

02.06.04

Nr.11 - LIN - Versuch

Praktikum Schnittstellen und Bussysteme



Vorwort:

Als Standard für die preisgünstige Vernetzung im Automobil hat sich der **LIN**- Subbus unterhalb von CAN etabliert. Die Industrielektronik beginnt sich für den **LIN** - Standard zu interessieren, verbindet der **LIN**- Subbus doch die Robustheitsanforderung im Automobil mit kostengünstigen Lösungen.

Der folgende Laborversuch soll nun die **LIN – BUS** Kenntnisse vertiefen.

Die zentralen Schwerpunkte dieses Laborpraktikums sind umfangreiche Funktionen in den Bereichen Test und Diagnose, die zusammen mit den laboreigenen Steuergeräten MC 68HC11 und den Master – Slave – Knoten als Hardware – Schnittstellen durchgeführt werden.

Die Diagnose kann durch die Beobachtung der Master - LCD – Analysefenster und dem PC – Softwarefenster erstellt werden.

Inhaltsverzeichnis

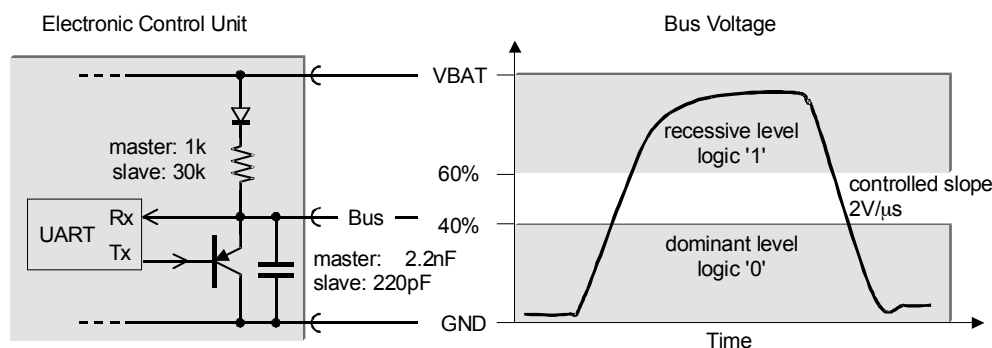
Grundlagen	3
Der LIN - BUS - Nachrichtenrahmen.....	4
Die 8 Bit Wortlängen - Tabelle.....	5
Von der Checksummenberechnung zum Checkbyte	6
Berechnung der Paritätsbits P0,P1	7
Übungsblatt zum Identifier - Feld	7
Das 256 Byte große Feld	8
Frontansicht des 68HC11 Minicomputer	9
Der Versuchsaufbau	10
Jumper Positionen der Schnittstellen - Karten	11
Bestückung der Schnittstellen - Karten	12
Knotenbeschreibungen (Master - Slave).....	13
Schaltplan	14

.....Messungen im Labo r
ab Seite 15

02.06.04

1. Grundlagen

Als neuer Standard für preisgünstige Vernetzung im Automobilbereich und im Industriebereich wurde im Jahre 2000 /2001 **LIN** (Local-Interconnect- Network) als Subbus zu CAN etabliert. Der LIN- Bus kann einen Entfernungsbereich von 40m abdecken, wobei die Daten bidirektional im seriellen UART- Format auf einem Draht übertragen werden. Die wenigen Berichte über LIN sind häufig fehlerhaft von einander abgeschrieben. Dieser Laborversuch soll alle Fragen über LIN beantworten.



Vorgesehen ist in diesem **Laborversuch** die Vernetzung einfacher Sensoren und Aktoren. Dies sind zum Beispiel : (kleine Elektromotoren, Leuchten und Temperaturfühler). Als Übertragungsraten sind 2400,9600,und 19200 Baud für die Praxis am wichtigsten. Die typische Spannungsversorgung liegt zwischen 8V-18V, in unserem Versuch bei 12 V und GND

Der LIN- Bus arbeitet nach dem Master- Slave- Konzept, dies bedeutet es gibt einen Master-Knoten der die Datenübertragung koordiniert, wobei die Slaves auf die gesendeten Nachrichten reagieren .

Die Slaves sind in der Lage, die Geschwindigkeit des Masters zu ermitteln und sich darauf zu synchronisieren.

Die Nachrichtenrahmen bestehen aus einer festen Abfolge von Datenbits, der **Header – Message** und der **Response Message** und ergeben damit den **LIN- BUS- Nachrichtenrahmen**. (siehe Seite 3)

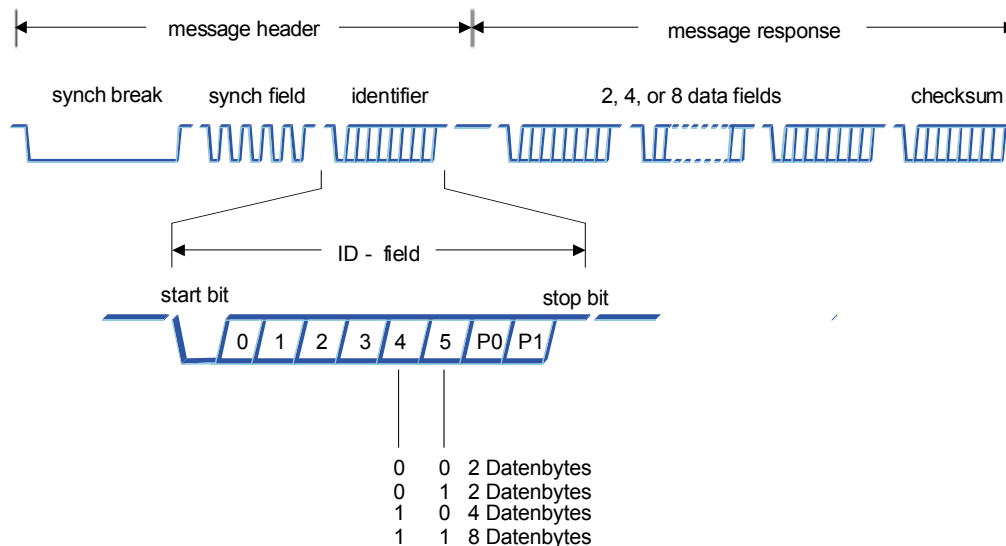
Die **Header – Message** besteht aus einem **Synchronisationsbreak**, einem **Synchronisationsfeld** und dem **Identifizierfeld**.

Die **Response- Message** (Antwort) besteht aus einer variablen Anzahl von zwei , vier oder acht **Datenbytes** und einem **Checkbyte**. Das Response –Feld kann eine Nachrichtenübertragung vom Master zum Slave oder vom Slave zum Master enthalten.

02.06.04

(siehe das Beispiel Seite 6)

Der LIN- BUS- Nachrichtenrahmen



Die Header - Message

Die Übertragung des Nachrichtenrahmens beginnt mit dem **Synchronisations-Break** , also einem Null- Pegel (GND dominanter Pegel von 13 Bitzeiten) und einem Eins- Pegel (rezessiver Pegel von bis zu 4 Bitzeiten)

einem darauf folgenden **Synchronisationsfeld** , eingeleitet von einem Startbit ,dem Wert **0x55** und einem Stopbit .Dieses Feld ermöglicht den Slaves die Ermittlung der Übertragungsrate des Masters.

Die **Header- Message** endet mit dem **Identifizierfeld** und bestimmt damit den weiteren Ablauf des Nachrichtenrahmens. Auch das **ID- Feld** wird, wie alle Bytefelder, von Start- und Stoppbit eingerahmt.

Nach dem **Startbit** folgen vier **Adressbits (ID0 bis ID3)** (also 16 Adressen) und zwei **Wortlängen- Bits (ID4 und ID5)** für die Anzahl der folgenden Datenbytes (also 2,4,8 Datenbytes) und zwei **Paritätsbits**.

P0 gerade Parität (also antivalenz) (aus 01 und 10 wird 1)

und **P1** ungerade Parität also äquivalenz) (aus 00 und 11 wird 1)

Die Paritätsbit berechnet sich wie folgt : (**P0** aus ID0,ID1.ID2,ID4) und (**P1** aus ID1,ID3,ID4.ID5)

(siehe das Beispiel Seite 7)

Von der 8 Bit verschlüsselten Wortlängen – Tabelle zum 256 Byte großen Darstellungsfeld.

Aus den **unteren 6 Bits (ID0 bis ID5)** ergibt sich also eine Adress-Wortlängen-Tabelle von **64 gültigen ID -Feldern**.

Nach der Berechnung der beiden zusätzlichen **Bits PO und P1** entsteht nun ein **Verschlüsselungsbyte von 8 Bit**, deren Werte nun im **256 Dec** Bereich liegen.

Beispiel:

Aus der 6 Bit Adresse **62 Dec** (Wortlänge von 8 Datenbytes) wird z.B. durch die Berechnung der zusätzlichen Bits P0 und P1 der verschlüsselte Wert **254 Dec**.

(Aus **62 Dec** = 0x3E Hex = **0011 1110** binär wird **1111 1110** = 0xFE = **254 Dec**)

Datenbits ID4 , ID5.

Die Bits ID4 und ID5 geben die Anzahl der folgenden Datenbytes an:

ID5,ID4 = **00 oder 01** => 16 + 16 = **32 gültige 2-Byte Datenbytes**,

ID5,ID4 = **10** => **16 gültige 4-Byte Datenbytes** und

ID5,ID4 = **11** => **16 gültige 8-Byte Datenbytes**.

Fazit:

Nur diese 64 verschlüsselten Adressen werden von den anderen Slave –Knoten als fehlerfrei erkannt. **Alle anderen 192** möglichen Bitkombinationen werden als Fehler erkannt.

Allein dieser Teil der aufwendigen Sicherung (es fehlt noch die Checksummen- Berechnung) der Übertragungsintegrität durch mehrfache Fehlererkennung lässt Anwendungen in problematischen Umgebungen zu.

Die komplette Tabelle finden Sie auf Seite 8

Das folgende **Response-Feld** wird generell als Antwort bezeichnet, auch wenn der Master-Knoten mit der Übertragung fortfährt..

In der Antwort werden zwei, vier oder acht Datenbytes versendet .

Die Anzahl der versendeten Datenbytes ermittelt der empfangende Knoten aus zwei Bits (ID4 und ID5) des Identifiers.

Als letztes Byte der Antwort wird ein „**Checkbyte** „versendet.

Die **Checksummen-Berechnung** besteht aus einer sequentiellen Summation der übertragenen **Datenbytes**. Findet während der Addition ein Übertrag statt, so wird dieser zur niedrigsten Stelle des Ergebnisses hinzuaddiert. So wird verhindert, dass das Ergebnis die Länge von acht Bits überschreitet. Das **Checkbyte** ist die inverse Checksumme!

(siehe Beispiel Seite 6)

02.06.04

Von der Checksummenberechnung zum Checkbyte

4 Byte - Daten Beispiel

Data0 = 0x4A
Data1 = 0x55
Data2 = 0x93
Data3 = 0xE5

	hex	CY	D7	D6	D5	D4	D3	D2D	1D	0
0x4A	0x4A		0	1	0	0	1	0	1	0
+0x55 =	0x9F	0	1	0	0	1	1	1	1	1
(Add Carry)	0x9F		1	0	0	1	1	1	1	1
+0x93 =	0x132	1	0	0	1	1	0	0	1	0
Add Carry	0x33		0	0	1	1	0	0	1	1
+0xE5 =	0x118	1	0	0	0	1	1	0	0	0
Add Carry	0x19		0	0	0	1	1	0	0	1
Invert	0xE6		1	1	1	0	0	1	1	0
0x19+0xE6 =	0xFF		1	1	1	1	1	1	1	1

Checksumme = 0x19

Als **letztes Byte** der Nachricht wird das **Checkbyte E6 Hex** versendet.

Der Empfänger der Nachricht empfängt das **Checkbyte**.

Das Checkbyte ist die invertierte Checksumme (**im obigen Beispiel E 6**)

Somit muss sich, bei einer einfachen Addition von Checksumme und Checkbyte immer ein Wert von **0xFF** ergeben. Dies wird auch immer überprüft.

02.06.04

Übungsblatt zum Identifier - Feld

Das Paritätsbit P0 (ID6)

entsteht durch die **antivalente Verknüpfung** von ID0, ID1, ID2 und ID4,

Das Paritätsbit P1 (ID7)

entsteht aus der **äquivalenten Verknüpfung** von ID1, ID3, ID4 und ID5,

Antivalenz

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Äquivalenz

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

Ermitteln Sie P0, P1, das vollständige Identifier – Feld und die Anzahl der Daten-Bytes.
Die Lösungen tragen Sie in die untere Tabelle ein.

ID (0..5)		P1	P0	Identifier - Feld								IP - Feld		Daten
dez	hex			7	6	5	4	3	2	1	0	dez	hex	Bytes
35	23	1	0	1	0	1	0	0	0	1	1	163	A3	4
01														
26														
55														
61														

02.06.04

Das 256 Byte große „IP - Feld“.

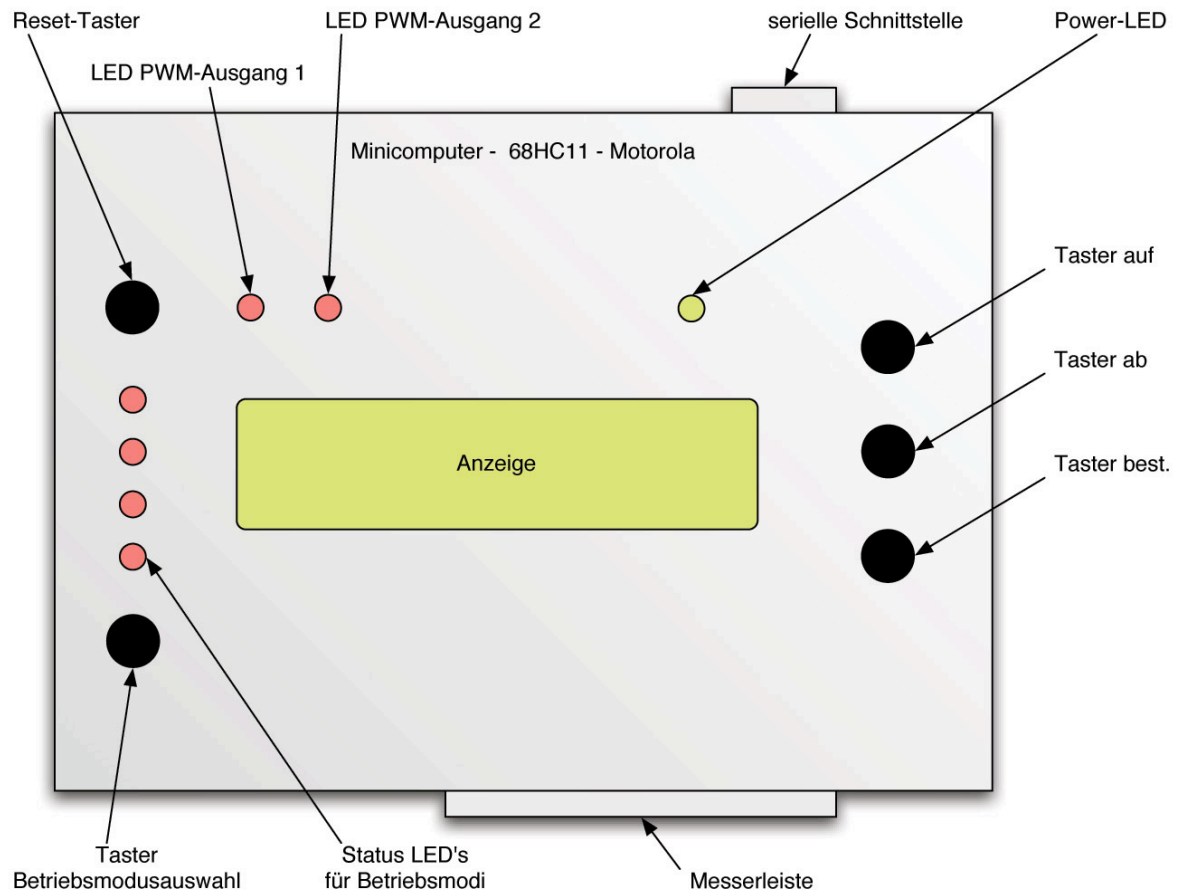
Die untere Tabelle zeigt alle **64 verschlüsselten gültigen Identifier „IP“** mit der dazugehörigen Anzahl der Datenbytes. Die restlichen 192 Datenbytes werden als **Fehler** erkannt.

Identifier IP		Low-Nibble															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
High-Nibble	0	-	-	-	2	-	-	2	-	2	-	-	-	-	2	-	-
	1	-	2	-	-	2	-	-	-	-	-	2	-	-	-	-	2
	2	4	-	-	-	-	4	-	-	-	-	-	4	-	-	4	-
	3	-	-	8	-	-	-	-	8	-	8	-	-	8	-	-	-
	4	-	-	2	-	-	-	-	2	-	2	-	-	2	-	-	-
	5	2	-	-	-	-	2	-	-	-	-	-	2	-	-	2	-
	6	-	4	-	-	4	-	-	-	-	-	4	-	-	-	-	4
	7	-	-	-	8	-	-	8	-	8	-	-	-	-	8	-	-
	8	2	-	-	-	-	2	-	-	-	-	-	2	-	-	2	-
	9	-	-	2	-	-	-	-	2	-	2	-	-	2	-	-	-
	A	-	-	-	4	-	-	4	-	4	-	-	-	-	4	-	-
	B	-	8	-	-	8	-	-	-	-	-	8	-	-	-	-	8
	C	-	2	-	-	2	-	-	-	-	-	2	-	-	-	-	2
	D	-	-	-	2	-	-	2	-	2	-	-	-	-	2	-	-
	E	-	-	4	-	-	-	-	4	-	4	-	-	4	-	-	-
	F	8	-	-	-	-	8	-	-	-	-	-	8	-	-	8	-

4 Felder sind reserviert. z.B.: (**ID = 3C = IP** ist die Sleepnachricht.)

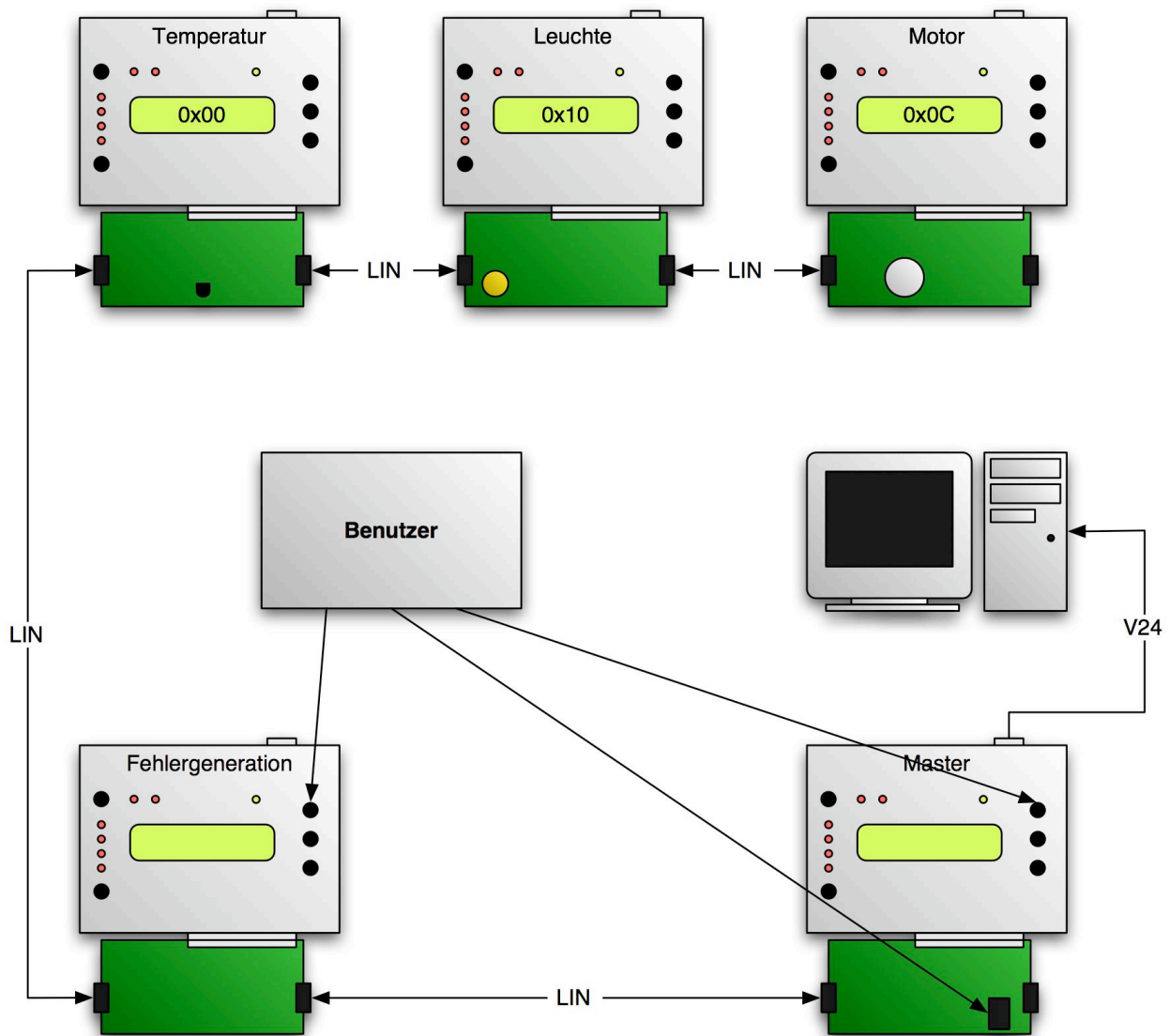
02.06.04

Mikrocomputersystem 68HC11



02.06.04

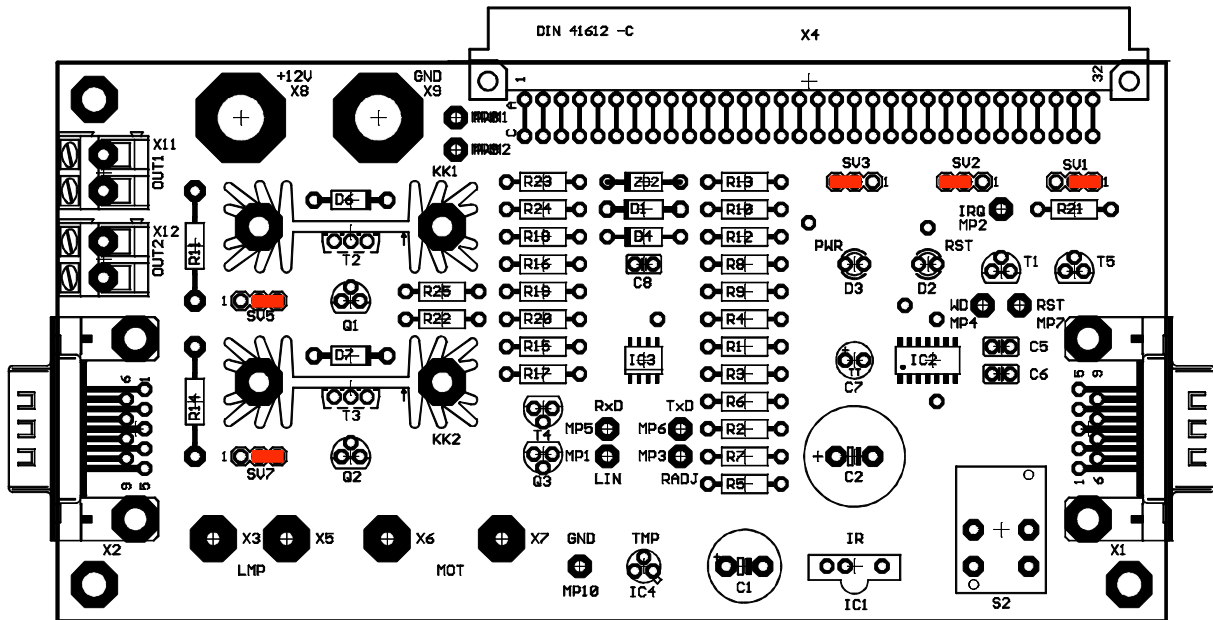
Der Versuchsaufbau



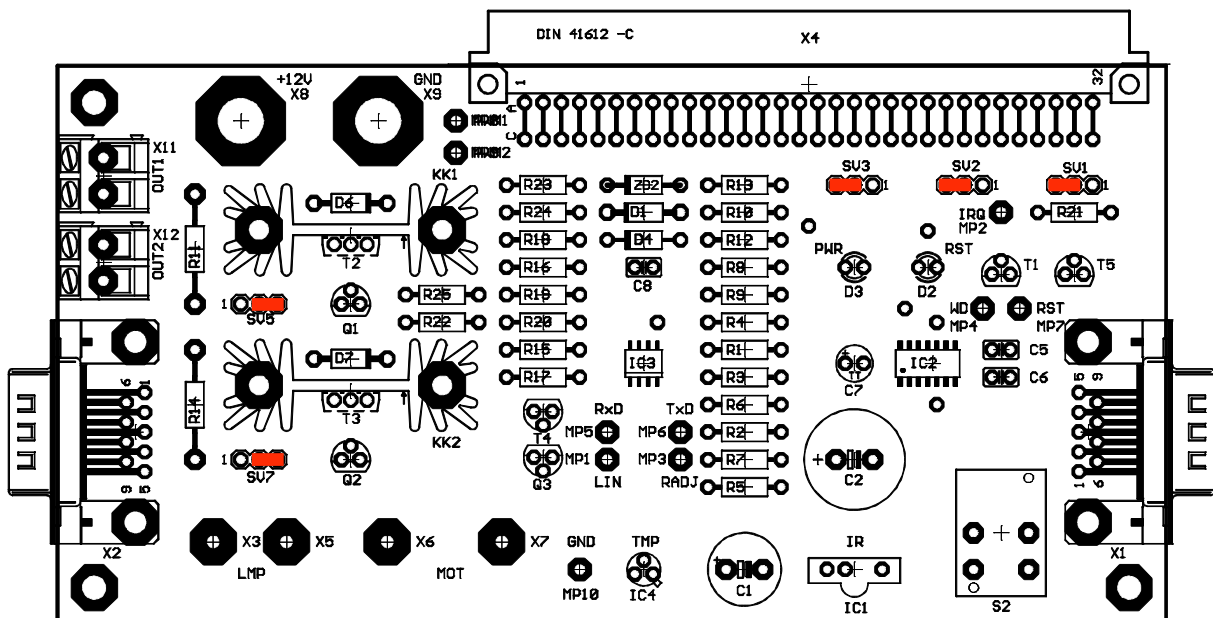
Stecken Sie die Schnittstellen-Karten auf die Mikrocomputersysteme.
Überprüfen Sie, ob die Jumper der Schnittstellenkarten richtig gesteckt sind.

02.06.04

Jumperpositionen Master- und Fehlergenerationsknoten



Jumperpositionen der Slave-Knoten

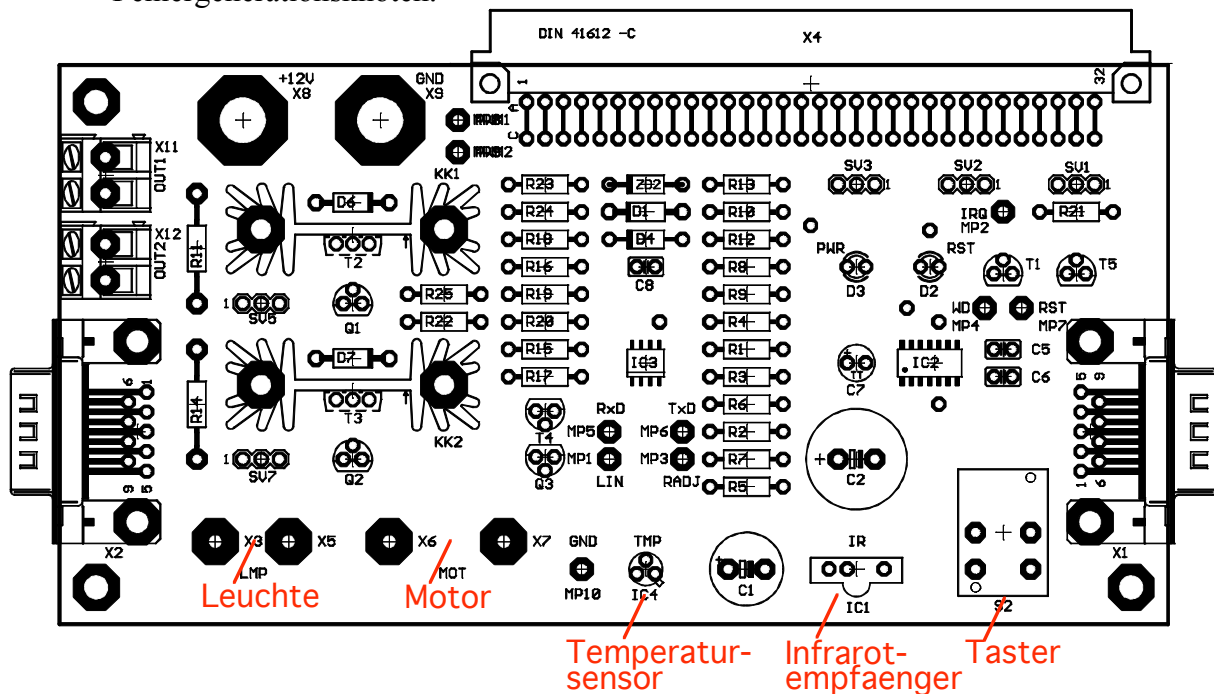


02.06.04

Bestückung der Schnittstellenkarten

Identifizieren Sie die einzelnen Schnittstellenkarten anhand der Bestückung. Es gibt fünf Knoten:

Master-Knoten
 Slave-Knoten „Leuchte“
 Slave-Knoten „Motor“
 Slave-Knoten „Temperatur“
 Fehlergenerationsknoten.



Auf dem Master-Knoten ist ein Infrarotempfänger (wird nicht benutzt) und eine **Weck - Taste** eingebaut.

(Nur mit dieser Taste kann das gesamte System aus dem „Sleepmodus“ geweckt werden!)

Auf den Slave- Knoten sind die jeweiligen Funktionsbauteile bestückt, also zum Beispiel auf dem Temperatur-Knoten der Temperatursensor u.s.w.

Auf dem Fehlergenerationsknoten fehlen sämtliche Knoten- Bauteile. (siehe Seite 10)

Knotenbeschreibungen:

Master-Knoten

Am Minikomputer des Master-Knoten sind über eine Menüsteuerung sämtliche Netzwerkfunktionen einstellbar. Dies sind die Einstellungen für die einzelnen Knoten, sowie eine Lüftersteuerung und die Anzeige der Register. Die einzelnen Menüpunkte sind mit den Tasten „Auf“, „Ab“ und „Bestätigen“ auswählbar (siehe Seite

Ist das System im „Sleepmodus“ eingestellt, so leuchtet die **rote Led**, die grüne Led ist dunkel. Die LCD – Anzeige aller beteiligten Knoten zeigt den Text „Standby“ an und alle Tasten am Minikomputer sind unwirksam. Das System kann nur noch mit der „**Wecktaste**“, am Masterknoten durch einen Weckimpuls geweckt werden

.

Slave-Knoten „Leuchte“

Der Slave-Knoten „Leuchte“ empfängt Daten vom Master-Knoten und stellt die Helligkeit der Leuchte entsprechend dem empfangenen Wert ein. Der Wert wird prozentual auf der Anzeige ausgegeben. Optional lassen sich bei gedrückter Taste „Bestätigen“ auch die übertragenen Daten (ID-B1-B2-CB-IP) in hexadezimaler Darstellung anzeigen.

Slave-Knoten „Motor“

Der Slave-Knoten „Motor“ hat im Wesentlichen die gleiche Funktionalität wie der Slave-Knoten „Leuchte“, die Einstellungen werden nur für den Motor vorgenommen.

Slave-Knoten „Temperatur“

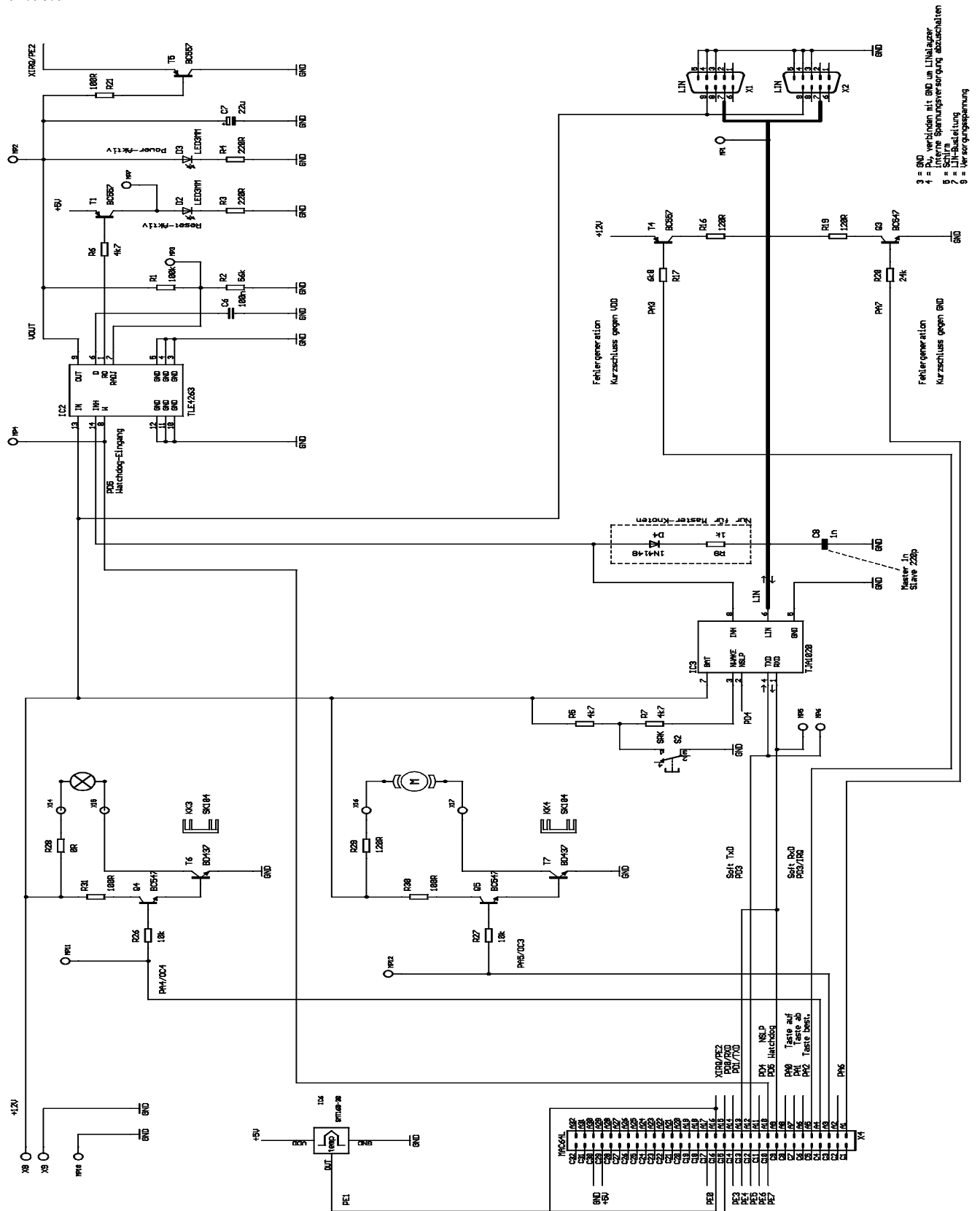
Der Slave-Knoten „Temperatur“ empfängt eine Temperaturanfrage und schickt daraufhin die aktuell ermittelte Temperatur zurück. Die Daten lassen sich wiederum in hexadezimaler Darstellung anzeigen. (Durch Drücken der „Bestätigen“ Taste.) So wird z.B. der Temperaturwert durch das Hex- Byte **B1** dargestellt. (**1C Hex** bedeutet 16 und 12 = **28grd**)

Fehlergenerations-Knoten

Mit dem Fehlergenerationsknoten lassen sich Kurzschlüsse auf der LIN - Busleitung erzeugen. Diese sind wahlweise gegen GND oder 12V Pegel einstellbar. Es lassen sich dauerhafte Kurzschlüsse generieren, wie auch Kurzschlüsse einzelner Bits in den verschiedenen Nachrichten-Bytes.

Siehe Schaltplan Seite 14

02.06.04



Messungen im Labor



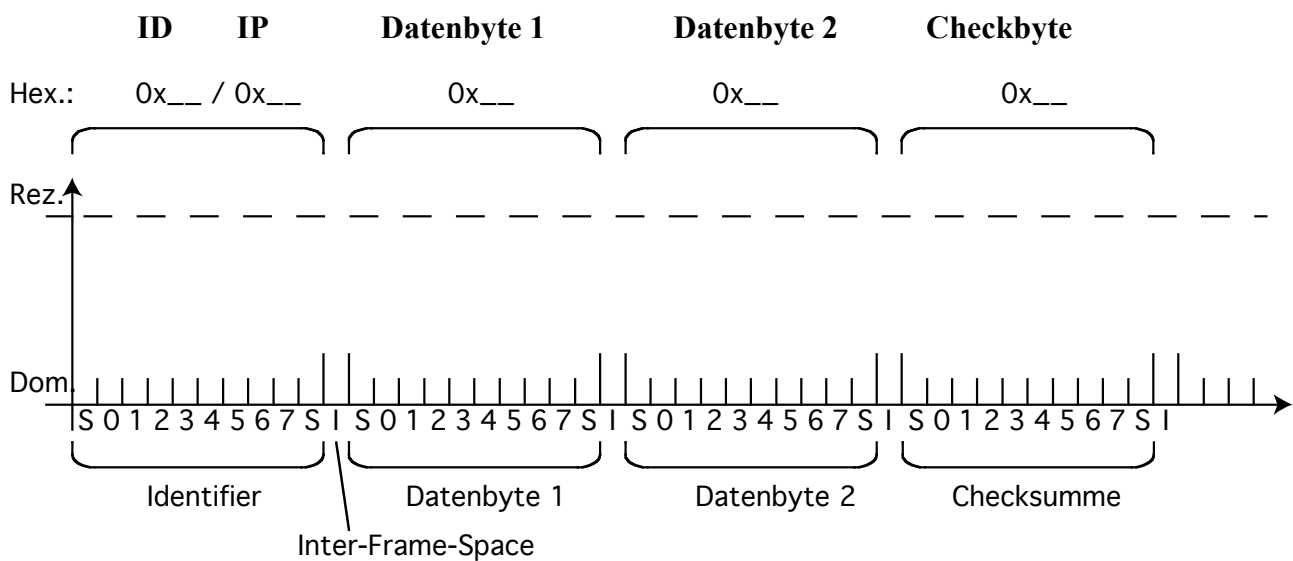
Der - LIN - BUS- Nachrichtenrahmen

Arbeitsblatt	Zeitdiagramm des seriellen Bitstromes	16
Programmübersicht im PC		17
Softwarebeschreibung		18
Versuch 1.....	Der "Slave - Knoten - Motor"	19 20
Versuch 2.....	Der "Slave - Knoten - Leuchte "	21
Versuch3.....	Der "Slave - Knoten - Temperatur "	22
Versuch 4.....	" Lüfterregelung "	23
Versuch 5.....	" Sleep – Mode "	24
Versuch 6.....	" Fehlergeneration "	25

02.06.04

Zeitdiagramm des seriellen Bitstromes.

Der LIN – Bus - Nachrichtenrahmen



Dom.: Dominanter Pegel (0V)
 Rez.: Rezessiver Pegel (+12V)
 Hex.: Bytes in hexadezimaler Darstellung
 S.: Start- bzw. Stop-Bits
 0...7.: Bits 0 bis 7

Datenbyte 1HexDez.

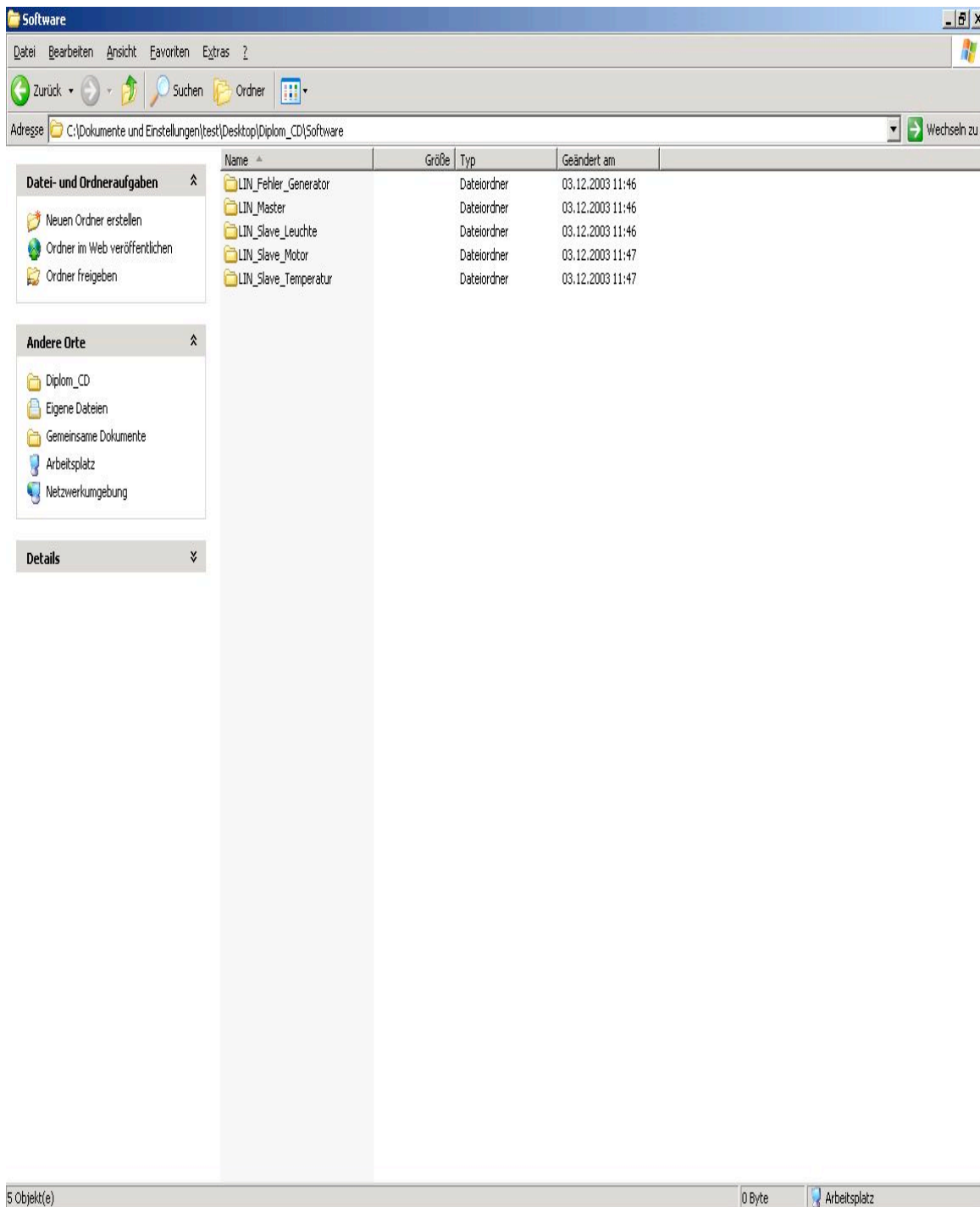
Datenbyte 2Hex

ChecksummeHex

CheckbyteHex

IPHex

02.06.04



Softwarebeschreibung

Laden der einzelnen Softwarepakete vom PC auf die entsprechenden Mikrocomputersysteme. Die einzelnen Softwarepakete liegen vorcompiliert im Motorola .S19-Format vor und liegen im Verzeichnis / Software / LIN _ Master, / Software / LIN _ Slave _ Leuchte, usw.

Laden Sie die Softwarepakete mit dem Download-Tool des Compilers „ICC11“ auf die einzelnen Minicomputersysteme. Dazu verbinden sie das V24 Übertragungskabel einerseits mit einem freien COM -Port des PC's und andererseits mit der seriellen Schnittstelle des Minicomputersystems.

Betriebsart „BS „ Bootstrap Programm vom PC laden

Betriebsart „EX „ Extended Reset = Programmstart.

Das Mastermenü:

Am Master – Knoten sind über eine Menüsteuerung alle 7 Netzwerkfunktionen einstellbar.

Die folgenden Funktionen sind mit den Tasten „AUF AB „ anwählbar. Mit der Taste

„Bestätigen „ wird das Programm gestartet.

Durch nochmaliges Drücken der Taste „Bestätigen „ kommt man wieder in das Menü zurück.

- 1 Slave 1 **Leuchte** ID = 10
- 2 Slave 2 **Motor** ID = 0C
- 3 Slave 3 **Temperaturfühler** ID = 00
- 4 Demonstration einer **Lüfter-Regelung** (Motor , Temperatur und Leuchte)
- 5 **Sleep – Modus** auswählen.
- 6 **Debug – Menü / S/E Register.** Das LCD – Fenster zeigt alle wichtigen LIN – BUS Daten an.
ID = Identifier , B1,B2 = Datenbytes , CB = Checkbyte, IP = Identifier.
- 7 Anzeige des **Fehlerregisters.** (maximaler Fehlerzähler ist **08**)

Der Masterknoten :

Der Minicomputer des Masters hält ständig die softwaremäßige Verbindung über die V24-Leitung zum Programm des PCs aufrecht . Dies bedeutet ,dass der Master immer seine momentanen Messwerte im PC -Fenster auflistet .Man kann damit jede Aktivität des LIN-Busses in Echtzeit beobachten .Dazu die folgenden Beispiele :

02.06.04

Die 6 Menüpunkte des Masters :

- 1 Einstellen der Helligkeit : ok**
Der Master liest die eingestellte Helligkeit der Leuchte aus. **IP DB1 DB2 CB**
50 00 00 FF für 0%
50 E6 00 19 für 11%
- 2 Einstellen der Drehzahl : ok**
Der Master liest die eingestellte Umdrehungszahl des Motors aus.
4C 00 00 FF für 0%
4C E6 00 19 für 11%
- 3 Auslesen der Temperatur : ok**
Der Master liest die Temperatur des Fühlers aus.
80 16 00 E9 für 22°C
- 4 Demonstration Lüfterregelung ok**
Der Master liest alle beteiligten Messdaten aus. Stopp mit ok.
50 00 00 FF Leuchte
4C D0 02 2D Motor
80 16 00 E9 Temperatur
- 5 Auslösen des Sleep-Modus ok Die rote Led leuchtet**
Der Master liest seine Kennbytes aus.
3C 00 00 00 00 00 00 00 00 FF
Alle Minicomputer sind im Stand by Modus der Fehlergenerator aber nicht.

02.06.04

Aufgabe 1 : Der Slave – Knoten – „Motor“



1. **Aufbau.**
2. **Zubehör :** Master-Knoten ,Slave –Knoten -Motor, Speicheroszilloskop , Netzteil und PC.
3. **Programmübertragungen vom PC**
4. **Aufgabenstellungen und Messungen**
5. **Auswertungen und Fragen**

- 1.1 **Verbinden** Sie den PC mit der seriellen Leitung zum Master.
- 1.2 **Verbinden** Sie den Master-Knoten und den Slave- Knoten „Motor“ mittels einer LIN -Bus – Verbindungsleitung.
- 1.3 **Versorgen** Sie den Master Knoten mit einer 12V Verbindungsleitung zum Netzteil. Alle anderen Knoten werden dann über die LIN –Busleitung mit 12V versorgt.
- 1.4 **Klemmen** Sie den Tastkopf des Speicheroszilloskops (Typ 2212 von Tektronix) an den „LIN – Messpunkt“ auf einer der beiden Schnittstellenkarten.
- 3.1 **Übertragen** Sie das Programm „LIN_Master“ vom PC zum Minikomputer des „Master –Knoten“. (Hinweis : “BS - Mode“...Seite 18...)
- 3.2 **Übertragen** Sie das Programm “LIN _ Slave _ Motor“ zum Minikomputer des „Slave.

02.06.04

4 Messungen Motor :

- 4.1 Die Minikomputer sind in der Betriebsart „EX-Mode“ und der Motor ist über das Menü des Minikomputers auf eine Umdrehungsgeschwindigkeit von **10%** eingestellt. Der PC empfängt serielle Daten.
- 4.2 Mit dem **Speicheroszilloskop** soll nun der gesamte serielle Bitstrom aufgezeichnet werden. Dieser Bitstrom ist in das **Arbeitsblatt** Seite 16 einzutragen.
Hinweis:.....

5. Auswertungen und Fragen.

Sie haben nun den gesamten **LIN – BUS – Nachrichtenrahmen** binär aufgezeichnet.

- 5.1. **Übertragen** Sie die seriellen Daten auch im **Hex-Format** in das Arbeitsblatt und **interpretieren** Sie den gesamten Datenstrom und **beantworten** Sie folgende Fragen:

ID0C. Hex,	IP	Hex
Anzahl der Datenbytes		
Checksumme		Hex
Checkbyte 10%		Hex

- 5.2 **Berechnen** Sie das Identifizierungsfeld **IP** (mit P0,P1), die Checksumme und das Checkbyte aus den obigen Messwerten und vergleichen Sie die Werte mit Ihrer Messung!

- 5.3 **Unterbrechen** Sie die LIN-BUS Verbindungsleitung und beobachten Sie den Datenstrom.

- 5.4 Der Slave Knoten **Motor** ist ein Sensor Aktor.

Weil das Datenbyte 1.....

- 5.5 **Erhöhen** Sie die Umdrehungsgeschwindigkeit auf **30 %** und lesen Sie das S/E Register des Minikomputers aus.

- 5.6 Zeichnen Sie den Bitstrom und **alle** Hex Daten in das Arbeitsblatt Seite 16 ein.

- 5.7 Überprüfen Sie das Checkbyte durch eigene Berechnung ! (siehe Seite 8)

Debug – Menü / SE Register 30%

.ID.....B1.....B2.....CB.....IP.....

Bemerkungen:

02.06.04

Aufgabe 2 : Der Slave – Knoten – „Leuchte“

1. Aufbau.
2. Zubehör : Master-Knoten , Slave – Knoten - Motor, Speicheroszilloskop , Netzteil und PC.
5. Programmübertragungen vom PC
6. Aufgabenstellungen und Messungen
5. Auswertungen und Fragen

- 1.1 Verbinden Sie nun den Master-Knoten und den Slave - Knoten „Leuchte“ mittels einer LIN- Bus Verbindungsleitung

4 Messungen Leuchte :

Führen Sie die gleichen Messungen wie unter der Aufgabe 1 mit **10%** Helligkeit durch.

- 5.1 Übertragen Sie die seriellen Daten auch im **Hex-Format** in das Arbeitsblatt (Seite 16) und interpretieren Sie den gesamten Datenstrom und beantworten Sie folgende Fragen:

ID = 10 Hex IP Hex,

Anzahl der Datenbytes

Checksumme Hex Checkbyte Hex

- 5.2 ja 5.3 ja

- 5.4 Der Slave Knoten Leuchte ist ein Sensor Aktor

Weil das Datenbyte 1.....

- 5.5 Erhöhen Sie die Helligkeit auf **50%** und lesen Sie das S/E Register aus!

- 5.6 ja 5.7 ja

Debug – Menü / SE Register 50% Helligkeit

.ID.....B1.....B2.....CB.....IP.....

Bemerkungen:

02.06.04

Aufgabe 3 : Der Slave – Knoten – „Temperatur „

1.1 Verbinden Sie nun den Master – Knoten und den Slave Knoten „Temperatur“ mittels einer LIN - Busleitung.

5 Messungen Temperatur :

Führen Sie die gleichen Messungen wie unter der Aufgabe 1 bei Raumtemperatur durch.

5.1 Übertragen Sie die seriellen Daten auch im **Hex-Format** in das Arbeitsblatt (Seite 16)und **interpretieren** Sie den gesamten Datenstrom und **beantworten** Sie folgende Fragen:

ID = 00 Hex **IP** **Hex,**
Anzahl der Datenbytes
Checksumme **Hex** **Checkbyte** **Hex**

5.2 Berechnen Sie das Identifizierfeld **IP** (mit P0,P1) ,die Checksumme und das Checkbyte aus den obigen Messwerten und vergleichen Sie die Werte mit Ihrer Messung!

5.3 Unterbrechen Sie die LIN-BUS Verbindungsleitung und beobachten Sie den Datenstrom.

Bemerkungen:.....

5.4 Der Slave Knoten Temperatur ist ein Sensor Aktor .,

weil das Datenbyte 1

.....

5.5 Verändern Sie die Temperatur für eine neue Messung!

Debug – Menü / SE Register Temp.neu

.ID.....B1.....B2.....CB.....IP.....

Bemerkungen:

02.06.04

Aufgabe 4 : „Lüfterregelung mit allen bisherigen Knoten „

Verbinden Sie nun den Master – Knoten und die Slave - Knoten „Temperatur“, „Leuchte“ und „Motor „ mittels einer LIN – Busleitung.

Übertragen Sie die Programme zu allen Minicomputer 68 HC11 und beobachten Sie nun den seriellen Datenaustausch zum PC , der im Terminal – Fenster ON - Line zu sehen ist. Erklären Sie die Abläufe.

Temperatur.....Motor.....

Leuchte.....

.....

Hinweise:

Durch das Halten der Return – Taste des Sklaven kann sein Messwert gelesen werden. .
Das Programm kann durch die Return – Taste des Masters gestoppt werden .
Im S/E Register steht die letzte Messwertanalyse des Regelkreises .

Programmablauf der Lüfterregelung

IP.....

IP.....

IP.....

Wie reagiert der Regelkreis auf Temperaturänderung ?

Bemerkungen.

.....

.....

.....

02.06.04

Aufgabe 5 : „Menüpunkt Sleep - Mode „

Führen Sie den Sleep – Mode durch und interpretieren sie die Sleep - Nachrichtenanalyse .

SleepnachrichtID.....IP.....

Bemerkungen:

Alle Knoten sind nun.....

Led rot

Led grün

Die Sleepnachricht kann nur.....

.....

IP = IP = weil.....

Checkbyte = weil.....

Die Aufwecktaste bewirkt.....

.....

.....

02.06.04

Aufgabe 6 : „ Fehlergeneration „

Fehlerregister: **BE PB CS ID SR**

BE: Bit Error **PB:** Physik Bus Error **CS:** Checksummen Error **ID:** Identifier Error
SR: Slave not respondet

Über das Fehlermenü der Fehlerkarte können nun alle Bits an beliebigen Stellen des LIN-BUS-Datenstromes verfälscht werden.

Gehen Sie alle Fehler der Reihe nach durch und beobachten Sie die Fehlermeldungen der Fehlerregister und die seriellen Daten zum PC.
Gleichzeitig können die Sie am Speicheroszilloskop die Bit-Fehler sehen.

Erklären Sie den Adressfehler und den Datenfehler.

Bemerkungen:

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ende

