Modul-Design S88-Treiber

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	14.02.2011
Quelle	git → docs → Gesamtprojekt → Dokumente → 02_Design → 02.02_Moduldesign
Autoren	Kai Dziembala
	Norman Nieß
	Tilo Bellmer
Version	2.0
Status	freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.1	24.11.2009	Jan-Christopher Icken	Initial
0.2	30.11.2009	Jan-Christopher Icken	Einleitung, Architektur, Referenzierte Dokumente
0.3	07.12.2009	Jan-Christopher Icken	Ports festgelegt, Zeiten für Abfragen festgelegt
0.4	14.12.2009	Jan-Christopher Icken	Kapitel 6 erstellt
0.5	21.12.2009	Jan-Christopher Icken	Anregungen aus dem Plenum eingefügt
0.6	29.04.2010	Kai Dziembala Norman Nieß	Kapitel 3 – 6 überarbeitet; Kapitel 7 entfernt; Layout-Anpassung
0.7	06.05.2010	Kai Dziembala Norman Nieß	Korrektur von Rechtschreibfehlern; neuer Dokumentenstatus: freigegeben
1.0	17.06.2010	Kai Dziembala	Freigabe des Moduldesign 'S88-Treiber'
1.1	06.12.2010	Tilo Bellmer	Überarbeitung der Moduldokumentation
2.0	17.01.2010	Robert Lucke	Review und Freigabe

2 Inhaltsverzeichnis

3 Referenzierte Dokumente

Software-Design:	Google Code → Dokumente → 02_Design → 02.01_Subsystemdesign → Software-Design
S88-Bus:	Aulis → F4 TI PROJEKT Bredereke WiSe0910 → Projektverzeichnis → Schnittstellendokumentation → S88-Bus

4 Einleitung

Der Zweck des Busses ist die Übermittlung von Informationen aus der Modellbahnanlage an die Steuerung. Der Bus beruht auf einem einfachen Prinzip: Der S88-Bus ist ein serielles Schiebe-Register mit parallelem PS-Eingang.

Weitere Teilnehmer dieses Busses werden durch einfaches Kaskadieren angeschlossen, so entsteht ein langes Schieberegister, in dem alle auszulesenden Bits in einer langen Kette liegen. Diesem Vorteil des einfachen Aufbaus stehen allerdings Nachteile gegenüber: Es ist keine Adressvorgabe der Rückmelder möglich und die Übertragung erfolgt vollkommen ungeschützt, d.h. es gibt weder Parity, Prüfsumme oder CRC.

Bei einer Übertragung geht die PS-Leitung auf 1, darauf erfolgt ein Schiebetakt; alle Register in der Schreibkette übernehmen bei diesem Takt die Information an ihren Paralleleingängen. Als nächstes erfolgt ein RESET-Puls (auch aktiv high), dieser löscht die den Paralleleingang vorgeschalteten Latches, welche damit wieder bereit für die Übernahme neuer Information sind. Die Latches speichern auch kurze Signale bis zur nächsten Abfrage.

Nun wird das Schieberegister (mit PS = 0) mittels eines CLK-Pulses geschoben. Dadurch dass jeweils der Datenausgang eines S88-Moduls mit dem Dateneingang des nächsten verbunden ist, kommen so die beim ersten Takt geladenen Zustände der Latches nach und nach zur Zentrale.

Die Steckerbelegung des S88-Busses:

PIN	Signal
1	Data
2	Ground
3	Clock
4	PS
5	Reset
6	5V

Um die an den S88-Rückmeldemodulen anliegenden Sensordaten auszulesen, wird ein S88-Treiber benötigt. Dieses Modul befindet sich in der Treiberschicht und stellt die Daten über den Shared Memory dem Modul Befehlsvalidierung zur Verfügung.

Außer der reinen Abfrage der Sensordaten, findet auch noch eine Validierung der ankommenden Daten statt.

Um die Rückmeldemodule abzufragen, ist es notwendig, ein Taktsignal (CLOCK) für deren Ansteuerung zu erzeugen. Des Weiteren müssen die Signale für RESET und LOAD erzeugt werden. Die Daten werden seriell auf der Leitung DATA eingelesen.

5 Architektur

5.1 Funktionshierarchie

Der S88-Treiber befindet sich in der Treiberschicht und greift auf einen definierten Speicherbereich im Shared Memory zu. Damit keine Daten anderer Module beschädigt werden, ist auf die Ansteuerung des Speichers besondere Aufmerksamkeit zu legen.

5.2 Daten

2 Byte Sensordaten (1 Bit pro Sensor)

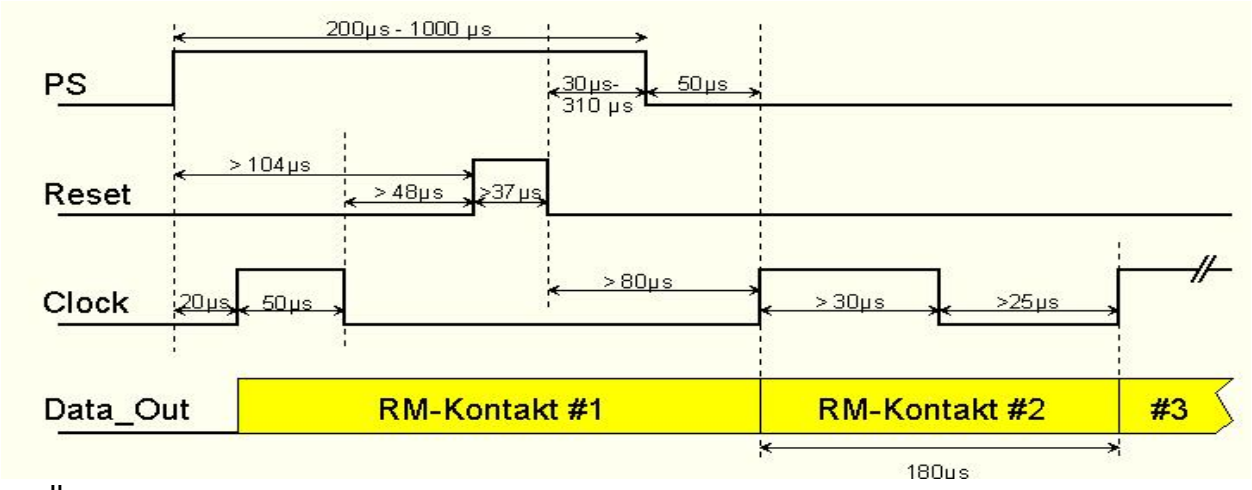
Jedes Bit in den 2 Bytes steht für den Zustand eines Sensors, wobei das MSB für den Sensor Nr. 16 und das LSB für den Sensor Nr.1 steht. Ein gesetztes Bit steht hierbei für ein Ereignis am Sensor, ein nicht gesetztes Bit dementsprechend für kein Ereignis am Sensor.

5.3 Abhängigkeiten von anderen Modulen

Damit der Software-Watchdog das Modul S88-Treiber eindeutig zuweisen kann, hat dieses die Modul-ID 3. Gibt das Modul die Ressourcen des Mikrocontrollers frei, obwohl eine Aufgabe noch nicht abgearbeitet ist, so wird dem Watchdog eine Nachricht mit einem Statusbyte gesendet, das den Fortschritt der Abarbeitung wiedergibt.

Um die Daten von den S88-Rückmeldemodulen abzurufen ist es notwendig, dass mehrere Signale erzeugt werden. Zur Erzeugung des Taktsignals wird die eigene Funktion 'wait(byte times)' genutzt.

Bei der Generierung des Taktsignals ist darauf zu achten, dass hierbei eine Taktperiode von 30µs nicht unterschritten wird. Um dies zu gewährleisten, ist eine Taktperiode von 40µs



sinnvoll.

Die Abfrage des ersten Sensors, benötigt zwecks Generierung des Signals PS und Reset eine Gesamtzeit von ~280µs, jeder weitere Sensor benötigt ~40µs. Dadurch ergibt sich eine Gesamtlesezeit von ~860µs.

Zusätzlich wird eine Validierung durchgeführt. Diese ist in eine extra Codedatei ausgelagert für eine bessere Übersicht. Sie dient dazu nur Veränderungen der Sensorzustände in den Shared Memory zu schreiben. Die Daten werden hierfür zweifach zwischengespeichert. Es gibt alte sowie neue Datenspeicher, die dann miteinander verglichen werden können. Sind die Daten unterschiedlich wird die Veränderung ins Memory geschrieben, falls nicht wird nichts im Memory geändert.

Der genaue Ablauf des Sensorauslesens ist im folgenden Diagramm zu sehen: