Modbus Funktionen des Antennentuners

Inhaltsverzeichnis

1 Übersicht

Dieses Dokument beschreibt die implementierten ModBus Funktionen des Antennentuners. Dieser basiert auf dem ATMega644PA-AU Board mit RS485 Interface in der Hardware-Version 1.0 vom 11.11.2020 oder der Version 1.1 vom 19.10.2021. Die Boards unterscheiden sich nicht funktional. Das Board V1.1 wurde maschinell teilbestückt und es erlaubt optional den Anschluß einer Batterie. Außerdem ist der Trimmer für die Quarzfrequenz entfallen.

1.1 Kommunikationsparameter nach der Initialisierung

Modbus-Adresse: 20

Baudrate: 9600 Bd, even parity, 2 stop bits

2 Implementierte Modbus-Funktionen

Die folgenden Modbus-Funktionen sind implementiert:

Read Coils: 0x01

Read Holding Registers: 0x03

Read Input Registers: 0x04

Write Single Coil: 0x05

Write Single Register: 0x06

Write Multiple Registers: 0x10

Read File Record: 0x14

Einige der oben genannten Funktionen haben Einschränkungen. Sie unterstützen möglicherweise nicht alle Zugriffsarten (RAM, PROGMEM, EEPROM).

Sende- und Empfangspuffer sind 255 Bytes lang und müssten damit alle Modbus Transaktionen unterstützen. Allerdings wurden keine Tests mit sehr langen Nachrichten durchgeführt.

Alle Adressen in den nachfolgenden Tabellen sind dezimal.

2.1 Read/Write Coils

Die Modbus Nomenklatur bezeichnet mit Coils das, was gewöhnlich ein binäres I/O-Port genannt wird. Man kann eine oder mehrere Coils ein- oder ausschalten sowie ihren Zustand lesen. Folgende Adressen sind für die Coils vergeben:

Adresse	ATMega644 Port	Anwendung
1000..1007	PA0..PA7	nur zum Testen
1008..1015	PB0..PB7	nur zum Testen
1016..1023	PC0..PC7	nur zum Testen
1024..1031	PD0..PD7	nur zum Testen

Hinweis: Zur Konfiguration des Antennentuners sollen die Holding Register 3300 und 3301 verwendet werden. Die Read und Write Coil Funktionen werden hier nur zu Testzwecken unterstützt. Sie sollten in der endgültig freigegebenen Version entfernt werden.

2.2 Modbus Input Register

Modbus definiert Input-Register, die nur lesbar sind und Holding Register, die les- und schreibbar sein können, aber nicht müssen. Modbus Register sind 16-bit breit. Werden breitere Register benötigt, müssen mehrere 16-bit Register benutzt werden. Alle Register sind als Big-Endian implementiert. Das heißt, daß das höherwertige Byte oder Wort an der niedrigeren Adresse steht und bei der Übertragung als erstes übertragen wird.

Folgende Input Register sind implementiert:

Adresse	Anwendung	Datenbreite	R/W
2000	Quarzfrequenz in kHz	16-bit	R
2001	Modultyp: 0x0202 (V1.0) 0x0203 (V1.2)	16-bit	R
2002	Modulnummer (eindeutige Nummer des Moduls)	16-bit	R
2003	Hardware-Version	16-bit	R
2004	Software-Version	16-bit	R
2005	CPU-Takt in kHz	16-bit	R
2006	I/O Takt in kHz	16-bit	R
2007	Anzahl der Flash Pages für die Applikation (RWW Sektion)	16-bit	R
2008	Anzahl der Flash Pages für den Bootloader (NRWW Sektion)	16-bit	R
2010 .. 2026	Unterstützte Baudraten / 10, 0 = Ende der Liste	16-bit	R
2100 .. 3123	Lesen des EEPROM Inhalts (2 kB)	16-bit	R
4000 .. 36767	Lesen des Flash Inhalts (64 kB)	16-bit	R

Diese Input Register können mit der Modbus Funktion „Read Input Registers“ (Function Code 04) gelesen werden. Die Anzahl der mit einem Kommando zu lesenden Register ist auf 125 begrenzt.

2.3 Modbus Holding Register

Holding Register sind je nach ihrer Funktion in mehrere Adressbereiche unterteilt. Es gibt einen Bereich für solche Register, die auf allen Modulen dieser Art implementiert sind und einen weiteren Bereich mit Registern für die speziellen Funktionen eines Moduls. Die einzelnen Bereiche können nochmals in logische Gruppen unterteilt sein.

Einzelne oder mehrere Holding Register können mit der Modbus Funktion „Read Holding Register“ (Function Code 03) gelesen werden. Wenn mehrere Register mit einem einzigen Funktionsaufruf gelesen werden, werden Interrupts unterbunden, während die Daten in den Sendepuffer kopiert werden. Damit ist sichergestellt, daß alle Daten aus demselben Zeitintervall stammen. Das ist wichtig, wenn RTC-Daten oder Zähler gelesen werden, die mehr als 16-bit umfassen.

Zum Schreiben der Holding Register wird die Funktion „Write Single Register“ (Function Code 06) benutzt. „Write Multiple Registers“ (Function Code 16) ist nur zum Zugriff auf Daten im RAM implementiert, also nicht für EEPROM Zugriffe ab der Adresse 3000 und auch nicht für die Verwaltungsregister von 3020 bis 3022. Write Multiple Registers sollte zum Setzen der Uhrzeit verwendet werden, da es eine geringere Latenzzeit hat, als das einzelne Setzen der Register mit der „Write Single Register“ Funktion. Write Multiple Registers verhindert anders als das Lesen keine Interrupts. Vor dem Setzen der Uhr sollte sie daher disabled werden.

2.3.1 Gemeinsame Holding Register

Folgende Holding Register sind, falls nicht anders angegeben, auf allen Modulen dieser Familie implementiert:

2.3.1.1 Konfiguration

Adresse	Anwendung	Datenbreite	R/W
3000	Anzahl der Konfigurationsregister	16-bit	R
3001	ModBus Adresse, nur die unteren 8 Bits sind gültig, die oberen 8 Bits sind immer 0.	16-bit	R/W
3002	Baudrate/10. (11=110 Bd, 30=300 Bd, 60=600 Bd, 120=1.2 kBd, 240=2.4 kBd, 480=4.8 kBd, 960=9.6 kBd, 1440=14.4 kBd, 1920=19.2 kBd, 2880=28.8 kBd, 3840=38.4 kBd, 5760=57.6 kBd, 7680=76.8 kBd, 11520=115.2 kBd, 12800=128 kBd, 25600=256 kBd)	16-bit	R/W
3003	RTC Korrektur. Positiver Wert: alle n Sekunden werden 10ms hinzuaddiert (Uhr läuft schneller). Negativer Wert: alle n Sekunden werden 10ms abgezogen (Uhr läuft langsamer).	16-bit	R/W

Adresse	Anwendung	Datenbreite	R/W
3004	EEPROM Programmierzähler (high). Zählt die Anzahl der EEPROM Programmierzyklen	16-bit	R
3005	EEPROM Programmierzähler (low)	16-bit	R
3006	Port A Status (nur Antennenumschalter)	16-bit	R/W

2.3.1.2 Datum und Uhrzeit

Adresse	Anwendung	Datenbreite	R/W
3010	Jahr	16-bit	R/W
3011	Monat	16-bit	R/W
3012	Tag	16-bit	R/W
3013	Stunde	16-bit	R/W
3014	Minute	16-bit	R/W
3015	Sekunde	16-bit	R/W
3016	Centisekunde (hundertstel Sekunde)	16-bit	R/W
3017	Enable Real Time Clock, 0=stopped, 1=run	16-bit	R/W

2.3.1.3 Verwaltung

Adresse	Anwendung	Datenbreite	R/W
3020	Neustart, schreibe 0x39f1. WDT Reset wird nach etwa 16 ms ausgelöst. Fehlerzähler bleiben erhalten. Keine Antwort!	16-bit	W
3021	Alle Fehlerzähler löschen. Schreibe 0x39f1.	16-bit	W
3022	Rücksetzen der Device ID im EEPROM auf den Default Wert. Schreibe 0x39f1.	16-bit	W
3023	Start des Bootloaders, schreibe 0x39f1. Keine Antwort!	16-bit	W

2.3.1.4 Fehlerzähler

Adresse	Anwendung	Datenbreite	R/W
3100	CPU Status Register nach Reset. Die oberen 8 Bits sind immer 0.	16-bit	R
3101	Gesamtzahl von Resets seit Power-on	16-bit	R/W
3102	Anzahl von Watchdog Resets seit Power-on	16-bit	R/W
3103	Anzahl von Software Resets seit Power-on (wegen Neustart mittels Register 3020)	16-bit	R/W
3104	Anzahl von externen Resets seit Power-on	16-bit	R/W
3105	Anzahl von brown-out Resets seit Power-on	16-bit	R/W
3106	Gesamtanzahl fehlerfrei empfangener Nachrichten seit Power-on (upper halfword)	16-bit	R/W
3107	Gesamtanzahl fehlerfrei empfangener Nachrichten seit Power-on (lower halfword)	16-bit	R/W
3108	Anzahl fehlerfrei empfangener Nachrichten für dieses Modul seit Power-on (upper halfword)	16-bit	R/W
3109	Anzahl fehlerfrei empfangener Nachrichten für dieses Modul seit Power-on (lower halfword)	16-bit	R/W
3110	Anzahl fehlerhafter Nachrichten (CRC-Fehler) seit Power-on (upper halfword)	16-bit	R/W
3111	Anzahl fehlerhafter Nachrichten (CRC-Fehler) seit Power-on (lower halfword)	16-bit	R/W
3112	Anzahl fehlerhafter Nachrichten seit Power-on; frame error, rx buffer overrun (upper halfword)	16-bit	R/W
3113	Anzahl fehlerhafter Nachrichten seit Power-on; frame error, rx buffer overrun (lower halfword)	16-bit	R/W
3114	Anzahl empfangener Bytes seit Power-on (upper halfword)	16-bit	R/W
3115	Anzahl empfangener Bytes seit Power-on (lower halfword)	16-bit	R/W
3116	Anzahl gesendeter Bytes seit Power-on (upper halfword)	16-bit	R/W
3117	Anzahl gesendeter Bytes seit Power-on (lower	16-bit	R/W

Adresse	Anwendung	Datenbreite	R/W
	halfword)		
3118	Vergangene Zeit seit dem letzten Einschalten in hundertstel Sekunden (upper halfword)	16-bit	R/W
3119	Vergangene Zeit seit dem letzten Einschalten in hundertstel Sekunden (middle halfword)	16-bit	R/W
3120	Vergangene Zeit seit dem letzten Einschalten in hundertstel Sekunden (lower halfword)	16-bit	R/W

2.3.1.5 Betriebsparameter

Adresse	Anwendung	Datenbreite	R/W
3200	Temperatur in °C * 16, signed	16-bit	R
3201	Versorgungsspannung (Vin). Gemessen über einen 4k7-1k Spannungsteiler, Referenzspannung des 12-bit ADC: 2,5 V. ==> Vin = 2,5 * 5,7 * ADC / 1024 = 0,013916015625 * ADC	16-bit	R

2.3.2 Spezielle Funktionen des Antennentuners

Die Register in der nachfolgenden Tabelle sind nur im Antennentuner implementiert.

Adresse	Anwendung	Datenbreite	R/W
3300	Setzen der Spulen-Relais. Nur die unteren sechs Bits sind gültig, alle anderen werden ignoriert.	16-bit	R/W
3301	Setzen der Kondensatoren-Relais. Nur die unteren sechs Bits sind gültig, alle anderen werden ignoriert.	16-bit	R/W

2.3.3 Konfigurierbare EEPROM Parameter

Hier sind Parameter des Moduls abgespeichert, die geschrieben und gelesen werden können. Dabei handelt es sich um Datentypen, die das Modbus-Protokoll nicht kennt. Sie werden daher in Blöcke von 16-bit Worten aufgeteilt und in der Reihenfolge übertragen, in der sie im EEPROM abgelegt sind.

2.3.3.1 TCXO-Register

Adresse	Anwendung	Datenbreite	R/W
3500 .. 3501	Nominale Quarzfrequenz f_n in Hz (uint32_t)	2 * 16-bit	R/W
3502 .. 3503	Maximale Frequenzabweichung des Quarzes in ppb (uint32_t)	2 * 16-bit	R/W
3504 .. 3505	Ausgleichsparameter a0 (float)	2 * 16-bit	R/W
3506 .. 3507	Ausgleichsparameter a1 (float)	2 * 16-bit	R/W
3508 .. 3509	Ausgleichsparameter a2 (float)	2 * 16-bit	R/W
3510 .. 3511	Ausgleichsparameter a3 (float)	2 * 16-bit	R/W
3512 .. 3513	Ausgleichsparameter a4 (float)	2 * 16-bit	R/W
3514 .. 3515	Ausgleichsparameter a5 (float)	2 * 16-bit	R/W

Die TCXO-Parameter gestatten eine Ausgleichsrechnung der Quarzfrequenz über der aktuellen Temperatur. Dazu bietet sich die Methode der kleinsten Fehlerquadrate an. Aus Messungen der Frequenz über der Temperatur errechnet man ein Polynom, dessen Parameter a0 bis a5 hier abgespeichert werden. Anders als mit der RTC-Korrektur kann damit eine temperaturabhängige kontinuierliche Abweichung von der Nominalfrequenz errechnet werden. Das kann entweder aus dem Host oder auf dem Modul geschehen.

2.3.4 File Records

Modbus benutzt File Record Funktionen um Datensätze zu lesen und zu schreiben. Read File Record ist implementiert, um einige Textdaten zur Diagnose als ASCII Strings zu lesen.

File Number	File Record	Bedeutung
1	1000	Dateiname der Quelldatei des Hauptprogramms
1	1001	Datum und Uhrzeit der Quelldatei des Hauptprogramms
1	1002	Dateiname der Quelldatei der Modbus Bibliothek
1	1003	Datum und Uhrzeit der Quelldatei der Modbus Bibliothek
1	1004	Datum des Compilerlaufs
1	1005	Uhrzeit des Compilerlaufs

Nur File Number 1 wird unterstützt und die Datensätze müssen im Programmspeicher stehen. Write File Record ist nicht implementiert.

2.4 Diagnostic Funktionen

Diagnostic Funktionen werden mit der Modbus Funktionscode 0x08 ausgeführt. Folgende Funktionen sind implementiert, aber auskommentiert und bisher ungetestet (da nicht von QModMaster unterstützt):

Funktionscode	Funktion
00	Return Query Data
01	Restart Communications Option
04	Force Listen Only Mode
10 (0x0A)	Clear Counters and Diagnostic Register
11 (0x0B)	Return Bus Message Count
12 (0x0C)	Return Bus Communication Error Count

3 Bootloader Funktionen

3.1 Übersicht

Der Bootloader ist ein eigenständiges Programm, das in einem speziellen Bereich des Flash-Speichers abgelegt ist. Der Bootloader ist optional und gestattet einen Update des Applikationsprogramms über die serielle Modbus-Schnittstelle. Dazu ist der Bootloader mit einem reduzierten Umfang an Modbus-Funktionen zur Kommunikation mit einem Host-PC ausgestattet.

Nach einem Hardware-Reset startet der Bootloader und prüft über einen CRC die Gültigkeit des Applikationsprogramms im Flash-Speicher. Ist diese Prüfung erfolgreich, wird sofort die Applikation gestartet, andernfalls bleibt der Bootloader aktiv und wartet auf Host-Kommandos.

Dieses Kapitel beschreibt die implementierten ModBus Funktionen des Bootloaders. Sie gestatten im wesentlichen das Löschen und Schreiben des Applikationsspeichers, dessen Gültigkeitsprüfung und den Start der Applikation.

3.2 Kommunikationsparameter nach der Initialisierung

Die Kommunikationsparameter sind immer die im Abschnitt 1.1 (Seite 3) beschriebenen Default-Parameter der Applikation. Eine Änderung dieser Parameter wirkt sich nicht auf den Bootloader aus.

3.3 Implementierte Modbus-Funktionen der Bootloaders

Die folgenden Modbus-Funktionen sind implementiert:

Read Holding Registers: 0x03

Read Input Registers: 0x04

Write Single Register: 0x06

Write Multiple Registers: 0x10

Sende- und Empfangspuffer sind 255 Bytes lang und müssten damit alle Modbus Transaktionen unterstützen. Allerdings wurden keine Tests mit sehr langen Nachrichten durchgeführt.

Alle Adressen in den nachfolgenden Tabellen sind dezimal.

3.4 Modbus Input Register des Bootloaders

Folgende Input Register sind implementiert:

Adresse	Anwendung	Datenbreite	R/W
2000	Quarzfrequenz in kHz	16-bit	R
2001	Modultyp (*): 0x8202 (V1.0) 0x8203 (V1.2)	16-bit	R
2002	Hardware-Version	16-bit	R
2003	Software-Version	16-bit	R
2004	CPU-Takt in kHz	16-bit	R
2005	I/O Takt in kHz	16-bit	R
2006	Anzahl der Flash Pages für die Applikation (RWW Sektion)	16-bit	R
2007	Anzahl der Flash Pages für den Bootloader (NRWW Sektion)	16-bit	R
2100 .. 3123	Lesen des EEPROM Inhalts (2 kB)	16-bit	R
4000 .. 36767	Lesen des Flash Inhalts (64 kB)	16-bit	R

Diese Input Register können mit der Modbus Funktion „Read Input Registers“ (Function Code 04) gelesen werden. Die Anzahl der mit einem Kommando zu lesenden Register ist auf 125 begrenzt.

*: Beim Bootloader ist das obere Bit des Modultyps auf 1 gesetzt. Damit kann die Host Software unterscheiden, ob gerade die Applikation oder der Bootloader ausgeführt wird.

3.5 Modbus Holding Register des Bootloaders

Einzelne oder mehrere Holding Register können mit der Modbus Funktion „Read Holding Register“ (Function Code 03) gelesen werden. Zum Schreiben der Holding Register wird die Funktion „Write Single Register“ (Function Code 06) benutzt. „Write Multiple Registers“ (Function Code 16) ist zum Zugriff auf den Page Buffer und zum Schreiben des EEPROMs implementiert. Dabei ist aber zu beachten, daß die Länge eine Modbus Nachricht auf 125 Datenbytes begrenzt ist. Wenn die Anzahl der Register eine Zweierpotenz sein soll, dann können also nur höchstens 32 Register mit einer Nachricht übertragen werden, denn für 64 Register wären 128 Bytes notwendig.

3.5.1 Verwaltung

Adresse	Anwendung	Datenbreite	R/W
3020	Start der Applikation an Flash-Adresse 0. Schreibe 0x39f1, keine Antwort!	16-bit	W
3023	Start des Bootloaders, schreibe 0x39f1. Keine Antwort!	16-bit	W
3024	Enable Flash erase and programming, schreibe 0x39f1. Wird nach 10 Sekunden Inaktivität zurückgesetzt. Liest verbleibende Zeit in cs.	16-bit	R/W
3025	Application start (noapp_start)	8-bit	R/W

Das Löschen und Programmieren des Flash Speichers muß durch Schreiben des Registers 3024 mit 0x39f1 freigegeben werden. Diese Freigabe wird nach dem Ablauf eines Timeouts (z. Zt. 10 Sekunden) automatisch wieder zurückgezogen. Beim Löschen oder Schreiben einer Page wird der Timeout wieder auf seinen Anfangswert zurückgesetzt. Durch Lesen des Registers 3024 kann die restliche Laufzeit ermittelt werden.

Das bool’sche Register an der Adresse 3025 bestimmt das Startup-Verhalten des Bootloaders. Es wird beim Start des Bootloaders auf 0 (false) gesetzt. Gleichzeitig wird ein Timer gestartet, mit dem der automatische Start der Applikation verzögert wird (derzeit um 2 Sekunden). Wird noapp_start innerhalb dieser Zeit auf einen Wert ungleich 0 (true) gesetzt, dann wird die Applikation nicht gestartet und der Bootloader bleibt aktiv. Diese Funktion dient zum Wiederherstellen eines Systems mit intakter Applikation, aber ungültigen Parametern, z.B. der Baudrate.

3.5.2 Flash Programmierung

Adresse	Anwendung	Datenbreite	R/W
6000	Flash Page löschen. Schreibe Page Nummer n. n=0xffff für alle Pages der RWW Sektion (dauert etwa 1 sek).	16-bit	W
6001	Flash Page n mit Daten aus dem Page Puffer programmieren.	16-bit	W
6100 .. 6228	Page Puffer, „Write Multiple Registers“ (Function Code 16) wird unterstützt	16-bit	R/W

3.5.3 EEPROM Programmierung

Adresse	Anwendung	Datenbreite	R/W
7000 .. 8023	EEPROM programmieren, „Write Multiple Registers“ (Function Code 16) wird unterstützt	16-bit	R/W

Das EEPROM wird durch Schreiben der oben genannten Register direkt neu geschrieben. Ein separates Enable wie beim FLASH Speicher ist nicht notwendig.

4 To Do

Die nachfolgenden Funktionen sollen gelegentlich noch implementiert werden. Ein Teil davon ist nach der ModBus Spezifikation „mandatory“. Mangels Unterstützung von QModMaster wurden sie bisher nicht implementiert.

Read Device ID (ASCII Strings):

0x00: Vendor Name

0x01: Product Code

0x02: Major/Minor Revision

0x03: Vendor URL

0x04: Product Name

0x05: Model Name

0x06: User Application Name

Module:

Temperaturabhängige RTC Korrektur

5 Unterstützte Baudraten

Abhängig vom verbauten Quarz können nur bestimmte Baudraten unterstützt werden. Die nachfolgende Tabelle gibt einen Überblick:

eingestellte Baudrate	gemessene Baudrate	verwendbar bei Quarzfrequenz		
		8 MHz	11,0592 MHz	12 MHz
110	115	✘	✘	✘
300	300	✔	✔	✔
600	600	✔	✔	✔
1200	1200	✔	✔	✔
2400	2400	✔	✔	✔
4800	4800	✔	✔	✔
9600	9592	✔	✔	✔
14400	14390	✘	✔	✔
19200	19209	✔	✔	✔
28800	9591	✘	✘	✘
38400	38432	✘	✔	✘
57600	57703	✘	✔	✔
115200	115473	✘	✔	✘
128000	9589	✘	✘	✘
256000	9593	✘	✘	✘
921600	943396	✘	✘	✘

Die Spalte der gemessenen Baudraten bezieht sich auf von QModMaster und „Waveshare USB to RS485 Konverter“ gesendete ModBus Nachrichten bei den jeweils eingestellten Baudraten. 28.8 kBd, 128 kBd und 256 kBd werden offensichtlich nicht unterstützt und stattdessen wird mit 9.6 kBd gearbeitet.
Bei einer eingestellten Baudrate von 110 Bd weicht die tatsächliche Baudrate so weit ab, daß keine Kommunikation mehr möglich ist.

Hinweis: wegen der eingebauten Tiefpässe in den RS-485-Kommunikationsleitungen ist die Baudrate im Antennentuner auf 38.4 kB begrenzt.