

Wi-Sun Add-onボード(WISUN-EVK-701)

更新履歴

版数	更新内容	更新日
0.2	Refsを追加した。	2023/01/12
0.1	初版を0.1版とする。	2022/08/05

目次

1. Wi-Sun Add-onボード（SPRESENSE用）概要
2. Spresense + Wi-SUN Add-onボード + Host PC システム構成図
3. 1対1通信サンプルコード動作確認
 1. 通信不能(Ubuntu開発環境)
 2. サンプルのバグを修正
 3. 修正後のサンプル動作確認
4. 1対1通信ログ確認参照資料
 1. 状態遷移図
 2. HANペアリングシーケンス
 3. データ送受信 シーケンス

Wi-SUN Add-onボード SPRESENSE-WiSUN-EVK-701 概要

- Wi-SUN Add-onボード「SPRESENSE-WiSUN-EVK-701」はSPRESENSEにWi-SUN通信の機能を追加する拡張ボードです。Wi-SUN規格はHEMSやBEMSに適した通信であり、搭載しているファームウェアはリレー通信やスリープ機能を含む、より汎用的に使用できる規格である「Wi-SUN Enhanced HAN」に対応しています。従来の1対多のスター型接続に加え、1対多対多のツリー型接続ができるため、低消費電力かつ長距離通信が可能なセンサノードを実現できます。

※詳細は

<https://www.rohm.co.jp/support/spresense-add-on-board>

【用語】

HAN : Home Area Network

HEMS : Home Energy Management System

BEMS : Building Energy Management System

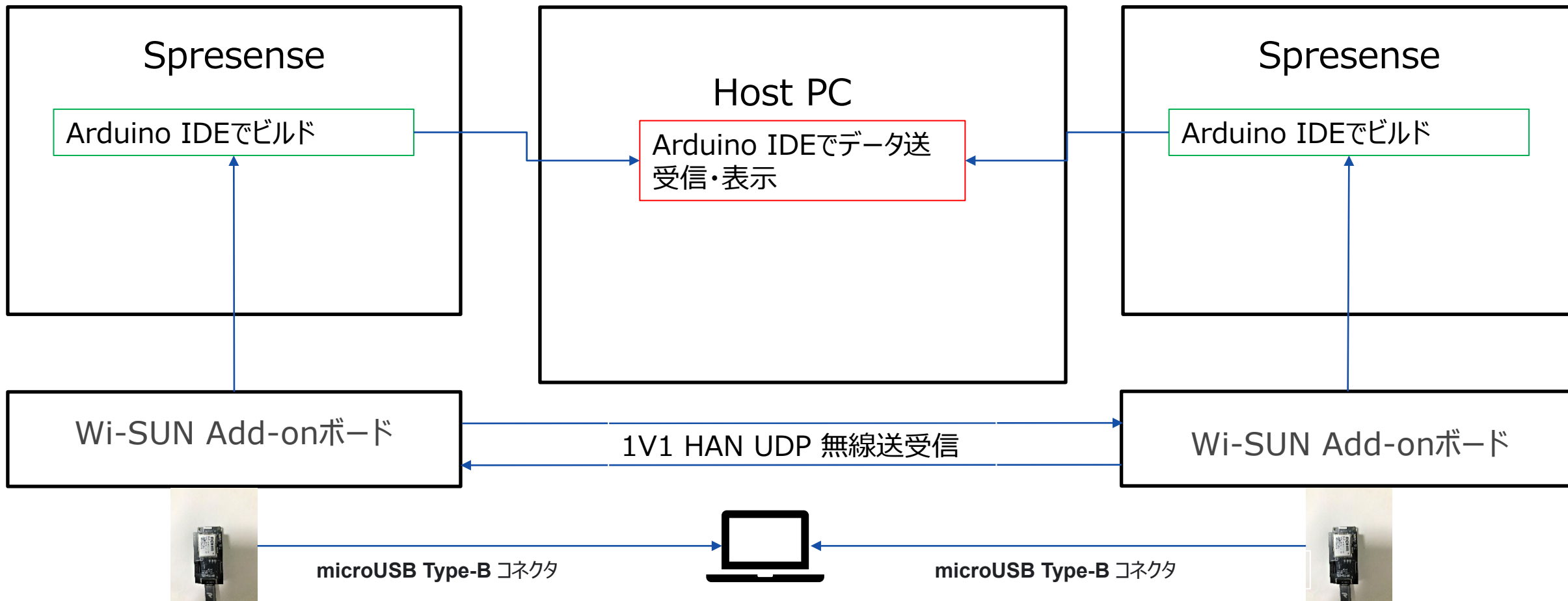


Spresense + Wi-SUN Add-onボード + Host PC システム構成図

1. 概略図

役割 : PAN コーディネータ

役割 : エンドデバイス



1対1通信サンプルコード動作確認：通信不能(Ubuntu開発環境)

- ROHM社が提供した1V1のサンプルコードはWindowsで動作確認できましたが、Ubuntuでコンパイルと動作はできませんでした。原因はサンプルコードはバグがありました。
 - 詳細は次のページで参照してください。

Ubuntuで動作確認したときに、通信不能のログ確認

https://fscdn.rohm.com/jp/products/databook/applinote/module/wireless/bp35c0-j11_uartif_specification_tr-j.pdf

上記のWi-sun Uart-IF仕様書を参照して下記のことわかりましたが、初期化コマンドを送信してから、ずっとレスポンスが来なくて、そして、送信するメッセージがずっと送りっぱなしのように見えます。

```
RESET
D0 F9 EE 5D 60 19 0 4 3 91 0 0
```

こちらはWi-sunモジュールにResetを出して、Wi-sunモジュールはUart2経由でレスポンスしたログです。

```
D0 EA 83 FC 0 5F 0 8 3 A0 0 8 3 0 5 0
```

こちらはSpresense→Wi-sunモジュールに初期化コマンドを出しています。
成功の場合はレスポンスは来るはずですが、現状はこないまま下記のログ出力し続けます。

```
D0 EA 83 FC 0 5F 0 6 3 44 0 24 1 23
D0 EA 83 FC 0 5F 0 6 3 44 0 24 1 23
```

1対1通信サンプルコード動作確認：サンプルのバグを修正

ソースコードを修正前後の差分

```
resources/WISUN-EVK-701/SPRESENSE-WISUN-EVK-701-DEV2/bp35c0-j11.cpp

@@ -17,7 +17,7 @@
17 17 OUT OF OR IN CONNECTION WITH THE
18 18 THE SOFTWARE.
19 19 *****
20 20 - #include <arduino.h>
20 20 + #include <Arduino.h>
21 21 #include "bp35c0-j11.h"
22 22
219 219 boolean BP35C0J11::cmd_send(unsigned short cmd) {
220 220     unsigned short hdr_chksum = uni_req[0] + uni_req[1] + uni_req[2] + uni_req[3] ;
221 221     unsigned short dat_chksum = 0 ;
222 222     unsigned short msg_length = 0 ;
223 223     unsigned short dat_length = 0 ;
224 224     unsigned short send_dat_size = 0 ;
225 225     unsigned char data[128] = {0};
226 226
227 227     unsigned char send_data[128] = {0} ;
228 228 - boolean rc = FALSE;
228 228 + boolean rc = TRUE;
229 229     unsigned char cnt = 0 ;
230 230
231 231     switch (cmd) {
232 232
233 233     @@ -364,6 +364,7 @@ boolean BP35C0J11::cmd_send(unsigned short cmd) {
364 364
365 365         break;
366 366     }
367 367 + return rc;
367 368 }
```

Spresense公式サイトはArduino.hを使っています。Windowsで上記大小文字区別しないでコンパイルOKでしたが、Ubuntuだとコンパイルエラーが発生します。OSを依存しないように、正確なヘッダを使ったほうがいいです。

【Spresense Arduino LibraryのArduino.h】
<https://github.com/sonydevworld/spresense-arduino-compatible/blob/master/Arduino15/packages/SPRESENSE/hardware/spresense/1.0.0/cores/spresense/Arduino.h>

cmd_send()戻り値はbool型ですが、もともとリターンしていなかった。Windowsで確認して特に問題がないことを確認しましたが、Ubuntuだと、本プログラムは動作不能の状態となりますので、OSを依存しないように、ちゃんと正確なリターン型の値に修正しました。そして、Ubuntuでも動作できるようになったことを確認しました。

バグを修正して1対1通信サンプルコード動作確認できた：下記のUART IF仕様書により、ログ確認

https://fscdn.rohm.com/jp/products/databook/applinote/module/wireless/bp35c0-j11_uartif_specification_tr-j.pdf

RESET **success_log_dev2.txt**

D0 F9 EE 5D 60 19 0 4 3 91 0 0 起動完了通知

State = 1

D0 EA 83 FC 0 5F 0 8 3 A0 0 8 3 0 5 0
D0 F9 EE 5D 20 5F 0 5 3 98 0 1 1 初期化成功

Init Success

State = 2

D0 EA 83 FC 0 2C 0 14 3 79 3 28 31 31 31 31 32 32 32 32 33 33 33 33 34 34 34 34
D0 F9 EE 5D 20 2C 0 5 3 65 0 1 1 HAN PANA 認証情報設定成功

PANA Password set Success

State = 4

D0 EA 83 FC 0 A 0 C 3 4F 7 F8 FF FF FF FF FF FF FF
D0 F9 EE 5D 20 A 0 11 3 4F 3 1B 1 5 56 78 0 1D 12 91 0 3 5E 48 DE HAN 動作開始：PANCのペアリング待ち

HAN Act Success

PANC: 0 1D 12 91 0 3 5E 48

State = 5

D0 EA 83 FC 0 3A 0 4 3 77 0 0
D0 F9 EE 5D 20 3A 0 D 3 7B 1 6A 1 0 1D 12 91 0 3 5E 48 HAN PANA 開始

PANA Act Success

D0 F9 EE 5D 60 28 0 D 3 A9 1 6A 1 0 1D 12 91 0 3 5E 48 PANA 認証結果通知成功

PANA Connect Success

State = 6

D0 EA 83 FC 0 5 0 6 3 44 0 28 E 1A
D0 F9 EE 5D 20 5 0 5 3 3E 0 1 1 D0 F9 EE 5D 60 18 0 23 3 AF 6 2B FE 80 0 0 0 0 0 2 1D 12 91 0 3 5E 48 E 1A E 1A 56 78 0 2 DE 0 4 54 45 53 54 UDP ポート OPEN成功

UDP port open Success

State = 7

D0 F9 EE 5D 60 18 0 23 3 AF 6 2B FE 80 0 0 0 0 0 2 1D 12 91 0 3 5E 48 E 1A E 1A 56 78 0 2 DE 0 4 54 45 53 54 データ受信成功

UDP rcv Success

CMD UDPSND

D0 EA 83 FC 0 8 0 1E 3 5F 4 7D FE 80 0 0 0 0 0 2 1D 12 91 0 3 5E 48 E 1A E 1A 0 4 54 45 53 54 データ送信成功

D0 F9 EE 5D 20 8 0 A 3 46 1 41 1 0 54 45 53 54

UDP send Success

RESET **success_log_panc2.txt**

D0 F9 EE 5D 60 19 0 4 3 91 0 0 起動完了通知

State = 1

D0 EA 83 FC 0 5F 0 8 3 A0 0 6 1 0 5 0
D0 F9 EE 5D 20 5F 0 5 3 98 0 1 1 初期化成功

Init Success

State = 2

D0 EA 83 FC 0 A 0 6 3 49 0 CE 56 78
D0 F9 EE 5D 20 A 0 8 3 46 0 D4 1 5 56 78 HAN 動作開始

HAN Act Success

State = 3

D0 EA 83 FC 0 5 0 6 3 44 0 28 E 1A
D0 F9 EE 5D 20 5 0 5 3 3E 0 1 1 UDP ポート OPEN成功

UDP port open Success

State = 4

D0 EA 83 FC 0 25 0 5 3 63 0 1 1
D0 F9 EE 5D 20 25 0 5 3 5E 0 1 1 HAN 受入れ接続モード切り替え

Normal connect mode

D0 F9 EE 5D 60 1A 0 E 3 9C 2 59 1 0 1D 12 91 0 3 58 5F DE 接続状態変更通知：MAC 接続完了

MAC connect Success

END DEVICE: 0 1D 12 91 0 3 58 5F

State = 5

D0 EA 83 FC 0 2C 0 1C 3 81 4 A2 0 1D 12 91 0 3 58 5F 31 31 31 31 32 32 32 32 33 33 33 33 34 34 34 34
D0 F9 EE 5D 20 2C 0 5 3 65 0 1 1 HAN PANA 認証情報設定成功

PANA Password set Success

State = 6

D0 EA 83 FC 0 3A 0 4 3 77 0 0
D0 F9 EE 5D 20 3A 0 5 3 73 0 1 1 HAN PANA 開始

PANA Act Success

D0 F9 EE 5D 60 1A 0 E 3 9C 2 5A 2 0 1D 12 91 0 3 58 5F DE 接続状態変更通知：PANA 認証完了

PANA authentication Success

State = 7

D0 EA 83 FC 0 8 0 1E 3 5F 4 8E FE 80 0 0 0 0 0 2 1D 12 91 0 3 58 5F E 1A E 1A 0 4 54 45 53 54
D0 F9 EE 5D 20 8 0 A 3 46 1 41 1 0 54 45 53 54 データ受信成功

UDP send Success

D0 F9 EE 5D 60 18 0 23 3 AF 6 3C FE 80 0 0 0 0 0 2 1D 12 91 0 3 58 5F E 1A E 1A 56 78 0 2 DE 0 4 54 45 53 54 データ送信成功

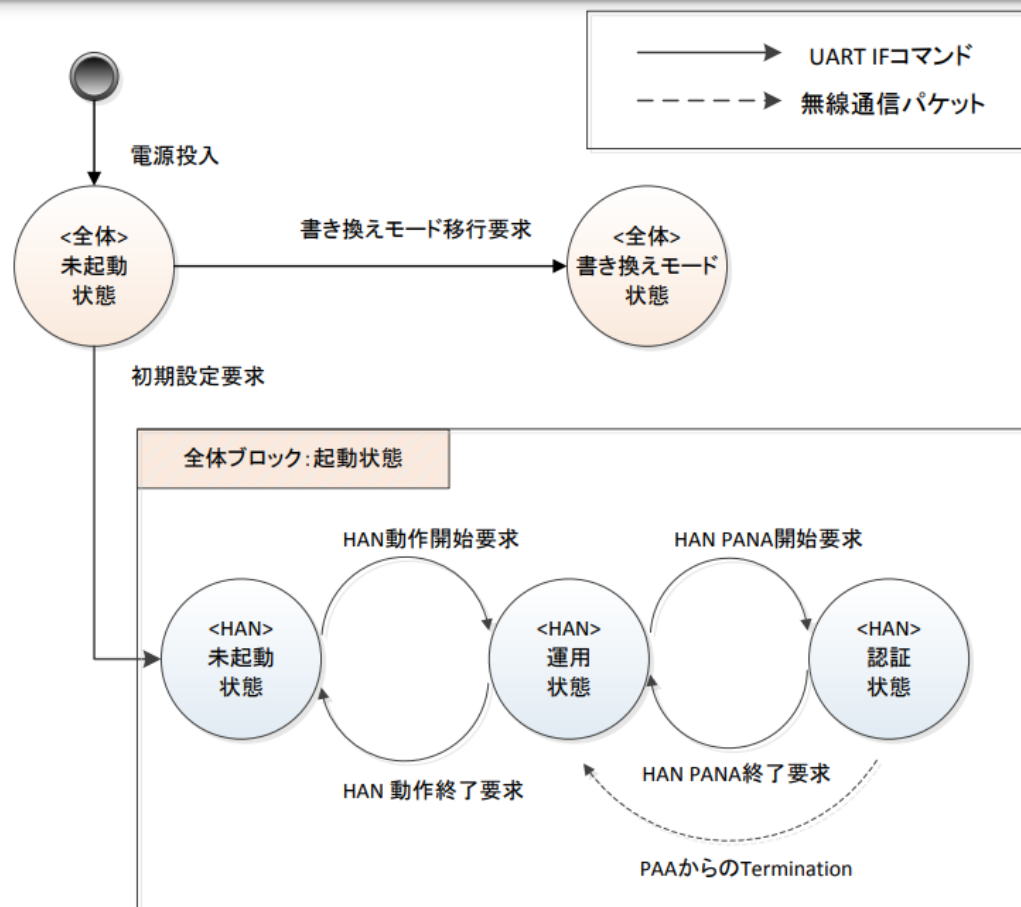
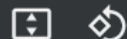
UDP rcv Success

※確認コツ：D0 F9 EE 5Dレスポンス後ろ2byte例：20 8→0x2008をUart IF仕様で
検査すれば、早く対応する内容がわかります。

1対1通信ログ確認参照資料：状態遷移図

17 / 192

100%



【参照箇所】

https://fscdn.rohm.com/jp/products/databook/applinote/module/wireless/bp35c0-j11_uartif_specification_tr-j.pdf

P17 状態遷移図

図 3 動作モードが PAN コーディネータ、コーディネータ、エンドデバイスモードでの状態遷移

1対1通信ログ確認参照資料： HANペアリングシーケンス

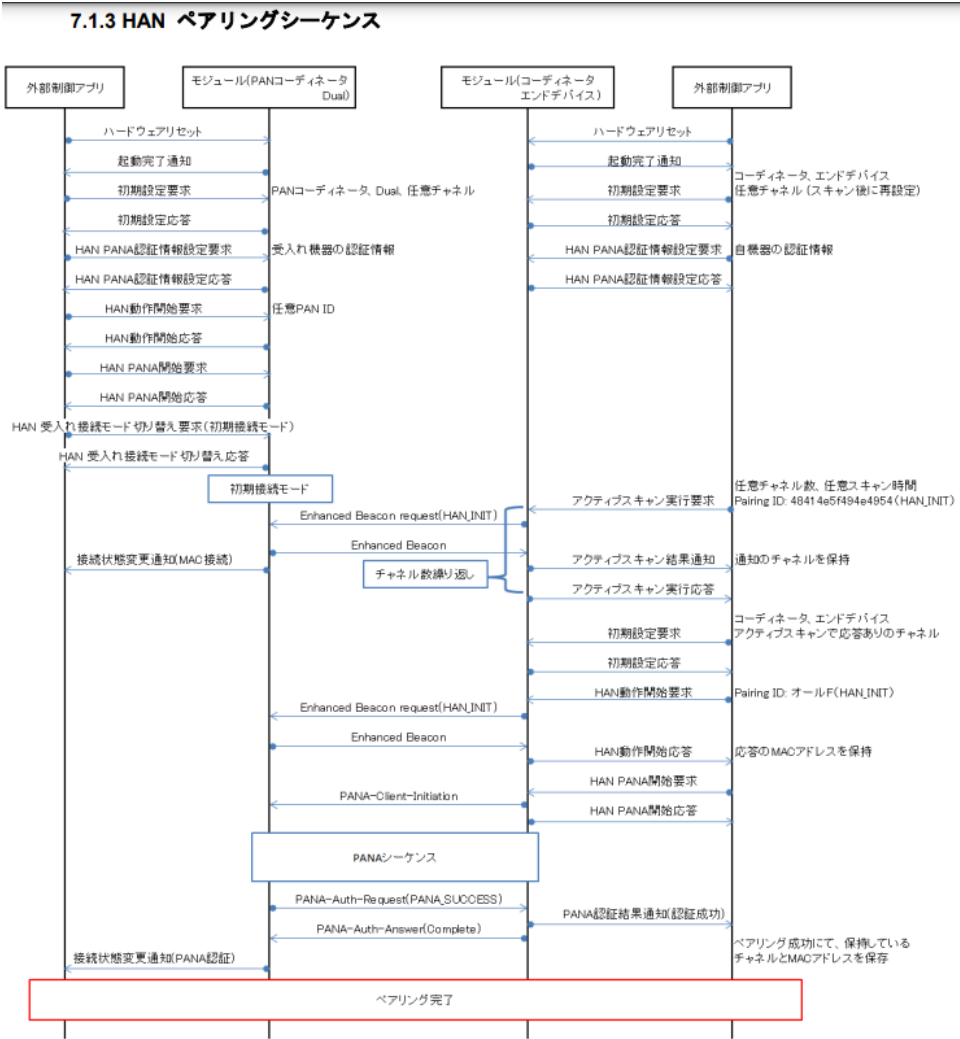


図 35 HAN ペアリングシーケンス

【参照箇所】

https://fscdn.rohm.com/jp/products/databook/applinote/module/wireless/bp35c0-j11_uartif_specification_tr-j.pdf
P161 HANペアリングシーケンス

1対1通信ログ確認参照資料：データ送受信 シーケンス

5.12 データ送受信 シーケンス

5.12.1 データ送受信(ND なし) シーケンス

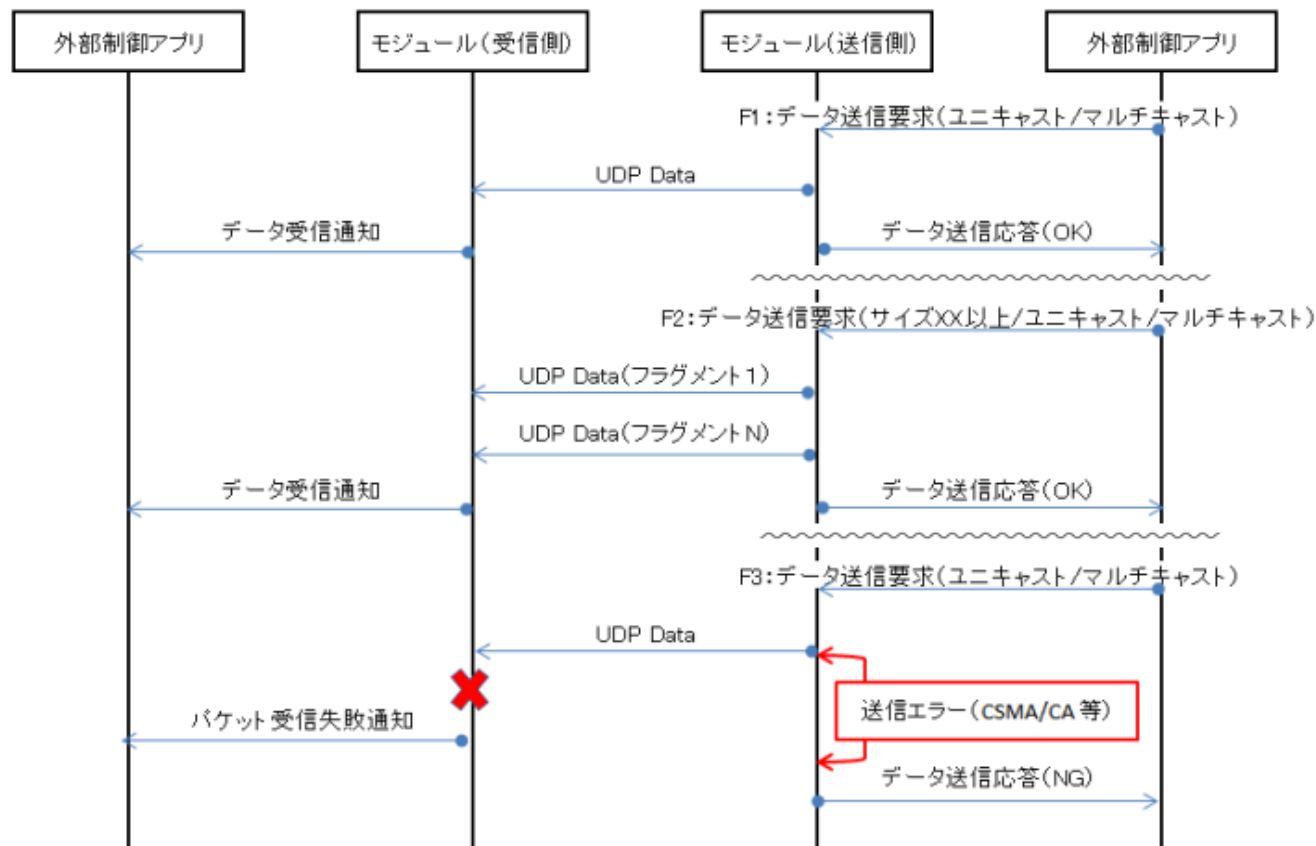


図 18 データ送受信(ND なし) シーケンス

【参照箇所】

https://fscdn.rohm.com/jp/products/databook/applinote/module/wireless/bp35c0-j11_uartif_specification_tr-j.pdf
P139 データ送受信 シーケンス