Kommunikationsmodell - NEUENTWURF

PROJEKT: PHOENIX

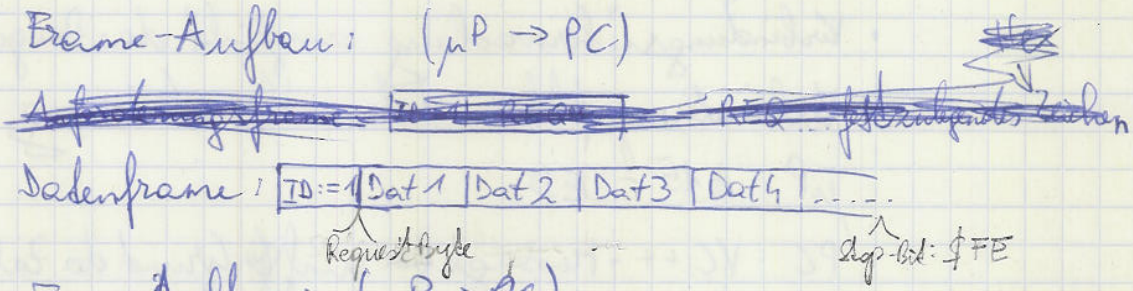
PROTOKOLL: WP2000

- Auftragsvergabe: Unterprogramm, welches aus verschiedenen Punkten im Programm aufgerufen wird (ev. INT 1 ← Serv. Routine; wenn INT 1 per Software auslösbar ist!)

  Auftragsnummer: 1 .... Telegramm mit Phone-Cabrid-Infos
  2 .... Statusabfrage für Live-Signal

Bei Änderung einer Phone-Cabrid-Information wird Auftrag 1 vergeben; sonst wird Auftrag 2 vergeben (Diese Auftragsvergabe erfolgt an allen Stellen im Programm an der Information von der Hardware erhalten werden)

Auftrag 1: Frame-Aufbau: ($\mu P \rightarrow PC$)



Auftrag 2: Frame-Aufbau: ($\mu P \rightarrow PC$)

Statusframe: ID=2 | Zahl: 39

39 ... Status-Dezimal wird ($\mu P \rightarrow PC$)

bei Auftrag 1:

~~Für jeden Auftrag gilt~~: (außer nach ID's. "generelles")

wird ein nach jedem übertragenen Byte vom PC gewartet (ACK-Abfrage mit 65ms-Timeout versehen; bei Überschreiten des Time-Outs wird Kommunikation abgebrochen) (Anzeige ev. über A15-Portpin ≙ Verbindungs~~abbruch~~ fehler)

bei Auftrag 2:

nach dem die Zahl 39 übertragen wurde wird auf die Zahl 93 gewartet (Timeout: 65ms); kommt 93 wird 93 zum PC gesendet → Verb.-überwachung auf der PC-Seite; kommt nichts oder etwas anderes zurück → setzen von A15-Portpin

PHOENIX - POI / WP2000

Frame-Aufbau:

R0	R1	R2	R3	R4	R5	R6	R7
ID	Adr. Register	Stat. Register	Daten #0	Daten #1	Daten #2	Daten #3	CHK-SUM

ID: 1) D1H .... PIN-Transfer POI → POS ! s. Rückseite !
2) D2H .... Pickup-Nr-Transfer POS → POI
3) D3H .... Status-Transfer
3) A1H .... ALIVE-Abfrage POS → POI
2) A2H .... Pickup-Nr-Anfrage POI → POS

Adr.Register: dest.Adr, src.Adr, z.B.: | A | 0 |
POI-Adr.: A-F (POI | POS)
POS → Adr.: 0-9

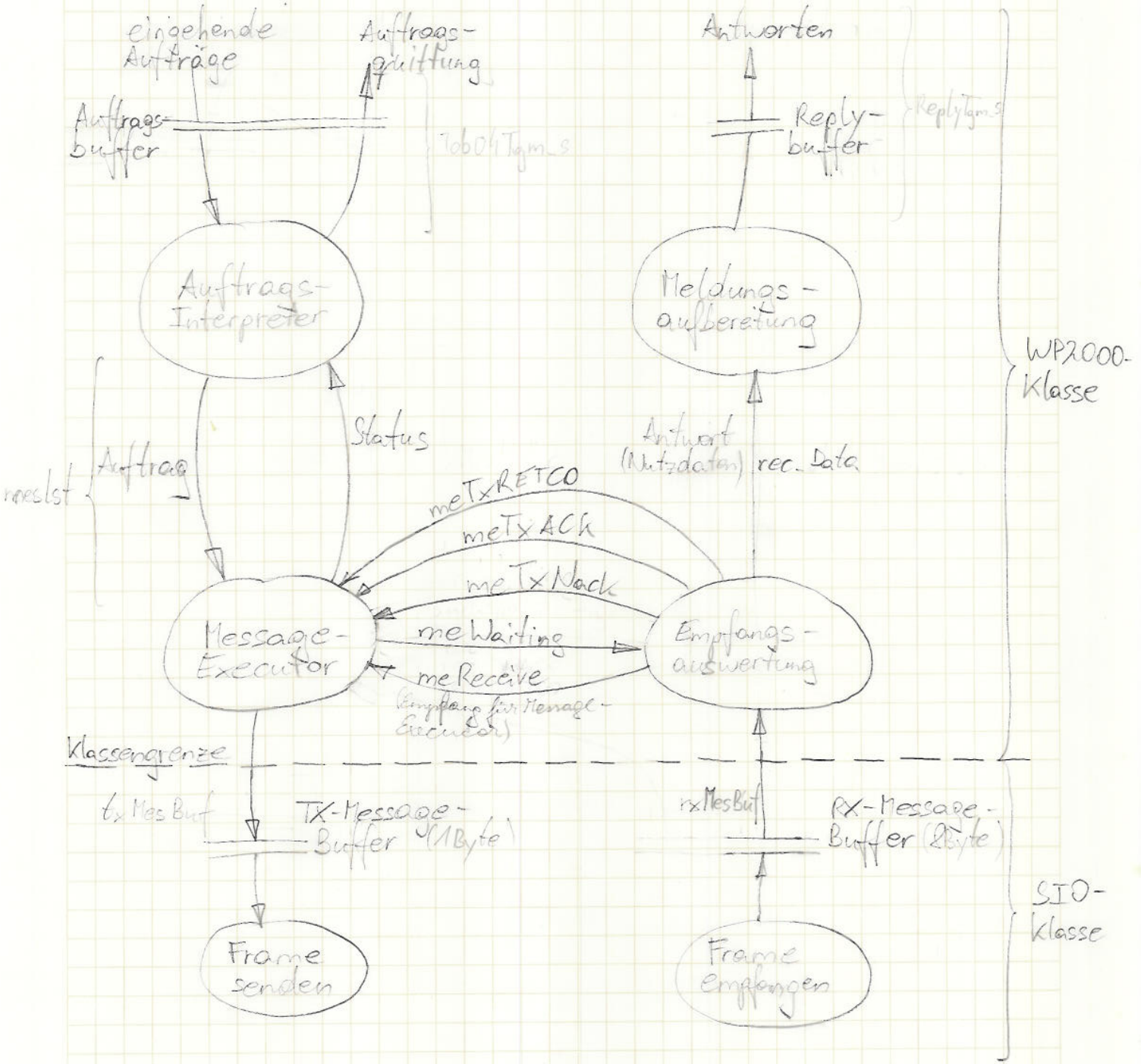
Status-Reg.:

Bit 7: Phone-Line Condition	1 → OFF HOOK	0 → ON HOOK
Bit 6: POS-Reset	1 → POS Reset auslösen	0 → kein POS Reset
Bit 5: Saftey-Cutoff	1 → Saftey Cutoff ausgelöst	0 → Saftey Cutoff inaktiv
Bit 4: Speech Recognition Unit	1 → SRU aktiv	0 → SRU inaktiv
Bit 3: Device 4	1 → Dev. 4 OFF	0 → Dev. 4 ON
Bit 2: Device 3	1 → Dev. 3 OFF	0 → Dev. 3 ON
Bit 1: Device 2	1 → Dev. 2 OFF	0 → Dev. 2 ON
Bit 0: Device 1	1 → Dev. 1 OFF	0 → Dev. 1 ON

Daten #0 = #3: bel. Daten

Check-Summe: Check-Summe (Byte-Addition)

WP2000 - Treiber — Funktionsschema:



Strukturen / WP200-Treiber:

Job04Tgm_s: (Aufträge, Anfragen)

hdr	WP2THdr_s
jobId	1 INT16
status	1 Byte
data	4 Byte

WP2TgHdr_s: (f. Job04Tgm_s)

tgmLen	1 WORD
jobNr	1 LONG
dstAdr	1 Byte
srcAdr	1 Byte

MesLst_s: (f. Message-Decoder, Empfangsauswertung)

mesId	1 Byte
mesData	8 Byte
mesStatus	1 INT16

ReplyTgm_s: } Job04Tgm_s

hdr	WP2THdr_s
jobId	1 INT16
status	1 Byte
data	4 Byte

- Empfangsauswertung: mit bei Nutzdaten holt er sich selbst
  - entleert Messagebuffer ~~und teilt Handshake-Signale dem Message-Executer mit~~
  - ~~& CS~~ Überprüfung
  - Nutzdaten werden zur Meldungsaufbereitung weitergeleitet (wenn nicht gerade auf ~~HS~~ Handshake-Signale gewartet wird) ⇒ keine Nutzdaten!
  - ~~versorgt~~ M.E. mit Handshake-Empfang
- Meldungsaufbereitung:
  - ~~leitet Antworten~~ weiter
  - ~~kann Anfragen~~ an die übergeordnete Applikation ~~stellen~~
  - je nach Antwort kann die Meldungsaufbereitung dem Auftrags-Interpreter Aufträge in

Job-Ids: Anfrage: ist Code richtig?

20 .... PIN-~~Daten-Transfer~~ ID_PIN_A
10 .... PIN-Quittung (Daten) ID_PIN_D
→ auflegen ja/nein?

11 .... PUN-Daten-Transfer ID_PUN_D
21 .... PUN-Anfrage ID_PUN_A

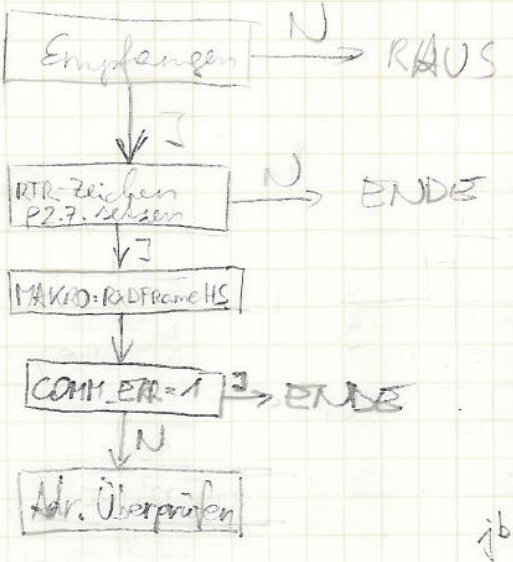
12 .... Status-Daten-Transfer ID_STS_D
22 .... Status-Anfrage ID_STS_A

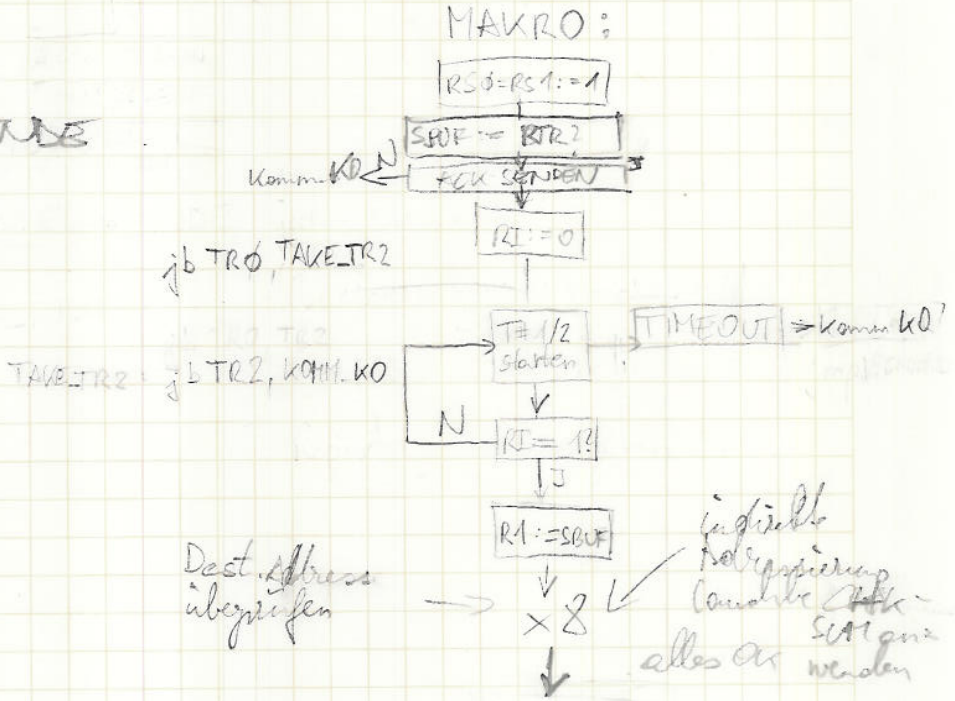
Job-Quittung: 0 ... Auftrag erfolgreich ausgeführt

Message-List: ⇒ 1 ... ~~A~~ neuer Auftrag anstehend

- -1 ... ~~no response from~~ no connection destination
- -2 ... destination not ready
- -3 ... retries failed
- -4 ... ~~invalid~~ invalid sign received
- -5 ... unexpectet receive
- -6 ... unexpectet sign
- -7 ... communication canceled
- -8 ... retrey request
- ~~-9 ... string too long~~
- -9 ... unknown ID received

ABLAUF-DIAGRAMM: EMPFANGEN





TIMER-Verwendung:

MACROS für Timer-Initialisierung

- T#0_INIT (LADEWERT)
- T#1_INIT (LADEWERT)