



1 SPI-Interface des 70RX-S2 Empfangsmoduls

1.1 Hardwareaufbau

Der neue Empfänger 70RX-S2 besitzt zusätzlich zur bewährten 8bit breiten Parallelschnittstelle eine neue 4bit breite SPI Schnittstelle. Es kann jeweils nur eine der beiden Betriebsarten benutzt werden, da sich die beiden Schnittstellen die Anschlüsse teilen. Die Beschaltung der 16-poligen Anschlußleiste erfolgt nach nebenstehendem Schema. In der Betriebsart „Parallel“ wird die Bitkombination am Eingang D0..D7 („0-aktiv“) als Kanalnummer interpretiert (0..255). Die interne CPU berechnet daraus die Parameter für den Synthesizer. Gleichzeitig ist an den Eingängen D0-D3 eine SPI-Schnittstelle aktiv. Dadurch eröffnen sich zusätzliche Einsatzmöglichkeiten für das Empfangsmodul.

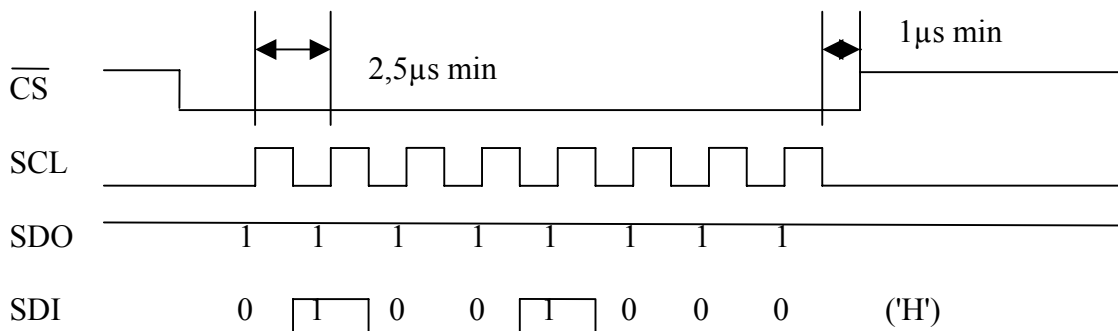
Wird über die SPI (D1-D3) das ASCII-Zeichen 'H' übertragen, schaltet das Modul auf seriellen Betrieb um und D0 wird als Ausgang aktiviert. Die serielle Betriebsart bleibt nun bis zum Neustart des Moduls erhalten. Über die serielle Schnittstelle kann die Frequenz des Empfängers eingestellt werden, Mute und RSSI ausgelesen werden und einige Parameter des Empfängers verändert werden. In der SPI Betriebsart lässt sich auch ein neuer Startkanal festlegen und dauerhaft im EEPROM ablegen.

Das Modul startet normalerweise im Parallelbetrieb und die 8 Leitungen D0..D7 sind zunächst als Eingang eingestellt. Die serielle Betriebsart lässt sich aber dauerhaft speichern (D0 ist dann sofort als serieller Datenausgang verfügbar und der 8bit Paralleleingang abgeschaltet). Die Parametrierung des Moduls ist grundsätzlich nur über die serielle Schnittstelle möglich.

Die Verbindungen zum Modul bestehen aus den Standardsignalen einer SPI-Schnittstelle

- \overline{CS} chip select (low aktiv)
- SCL serial clock (Datenübernahme mit steigender Flanke!)
- SDI data in (Modul)
- SDO data out (Modul)

Die einzelnen Zeichen müssen als ASCII-Zeichen übertragen werden. Jedes Byte wird mit einem \overline{CS} = "high" abgeschlossen (Interrupttrigger).



D0	-	SDO
D1	-	SCL
D2	-	SDI
D3	-	\overline{CS}
D4	-	
D5	-	
D6	-	
D7	-	
GND	-	
LD	-	
MOD	-	
Ub	-	
N.C.	-	
Audio	-	
RSSI	-	
Mute	-	



Wichtig: Zwischen den einzelnen ASCII-Zeichen muss eine „Pausenzeit“ von **50µs** eingehalten werden. Das Modul arbeitet generell im Slave-Betrieb.

Bei der Übertragung des ersten Zeichens ist der SPI-Ausgang u.U. noch nicht aktiviert (Start im Parallel-Mode). In diesem Fall ist das erste empfangene Zeichen '0xFF'!

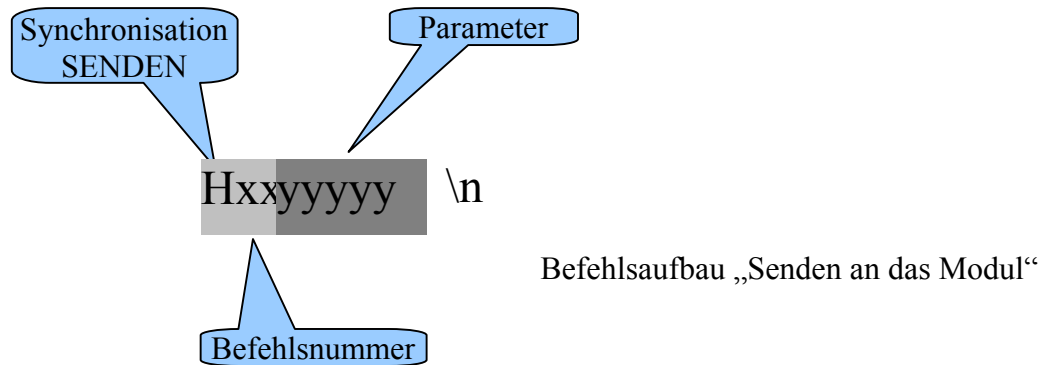
Bitte unbedingt beachten: Die Kommunikation sollte sich immer auf das Nötigste beschränken. Wenn keine Einstellung am Modul vorgenommen wird oder keine Abfrage getätigt wird, sollen insbesondere die Signale SCL und SDI in Ruhe sein. Jeder Flankenwechsel eines Rechtecksignals kann sonst das empfangene Signal beeinflussen.

1.2 Software

1.2.1 Befehlsaufbau

Der Befehlsaufbau beginnt immer mit dem Buchstaben 'H' gefolgt von Ziffern. Wird nach der Übertragung der Ziffern ein Zeichen kleiner ' ' oder 0x20 gelesen, wird die Befehlsdekodierung ausgelöst. Bei einer Pufferlänge von mehr als 10 Zeichen wird die Datenaufnahme abgebrochen und die Dekodierung **NICHT** ausgeführt.

Die Befehlsnummer (xx) MUSS MIT 2 ZIFFERN übergeben werden (führende '0')



Der Befehlsaufbau beim Empfang entspricht dem Befehlsaufbau beim Senden, jedoch ist das Synchronisationszeichen bei fehlerfreier Ausführung ein 'M' und bei fehlerhafter Ausführung ein 'E' (Error)!

Bei der Ausführung eines Befehl kann man insgesamt 4 Phasen (Zustände) unterscheiden:

- (0) warten auf Synchronisationszeichen
- (1) Datenaufnahme bis zum Abschlusszeichen
- (2) Befehl dekodieren/ausführen
- (3) Antwort senden



Wird ein Zeichen übertragen, so wird der letzte **ZUSTAND VOR EMPFANG** dieses Zeichens zurückgegeben (SPI)!

		Rückgabe des ASCII-Zeichens	
Beispiel:	1. Zeichen	<> 'H'	'0' → Zustand warten
	2. Zeichen	= 'H'	'0' → Zustand warten; Synchronisierung
	3. Zeichen	= '1'	'1' → Zustand Datenaufnahme Code 1. Digit
	4. Zeichen	= '1'	'1' → Zustand Datenaufnahme Code 2. Digit
	5. Zeichen	= '1'	'1' → Zustand Datenaufnahme Parameter 1. Digit
	6. Zeichen	= '2'	'1' → Zustand Datenaufnahme Parameter 2. Digit
	(6a. Zeichen	= '3'	'1' → Zustand Datenaufnahme Parameter 3. Digit)
	(6b. Zeichen	= '4'	'1' → Zustand Datenaufnahme Parameter 4. Digit)
	(6c. Zeichen	= '5'	'1' → Zustand Datenaufnahme Parameter 5. Digit)
	7. Zeichen	= '\n'	'1' → Zustand Datenaufnahme, Ende
	8. Zeichen	= ' '	'2' → Zustand Dekodieren/Ausführen
	9. Zeichen	= ' '	'2' → Zustand Dekodieren/Ausführen
	10. Zeichen	= ' '	'2' → Zustand Dekodieren/Ausführen
	11. Zeichen	= 'M'	'M' → Antwort senden ; Werte folgen
	12. Zeichen	= '1'	'1' → Code 1. Digit
	13. Zeichen	= '1'	'1' → Code 2. Digit
	14. Zeichen	= 'y'	'y' → Ergebnis Parameter 1. Digit
	15. Zeichen	= 'y'	'y' → Ergebnis Parameter 2. Digit
	(15a. Zeichen	= 'y'	'y' → Ergebnis Parameter 3. Digit)
	(15b. Zeichen	= 'y'	'y' → Ergebnis Parameter 4. Digit)
	(15c. Zeichen	= 'y'	'y' → Ergebnis Parameter 5. Digit)
	16. Zeichen	= '\n'	'\n' → Stringabschluss

1.2.2 Befehlscodes

Basierend auf dem in 1.2.1 beschriebenen Befehlsaufbau ist es möglich **99 Befehlsnummern** zu vergeben (00 ist dabei nicht erlaubt!). Die mit yyy gekennzeichneten Positionen sind die jeweiligen Parameterwerte!

Es sind zur Zeit nur einige Befehlsnummern verwendet.

ALLE ZEICHEN SIND ASCII-ZEICHEN !!!!!

1.2.2.1 GET_RSSI (01)

Abfrage des RSSI-Signals. Der Rückgabewert ist das RSSI-Signal in 10mV Einheiten.

Befehlsstring: „H010\n“

Antwort: „M01yyy\n“ yyy RSSI in 10mV

1.2.2.2 GET_F_KORR (02)

Abfrage der Frequenzkorrektur. Der Wert wurde beim Abgleich des Moduls ermittelt und im EEPROM abgelegt.

Befehlsstring: „H020\n“

Antwort: „M02yyy\n“ yyy Dezimalzahl (0...127)



1.2.2.3 GET_VE_KORR (03)

Abfrage des Arbeitspunktes der Vorstufenregelung. Der Wert liegt im Bereich 15...60 und wird durch das RSSI Signal nachgeführt (dies kann für Sonderanwendung abgeschaltet werden, siehe 1.2.2.10). Der Defaultwert ist im EEPROM abgelegt.

Befehlsstring: „H030\n“
Antwort: „M03yyy\n“ yyy Dezimalzahl (0...127)

1.2.2.4 GET_RASTER (04)

Abfrage der Kanalrasterfrequenz. Der Wert wird in 10Hz Einheiten zurückgegeben.

Befehlsstring: „H040\n“
Antwort: „M04yyyy\n“ yyyy Dezimalzahl in 10Hz

1.2.2.5 SET_MUTE (08)

Definition des Schwellwertes des MUTE-Signals. Alle Werte zwischen 1 und 254 setzen die Schwelle auf den übertragenen Wert (Spannung des RSSI-Signals in 10mV). Die Werte 0 und 255 stellen einen Sonderfall dar. Mit dem Wert 0 wird der aktuell gemessene RSSI-Wert als Schwellwert übernommen – mit dem Wert 255 wird der ab Werk voreingestellte Wert zurückgesetzt (ermittelt bei 20dB SINAD).

1...254 Schaltschwelle in 10mV Einheiten
0 = Übernahme des aktuellen Wertes
255 = Defaultwert

Befehlsstring: „H0890\n“ setzt die Schwelle auf 900mV
Antwort: „M0890\n“ eingestellter Wert wird bestätigt!

1.2.2.6 GET_CHANNEL (05)

Abfrage der aktuellen Kanalnummer.

Befehlsstring: „H050\n“
Antwort: „M05yyyy\n“ yyyy Dezimalzahl Kanalnummer

1.2.2.7 SET_CHANNEL (06)

Setzen der Kanalnummer.

Befehlsstring: „H06145\n“ setzt den Kanal auf Nr 145
Antwort: „M06145\n“ eingestellter Wert wird bestätigt!

1.2.2.8 GET_MUTE (07)

Lesen der aktuellen MUTE-Schwelle. Der eingestellte Wert wird in 10mV-Einheiten zurückgegeben.

Befehlsstring: „H07\n“ Aufforderung Mute-Schwelle zurückzugeben
Antwort: „M0792\n“ MUTE-Schwelle = 92 → 920mV



1.2.2.9 SET_SPIFLAG(09)

Wenn dieses Flag gesetzt ist, startet das Modul direkt im seriellen Betrieb. Mit dem Wert =1 als Parameter wird das Flag gesetzt mit Wert=0 wieder zurückgesetzt. Es werden alle Flags als 16-Bit Wert zurückgegeben

Befehlsstring:	„H091\n“	setzen des SPI-Flags
Antwort:	„M09yyyy\n“	yyyy = 16_bits (Flags)
Befehlsstring:	„H090\n“	zurücksetzen des SPI-Flags
Antwort:	„M09yyyy\n“	yyyy = 16_bits (Flags)

1.2.2.10 SET_V_DISABLE(10)

Dieses Flag schaltet die automatische Regelung der Vorstufe aus.

Befehlsstring:	„H101\n“	setzen des Flags der Regelsperre
Antwort:	„M10yyyy\n“	yyyy = 16_bits (Flags)
Befehlsstring:	„H100\n“	zurücksetzen des Flags der Regelsperre
Antwort:	„M10yyyy\n“	yyyy = 16_bits (Flags)

1.2.2.11 SET_START_CH(11)

Setzen der Kanalnummer mit der das Modul einstartet. Das Modul muss zusätzlich auf SPI Betrieb eingestellt werden (siehe 1.2.2.9)

Befehlsstring:	„H11145\n“	setzt den Startkanal auf Nr 145
Antwort:	„M11145\n“	eingestellter Wert wird bestätigt!

1.2.2.12 GET_LDLIMIT (51)

Lesen der Zyklenanzahl nach denen die Parametrierung des Synthesizers nach Ausbleiben des LD (Lock Detect) Signals wiederholt wird.

Befehlsstring:	„H51\n“	
Antwort:	„M51yyyy\n“	yyyy Dezimalzahl Lock Detect Auswert

1.2.2.13 GET_FLAGS (55)

Aufforderung alle Flags (wie in 1.2.2.15 und 1.2.2.16) zurückzusenden.

Befehlsstring:	„H55\n“	alle Flags zurücksenden
Antwort:	„M55yyyyyy\n“	

Aktuell definierte Flags:

0 SPI_FLAG	Betriebsart bei Start des Moduls SPI/Parallel
1 VORSTUFE_FLAG	Vorstufenregelung aus/ein

Für Ihre Tests können wir Ihnen gerne ein kleines C-Programm zukommen lassen, das die Kommunikation des Moduls über die SPI Schnittstelle vornimmt. Einfach eine kurze Mail an info@hmradiode mit Stichwort SPI schicken. Die Anbindung an vorhandene Prozessoren wird so möglicherweise etwas erleichtert.