CombiConceptXM1.x wireless-communication

1) Allgemeines

Das ursprüngliche conceptXM1-System ist mit jeweils einem BIM-433 Transceiver (Radiometrix) ausgestattet. Leider erweisen sich diese Bausteine mangels Frequenzstabilität (starker Temperaturdrift), ausreichender Sendeleistung, genügender Datenübertragungsrate und mehrerer Sende-, Empfangskanäle als eher ungeeignet für ein zuverlässiges Netzwerk, bestehend aus mehreren Hosts. Im Zuge der weiteren Entwicklung kommt künftig ein intelligentes Kommunikations-Interface zum Einsatz. Dieses Interface, bestehend aus Steuer-controller und rfm12-Transceiver ermöglicht einen ausreichend schnellen Daten-austausch - sowohl drahtlos als auch über den integrierten RS485-Transceiver.

2) Kommunikationsstruktur

Dem Komplettsystem, bestehend aus Steuercontroller, rfm12-Transceiver und CombiConceptXM1.x-System liegt ein definiertes Schichtmodell zu Grunde:

Layer 1 - Physikalische Interfaces (RS485 oder wireless)

Der physikalische Layer bestimmt u.a. über den drahtlosen oder drahtgebun-denen Datenaustausch zwischen den aktiven Netzwerkteilnehmern. Weitere Parameter, wie: Tranferrate, Sendeleistung, Sende-, Empfangsfrequenz etc. werden ebenfalls bestimmt.

Layer 2 - Netzwerkkommunikation (Interfaces)

Die Kombination aus Steuercontroller und rfm12-Transceiver (des Weiteren auch als Interface bezeichnet) erlaubt die ständige, vom CombiConceptXM1.x-System unabhängige Überwachung der Netzwerkqualität. Dies beinhaltet die qualitative, sowie quantitative Erfassung aller aktiven Teilnehmer (des Weiteren als Hosts bezeichnet) durch Austausch von speziellen Messages (Scan,Alive,Kill). Hierbei wird zwischen direkt erreichbaren (Direct-) Hosts und entfernt ansprechbaren (Mesh-) Hosts differenziert. Sinngemäß können Meshed-Hosts nur indirekt über einen erreichbaren Direct-Host kontaktiert werden. Bei genauerer Betrachtung der Thematik handelt es sich bei einem Meshed-Hosts um einen, für den direkten Zugriff zu weit entfernten Teilnehmer des Netzwerkes.
Welcher Host nun als „Direct“ oder „Meshed“ deklariert wird, bestimmt jedes Interface im Verlauf der Netzwerksuche (Scan-Mode) selbsttätig. Das Ergebnis beinhaltet die verfügbaren Adressen inkl. Qualitätsfaktor (0..99%), wird im Anschluß in die
„scan host-table“- gesichert und steht künftig dem dynamischen Datenaustausch zur Verfügung.
Außer o.g. Systemfunktionen übernimmt Layer 2 die Integration des serCOM-Datenpaketes (aus Layer 3) , bzw. Konvertierung des rfm12-Datenpaketes (nach Layer 3).

Layer 3 - Systemkommunikation (Interface / CombiConceptXM1.x)
Die Kommunikation zwischen Interface und CombiConceptXM1.x findet auf Basis der SPI-Interfaces beider Systeme statt. Layer 3 regelt Zugriffsvereinbarungen und beinhaltet Sende- und Empfangspufferspeicher zur Vermeidung von Datenverlust.

Layer 4 - CombiConceptXM1.x-System Control
Bestimmt den Datenverkehr zwischen Local-, Sub-Area-Bus und Layer 5. Kommunikation mit Sub-Area-Bus

Layer 5 - conceptXM1.2-Application
Ereinisabhängige Auswertung und Anforderung von Daten aus untergeordneten Layern.

3) Protokollstrukturen
Der Datenaustausch zwischen den jeweiligen Layern erfordert eine, den Anforderungen entsprechende Integration bzw. Extrahierung von Protokollen. Dies wird bei Gegenüberstellung des serCOM-Protokollstandards und des rfm12-Protokolles besonders deutlich:

Auszug serCOM-Protokoll:

Mode Ident	LAB TAR	SAB TAR	LAB SOU	SAB SOU	Frame Size	Store Type	Sou Dadr	Tar Dadr	Data	…
8bit	8bit	8bit	8bit	8bit	8bit	8bit	16bit	16bit	8bit	

rfm12-Basisprotokoll:

Rfm12 Instr	SOU Adr	DTar Adr	MTar Adr	[EMBEDDED DATA]	CK Sum
8bit	8bit	8bit	8bit	xBit	8bit

Der Darstellung entsprechend wird im Sendebetrieb durch Layer 2 das von Layer 3 gelieferte serCOM-Protokoll in das rfm12-Basisprotokoll „eingebettet“ oder bei Empfang das serCOM-Protokoll dem von Layer 1 gelieferten rfm12-Frame „extrahiert“ und an Layer 3 weitergegeben.
Das serCOM-Protokoll wird durch das Interface nicht verändert. Lediglich die Steuerparameter: Mode-Ident, LAB-Target,LAB-Source, Store-Type und Frame-Size werden durch das Interface verwendet, aber nicht modifiziert !

Modifikationen serCOM-Protokoll:
Das serCOM-Protokoll entspricht mit aktueller, nicht mehr der ursprünglichen Version .

- Der unter conceptXM1 beschriebenen Struktur wurden die Synchron-Bytes (3Bytes nach Frame-Size) sowie die Checksumme entnommen.
- Des weiteren wurde dem conceptXM1-System die komplexe Manchester-Codierung, Checksummenberechnung, als auch die Transceiver-Steuerung entnommen.
- Durch das im Mode-Identifier gesetzte Flag: <setup> wird dem Interface der Setup-Mode signalisiert. Der Wert des Type-Parameters entspricht der gewünschten Setup-Funktion:

  0 = invalid
  Rückgabewert: -
  1 = scan mode
  Rückgabewert: -
  2 = set new host adress, followed by a kill, alive & a scan message
  Rückgabewert: -
  3 = rf-transceiver off
  Rückgabewert: -
  4 = rf-transceiver on
  Rückgabewert: -
  5 = soft-uart (rs485) off
  Rückgabewert: -
  6 = soft-uart (rs485) on
  Rückgabewert: -
  7 = alive message
  Rückgabewert: -
  8 = kill message
  Rückgabewert: -
  9 = load host identifier (provides available host-adresses & names)
  Rückgabewert:*<mi><4xAdressbytes><framesize><type><16bit Sadr><16bit Tadr><host-adr><6xNamebytes>*

  Setupformat: *<mi><4xAdressbytes><frame-size><type>*

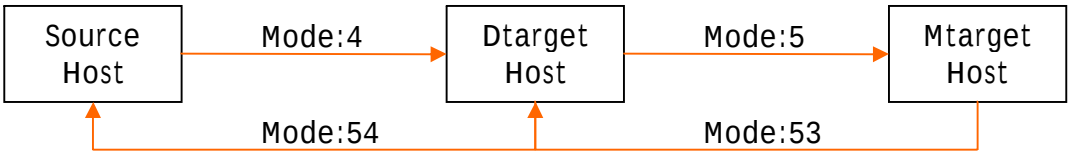
4) RFM12-Protokolle

Die Kommunikation zwischen den jeweiligen Interfaces wird auf Basis von rfm12-Protokollen abgewickelt. Grundsätzlich ist zwischen Setup-, und Transfer-Protokollen zu unterscheiden. Während Setup-Protokolle den qualitativen und quantitativen Netzwerkzustand bestimmen, werden Transferprotokolle für den Transport von serCOM-Frames benutzt. Die gewünschte Kommunikationsform ist im [rfm12-instr] Parameter festgelegt:

;----------
A) primary:
1 = scan (requests:50)
2 = embedded data to direct host (requests:52)
4 = embedded data to meshed host - first step to the direct host (requests:54)
5 = embedded data from the direct to the meshed-host (requests:53)
6 = embedded data to all available direct hosts (requests:52)
7 = alive message to the complete network (requests:51)
8 = kill message to the complete network (requests:51)
9 = lha change message to all direct hosts as a kind of broadcast (request:51)
10 = remote access message to a direct host (request:55)
NOTE: database for primary instruction is always the <RFMtxbuffer>

B)Secondary:
50 = available direct-host (incl.quality) to the calling host (in order of a received:1)
51 = simple acknowledge to the calling host (in order of a received:7/8)
52 = simple acknowledge to the calling host (in order of a received:2/6)
53 = mesh-acknowledge to the calling direct-host (in order of a received:5)
54 = mesh-acknowledge from the direct to the calling host (in order of a received:53)
55 = remote acknowledge to the direct host (in order of a received:10)
56 = reset acknowledge to the direct host (in order of a received:11)
;----------

Beispiel Meshed Transfer:

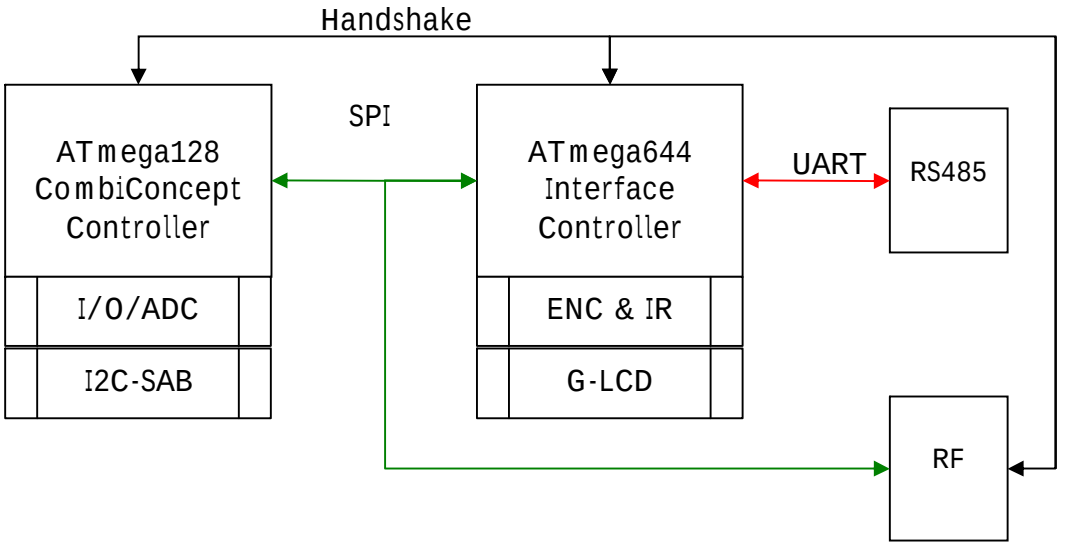


5) Zugriffsvereinbarungen
Der Netzwerkzugriff wird weitestgehend durch das Interface abgewickelt. Dies bedeutet das CokmbiConceptXM1.:
⇨ erwartet keine Acknowledges nach Datenversand
⇨ generiert keine Acknowledges nach Datenempfang

Acknowledges werden folglich, lediglich auf Interfaceebene ausgetauscht !
Bedingt durch diese Neuerungen wurden allen relevanten Softwarebausteinen des CombiConceptXM1.x-Systems:
⇨ die Acknowledgegenerierung und Auswertung entnommen
⇨ die damit verknüpften Warteschleifen entnommen

Zur Vermeidung von Datenverlust bzw. Kollisionen wurde dem CombiConceptXM1.x, sowie dem Interface ein Hardware-Handshaking integriert. Hierfür verbinden außer dem SPI-Port zwei weitere Handshake-Leitungen die beiden Systeme.

6) Hardwareschema



Wie der CombiConceptXM1.x Dokumentation zu entnehmen ist werden durch das Basissystems:

- ein S(ub)-A(rea)-B(us), der den I2C- Spezifikation entspricht
- und ein L(ocal)-A(rea)-B(us), der den serCOM- Spezifikationen entspricht,

unterstützt. Der SAB ist direkt am Basisystem zugänglich - der LAB über den separaten LAB-Controller (Interface).