Modul-Design Auditing-System

Für das studentische Projekt *Sichere Eisenbahnsteuerung*

Datum	21.04.2010
Quelle	Aulis: Design → Moduldesign
Autoren	Icken, Jan-Christopher
Version	0.2
Status	freigegeben

1 Historie

Version	Datum	Autor	Bemerkung
0.1	14.04.10	Icken, Jan-Christopher	Initiale Version
0.2	21.04.10	Icken, Jan-Christopher	Kapitel 5 Architektur überarbeitet

2 Inhaltsverzeichnis

3 Einleitung

Dieses Dokument dient zur Dokumentation der Erweiterung des bereits im WiSe09/10 entwickelten Moduls Auditing-System. Die Erweiterung soll es ermöglichen, die über den I2C-Bus übermittelten kodierten Nachrichten in eine für Menschen einfach zu lesende Form zu übersetzen. Zu diesem Zweck, wird der auf dem Arduino laufende Teil des Auditing-Systems angepasst. Der auf dem C515C-Mikrocontroller laufende Teil des Auditing-Systems bleibt von diesen Modifikationen unberührt.

4 Referenzierte Dokumente

- F4 TI PROJEKT Bredereke WiSe0910 > Projektverzeichnis > 02: Design > 03: Moduldesign > Auditing-System-Modul-Design
- F4 TI PROJEKT Bredereke WiSe0910 > Projektverzeichnis > 02: Design > 03: Moduldesign > Leitzentrale-Modul-Design
- F4 TI PROJEKT Bredereke WiSe0910 > Projektverzeichnis > 02: Design > 03: Moduldesign > Befehlsvalidierung-Modul-Design
- F4 TI PROJEKT Bredereke WiSe0910 > Projektverzeichnis > 02: Design > 03: Moduldesign > Ergebnisvalidierung-Modul-Design

5 Architektur

In diesem Dokument werden nur die Änderungen am Moduldesign beschrieben, die die Implementierung der Software auf dem Arduino-Mikrocontroller betreffen.

5.1 Funktionshierarchie

Name	**setup()**
Beschreibung	Initialisierung des Arduinos als Slave auf dem i2c-Bus und Initialisierung der Seriellen schnittstelle mit einer bestimmten Baudrate. Außerdem festlegung
Vorbedingung	Arduino an Strom angeschlossen
Parameter	-
Rückgabe	-
Nachbedingung	Initialisierungen durchgeführt

Name	**loop()**
Beschreibung	In der Arduino IDE vorgesehene Funktion die dauerhaft ausgeführt wird. Wird in diesem Fall nicht benötigt muss jedoch vorhanden sein.
Vorbedingung	-
Parameter	-
Rückgabe	-
Nachbedingung	-

Name	**receiveEvent(int HowMany)**
Beschreibung	Legt fest wie auf Daten die über den i2c-Bus empfangen werden reagiert werden soll.
Vorbedingung	Daten über den i2c-Bus empfangen
Parameter	Int HowMany = Anzahl der empfangenen Bytes
Rückgabe	-

Nachbedingung	Alle empfangenen Daten wurden umgewandelt und wurden über die serielle Schnittstelle ausgegeben

Name	**convert_message(int message_array[])**
Beschreibung	Diese Funktion nimmt das aus 7 Byte bestehende Array einer Nachricht an, prüft das erste Byte und ruft die jeweilige für die Umwandlung zuständige Funktion auf.
Vorbedingung	Funktion wurde von receiveEvent() aufgerufen
Parameter	Int message_array[7] = Aus 7 Byte bestehende Auditing Meldung
Rückgabe	-
Nachbedingung	-

Name	**convert_message_lz(int message_array[])**
Beschreibung	Wandelt Nachrichten der Leitzentrale in Klartext um und gibt diese auf der seriellen Schnittstelle aus.
Vorbedingung	Funktion wurde von convert_message() aufgerufen
Parameter	Int message_array[7] = Aus 7 Byte bestehende Auditing Meldung
Rückgabe	-
Nachbedingung	-

Name	**convert_message_bv(int message_array[])**
Beschreibung	Wandelt Nachrichten der Befehlsvalidierung in Klartext um und gibt diese auf der seriellen Schnittstelle aus.

Vorbedingung	Funktion wurde von convert_message() aufgerufen
Parameter	Int message_array[7] = Aus 7 Byte bestehende Auditing Meldung
Rückgabe	-
Nachbedingung	-

Name	**convert_message_ev(int message_array[])**
Beschreibung	Wandelt Nachrichten der Ergebnisvalidierung in Klartext um und gibt diese auf der seriellen Schnittstelle aus.
Vorbedingung	Funktion wurde von convert_message() aufgerufen
Parameter	Int message_array[7] = Aus 7 Byte bestehende Auditing Meldung
Rückgabe	-
Nachbedingung	-

5.2 Daten

Die oben genannten Funktionen benötigen folgende Variablen

Name	**Verwendung**
Int message_array[7]	Eine Auditing über den i2c-Bus empfangene Meldung bestehend aus 7 Byte. Bedeutung der 7 Byte ist den jeweiligen Moduldesign Dokumenten zu entnehmen: • Auditing-System-Modul-Design Kapitel 7.2 • Leitzentrale-Modul-Design Kapitel 5.5.3 (Auditing-System) • Befehlsvalidierung-Modul-Design Kapitel 5.6.3 (Auditing-System)

	• Ergebnisvalidierung-Modul-Design Kapitel 5.6 (Nachrichtenkodierung)

5.3 Abhängigkeiten von anderen Modulen

Abhängigkeiten siehe Modul-Design-Auditing-System-Dokument aus dem WiSe09/10

5.4 Schnittstellenbeschreibung

5.4.1 Interne Schnittstellenbeschreibung

Als Interne Schnittstelle dient die Verbindung über den i2c-Bus zwischen dem C515C und dem Arduino.

5.4.2 Externe Schnittstellen:

Als externe Schnittstelle dient ein auf dem an den Arduino angeschlossenen PC laufendes Terminal Programm wie z.B.

HyperTerminal

HTerm http://www.der-hammer.info/terminal/

Putty http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Dieses gibt die vom Arduino umgewandelten Auditing-Meldungen auf dem Bildschirm aus.

6 Dynamisches Verhalten

(TODO: Diagramm erstellen)

- Verbindungsanfrage auf modulinterner I2C-Schnittstelle (Handling über Wire-Library)
- Für 1 bis n Nachrichten
  - 7 Byte empfangen und in Array kopieren
- Verbindung über I2C-Schnittstelle von Gegenseite getrennt (Handling über Wire-Library)
- Array mit Nachrichten in Klartext umwandeln
  - Kodierung der Nachrichten siehe:
    - Auditing-System-Modul-Design Kapitel 7.2
    - Leitzentrale-Modul-Design Kapitel 5.5.3 (Auditing-System)
    - Befehlsvalidierung-Modul-Design Kapitel 5.6.3 (Auditing-System)
    - Ergebnisvalidierung-Modul-Design Kapitel 5.6 (Nachrichtenkodierung)
- Klartext über virtuelle serielle Schnittstelle des Arduino-Mikrocontrollers auf einem angeschlossenen PC mit Terminal Programm ausgeben

7 Anhang