

FH GÜNNI

Studiengang Mechatronik

Projektarbeit

Thema 36: Smart Reedimpulse

Vorgelegt von GÜNNIDIGGA

Matrikel-Nr. 503030

Dozent: Prof.Dr.Ing Günni

Erstellt am 5.09.2024

Erklärung

LOLSOOS

Inhaltsverzeichnis

1. Einleitung.....	1
2. Aufbau	1
3. Programmierung.....	1
4. Auslesen.....	2
5. Quellenverzeichnis	3
6. Abbildungsverzeichnis.....	3
7. Anhang	4
7.1 Schaltplan	4
7.2 Laboraufbau	4
7.3 Ausgabe in MBSheet.....	5

1. Einleitung

Im Rahmen des Elektrotechnik Praktikums habe ich mich für das Thema 36: Smart Reed Impulse entschieden. Das Projekt verlangte das Auslesen des Zustands eines Reedkontakts, und das anschließende Kommunizieren über M-Bus. Wir haben uns aufgrund der einfachen Auslese des Reedkontakts entschieden das Programm für einen Arduino Mega zu schreiben. Dazu wurde eine M-Bus Slave Platine eingesetzt, um die Serielle Kommunikation über M-Bus zu ermöglichen. Das Programm wurde in „C“ für einen Arduino geschrieben und mithilfe eines M-Bus Masters getestet. Das Programm soll auf Anfrage des Masters den Status des Reedkontakts abfragen und eine gültige Ausgabe kodieren. Diese wird dann mithilfe der M-Bus Slave Platine, über den M-Bus versendet.

2. Aufbau

Die benutzte M-Bus Slave Platine besteht aus einem TSS712A Chip. Dieser Chip wurde entwickelt, um normale Serielle Kommunikationen auf den Spannungen des M-Bus Systems zu ermöglichen.

Der Arduino schließt an die RX/TX-Pins der Platine an, und versorgt den Chip darüber hinaus mit 5V. Die Signale wandern durch Optokoppler und erreichen den Chip. Dieser reproduziert die Signale mit einer höheren Spannung. Die 0V des Arduinos für das binäre Signal 0, wird als 24V ausgegeben. Für eine binäre 1 gibt der Chip 38V aus. Die 2 Kabel des M-Bus schließen direkt an die Platine mit Schraubkontakten an. [TEX-10]

Der Reedkontakt ist auf einer Steckplatte befestigt. Eine Seite wird mit der 3.3V Stromversorgung des Arduinos verbunden. Die andere Seite wird wiederum mit einem Kabel in Pin 13 des Mikrokontrollers gesteckt. Außerdem wird neben dem Kabel auch ein 6,6kOhm Pull-down-Widerstand parallel in die Platte gesteckt. Dieser wird mit dem GND-Pin des Arduinos verbunden. Der Pull-down Widerstand sorgt für eine Spannung von 0V wenn der Reedkontakt geöffnet ist. Ist der Reedkontakt geschlossen, liegt eine Spannung von 3.3V an Pin 13, die von dem Arduino ausgelesen werden kann.

Zum Testen wurde der M-Bus Master und der Arduino gleichzeitig an einen Laptop angeschlossen.

3. Programmierung

Der Arduino wird beim Einschalten als M-Bus Slave konfiguriert. Er besitzt dabei eine ID und eine M-Bus-Adresse. Darüber hinaus wird die Baudrate auch sofort auf 9600 festgelegt. Der Arduino wartet nun auf den Initialisierungs-Befehl des Masters. Wenn ein

Magnet in die Nähe des Reedkontakts gebracht wird fängt zusätzlich die eingebaute LED des Mikrocontrollers (LED_BUILTIN) an zu leuchten um zu signalisieren das der Kontakt geschlossen ist.

Sobald der Initialisierungs-Befehl empfangen wurde, sendet der Arduino ein Acknowledgement-Paket zurück. Das Einrichten des Arduinos als M-Bus Zähler ist somit abgeschlossen. [WAC-17]

Wenn der Master einen Auslese Befehl versendet, beginnt der Arduino damit den Zustand des Reedkontakts auf Pin 13, mithilfe der reedState()-Funktion abzufragen. Dieser wird als TRUE oder FALSE zurückgegeben und zur Überprüfung in der Konsole ausgegeben. Daraufhin beginnt das Kodieren der Antwort. Das Payload Objekt der M-Bus Bibliothek wird mit payload.reset() zurückgesetzt und mit mehreren Text-Feldern befüllt. Darunter sind zuerst die Baudrate und Firmware Version enthalten. Danach wird das Textfeld „DIGITAL_INPUT“ erstellt und basierend auf der Ausgabe der reedState()-Funktion entweder mit einer 1 oder einer 0 kodiert.

Der Payload wird daraufhin mit einer Checksum in Bytes zusammengefasst, die über die Serielle Verbindung des Arduinos gesendet werden können. [HWH-24]

4. Auslesen

Das Auslesen des M-Bus erfolgt über MBSheet auf dem Laptop. Das Programm sucht Zähler, die an den M-Bus-Master angeschlossen sind über die M-Bus-Adresse. Die Adresse dieses Zählers ist 1, und wird somit sofort erkannt. MBSheet initialisiert den Arduino als Slave und gibt die erkannte ID, die Menge an verfügbaren Datensätzen und den Hersteller des M-Bus-Zählers oben aus.

Danach kann mit dem Befehl „M-BUS Lesen“ eine Abfrage an den Slave geschickt werden. Der Arduino überprüft den Zustand des Reedkontakts und beginnt die Ausgabe zu formatieren. Während der Mikrocontroller arbeitet, gibt er in der Konsole die zu übersenden Daten zur Überprüfung aus. Auch die gesendeten Bits werden zur Kontrolle in der Konsole ausgegeben.

MBSheet druckt danach die Antworten des Slaves in einer Liste aus. Hier kann man deutlich den Zustand des Reedkontakts ablesen. Der Wert ist „\$00“ für einen offenen Reedkontakt und „\$01“ für einen geschlossenen (Abbildung 3).

5. Quellenverzeichnis

- [WAC-17] Wachendorff Prozesstechnik GmbH & Co. KG: M_BUS Communication Protocol. Hamburg. Internet 2017-06
https://www.wachendorff-prozesstechnik.de/fileadmin/wp/fileserver/manual/ECM_WECM_Zaehler_M-BUS_Schnittstellenbeschreibung_Protokoll.pdf
- [HWH-24] GitHub: HWHardsoft: Arduino-MBUS-Meter. Köln. Internet 2024-01-24
<https://github.com/HWHardsoft/Arduino-MBUS-Meter/commits/main/>
- [TEX-10] ti.com: TEXAS INSTRUMENTS: TSS721A Meter-Bus Transceiver datasheet (Rev. B). Hamburg. Internet 2010
<https://www.ti.com/lit/ds/symlink/tss721a.pdf>

6. Abbildungsverzeichnis

Abbildung 1: Schematische Ansicht des Schaltplans	4
Abbildung 2: Aufbau der Schaltung	4
Abbildung 3: Auswertung per MBSheet, die zwei Möglichen Werte sind markiert	5

7. Anhang

7.1 Schaltplan

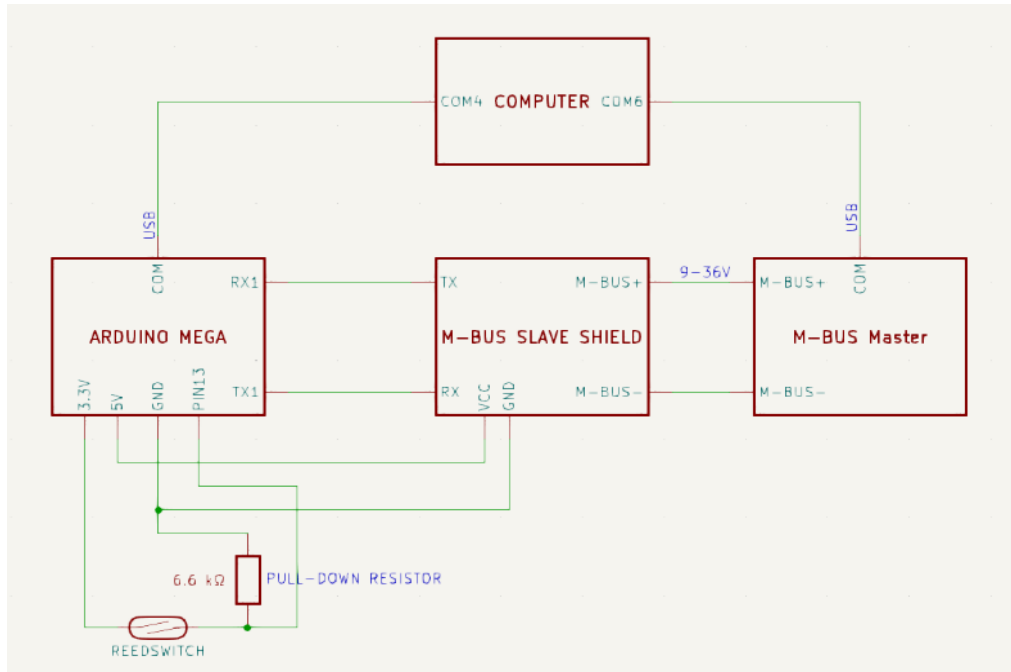


Abbildung 1: Schematische Ansicht des Schaltplans

7.2 Laboraufbau

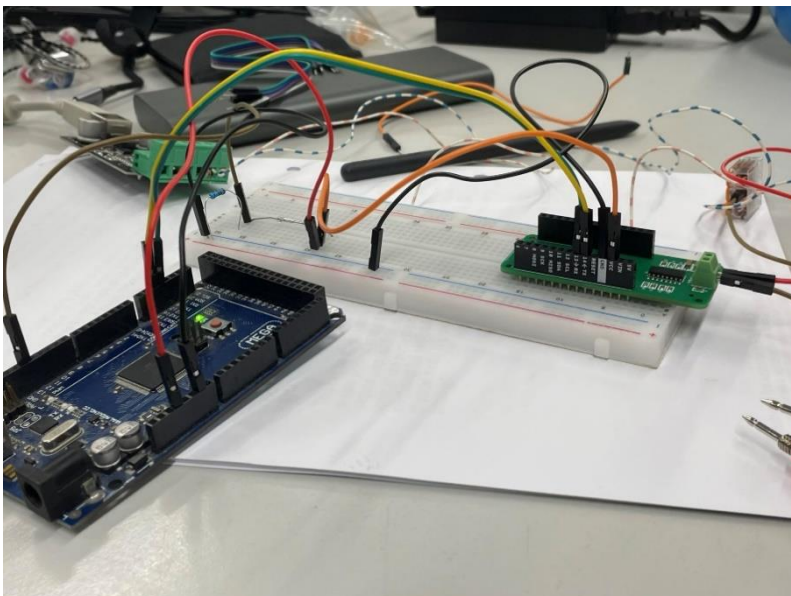


Abbildung 2: Aufbau der Schaltung

7.3 Ausgabe in MBSheet

M-Bus Sheet V2.2 Demo-Version

Einbauort	Adresse	ID	Baud	Herst.	Version	Medium	Datensätze	COM port
	1	50303023	9600	AEG	1	Electricity	1,2,3,4	COM6

Zähler: 1 COM-port: COM6
 Such-Baudrate(n): 9600
 GSM-Modem (BN1)
 SMD_NKE Suche Δdr. Stop
 Multi Tel. Suche ID
 App. Res.
 Zeitintervall [s]: 5 M-Bus Lesen Modem
 M-Bus Follen
 Status: idle M-Bus Loggen IDS+
 Schreibe Daten MDK
 Opto

Drucke Zählerliste Sichere Zählerliste Lade Zählerliste Duplikate löschen CSV Trennz. Zusätzl. Timeout [ms]: 1000

Einbauort	Datum	Zeit	Adresse	ID-Nr.	Herst.	Nr.	Wert	Einheit	Beschreibung	Art	Modul	SP-Nr.	Tarif
	13.06.24	17:15:53	1	50303023	AEG	1	9600		baudrate	instant.	0	0	0
	13.06.24	17:15:53	1	50303023	AEG	2	25		firmware version #	instant.	0	0	0
	13.06.24	17:15:53	1	50303023	AEG	3	\$00		digital input	instant.	0	0	0
	13.06.24	17:15:53	1	50303023	AEG	4	\$00		M-Bus Status				

Drucke Daten Sichere Daten Lade Daten Lösche Daten Import Registrieren Zoom Daten

10 7B 1C 97 16
 RX-Puffer:
 Keine Antwort von Adresse 28

Lösche Log
 Exit

Abbildung 3: Auswertung per MBSheet, die zwei Möglichen Werte sind markiert