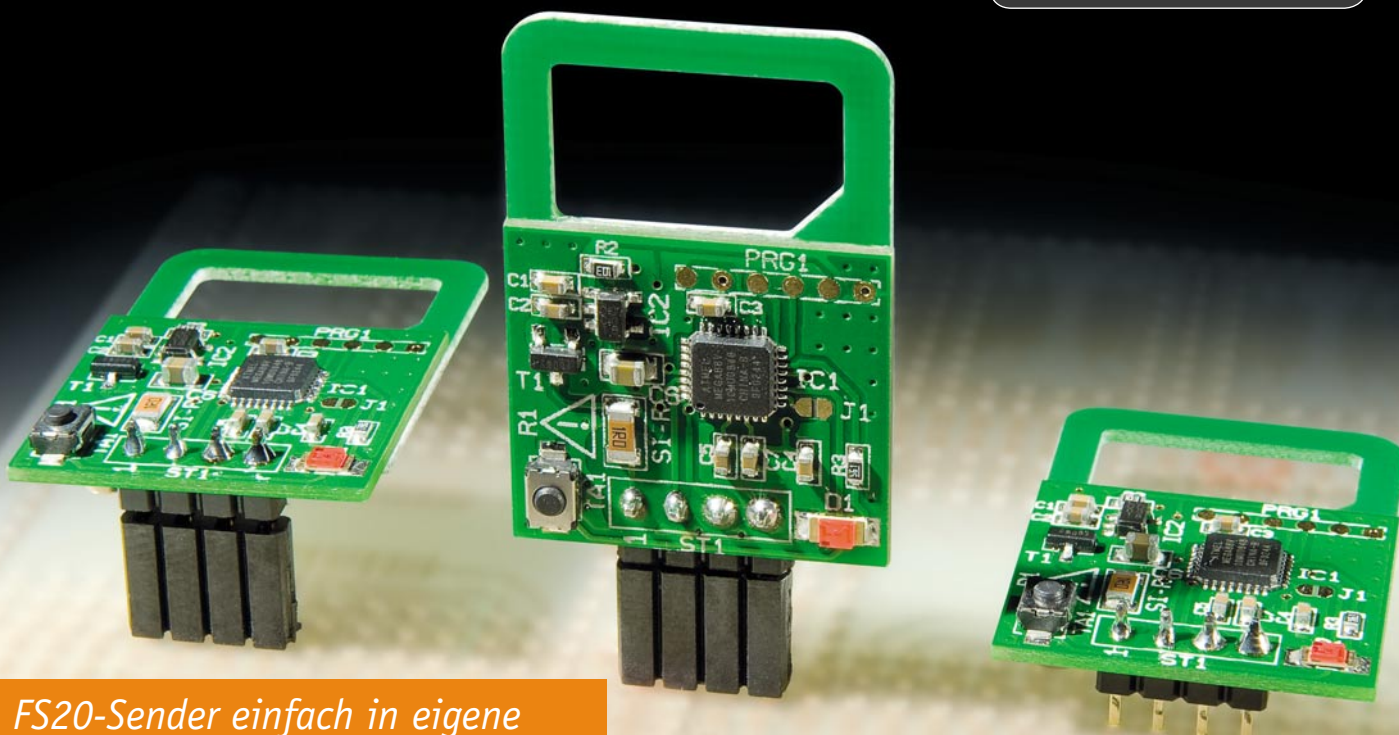




FS20-Sender einfach in eigene Mikrocontroller-Projekte einbinden

UART-FS20-Übersetzer – Mini-UART-FS20-Sender FS20 US

Die FS20-Welt hält eine riesige Anzahl an praktischen Empfängern und Aktoren bereit. Deren Nutzung für eigene Mikrocontroller-Projekte ist in doppelter Hinsicht eine interessante Sache – zum einen spart man sich die Entwicklung eigener Funkprotokolle, Empfangs-, Sendemodule und der Aktoren, und zum anderen erhält man mit dem FS20-System ein zugelassenes System, das alle Richtlinien erfüllt. Genau dies macht der Mini-UART-FS20-Sender FS20 US möglich: Mit einfachen Befehlen lässt sich von einem Mikrocontroller aus über dessen serielle Schnittstelle jeder FS20-Empfänger ansteuern.

FS20 einfach „anbinden“

Elektronikprojekte mit kleinen Mikroprozessoren gewinnen immer breitere Kreise von Anwendern. Nicht zuletzt die Verfügbarkeit spezialisierter und einfach erlernbarer Programmiersprachen wie Bascom oder die zum Arduino-Projekt gehörende Programmiersoftware machen es auch Nicht-Programmier-Profis möglich, eigene Softwareprojekte, verbunden mit typisch steiler Lernkurve, zu realisieren. Da die zugehörige Hardware

recht einfach ist und schnell „quick and dirty“ auf einer Lochrasterplatine ihren Platz findet, liegt es zur schnellen Fertigstellung eines Projektes nahe, auf bereits vorhandene oder schnell fertig einsetzbare Aktoren zurückzugreifen. Dies gilt insbesondere für Funkanwendungen und bei Geräten, die mit gefährlicher Netzspannung arbeiten.

Das FS20-System bietet hier preiswert eine geradezu unendliche Menge an Funk-Aktoren, vom RDS-Radio für die Unterputzmontage bis hin zum Power-Aktor, der drei 16-A-Schaltkanäle und einen Dimmkanal bedienen kann.

Für den PC-Bereich gibt es bereits einen Interface-Sender, den FS20 PCS. Er setzt auf eine Open-Source-PC-Programmiersoftware (EventGhost) auf und wird über einen USB-Port angesteuert.

Auch wenn das hier vorgestellte Mini-UART-Sendemodul mithilfe eines USB-UART-Wandlers wie dem UM2102 oder dem in diesem Journal vorgestellten UM-FT232H über einen PC angesteuert werden kann, ist es in erster Linie für die direkte Anbindung an den UART-Port eines Mikrocontrollers vorgesehen und bildet damit eine ideale Einbindung des FS20-Empfän-

Gerätetyp:	FS20-Sendemodul mit UART-Schnittstelle
Kompatibilität:	zu allen FS20-Empfängern
Kanäle:	beliebig
Frequenz:	868,35 MHz
Reichweite:	bis zu 100 m (Freifeld)
Schnittstelle:	UART (RxD, TxD), nicht-invertierter TTL-Pegel (abhängig von UB)
Versorgung:	3–5 V _{DC} , max. 30 mA
Abmessungen (B x H x T):	22 x 32 x 9 mm
Montage:	waagrecht oder senkrecht, gesteckt oder aufgelötet

gersystems in eigene μ C-Projekte. Damit verfügt man über einen kompletten und individuell per einfachem Steuerbefehl programmierbaren FS20-Sender, integriert ins selbst entwickelte Gerät.

Das kleine Elektronikmodul ist mechanisch so ausgeführt, dass es löt- oder steckbar ist (2,54-mm-Rastermaß) und in verschiedenen Einbaulagen verbaut werden kann. Somit findet sich quasi in jedem Gerät ein Platz mit geeigneter Antennenausrichtung für dieses kompakte und flache Modul.

Die universell ausgeführte UART-Schnittstelle des Moduls lässt sich mit (fast) jedem Mikrocontroller ansteuern, zudem kann die Ansteuerung flexibel erfolgen, ohne die Notwendigkeit einer Konfiguration des Moduls (bei Bedarf ist aber die Baudrate umstellbar).

Das Zurückgreifen auf ein solches Sendemodul erspart auch eine Menge eigener Entwicklungsarbeit, das Modul verfügt schließlich über eine Zulassung und garantiert somit für die Einhaltung aller geltenden Richtlinien wie z. B. Sendeleistung, Störimpfindlichkeit, Duty-Cycle-Management (siehe Abschnitt „Kommunikation und Befehle“).

Zur Kontrolle verfügt das FS20-US-Modul über eine rote LED zur Aussendungskontrolle. Dazu kommt ein Taster zur Testaussendung bzw. zum Zurücksetzen in den Auslieferungszustand. Zur Testaussendung wird nach dem ersten Einschalten durch Drücken des Tasters der Testbefehl „Wechsel zwischen Ein/Aus“ (Befehl: 0x12, Kanal: 1111, Hauscode: 11111111) ausgesendet. Im späteren Betrieb erfolgt bei Drücken des Tasters die Wiederholung des zuletzt gesendeten FS20-Befehls.

Die Möglichkeit des Zurücksetzens in den Auslieferungszustand (Werkseinstellung) ist hilfreich, wenn die

einmal eingestellte Baudrate nicht mehr bekannt ist.

Inbetriebnahme/Bedienung

Wie einfach es ist, das FS20-US-Modul mit eigener Hardware zu verbinden, zeigt **Bild 1** anhand der Anbindung an das aktuelle Arduino-Uno-Board. Hier bietet sich auch eine Montage als einfaches Shield (steckbarer Platinaufsatz) direkt auf dem Arduino-Board an.

Da das Modul so ausgeführt ist, dass es auf die verschiedensten Weisen gesteckt oder aufgelötet werden kann, ist die Montage sehr flexibel ausführbar. **Bild 2** zeigt, wie die Bestückung mit der geraden und der abgewinkelten Stiftleiste (liegen beide dem Bausatz bei) erfolgen kann, die sowohl auf der Vorder- als auch auf der Rückseite der Platine einsetzbar sind. Dabei kann man das Modul sowohl aufstecken als auch einlöten.

Bei der Betriebsspannung ist auf die zulässige Versorgungsspannung von 3 bis 5 V_{DC} (+/-10 %) und auf den richtigen Anschluss von RxD (Empfangen) und TxD (Senden) zu achten. Da beide Seiten der Hardware, also das FS20-UART-Sendemodul und die eigene Mikrocontrollerschaltung, senden und empfangen, ist natürlich die Sendeleitung (TxD) des einen „Gerätes“ an die Empfangsleitung (RxD) des anderen „Gerätes“ anzuschließen!

Soll der FS20 US an eine RS232-Schnittstelle, z. B. an einen PC angeschlossen werden, sind dafür zusätzlich ein Schnittstellen-Wandler wie der TTL- nach RS-232-Umsetzer (Artikel-Nr. JI-03 84 39) von ELV plus eine 5-V-Spannungsversorgung notwendig. Möchte man den FS20 US über USB an einen PC anschließen, so gelingt das mithilfe des UM2102 (Artikel-Nr. JI-09 18 59) oder mit dem in dieser Ausgabe vorge-

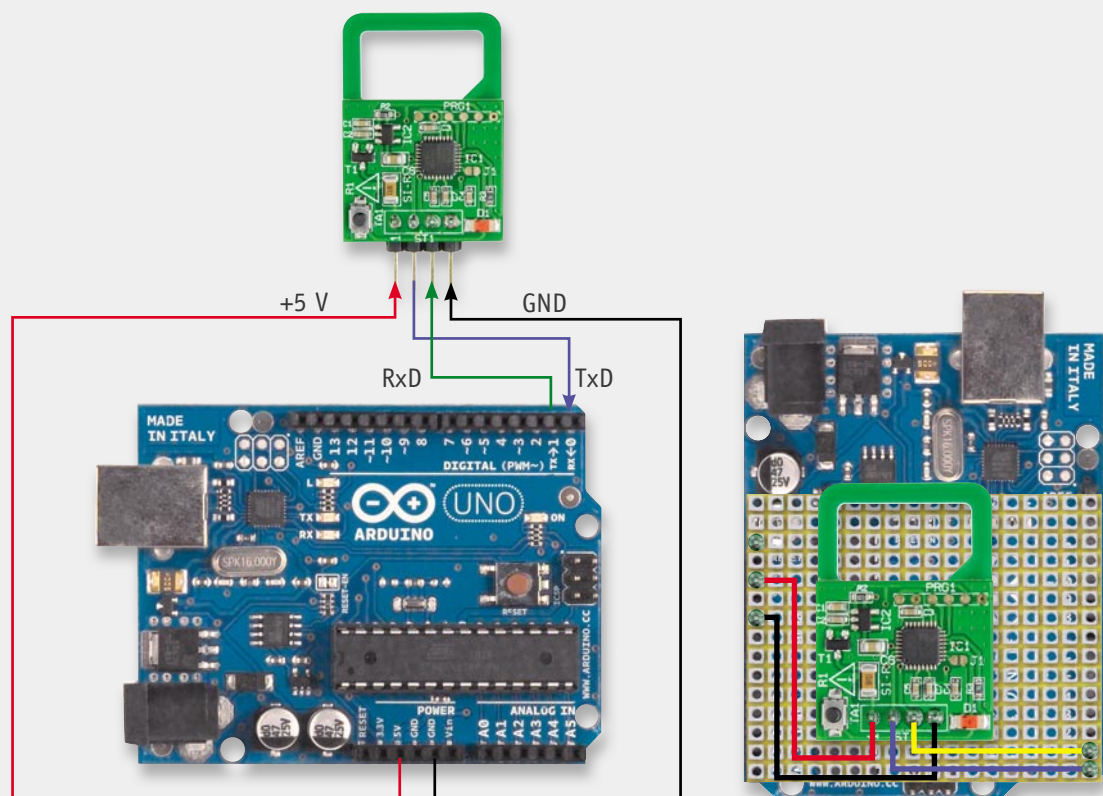
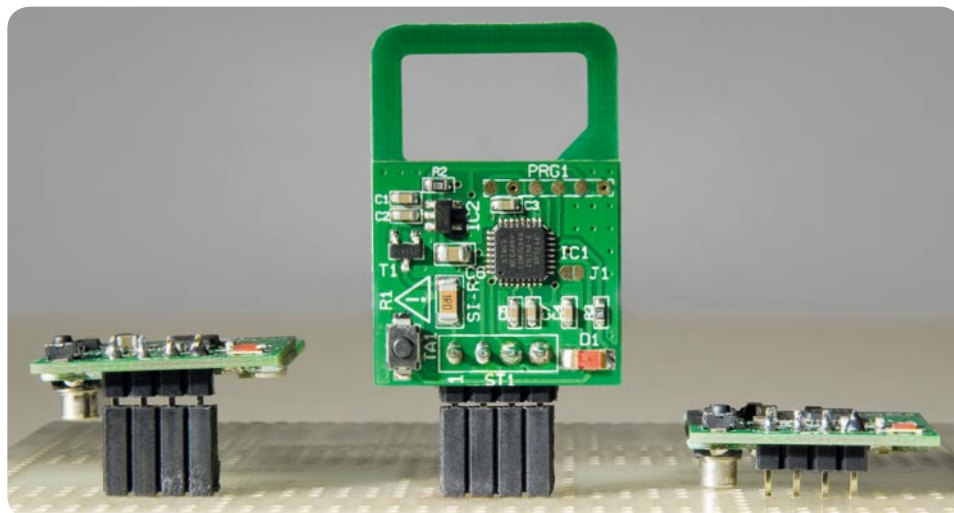


Bild 1: So wird der FS20 US an ein Mikrocontrollersystem, hier ein Arduino Uno, angeschlossen. Im Montagebeispiel rechts ist der Einsatz als Arduino-Shield zu sehen.

Bild 2: Das Modul ist entweder mit der geraden oder der abgewinkelten Stiftleiste flexibel im Host-System montierbar. Hier werden drei Beispiele gezeigt, wobei das Modul sowohl gesteckt (links und Mitte) als auch eingelötet (rechts) werden kann.
(Module vergrößert dargestellt. Originalgröße ohne Stiftleiste (H x B): ca. 32 x 22 mm)



stellten UM-FT2232H ohne eine zusätzliche Stromversorgung (USB-powered).

Sobald das FS20-UART-Sendemodul richtig angeschlossen und mit Spannung versorgt wird, kann ein erster Test erfolgen, indem durch kurzes Betätigen des Tasters TA 1 ein erster FS20-Befehl ausgesendet wird. Dies ist nach Inbetriebnahme immer der Befehl „Wechsel zwischen Ein/Aus“ (Befehl: 0x12) mit dem Hauscode 11111111 auf dem Kanal 1111. Versetzt man z. B. einen FS20-Empfänger (z. B. Funk-Schaltsteckdose FS20 ST) in den Anlernmodus und lernt den Testbefehl des FS20 US an, indem Taste TA 1 einmal kurz gedrückt wird, lässt sich der Empfänger anschließend bereits testweise schalten. Auch über das Diagnosetool FS20 DT ist die Aussendung gut kontrollierbar, hier werden zudem ja Hauscode und Adresse direkt angezeigt.

Sobald das FS20 US über die UART-Schnittstelle die Anweisung erhält, einen anderen FS20-Befehl auszusenden, ersetzt dieser den Testbefehl und wird bei Betätigen der Taste TA 1 wiederholt ausgesendet.

Kommunikation und Befehle

Die serielle Kommunikation mit dem FS20 US erfolgt mit 9600 Baud, 1 Stoppbit und ohne Paritätsbit (9600, 8, N, 1). Alle anwendbaren Befehle sind in [Tabelle 1](#) aufgelistet. Vorangestellt sind der Befehls-ID ein Startzeichen (STX = 0x02) und die Befehlslänge, wobei weder das Startzeichen noch das Byte der Befehlslänge mitzählen. Je nach Befehl folgen der Befehls-ID unterschiedlich viele Parameter-Byte, deren Inhalte in den [Tabellen 2 bis 5](#) aufgelistet sind. Ein Stoppzeichen am Ende einer Übertragung gibt es nicht, da innerhalb der Übertragung alle Werte für ein Parameter-Byte (0x00 bis 0xFF) zulässig sind und somit fälschlicherweise als Stoppzeichen interpretiert werden würden. Die Befehlsstelegramme werden stattdessen anhand des Startzeichens, der Befehlslänge und bestimmter Timeouts sicher erkannt.

Zwischen zwei Befehlsstelegrammen ist eine Sendepause von mindestens 10 ms einzuhalten, wodurch das Startzeichen eindeutig als ein solches erkannt werden kann. Nach der Übertragung des Startzeichens muss der Rest der Befehlsanweisung innerhalb von 10 ms übertragen sein, da sonst ein Timeout eintritt.

Befehlsliste des FS20 US

Tabelle 1

Startzeichen	Befehlslänge	Befehls-ID	Bedeutung / Funktion	Parameter Byte
0x02	0x01	0xF0	Status, Baudraten-Einstellung und Firmware-Version ausgeben	–
0x02	0x06	0xF1	FS20-Befehl einmal senden	HC1, HC2, ADR, BEF, ERW
0x02	0x07	0xF2	FS20-Befehl mehrfach wiederholt senden	HC1, HC2, ADR, BEF, ERW, Anzahl
0x02	0x01	0xF3	Letzten FS20-Befehl noch einmal senden	–
0x02	0x01	0xF4	Senden abbrechen	–
0x02	0x02	0xF5	Baudrate umstellen	Baudrate
0x02	0x01	0xF6	Reset ausführen	–
0x02	0x01	0xF7	Werkseinstellung wiederherstellen	–

Mögliche Hauscodes (HC1, HC2)

Tabelle 2

HC1	HC2	Hauscode
0x00	0x00	1111 1111
0x00	0x01	1111 1112
0x00	0x02	1111 1113
0x00	0x03	1111 1114
0x00	0x04	1111 1121
0x00	0x05	1111 1122
0x00	0x06	1111 1123
0x00	0x07	1111 1124
usw.	usw.	usw.
0xFF	0xF8	4444 4431
0xFF	0xF9	4444 4432
0xFF	0xFA	4444 4433
0xFF	0xFB	4444 4434
0xFF	0xFC	4444 4441
0xFF	0xFD	4444 4442
0xFF	0xFE	4444 4443
0xFF	0xFF	4444 4444

Mögliche Kanaladressen (ADR)

Tabelle 3

ADR	FS20-Adresse
0x00	1111
0x01	1112
0x02	1113
0x03	1114
0x04	1121
0x05	1122
0x06	1123
0x07	1124
usw.	usw.
0xF8	4431
0xF9	4432
0xFA	4433
0xFB	4434
0xFC	4441
0xFD	4442
0xFE	4443
0xFF	4444

FS20-Befehle (BEF) mit und ohne Nutzung einer Timerzeit (ERW)

BEF	ERW	FS20-Befehl
0x00	egal	Aus (auf 0 % dimmen)
0x01	egal	Einschalten auf Helligkeitsstufe 1 (6,25 %) (Minimum)
0x02	egal	Einschalten auf Helligkeitsstufe 2 (12,50 %)
0x03	egal	Einschalten auf Helligkeitsstufe 3 (18,75 %)
0x04	egal	Einschalten auf Helligkeitsstufe 4 (25,00 %)
0x05	egal	Einschalten auf Helligkeitsstufe 5 (31,25 %)
0x06	egal	Einschalten auf Helligkeitsstufe 6 (37,50 %)
0x07	egal	Einschalten auf Helligkeitsstufe 7 (43,75 %)
0x08	egal	Einschalten auf Helligkeitsstufe 8 (50,00 %)
0x09	egal	Einschalten auf Helligkeitsstufe 9 (56,25 %)
0x0A	egal	Einschalten auf Helligkeitsstufe 10 (62,50 %)
0x0B	egal	Einschalten auf Helligkeitsstufe 11 (68,75 %)
0x0C	egal	Einschalten auf Helligkeitsstufe 12 (75,00 %)
0x0D	egal	Einschalten auf Helligkeitsstufe 13 (81,25 %)
0x0E	egal	Einschalten auf Helligkeitsstufe 14 (87,50 %)
0x0F	egal	Einschalten auf Helligkeitsstufe 15 (93,75 %)
0x10	egal	Einschalten auf Helligkeitsstufe 16 (100,00 %) (Maximum)
0x11	egal	Einschalten auf letzten Helligkeitswert
0x12	egal	Wechsel zwischen „Aus“ und „An, alter Wert“ (Toggle)
0x13	egal	Eine Helligkeitsstufe heller dimmen
0x14	egal	Eine Helligkeitsstufe dunkler dimmen
0x15	egal	Heraufdimmen bis Maximum, Pause, Herabdimmen bis Minimum, Pause usw.
0x16	egal	Programmierung der internen Timerzeit starten bzw. stoppen
0x17	egal	(Kann bei einigen Empfängern zum Anlernen von Hauscode und Adresse genutzt werden)
0x18	egal	Aus für (interne) Timerzeit, danach alte Helligkeit
0x19	egal	An (100 %) für (interne) Timerzeit, danach Aus
0x1A	egal	An (alte Helligkeit) für (interne) Timerzeit, danach Aus
0x1B	egal	Auf Auslieferungszustand zurücksetzen (wird nicht von allen FS20-Empfängern unterstützt)
0x1C	egal	Befehl ungültig
0x1D	egal	Befehl ungültig
0x1E	egal	An (100 %) für (interne) Timerzeit, danach alter Zustand
0x1F	egal	An (alte Helligkeit) für (interne) Timerzeit, danach alter Zustand
0x20	0x00 – 0xFF	Dimme in Timerzeit bis auf 0 % (Aus)
0x21	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 1 (6,25 %)
0x22	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 2 (12,50 %)
0x23	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 3 (18,75 %)
0x24	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 4 (25,00 %)
0x25	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 5 (31,25 %)
0x26	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 6 (37,50 %)
0x27	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 7 (43,75 %)
0x28	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 8 (50,00 %)
0x29	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 9 (56,25 %)
0x2A	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 10 (62,50 %)
0x2B	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 11 (68,75 %)
0x2C	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 12 (75,00 %)
0x2D	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 13 (81,25 %)
0x2E	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 14 (87,50 %)
0x2F	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 15 (93,75 %)
0x30	0x00 – 0xFF	Dimme in Timerzeit bis auf Helligkeitsstufe 16 (100,00 %)
0x31	0x00 – 0xFF	Dimme in Timerzeit bis auf letzten Helligkeitswert herab/herauf
0x32	0x00 – 0xFF	Dimme auf „An, alter Wert“ herauf und dimme nach Timerzeit auf „Aus“ herab (im Wechsel)
0x33	0x00 – 0xFF	Dimme sofort eine Helligkeitsstufe herauf (bei langem Tastendruck mehrere Helligkeitsstufen solange Taste gedrückt wird) und schalte nach Timerzeit aus
0x34	0x00 – 0xFF	Dimme sofort eine Helligkeitsstufe herab (bei langem Tastendruck mehrere Helligkeitsstufen solange Taste gedrückt wird) und schalte nach Timerzeit aus
0x35	0x00 – 0xFF	Dimme im Wechsel sofort eine Helligkeitsstufe herauf bzw. herab (bei langem Tastendruck mehrere Helligkeitsstufen herauf bis Maximum und nach kurzer Pause wieder herab bis Minimum usw., solange Taste gedrückt wird) und schalte nach Timerzeit aus (im Wechsel)
0x36	0x00 – 0xFF	Programmierung der internen Timerzeit
0x37	egal	Befehl ungültig
0x38	0x00 – 0xFF	Aus für Timerzeit, danach auf alte Helligkeit
0x39	0x00 – 0xFF	An (100 %) für Timerzeit, danach aus
0x3A	0x00 – 0xFF	An (alte Helligkeit) für Timerzeit, danach aus
0x3B	egal	Befehl ungültig
0x3C	0x00 – 0xFF	Programmierung der internen Rampenzeit für Heraufdimmen
0x3D	0x00 – 0xFF	Programmierung der internen Rampenzeit für Herabdimmen
0x3E	0x00 – 0xFF	An (100 %) für Timerzeit, danach auf vorherigen Zustand
0x3F	0x00 – 0xFF	An (alte Helligkeit) für Timerzeit, danach auf vorherigen Zustand

und die zuvor empfangenen Bytes verworfen werden. Diese Angaben gelten unabhängig von der eingestellten Baudrate.

Der Vorteil dieses sehr einfachen Übertragungsprotokolls ist zum einen, dass es Programmierneulungen entgegenkommt, und zum anderen, dass dadurch der FS20-UART-Sender über einen Pegelwandler auch direkt vom PC mithilfe eines Terminalprogramms ansteuerbar ist.

Das Befehlstelegramm für das Einschalten (mit dem Befehl 0x11 = „Einschalten auf letzte Helligkeit“) eines FS20-Empfängers, der auf den Hauscode 4442 1121 und den Kanal 1113 angelernt wurde, sieht dann z. B. folgendermaßen aus:

Start	Länge	Befehls-ID	Parameter-Byte
0x02	0x06	0xF1	0xFD, 0x04, 0x02, 0x11, 0xFF

Als Antwort gibt der FS20-UART-Sender Folgendes zurück, wobei seine Übertragungsgeschwindigkeit 9600 Baud beträgt, er über die Firmware-Version 1.0 verfügt und nun gerade mit dem Aussenden des FS20-Befehls beschäftigt ist:

Start	Länge	Status-ID	Baud	Version
0x02	0x03	0x01	0x01	0x0A

Die möglichen Antworten des FS20 US und deren Bedeutung sind in [Tabelle 6](#) zusammengefasst. Mit der Status-ID 0x05 antwortet der FS20 US nur dann, wenn die zulässige Duty-Cycle-Zeit überschritten wurde. Darunter versteht man die von der Bundesnetzagentur vorgegebene maximale Aussendezeit pro Stunde. Laut geltender Duty-Cycle-Regelung darf der Sender innerhalb einer Stunde das 868,35-MHz-Band

Mögliche Timerzeiten im Erweiterungsbyte (ERW)

ERW	Timerzeit	ERW	Timerzeit	ERW	Timerzeit	ERW	Timerzeit
0x00	sofort/dauerhaft	0x2E	14 s	0x6A	2 min, 40 s	0x9F	32 min
0x01	0,25 s	0x2F	15 s	0x6B	2 min, 56 s	...	nicht nutzen
0x02	0,50 s	...	nicht nutzen	0x6C	3 min, 12 s	0xA8	34 min, 8 s
0x03	0,75 s	0x38	16 s	0x6D	3 min, 28 s	0xA9	38 min, 24 s
0x04	1,00 s	0x39	18 s	0x6E	3 min, 44 s	0xAA	42 min, 40 s
0x05	1,25 s	0x3A	20 s	0x6F	4 min	0xAB	46 min, 56 s
0x06	1,50 s	0x3B	22 s	...	nicht nutzen	0xAC	51 min, 12 s
0x07	1,75 s	0x3C	24 s	0x78	4 min, 16 s	0xAD	55 min, 28 s
0x08	2,00 s	0x3D	26 s	0x79	4 min, 48 s	0xAE	59 min, 44 s
0x09	2,25 s	0x3E	28 s	0x7A	5 min, 20 s	0xAF	1 h, 4 min
0x0A	2,50 s	0x3F	30 s	0x7B	5 min, 52 s	...	nicht nutzen
0x0B	2,75 s	...	nicht nutzen	0x7C	6 min, 24 s	0xB8	1 h, 8 min, 16 s
0x0C	3,00 s	0x48	32 s	0x7D	6 min, 56 s	0xB9	1 h, 16 min, 48 s
0x0D	3,25 s	0x49	36 s	0x7E	7 min, 28 s	0xBA	1 h, 25 min, 20 s
0x0E	3,50 s	0x4A	40 s	0x7F	8 min	0xBB	1 h, 33 min, 52 s
0x0F	3,75 s	0x4B	44 s	...	nicht nutzen	0xBC	1 h, 42 min, 24 s
...	nicht nutzen	0x4C	48 s	0x88	8 min, 32 s	0xBD	1 h, 50 min, 56 s
0x18	4,0 s	0x4D	52 s	0x89	9 min, 36 s	0xBE	1 h, 59 min, 28 s
0x19	4,5 s	0x4E	56 s	0x8A	10 min, 40 s	0xBF	2 h, 8 min
0x1A	5,0 s	0x4F	60 s	0x8B	11 min, 44 s	...	nicht nutzen
0x1B	5,5 s	...	nicht nutzen	0x8C	12 min, 48 s	0xC8	2 h, 16 min, 32 s
0x1C	6,0 s	0x58	1 min, 4 s	0x8D	13 min, 52 s	0xC9	2 h, 33 min, 36 s
0x1D	6,5 s	0x59	1 min, 12 s	0x8E	14 min, 56 s	0xCA	2 h, 50 min, 40 s
0x1E	7,0 s	0x5A	1 min, 20 s	0x8F	16 min	0xCB	3 h, 7 min, 44 s
0x1F	7,5 s	0x5B	1 min, 28 s	...	nicht nutzen	0xCC	3 h, 24 min, 48 s
...	nicht nutzen	0x5C	1 min, 36 s	0x98	17 min, 4 s	0xCD	3 h, 41 min, 52 s
0x28	8 s	0x5D	1 min, 44 s	0x99	19 min, 12 s	0xCE	3 h, 58 min, 56 s
0x29	9 s	0x5E	1 min, 52 s	0x9A	21 min, 20 s	0xCF	4 h, 16 min
0x2A	10 s	0x5F	2 min	0x9B	23 min, 28 s	...	nicht nutzen
0x2B	11 s	...	nicht nutzen	0x9C	25 min, 36 s		
0x2C	12 s	0x68	2 min, 8 s	0x9D	27 min, 44 s		
0x2D	13 s	0x69	2 min, 24 s	0x9E	29 min, 52 s		

Mögliche Antworten des FS20 US auf Befehle

Startzeichen	Befehlslänge	Status-ID	Baudraten-Einstellung	Firmware-Version
0x02	0x03	1 Byte Mögl. Werte: 0x00 = Ruhemodus 0x01 = Senden aktiv 0x02 = Unbekannter Befehl 0x03 = Falsche Befehlslänge 0x04 = Unbekannter Parameter 0x05 = Duty-Cycle-Sperre aktiv	1 Byte Mögl. Werte: 0x00 = 4800 0x01 = 9600 0x02 = 14.400 0x03 = 19.200 0x04 = 38.400	1 Byte Beispiele: 0x09 = Version 0.9 0x0A = Version 1.0 0x0B = Version 1.1 0x10 = Version 1.6 usw.

nur zu 1 % belegen, damit andere Teilnehmer dieses Frequenzbandes nicht unnötig gestört werden. Diese gesetzliche Vorgabe führt dazu, dass der FS20 US die genaue Dauer aller Sendezeiten protokolliert und bei Überschreitung ein weiteres Senden so lange sperrt, bis genug Zeit vergangen ist, dass das Senden wieder zulässig ist. Bei aktiver Sperre setzt der FS20 US keine Funkbefehle ab und antwortet mit der Status-ID 0x05. Zusätzlich zeigt er die Sperre durch mehrmaliges schnelles Aufblitzen der LED an.

Bei Bedarf kann man die Übertragungsgeschwindigkeit des FS20-UART-Senders verändern. Da die Werkseinstellung 9600 Baud ist, muss der Konfigurationsbefehl mit dieser Geschwindigkeit an das Modul gesendet werden. Möchte man statt mit 9600 Baud in Zukunft die Datenübertragung mit 38.400 Baud vornehmen, ist, wie in [Tabelle 1](#) dargestellt, das folgende Befehlsstelegramm zu senden:

Start	Länge	Befehls-ID	Parameter-Byte
0x02	0x02	0xF5	0x04

Die neu eingestellte Geschwindigkeit wird aber erst nach einem Reset/Neustart verwendet. Daher sollte man vor dem Senden des Reset-Befehls (0x02, 0x01, 0xF6) erst die neue Baudrate überprüfen, die z. B. in der Antwort auf den Konfigurationsbefehl mit enthalten ist. Erst danach sollte man den Reset ausführen und kann dann mit der neuen Einstellung weiterarbeiten.

Werkseinstellung wiederherstellen

Sollte man einmal vergessen, welche Einstellung im FS20 US aktiv ist oder klappt die Datenübertragung mit der neu gewählten Geschwindigkeit aus irgendeinem Grunde nicht fehlerfrei, ist die Werkseinstellung auch mit der Taste TA 1 direkt am Modul wiederherstellbar. Dafür ist das Modul erst spannungsfrei zu schalten. Dann sollte die Taste TA 1 gedrückt, die Spannungsversorgung wieder eingeschaltet und die Taste gleichzeitig noch drei Sekunden lang weiter ge-

drückt gehalten werden, bis die LED für ca. eine Sekunde aufleuchtet.

Schaltungsbeschreibung

Hauptbestandteile der in [Bild 3](#) gezeigten Schaltung des FS20-UART-Sendemoduls sind das auf 868,35-MHz-Basis sendende Funkmodul HFS1 und der Mikrocontroller IC 1, der quasi als Protokollwandler die über die UART-Schnittstelle empfangenen Befehle in FS20-kompatible Funkbefehle umwandelt.

Die Kommunikation mit externen Hardwarekomponenten, wie z. B. einem weiteren Mikrocontroller, erfolgt über die Datenleitungen TxD und RxD der Stiftleiste ST1. Auch die Stromversorgung des FS20 US erfolgt über diese 4-polige Schnittstelle, wobei hierfür eine stabilisierte Gleichspannung zwischen 3 und 5 V bereitgestellt werden muss. Das FS20-US-Modul nimmt keine Spannungswandlung vor, sondern verfügt lediglich über eine Sicherung (R 1) und einen aus einem MOSFET aufgebauten Verpolungsschutz (T 1 und R 4) mit sehr geringem Spannungsabfall. Der Verzicht auf einen Spannungswandler ist damit begründet, dass mit der gewählten Versorgungsspannung gleichzeitig der Spannungspegel auf der UART-Schnittstelle vorgegeben wird und hier dadurch kein zusätzlicher Pegelwandler notwendig wird. Natürlich ist hierbei besondere Vorsicht geboten, da zu hohe Spannungspegel auf den Datenleitungen oder auf der Versorgungsleitung bereits nach kürzester Zeit zu einem Defekt der Elektronik führen. Als absolutes Maximum gilt ein Spannungspegel von +5 V (+10 %) für die Versorgungsspannung und die Datenleitungen, wobei die Spannung auf den Datenleitungen niemals die Höhe der Versorgungsspannung überschreiten darf!

Getaktet wird der Mikrocontroller IC 1 vom 4-MHz-Resonator Q 1. Wie es unter „Elektronikwissen“ genauer erläutert ist, können mit diesem Quarz die wichtigsten Baudraten fehlerfrei verwendet werden.

Damit die Programmausführung auf dem Mikrocontroller IC 1 im Falle einer zu geringen Versorgungsspannung nicht in einem undefinierten Betriebszu-

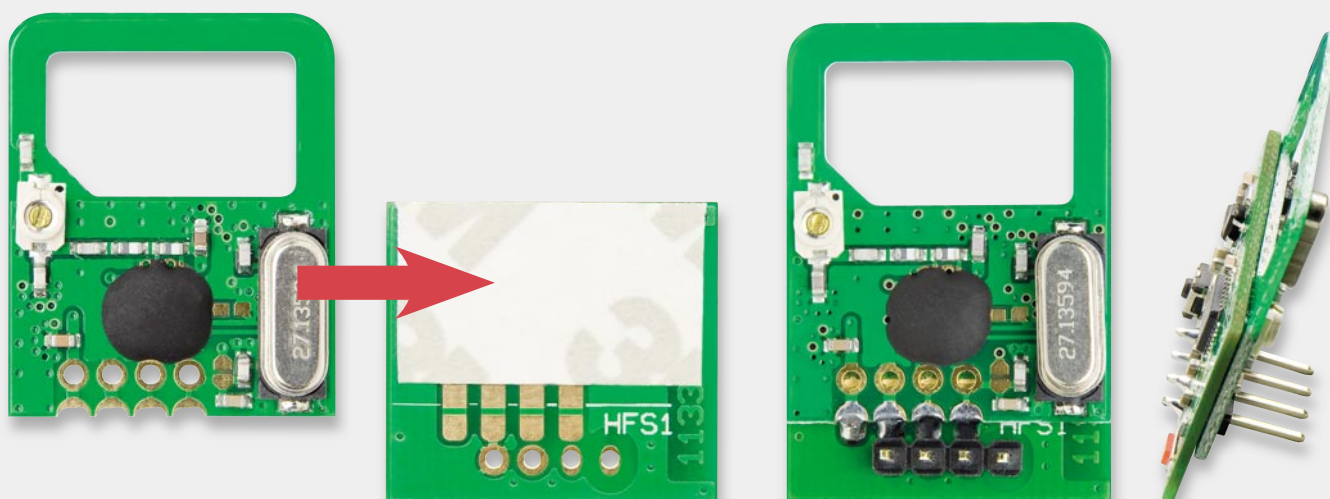


Bild 4: Das Sendemodul ist zunächst per doppelseitigem Klebepad auf der Rückseite der FS20 US-Platine aufzukleben und dann über die am Platinenrand gelegenen SMD-Pads mit dieser zu verlöten. Rechts sieht man die fertig verlötete Kombination und die bündig miteinander verklebten Platinen. (Module vergrößert dargestellt.)

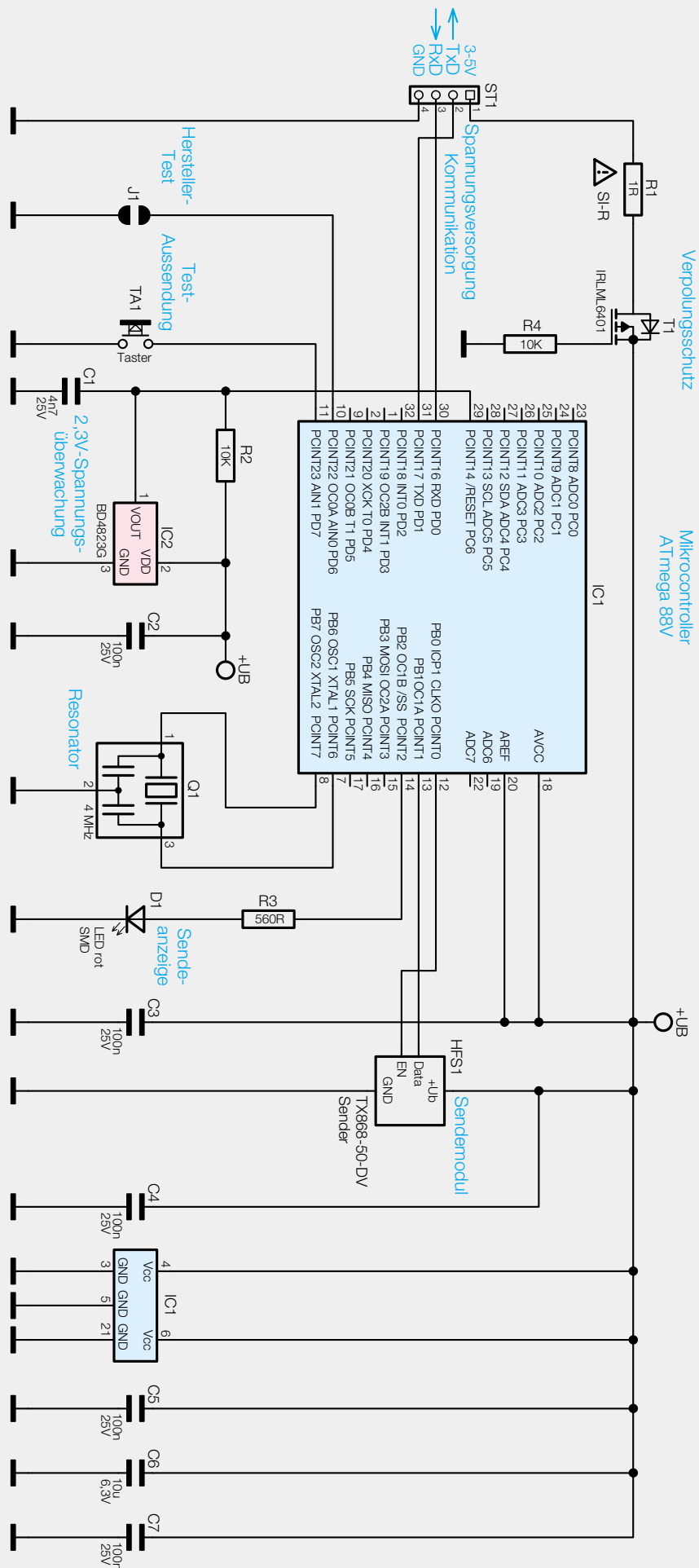
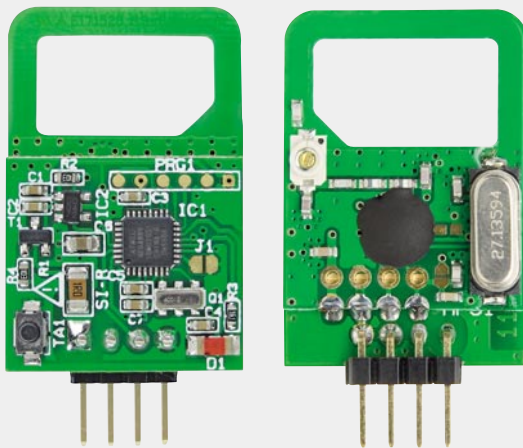
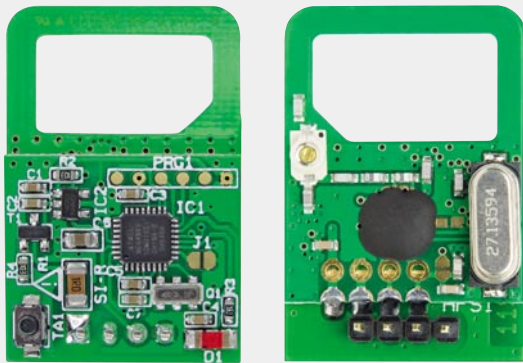
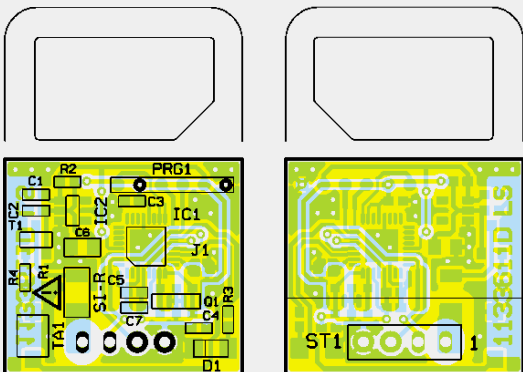


Bild 3: Schaltbild des Mini-UART-FS20-Umsetzers



HFS1



Die komplett bestückte Platine des FS20 US, oben mit gerader und in der Mitte mit abgewinkelter Stiftleiste. Unten befindet sich der Bestückungsplan (Maßstab 2:1).

Widerstände:

Sicherungswiderstand 1 Ω /SMD/1206	R1
560 Ω /SMD/0603	R3
10 k Ω /SMD/0603	R2, R4

Kondensatoren:

4,7 nF/SMD/0603	C1
100 nF/SMD/0603	C2–C5, C7
10 μ F/SMD/0805	C6

Halbleiter:

ELV101015	IC1
BD4823G/SMD	IC2
IRLML6401/SMD	T1
LED, SMD, Rot, Low Current	D1

Sonstiges:

Resonator, keramisch, 4 MHz	Q1
Sendemodul TX868-50-DV eQ-3, 868 MHz	HFS1
Mini-Drucktaster, 1x ein, Höhe: 2 mm	TA1
Stiftleiste, 1x 4-polig, gerade, print	ST1
Stiftleiste, 1x 4-polig, winkelprint	ST1
Klebepad, doppelseitig klebend	

Stückliste

stand weiterläuft, was den FS20 US im schlimmsten Falle zum Dauerstrich-Sender macht (wodurch das Funkband dauerhaft belegt und jede weitere Datenübertragung anderer Geräte verhindert wird), hält der Reset-Baustein IC 2 den Mikrocontroller im Reset-Zustand, sobald die Versorgungsspannung +2,3 V unterschreitet. Der eingesetzte Mikrocontroller verfügt zwar intern über einen ähnlichen Schaltkreis (Brown-out-Detection genannt), jedoch wird dieser je nach Einstellung bereits bei 2,7 V oder erst bei 1,8 V aktiv. Die 2,7 V entsprechen jedoch genau der gerade noch zulässigen Mindestspannung von 3 V minus 10 %, und die 1,8 V liegen bereits unterhalb der Mindestspannung, die das Sendemodul HFS 1 zur sicheren Übertragung benötigt.

Die Bedienelemente Taster TA 1 und die LED D 1 sind direkt am Mikrocontroller angeschlossen und bedürfen keiner weiteren Erklärung. Der Lötjumper J 1 dient nur werkseitigen Testzwecken und darf nicht geschlossen werden.

Nachbau

Der Nachbau des FS20-UART-Sendemoduls gestaltet sich aufgrund der fertig bestückten SMD-Komponenten und der wenigen noch zu bestückenden Bauteile schnell und einfach. Lediglich zwei Bauteile sind noch aufzulöten. Zum einen ist dies das Sendemodul HFS 1, das zuerst, wie in Bild 4 gezeigt, mithilfe des mitgelieferten Klebepads auf die Platinenrückseite auf-



zukleben und anschließend über die am Platinenrand befindlichen SMD-Lötpads anzulöten ist. Die daneben liegenden und mit Bohrungen versehenen Lötpads auf dem Modul HFS 1 werden in diesem Fall nicht verwendet und bleiben frei von Lötzinn. Das Ergebnis sollte wie in Bild 4 rechts gezeigt, aussehen.

Für ST 1 können verschiedene gerade oder abgewinkelte Stiftleisten oder Buchsen verwendet werden, wobei dem Bausatz FS20 US bereits eine abgewinkelte und eine gerade Stiftleiste beiliegt, mit denen das FS20-UART-Sendemodul entweder in aufrechter oder in waagerechter Position aufgelötet oder aufgesteckt werden kann.

Für den anschließenden Einbau des FS20 US sind folgende wichtige Hinweise zu beachten: Das Modul ist zum Schutz vor elektrostatischen Entladungen, wie

sie bei Berührung durch Personen häufig auftreten, in ein geschlossenes Gehäuse einzubauen.

Der Testtaster darf aus demselben Grund nur über einen isolierten Druckknopf/Taststößel/Tastfolie oder Ähnliches von außerhalb des Gehäuses zu betätigen sein.

Die Signal-LED darf ebenfalls nicht berührbar sein. Soll sie außerhalb des Gehäuses sichtbar sein, ist ein transparentes Gehäuse, ein transparentes Gehäusefenster oder ein Lichtleiter zu verwenden.

Die Versorgungsspannung von max. 5 V (+10 %) darf in keinem Fall überschritten werden. Gleichzeitig bestimmt die gewählte Versorgungsspannung auch den zulässigen Spannungspegel auf den Datenleitungen RxD und TxD der UART-Schnittstelle.

Die Länge der an ST 1 angeschlossenen Datenleitungen RxD und TxD darf 1 m nicht überschreiten. Die maximale Länge ist abhängig von der verwendeten Baudrate; je größer die Baudrate, desto kürzer sollten die Leitungen für eine sichere Datenübertragung gewählt werden. **ELV**

Zusammenhang von Baudrate und Prozessortakt

Die UART-Schnittstelle arbeitet mit einem seriellen Datenstrom auf je einer Leitung für das Senden und Empfangen von Daten. Dieser Datenstrom setzt sich aus einem Datenrahmen mit einem Startbit, 5 bis 9 Datenbits, bei Bedarf einem Paritybit (dient zur Feststellung von Übertragungsfehlern) und einem abschließenden Stoppbit zusammen.

UART-Schnittstellen (auch RS232) arbeiten asynchron, also ohne eigenes Taktsignal auf der Übertragungsleitung, mit einer Taktrate (Baudrate), die beim Sender und beim Empfänger identisch gewählt und von diesen auch unterstützt werden muss. Die Synchronisierung zwischen Sender und Empfänger erfolgt über das Start- und Stoppbit des Datenrahmens sowie über die beim Sender und Empfänger eingestellte Baudrate.

Dafür muss der Prozessortakt so herunterteilbar sein, dass die Datenbits mit der jeweiligen Baudrate erzeugt und auch wieder abgetastet werden können. Kleinere Abweichungen sind dabei kein Problem, da sich der Empfänger auf das Startbit jedes Datenwortes immer wieder neu synchronisiert. Dabei ist allerdings zu bedenken, dass weitere Abweichungen durch Fehler in der Takterzeugung hinzukommen können. Der intern z. B. über ein RC-Netzwerk erzeugte Takt vieler Mikrocontroller (z. B. ATmega) ist herstellerseitig nur auf 10 % Genauigkeit kalibriert und zudem stark temperaturabhängig. Daher sollten für eine sichere Datenübertragung besser externe Quarze eingesetzt werden und die Baudraten mit bereits rechnerisch vorhandenen deutlichen Abweichungen von einigen Prozent eher nicht verwendet werden.

Welche Baudraten mit welchen eingesetzten Prozessortakten möglich sind, findet sich in den Datenblättern zum Mikrocontroller (z. B. für den ATmega88V in [1]) oder man nutzt zur

Ermittlung eines geeigneten Quarzes das Internet, wofür ein Baudraten-Rechner, wie in [2] sehr hilfreich ist.

Mithilfe dieses Rechners wurde für den Fall des 4-MHz-Prozessortakts, wie er z. B. beim FS20 US eingesetzt wird, die oben stehende Tabelle errechnet.

In diesem Beispiel können mit dem 4-MHz-Quarz, also mit geringem relativen Fehler 300, 1200, 2400, 4800, 9600, 14k4, 19k2, 38k4, 250k und 500k Baud gesendet und empfangen werden. Auf die langsamsten Baudraten wird bei vielen Systemen häufig verzichtet, da diese selten sinnvoll sind, während die höchsten Baudraten meist zugunsten der Störfestigkeit nicht realisiert werden. Je höher die Übertragungsgeschwindigkeit ist, desto kürzer sollten die Leitungen bleiben und desto wichtiger wird eine geringe Leitungskapazität.

Eine umfassende Einführung zum Thema UART findet sich unter [3].

Baud Rate (bps)	$f_{osc} = 4 \text{ MHz}$			
	$U2X = 0$		$U2X = 1$	
	UBRR	Error (%)	UBRR	Error (%)
300	832=0x340	0	1666=0x682	0
1200	207=0xcf	0.2	416=0x1a0	-0.1
2400	103=0x67	0.2	207=0xcf	0.2
4800	51=0x33	0.2	103=0x67	0.2
9600	25=0x19	0.2	51=0x33	0.2
14.4k	16=0x10	2.1	34=0x22	-0.8
19.2k	12=0xc	0.2	25=0x19	0.2
28.8k	8	-3.5	16=0x10	2.1
38.4k	6	-7	12=0xc	0.2
57.6k	3	8.5	8	-3.5
76.8k	2	8.5	6	-7
115.2k	1	8.5	3	8.5
230.4k	0	8.5	1	8.5
250k	0	0	1	0
460.8k	–	–	0	8.5
500k	–	–	0	0
921.6k	–	–	–	–
1M	–	–	–	–

[1] http://www.atmel.com/dyn/resources/prod_documents/doc2545.pdf (Datenblatt „ATMEGA_88V.pdf“, Seite 193)

[2] www.gjlay.de/helferlein/avr-uart-rechner.html

[3] <http://de.wikipedia.org/wiki/UART>