht aus 4 Signalleitungen: MOSI (Master Out Slave IN), MISO (Master In Slave Out), SCK (Serial Clock), /SS (Slave Select neg.). Die eingelegte Karte wird an der SPI-Schnittstelle mit einer Bruttogeschwindigkeit von 12 Mbit/s beschrieben und gelesen. Die

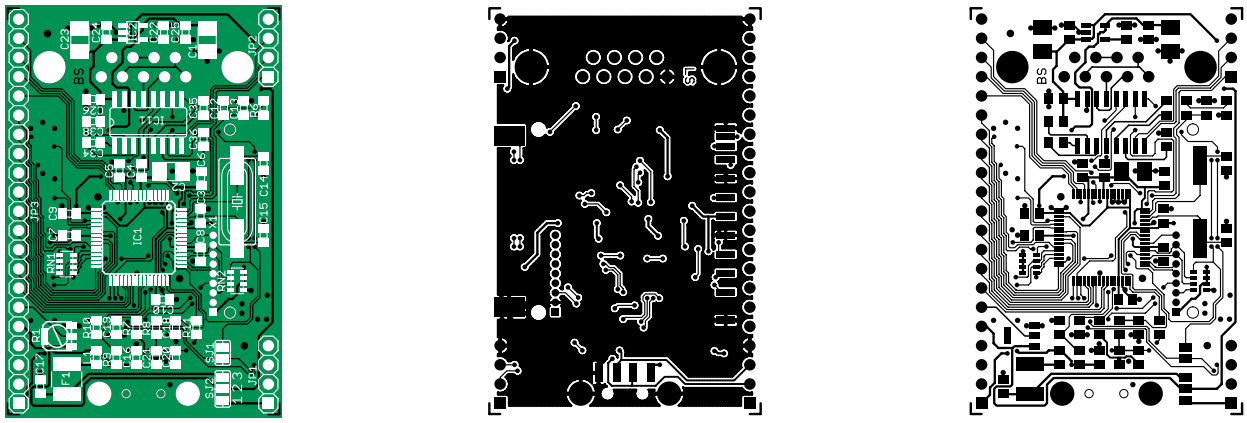


Bild 2. Layout und Bestückungsplan der SMD-bestückten Platine.

Stückliste

Widerstände:
R1 = SMD-Potentiometer 1 k, 1 % (optional, siehe Text)
RN1, RN2 = Widerstandsnetzwerk 4x10 k, 5%
R6, R11 = 1k5 (SMD 0805, 1%)
R7, R8 = 22 Ω (SMD 0805, 1%)
R9 = 27 k (SMD 0805, 1%)
R10 = 39 k (SMD 0805, 1%)

Kondensatoren:
C1, C2, C23 = 3µ3 (SMD 3528, Tantal, 20%)
C3, C5, C6, C7, C8, C9, C10, C17, C24, C25 = 100 n (SMD 0805, 10%)
C4, C13, C22 = 1 n (NP0, SMD 0805, 5 %)
C12, C16, C17 = 10 n (SMD 0805, 10%)
C14, C15, C18, C19, C20, C21 = 15 p (SMD 0805, 5%)
C26, C34, C35, C36, C38 = 1 µ (SMD 0805, 10%)

Halbleiter:
IC1 = AT91SAM7S64 (programmiert, EPS060006-41)
IC2 = LP2985A-33DBVT (SOT23, TI)
IC11 = MAX232 (SO16, Maxim)

Außerdem:
X1 = 12-MHz-Quarz (SM49)
F1 = Polyswitch 140 mA
CON1 = Kartenhalter für SD/MMC-Karten
CON2 = Sub-D-Buchse, 9-polig, gewinkelt, für Platinenmontage
L1 = Drossel MLB-201209-0080AI (Kitagawa)
ST1 = USB-A-Stecker für Platinenmontage, z.B. Assmann A-USB-A-SMT (Reichelt USB AGF)
JP1, JP2 = 4-polige Stiftleiste, Raster 2,54 mm
JP3 = 21-polige Stiftleiste, Raster 2,54 mm
Platine EPS060006-1 (unbestückt) oder EPS060006-91 (bestückt und getestet, siehe Elektor-Shop-Anzeige am Heftende)
Software (Gratis-Download von www.elektor.de)

Tabelle 1. Befehlsübersicht

Gruppe der USB-Befehle:

Mount	UnMount	CardAvailable	CardNotAvailable
Connect	DisConnect	WriteDisable	WriteEnable
Ping			

Gruppe der Dateibefehle:

FileOpen	FileClose	FileRead	FileWrite
FileSeek	FileTell	Dir	ChangeDir
CreateDir	RenameFile	RemoveFile	QuickFormat

Gruppe der Peripherie-Befehle:

InitDisplay	ClearDisplay	DisplayOff	DisplayWrite
Baudrate	SetDirection	SetOutput	GetInput
SetPullup	GetCardStatus	GetFirmwareVersion	

Beispiel

MOUNT-Anweisung an den Controller:

41h 04h 00h 45h

- 45h: Checksumme 41h XOR 04h XOR 00h ergibt 45h
- 00h: High-Byte der Gesamtlänge der Anweisung
- 04h: Low-Byte der Gesamtlänge der Anweisung
- 41h: Anweisungsbyte

Antwort vom Controller:

41h 01h 40h

- 40h: Checksumme 41h XOR 01h ergibt 40h
- 01h: Ist 1, die MOUNT-Anweisung wurde erfolgreich ausgeführt
- 41h: Wiederholung des Anweisungsbytes

Karte darf während des Betriebs **nicht** entnommen oder eingesteckt werden!

Stromversorgung

Die Platine erhält ihre Betriebsspannung von 5 V wahlweise über den USB-Anschluss oder über den externen Anschluss JP1.
Die Auswahl der Spannungsquelle erfolgt dabei über den Löt-Jumper (SJ2). Ein Spannungsregler (IC2) sorgt für eine geregelte 3,3-V-Systemspannung, die den größten Teil der Schaltung versorgt. Für Anwendungen, die eine Versorgung sowohl vom USB-Port als auch über JP1 erforderlich machen, können beide Punkte des Löt-Jumpers SJ2 geschlossen werden. Doch Vorsicht: Das ist nur dann zulässig, wenn zweifelsfrei sichergestellt ist, dass das Board **nicht gleichzeitig** vom USB-Bus und von einer externen Spannungsquelle versorgt wird!
Die Spule L1 und der Kondensator C17 dienen dem Aussieben von Störsignalen. Die gleiche Aufgabe haben auch die Kondensatoren am USB-Bus und am Spannungsregler. Eine Polyswitch-Sicherung (F1, 150 mA) schützt die Schaltung bei zu hoher Stromaufnahme im Fehlerfall.

Bild 3. Auf der Platinen-Oberseite befinden sich nur drei Bauteile: Der USB-Stecker, die RS232-Buchse und der MMC/SD-Kartenhalter.

Platine

Im Wesentlichen ist die Platine (**Bild 2**) nur auf einer Seite mit SMD-Bauteilen bestückt. Lediglich der USB-Stecker, die RS232-Buchse und der MMC/SD-Kartenhalter befinden sich auf der Oberseite (siehe **Bild 3)**. Die größten Bauteile auf der Unterseite (**Bild 4**) sind der Mikrocontroller AT91SAM7S64, der MAX232 und der 12-MHz-Quarz.

Das Bestücken der SMD-Bauteile ist nur erfahrenen Elektronikern anzuraten. Insbesondere der Controller im 64-poligen LQFP-Gehäuse und die Widerstandsnetzwerke mit der Bauform 1206 sind nicht einfach zu löten. Im ELEKTOR-Shop wird daher neben der unbestückten Platine auch eine fertig bestückte und getestete Version angeboten.

Software

Die Software zu diesem Projekt ist in einer ZIP-Datei unter der Nummer 060006-81 auf der Projektseite bei www.elektor.de zum Download zusammengefasst. Der Textkasten „Software-Dateien" gibt eine Übersicht über die verfügbaren Programme und Dokumente. Es handelt sich hauptsächlich um Treiber- und Beispieldateien für das an die RS232-Schnittstelle der Flash-Platine angeschlossene Mikrocontroller-System.
Die Software im (programmiert erhältlichen) ARM-Controller der Speicherplatine verwaltet das Dateisystem und verarbeitet Befehle, die von der USB- oder RS232-Schnittstelle kommen. Über die USB-Schnittstelle sendet das Betriebssystem des PCs Befehle und spricht den Flashdrive als Laufwerk an. Über die RS232-Schnittstelle erhält die Speicherplatine Befehle vom angeschlossenen Mikrocontrollersystem und sendet Antworten zurück. Wie die Übersicht in **Tabelle 1** zeigt, lassen sich diese Befehle in drei Gruppen einteilen:

- **USB-Befehle**

Hier sind alle Kommandos zusammengefasst, die das Zusammenspiel zwischen der USB-, der RS232-Schnittstelle, dem PC und dem ARM7-Controller regeln.

- **Dateibefehle**

Sie reichen vom Öffnen und Schließen einer Datei über das Anlegen und Wechseln von Verzeichnissen bis zum Formatieren der eingelegten MMC/SD-Card.

- **Peripherie-Befehle**

Anweisungen zur Steuerung des Displays, der freien Ein- und Ausgabe-Pins und der Übertragungsgeschwindigkeit.

Alle Anweisungen haben eine einheitliche Struktur. Jede Anweisung besteht aus einem Befehl zum Controller und einer Antwort vom Controller. Der Befehl zum Controller besitzt einen Befehlsteil, einen Parameterteil (falls Parameter vorhanden sind) und eine Checksumme. Unterschieden wird zwischen Befehlen ohne Parameter und Befehlen mit bis zu vier Parametern. Die Befehle zum Controller haben folgendes Format:

[Befehlsbyte - [0 bis 4 Parameter] - Checksumme]

1. Byte Anweisungsbyte

2. Byte LowByte der Gesamtlänge der Anweisung

3. Byte HighByte der Gesamtlänge der Anweisung

4. Byte LowByte Länge des 1. Parameters
5. Byte HighByte Länge des 1. Parameters
6. Byte Daten des 1. Parameters (1 - 512 Byte)

... bis zu 3 weitere Parameter

n. Byte 8 Bit XOR Checksumme über die gesamte Anweisung

Die Antwort des Controllers sieht ähnlich aus. Der Controller gibt als Antwort auf die meisten Befehle eine drei Byte lange Sequenz zurück. Die Sequenz besteht aus der Wiederholung des Befehlsbytes (1. Byte), der Meldung über Erfolg oder Misserfolg der Ausführung (2. Byte) und einer Checksumme (3. Byte). Bei Befehlen, die Daten anfordern, werden die Daten vor der Sequenz übertragen.

Die Antworten des Controllers haben folgendes Format:

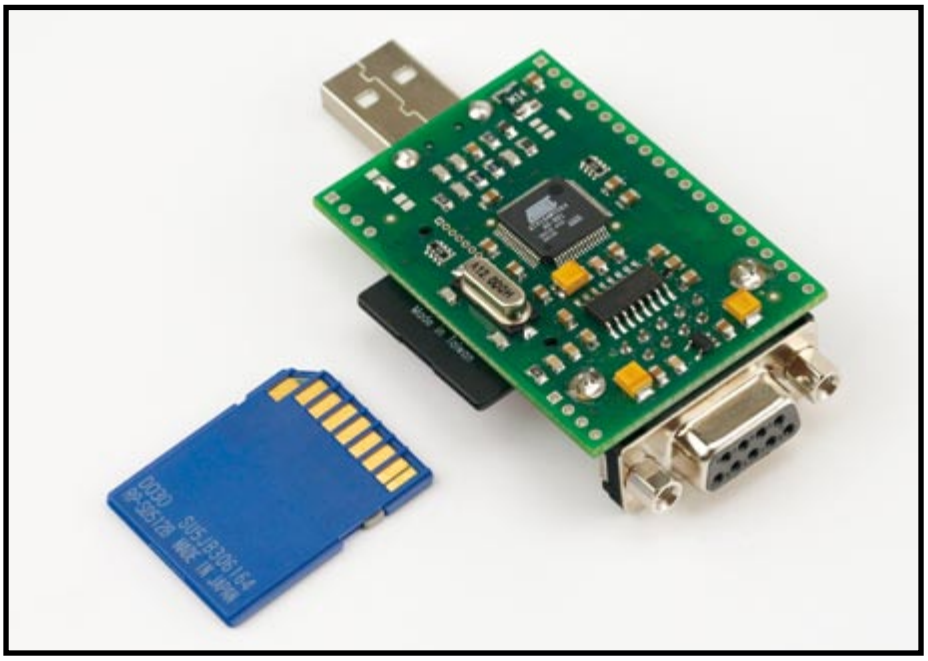


Bild 4. Die größten Bauteile auf der SMD-bestückten Unterseite der Platine sind der Mikrocontroller AT91SAM7S64, der MAX232 und der 12-MHz-Quarz.

NAVTEX

Seine Entstehung verdankt der „USB-Stick für Mikrocontroller" dem Seewetterdienst NAVTEX (Navigational Warnings by Telex). Über das weltweite NAVTEX-Sendernetz werden Seewetterberichte, Sturmwarnungen und nautische Hinweise in unregelmäßigen Abständen ausgegeben. Der internationale Dienst sendet NAVTEX auf der Mittelwellenfrequenz 518 kHz, die nationalen Dienste senden auf 490 kHz [3].

Für NAVTEX-Meldungen wurde ein spezieller Empfänger (siehe Foto) entwickelt. Er nimmt die Berichte in Form von Textnachrichten entgegen und legt sie in einem Flashspeicher ab, der als „Datenträger" dient. Auf einem via USB angeschlossenen PC können die Nachrichten simultan verfolgt werden.

Der Entwicklung des NAVTEX-Empfängers folgte der Wunsch, diese Funktionen auch auf beliebige Mikrocontroller-Systeme zu übertragen. Erweitert um das eine oder andere nützliche Detail entstand so die in diesem Artikel vorgestellte Platine mit RS232- und USB-Schnittstelle sowie MMC/SD-Card als Massenspeicher.

Software-Dateien

060006-81.zip
(Download von www.elektor.de und auch auf CD erhältlich):

Quelltexte/Treiber für C
Ordner Stickdrv:

double.cpp	Befehle an den Stick
double.h	Headerdatei
serial.cpp	serielle Schnittstelle
serial.h	Headerdatei

Ordner Stick51:
C-Beispiel für die 51er-Familie

Quelltexte und Treiber von DOS-ähnlichen Befehlen:
Ordner Stickdos:

COPYS.PAS	Copybefehl
DELS.PAS	Deletebefehl
DIRS.PAS	Dirbefehl
TYPES.PAS	Typebefehl
MISC.PAS	Dies & Das
PARAM.PAS	Parameterbehandlung der obigen Befehle
RS232.PAS	serielle Schnittstelle
STICK.PAS	Befehle an den Stick

Windows-Programm (inkl. Quelltexte)
Ordner Sticktest mit Testsuite für den Stick

Dokumentation:
Ordner Stickdok: Zusatzinformation mit Beschreibung der Befehle

[Befehlsbyte - Ausführung - Checksumme]

1. Byte Wiederholung der Anweisung.

2. Byte Ist „1“, wenn die Anweisung erfolgreich ausgeführt wurde, sonst „0“.

3. Byte 8-bit-XOR Checksumme über die zwei Bytes.

In Tabelle 1 ist als Beispiel der MOUNT-Befehl angegeben. MOUNT ist im Allgemeinen der erste Befehl in einem Programm und enthält keinen Parameterteil. Mit diesem Befehl wird der Controller aufgefordert, einen Teil des Dateisystems der MMC/SD-Card (FAT, Verzeichnis) in das RAM des Controllers zu laden. Erst nach diesem Befehl ist der Controller bereit, andere Befehle aus der Gruppe der Dateibefehle entgegenzunehmen.
Werden alle Bytes eines Befehls an den Controller beziehungsweise einer Antwort des Controllers XOR verknüpft, so muss das Ergebnis 00h ergeben. Bei der Antwort bedeutet dies, dass die Bytefolge mit großer Wahrscheinlichkeit korrekt übertragen wurde. Damit ist aber noch nicht gesagt, dass der Befehl auch ausgeführt wurde. Dafür muss zusätzlich zur Checksumme noch das mittlere Byte geprüft werden. Ist es 01h, wurde der Befehl erfolgreich ausgeführt, ist es hingegen 00h, ist die Ausführung misslungen.

Eine detaillierte Beschreibung der einzelnen Befehle ist ebenfalls im Software-Download (060006-81.zip) unter [1] beziehungsweise in der Dokumentation unter [2] enthalten.

(060006-1e)

Weblinks:

[1] www.elektor.de

[2] www.engelmann-schrader.de

[3] http://de.wikipedia.org/wiki/NAVTEX

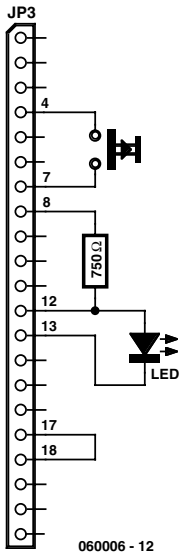
Datenlogger

Der USB-Flashspeicher lässt sich sehr einfach als Datenlogger betreiben. Dafür ist in der Firmware des ARM-Controllers ein Datenlog-Modus vorgesehen, der durch Stecken eines Jumpers an Pin 17 und 18 der 21-poligen Stiftleiste JP3 aktiviert wird. An JP3 werden dann noch drei Bauteile angeschlossen:

- Ein Mikrotaster zwischen Pin 4 (GND) und Pin 7 (PA30),
- ein 750-Ω-Widerstand zwischen Pin 8 (PA29) und Pin 12 (PA16),
- eine rote Low-current-LED (2 mA) mit der Anode an Pin 12 (PA16) und der Kathode an Pin 13 (PA24).

Nach dem Aufstecken des Jumpers leuchtet die LED und zeigt damit an, dass der Datenlog-Modus aktiv ist. Betätigt man nun den Taster, dann wird das Verzeichnis ‚DATEN‘ angelegt (wenn noch nicht vorhanden). Die LED blinkt und der Flashspeicher wartet auf Daten von der RS232-Schnittstelle (9600 bit/s). Diese Daten werden in die Datei ‚DATEN001.TXT‘ geschrieben. Existiert die Datei ‚DATEN001.TXT‘, so wird die Datei ‚DATEN002.TXT‘ angelegt und beschrieben und so weiter. Die Dateinamen reichen von ‚DATEN001.TXT‘ bis ‚DATEN999.TXT‘.

Durch nochmaliges Betätigen des Tasters wird die Datei geschlossen. Die LED leuchtet dann wieder dauernd.



Beschaltung der 21-poligen Stiftleiste JP3 im Datenlog-Modus. Pin 1 ist die quadratische Lötinsel.