

I²C-Real-Time-Clock für Mikroprozessorsysteme

Dieser Uhrenbaustein, auch Real-Time-Clock (RTC) genannt, wird über den bewährten I²C-Bus angesteuert. Die interne Uhr läuft autark und dank Pufferung durch einen Goldcap auch dann weiter, wenn die Betriebsspannung abgeschaltet wird. Es wird ein hochgenauer Sekundentakt generiert, der z. B. zur Weiterverarbeitung mit einem Mikrocontrollersystem einsetzbar ist. Die Platine ist auch für den direkten Einsatz mit einem Arduino-Board ausgelegt.

Genauere Zeit auch ohne Funk

Geht es heute um eine genaue Zeitdatenausgabe, greift man schnell zum Funkuhrsystem. Doch nicht immer ist der Betrieb einer Funkuhr das beste Mittel der Wahl, denn in den beengten Verhältnissen eines Mikroprozessorsystems, zudem vielleicht noch mit einer großflächigen Anzeige, sind die Empfangsbedingungen sehr eingeschränkt. Zudem ist es bei solchen Geräten bei Weitem nicht immer möglich, die Antenne in kritischen Empfangssituationen optimal auszurichten. Ein solches, auch in unserem ELV-Forum diskutiertes Beispiel ist die neue FS20-Zeitschaltuhr FS20 ZSU, die wir in den letzten Ausgaben des ELVjournal vorgestellt

haben. Hier ist ein Display eingebaut, das die Frontseite der Uhr nahezu vollflächig einnimmt. Unter ungünstigen Empfangsbedingungen trägt das – wie auch die räumlich sehr nahe liegende Controllerschaltung – dazu bei, dass es zu massiven Empfangsstörungen kommen kann. Deshalb haben wir hier z. B. auf das gute, alte Quarzuhren-Prinzip zurückgegriffen, aber uns dabei auch der Trickkiste bedient und eine sehr einfache Korrekturmöglichkeit geschaffen, bei der die Abweichung, die ja auch von den klimatischen Bedingungen des in der Regel festen Standorts der Uhr abhängt, so präzise korrigierbar ist, dass die Gangabweichung nach einiger Zeit tatsächlich zu vernachlässigen ist.

Während es bei der FS20 ZSU als Beispiel einfach ist, den Mikroprozessor samt Taktquarz als Zeitmesser zu beschäftigen – schließlich ist er, funktionell bedingt, immer in Betrieb –, sind andere Systeme, z. B. ein PC oder ein tragbarer Computer, meist länger ausgeschaltet als in Betrieb. Hier bewährt sich seit Erscheinen des Personalcomputers ein kleiner Baustein, den man

Daten

Spannungsversorgung:	2–5 Vdc
Stromaufnahme (Stand-by):	max. 2 µA
Interface:	I ² C-Bus
Abmessungen mit Shield-Adapter:	52 x 43 mm
Abmessungen ohne Shield-Adapter:	32 x 27 mm

Die Funktion der Interrupt-Ausgänge

INTRA	Interrupt-Ausgang für Alarm_A, Alarm_B sowie programmierbarer Taktausgang für folgende Intervalle: 0,5 s/1 s/1 min/1 h und jeden Ersten des Monats
INTRB	Ausgang für das Oszillatorsignal (32768 Hz) oder Interrupt Alarm_B

auf jedem Mainboard findet – der RTC-Chip. Das ist ein hochspezialisierter Chip, der mit Hilfe eines externen oder auch internen Quarzes genaue Zeitinformationen inklusive Datum bereitstellt, oft sogar unter Berücksichtigung des Schaltjahres. Näheres zur Technik und zum Aufbau des RTC-Bausteins ist unter „Elektronikwissen“ zusammengefasst.

Ein ganz wesentliches Merkmal des RTC-Bausteins ist die Fähigkeit, unterstützt von einer kleinen Spannungsquelle, die Zeitdaten auch ohne reguläre Betriebsspannung für einen langen Zeitraum weiter zu erzeugen und beim Wiedereinschalten des Systems sofort zur Verfügung zu stellen. Als Spannungsquelle dienen dabei entweder die bei PCs üblichen Backup-Batterien, heute meist als 3-V-Knopfzelle ausgeführt, oder aber Goldcaps, die mit ihrer hohen Kapazität auch lange Abschaltungen überbrücken können. Letzteren kommt entgegen, dass ein RTC im Schlafmodus nur sehr geringe Ströme im Nano-Ampere-Bereich benötigt. Auf älteren PC-Mainboards findet man mitunter sogar RTC-Chips mit direkt im Chipgehäuse untergebrachten, also nicht wechselbaren Knopfzellen, so sicher war man sich der langjährigen Haltbarkeit (bzw. Wartung/Betreuung für einen eventuellen Wechsel) des Systems.

Mit den hier beschriebenen Eigenschaften sowie denen, auf die wir noch in der Folge eingehen werden, ist ein RTC-Chip die ideale Grundlage für das Zeitmanagement eines Mikrocontrollersystems, das zudem

nicht ständig in Betrieb sein muss, da ja der RTC-Baustein, wie gerade erläutert, gepuffert wird.

Der in unserer Schaltung zum Einsatz kommende RTC mit der Typenbezeichnung RS5C372A [1] stammt vom Hersteller Ricoh [2]. Der im achtpoligen SSOP-Gehäuse untergebrachte Baustein bietet folgende Features:

- Uhrzeit: Sekunden/Minuten/Stunden
- Datum: Tage/Monate/Jahre (mit automatischer Berücksichtigung der Schaltjahre)
- Interrupt programmierbar für: 0,5 s/1 s/1 min/1 h und jeden Ersten des Monats (INTRA und INTRB)
- Zwei Alarmwecker für Woche/Tag und Stunde
- Schaltjahreerkennung bis 2099
- 12- und 24- Stunden-Modus
- Geringe Stromaufnahme (<2 μ A)

Die Kommunikation mit dem Baustein erfolgt über ein I²C-Interface. So ist die Schaltung einfach in Mikrocontrollersysteme integrierbar.

Die Takterzeugung geschieht mittels eines Quarzoszillators mit einer Frequenz von 32,768 Hz.

Die Uhr bietet zwei unabhängige Interrupt-Ausgänge, deren Funktion programmiert werden kann. Die Funktionen dieser Ausgänge sind in [Tabelle 1](#) dargestellt. In Verbindung mit einem Mikrocontroller kann z. B. an INTRA ein Sekundentakt generiert werden, der dann die interne Softwareuhr des Mikrocontrollers steuert. Der zweite Interrupt-Ausgang INTRB kann als Wecker genutzt werden, wobei Tag der Woche, Stunde und Minute einprogrammiert werden können. Es kann aber auch jeden Tag der Woche zu einer bestimmten Uhrzeit ein „Alarm“ ausgelöst werden.

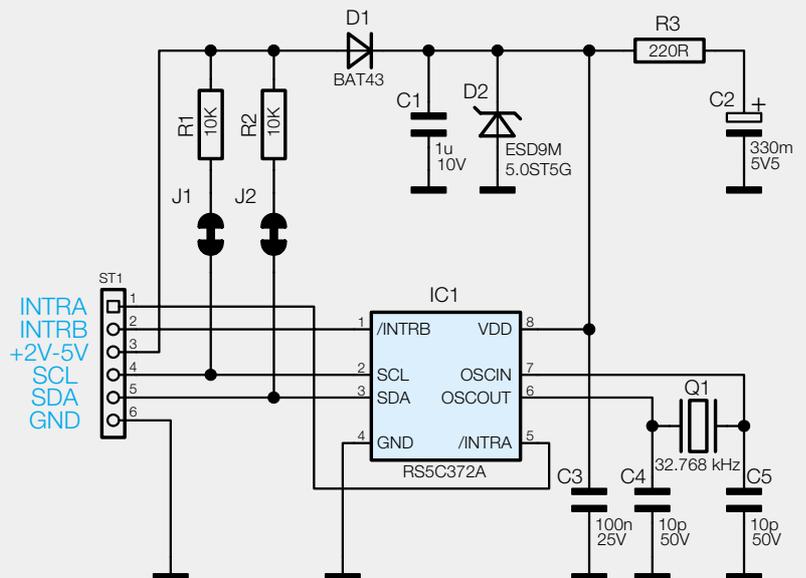
Wie man den hier vorgestellten RTC auch in eigene Mikrocontrollersysteme wie z. B. Arduino einbindet, wird hier in der Folge beschrieben.

Rund um den RTC entstand eine kleine Platine, die auch den Goldcap trägt, der die Pufferung bei ausgeschalteter Betriebsspannung übernimmt. Konstruktiv ist die Platine so ausgeführt, dass sie sowohl als kleine, heraustrennbare Tochterplatine in einem eige-

	Internal address				Contents	Data							
	A3	A2	A1	A0		D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	Second Counter	—	S ₄₀	S ₂₀	S ₁₀	S ₈	S ₄	S ₂	S ₁
1	0	0	0	1	Minute Counter	—	M ₄₀	M ₂₀	M ₁₀	M ₈	M ₄	M ₂	M ₁
2	0	0	1	0	Hour Counter	—	—	H ₂₀ P/Ä	H ₁₀	H ₈	H ₄	H ₂	H ₁
3	0	0	1	1	Day of the Week Counter	—	—	—	—	—	W ₄	W ₂	W ₁
4	0	1	0	0	Day Counter	—	—	D ₂₀	D ₁₀	D ₈	D ₄	D ₂	D ₁
5	0	1	0	1	Month Counter	—	—	—	MO ₁₀	MO ₈	MO ₄	MO ₂	MO ₁
6	0	1	1	0	Year Counter	Y ₈₀	Y ₄₀	Y ₂₀	Y ₁₀	Y ₈	Y ₄	Y ₂	Y ₁
7	0	1	1	1	Time Trimming Register	XSL	F ₆	F ₅	F ₄	F ₃	F ₂	F ₁	F ₀
8	1	0	0	0	Alarm_A (Minute Register)	—	AM ₄₀	AM ₂₀	AM ₁₀	AM ₈	AM ₄	AM ₂	AM ₁
9	1	0	0	1	Alarm_A (Hour Register)	—	—	AH ₂₀ AP/Ä	AH ₁₀	AH ₈	AH ₄	AH ₂	AH ₁
A	1	0	1	0	Alarm_A (Day of the Week Register)	—	AW ₆	AW ₅	AW ₄	AW ₃	AW ₂	AW ₁	AW ₀
B	1	0	1	1	Alarm_B (Minute Register)	—	BM ₄₀	BM ₂₀	BM ₁₀	BM ₈	BM ₄	BM ₂	BM ₁
C	1	1	0	0	Alarm_B (Hour Register)	—	—	BH ₂₀ BP/Ä	BH ₁₀	BH ₈	BH ₄	BH ₂	BH ₁
D	1	1	0	1	Alarm_B (Day of the Week Register)	—	BW ₆	BW ₅	BW ₄	BW ₃	BW ₂	BW ₁	BW ₀
E	1	1	1	0	Control Register 1	AALE	BALE	SL ₂	SL ₁	TEST	CT ₂	CT ₁	CT ₀
F	1	1	1	1	Control Register 2	—	—	12/24	ADJ XSTP	CLEN	CTFG	AAFG	BAFG

Bild 1: Die wichtigsten Register des RS5C372A

Bild 2: Das Schaltbild des I²C-RTC



nen Mikroprozessorsystem eingesetzt werden kann als auch als Arduino-Shield betreibbar ist.

Gerade letzteres System erfreut sich ja wegen der einfachen Programmierbarkeit steigender Beliebtheit, da kann man nicht genug schnell einsetzbare Peripherie haben.

Programmierung

Die Programmierung bzw. die Kommunikation des RTC-Bausteins findet über den I²C-Bus statt. Eine detaillierte Beschreibung aller Register des RS5C372A (Bild 1) und die genaue Programmierung würde den Rahmen dieses Artikels sprengen, so dass ein kostenloses Demoprogramm und eine komplette Library für das Arduino-Board unter [3]/[4] zum Download bereitsteht. Hier wird an Beispielen gezeigt, wie man z. B. die Uhrzeit setzt oder den Baustein initialisiert. Der Quellcode und die Library sind in C++ für das Arduino-Board geschrieben und sind leicht in „herkömmliches“ C+ übertragbar, z. B. für den IAR- oder einen anderen C-Compiler.

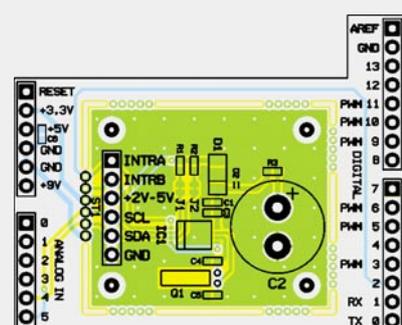
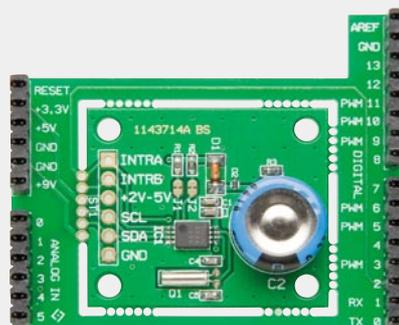
Schaltung

Das Schaltbild des I²C-RTC ist in Bild 2 dargestellt.

Der Uhren-Baustein IC 1 benötigt nur wenige externe Bauteile. Die Versorgungsspannung (2–5 V) wird über die Diode D 1 entkoppelt, so dass nach Abschalten der externen Betriebsspannung der Uhrenbaustein über den Goldcap C 2 weiter versorgt wird. C 2 ist im Prinzip ein Kondensator mit relativ hoher Kapazität (0,33 Farad). C 2 lädt sich sehr schnell über den Widerstand R 3 auf und gibt genügend Energie ab, um eine Gangreserve von ungefähr einer Woche zu erreichen.

Die Transildiode D 2 schützt die Schaltung vor Überspannungsspitzen. Die Widerstände R 1 und R 2 sind Pull-up-Widerstände für die beiden Signalleitungen SDA und SCL und werden bei Bedarf über die Jumper J 1 und J 2 deaktiviert.

Die Frequenz des internen Oszillators wird durch den externen Quarz Q 1 bestimmt und liegt bei 32,768 Hz. Die beiden Kondensatoren C 4 und C 5 dienen zur Anpassung der Lastkapazität des Quarzes. IC 1 ist normalerweise für Quarze mit einer Lastkapazität (CL) von 6 pF bis 8 pF ausgelegt. Durch die Verwendung eines Standard-Quarzes mit einer Lastkapazität von 12 pF sind zwei externe Kondensatoren (C 4, C 5) notwendig.



Die fertig bestückte Platine der I²C-Real-Time-Clock mit dem zugehörigen Bestückungsplan

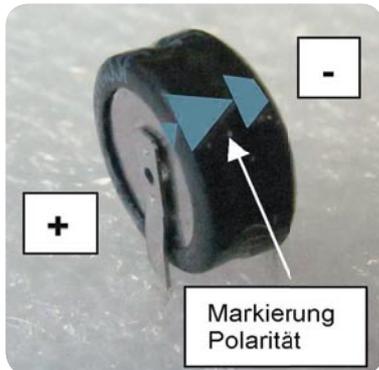


Bild 3: Kennzeichnung der Polarität beim Goldcap

Nachbau

Der Nachbau gestaltet sich durch die vorbestückten SMD-Bauteile recht einfach und ist schnell bewerkstelligt. Lediglich der Goldcap C 2 und die Stift- bzw. Buchsenleisten und der Quarz müssen bestückt und verlötet werden. Bei Einsetzen des Goldcaps ist unbedingt auf die richtige Polung zu achten. Es können sowohl stehende als auch liegende Varianten eingesetzt werden. Bei der liegenden Version ist die Kennzeichnung der



Bild 5: Stapelbare Buchsenleisten

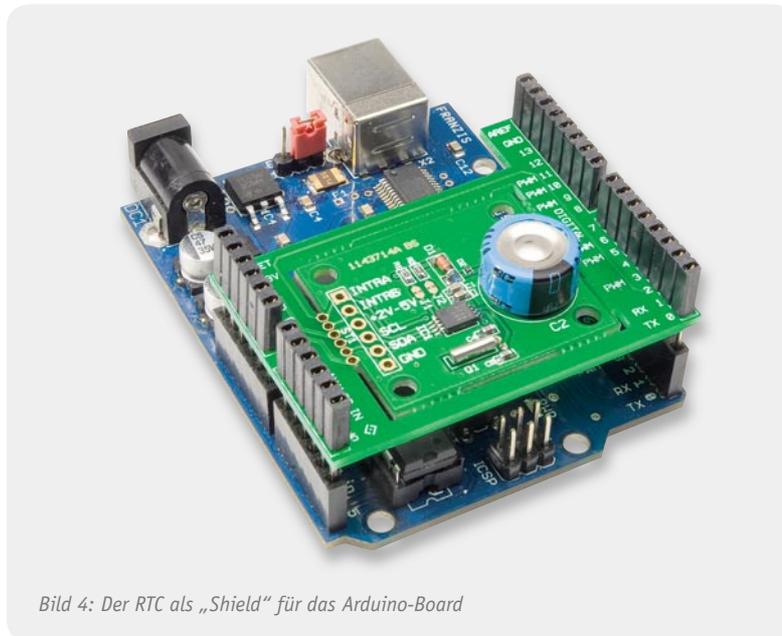


Bild 4: Der RTC als „Shield“ für das Arduino-Board

Polarität durch aufgedruckte Pfeile erkennbar (Bild 3), die in Richtung Minuspol zeigen.

Der Quarz wird auf der Platinenoberseite eingesetzt (siehe Platinenfoto). Um die elektrischen Eigenschaften nicht negativ zu beeinträchtigen, sollte die Lötzeit so kurz wie möglich gehalten werden (<3 Sekunden).

Inbetriebnahme

Die Platine ist sowohl für den Einsatz mit einem Arduino-Board (Bild 4) als auch für den universellen Einsatz ausgelegt. Soll die Platine als „Shield“ für das Arduino-Board eingesetzt werden, sind spezielle Buchsenleisten einzulöten, die gleichzeitig als Buchsen- und Stiftleisten dienen. Diese stapelbaren Buchsenleisten (Bild 5) bieten den Vorteil, dass keine separaten Bohrungen für die Buchsen und Stiftleisten notwendig sind. Mehrere „Shields“ (Zusatzmodule) lassen sich so direkt übereinander stapeln. Diese Buchsenleisten werden von oben in die Platine eingesetzt und auf der Platinenunterseite verlötet.

Möchte man die Platine für eine andere Mikrocontrollerschaltung verwenden, kann der mittlere Teil, auf dem sich die Elektronik befindet, herausgetrennt werden. Die Platine besitzt hierzu eine Perforation, so dass der mittlere Teil einfach herausgebrochen werden kann. Die Platine (Bild 6) kann dann mit einer „normalen“ Stiftleiste bestückt werden. Hierbei ist darauf zu achten, dass die Verbindungsleitung nicht länger als 3 m sein darf.

Der I²C-Bus benötigt für eine korrekte Funktion Pull-up-Widerstände an beiden Busleitungen SDA und SCL. Wird der RTC-Baustein in einem System eingesetzt, in dem schon relativ niederohmige Pull-up-Widerstände vorhanden sind, müssen eventuell die auf der Platine befindlichen Pull-up-Widerstände (R 1 und R 2) deaktiviert werden. Dies geschieht durch Auftrennen der beiden Jumper J 1 und J 2. Hierzu wird die Verbindung

Widerstände:

220 Ω/SMD/0603	R3
10 kΩ/SMD/0603	R1, R2

Kondensatoren:

10 pF/SMD/0603	C4, C5
100 nF/SMD/0603	C3
1 µF/SMD/0603	C1
Goldcap, 0,33 F, 5,5 V	C2

Halbleiter:

RS5C372A/SMD/Ricoh	IC1
BAT43/SMD	D1
ESD9M5.0ST5G/SMD	D2

Sonstiges:

Quarz, 32,768 kHz, 12,5 pF	Q1
1 Ferrit-Ringkern, 25 x 12 mm	
1 Stiftleiste, 6-polig, stehend	ST1

auf der Platine mit einem scharfen Messer getrennt. **Hinweis:** Im Regelfall, d. h. auch beim Einsatz der Schaltung mit einem Arduino-Board, brauchen diese Brücken (J 1 und J 2) nicht getrennt zu werden.

Beim Einsatz mit dem Arduino ist im Hinblick auf Störaussendung noch etwas zu beachten. Da die kommerziell erhältlichen Arduino-Boards beim Anschluss von externen Komponenten dazu neigen, Störsignale über die Zuleitungen auszusenden, muss aus diesem Grund ein Ferritring in die Zuleitung der Versorgungsspannung eingebracht werden. Hierzu wird die Zuleitung viermal durch den Ferritring geführt (Bild 7). **ELV**



Bild 6: Herausgetrennte Platine für den universellen Einsatz

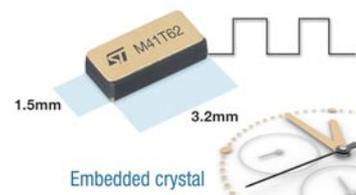
i Weitere Infos:

- [1] Download Datenblatt RS5C372: www.ricoh.com/LSI/product_rtc/2wire/5c372/5c372a-e.pdf
- [2] www.ricoh.com/LSI/product_rtc
- [3] Download „Arduino Demo und Library “: www.elv.de: Webcode #1202
- [4] I²C-RTC im ELV-Web-Shop: www.elv.de: Webcode #1203

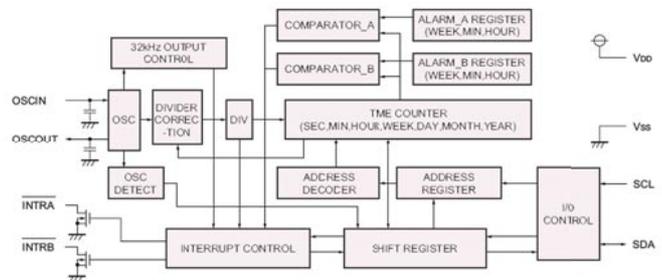


Bild 7: So wird der Ferritring in die Versorgungsspannungszuleitung des Arduino-Boards eingefügt.

Real-time clock



Noch brandneu: extrem kleiner RTC-Baustein M41T62 von STMicroelectronics mit integriertem Quarz, gerade einmal 350 nA Stromaufnahme und zahlreichen programmierbaren Zusatzfunktionen wie Watchdog oder Zeitstempel bei Power-down-Prozessen
Quelle: STMicroelectronics



Das Blockschaltbild des in unserem Projekt verwendeten RS5C372A von Ricoh. Man erkennt deutlich die einzelnen Funktionsgruppen Oszillator/Teiler, Korrekturreinrichtung, Zähler, Alarmerzeugung, Decodierung, Register, Interrupt-Ausgabe und Buscontroller. Auch dieser Baustein benötigt typisch nur 500 nA (3 V).
Quelle: RICOH

Real-Time-Clock (RTC)

Eine RTC (Real-Time-Clock, Echtzeituhr) ist ein auf die Messung der physikalischen Zeit spezialisierter Baustein, der intern erzeugte Zeitdaten über einen Datenbus ausgibt. Als Taktquelle dient vorwiegend ein sogenannter Uhrenquarz (32,768 kHz), der entweder extern angeschlossen wird oder bereits in den RTC-Baustein integriert ist. Im internen Speicher werden Sekunden, Minuten, Stunden, Tage, Monate und Jahre gespeichert. Je nach Hersteller werden dabei auch die Schaltjahre berücksichtigt. Die Genauig-

keit ist vom verwendeten Quarz abhängig. Da ein Quarz gewisse Fertigungstoleranzen aufweist, muss die Uhr abgeglichen oder synchronisiert werden. Anwendung finden RTCs in PC- oder Mikrocontrollersystemen. Ein wesentlicher Vorteil ist, dass ein solcher RTC-Baustein in der Regel autark läuft, d. h. auch ohne extern zugeführte Betriebsspannung wird die Zeitmessung für einen gewissen Zeitraum fortgesetzt (Gangreserve). Dies wird durch eine Batterie (3 V), einen Akku oder sonstigen Stromspeicher erreicht, der bei ausgeschalteter Betriebsspannung den RTC-Baustein weiter versorgt.