Xbee単体による I2C通信の検討

技術要素確立（Xbee-I2C）

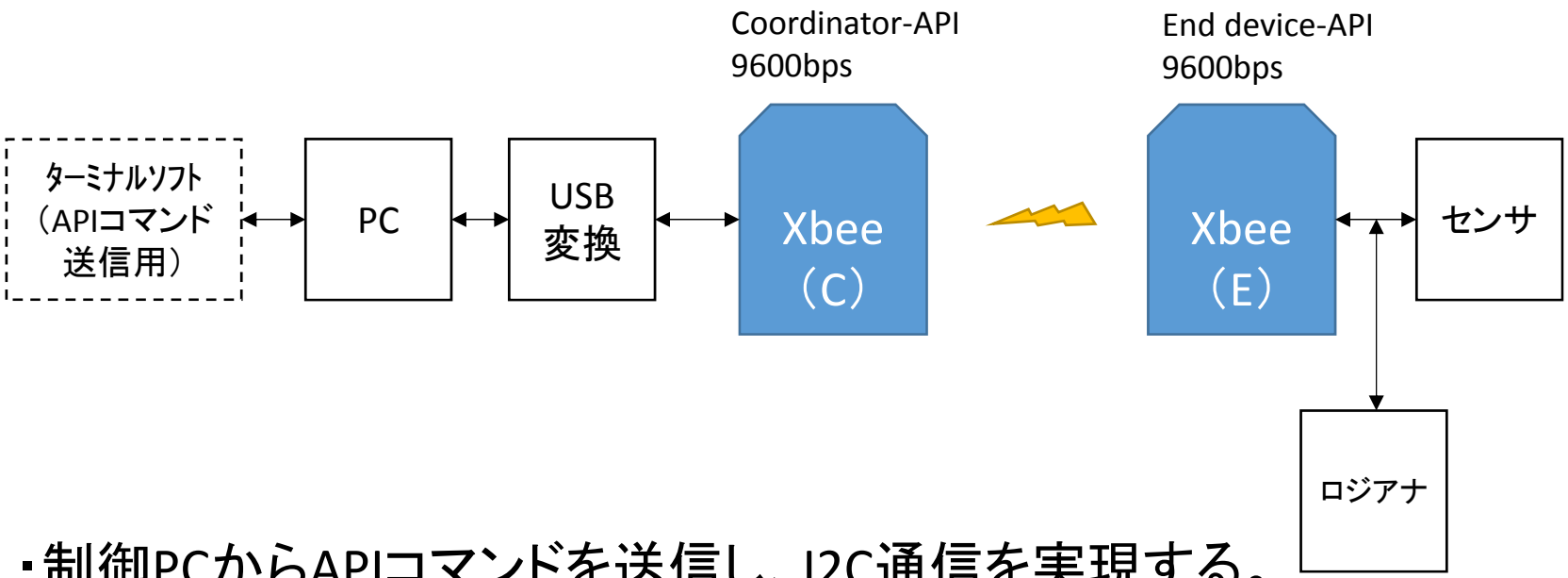
1. 概要

- Xbee単体には、I2CやSPIのペリフェラルは存在しない。そこで、DIOの機能を駆使し、I2C通信機能を実現する方法について検討を行った。

参考文献:トランジスタ技術 2012年12月号

2. 検証コンフィギュレーション

- ターミナルソフトからAPIコマンドを発行し、エンドデバイスのセンサと通信を行う。



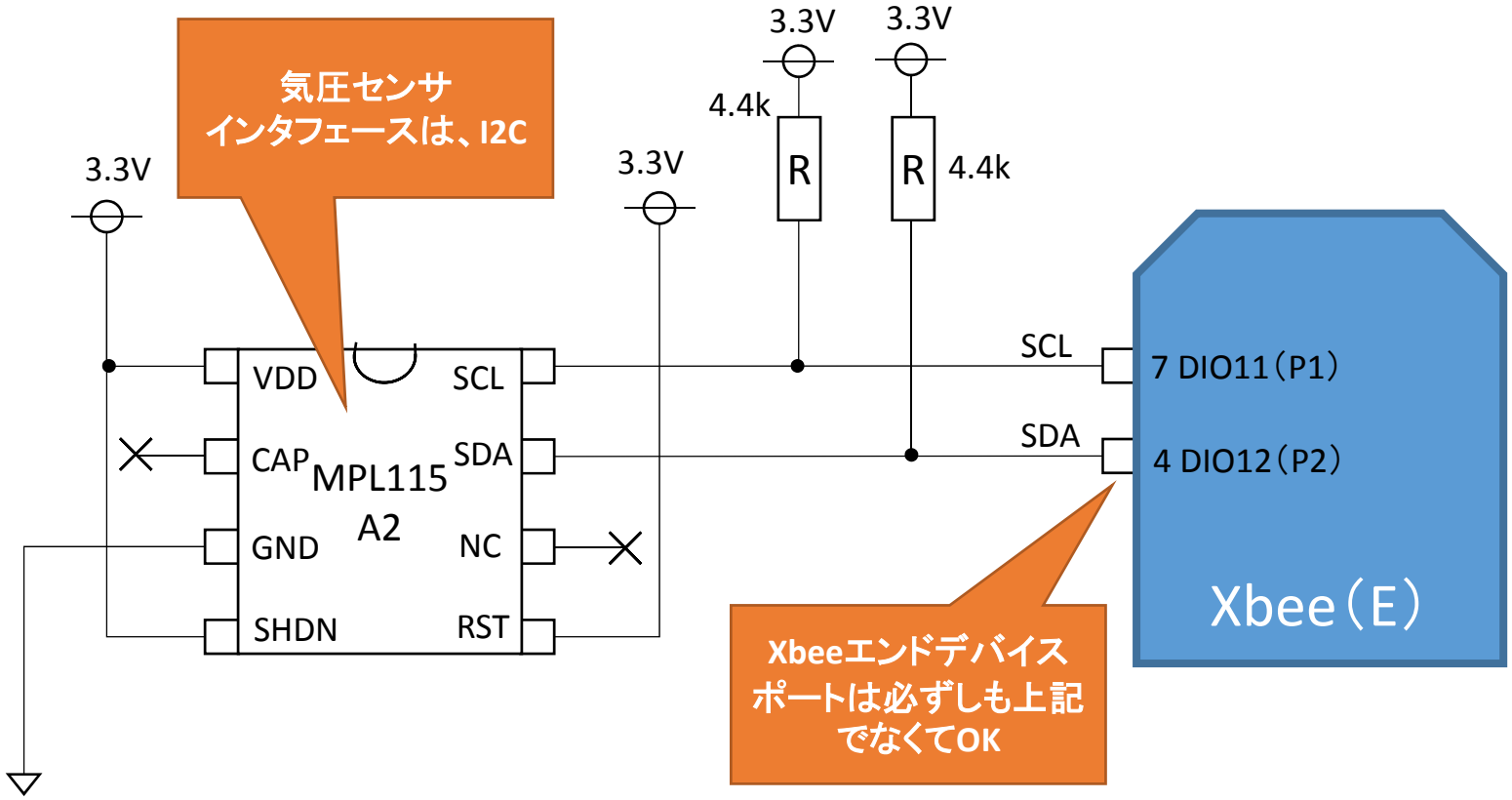
・制御PCからAPIコマンドを送信し、I2C通信を実現する。

・解析のため、間にロジアナを設置した。

・Xbeeは、S2のPRO版を使用した。

3. エンドデバイス回路図

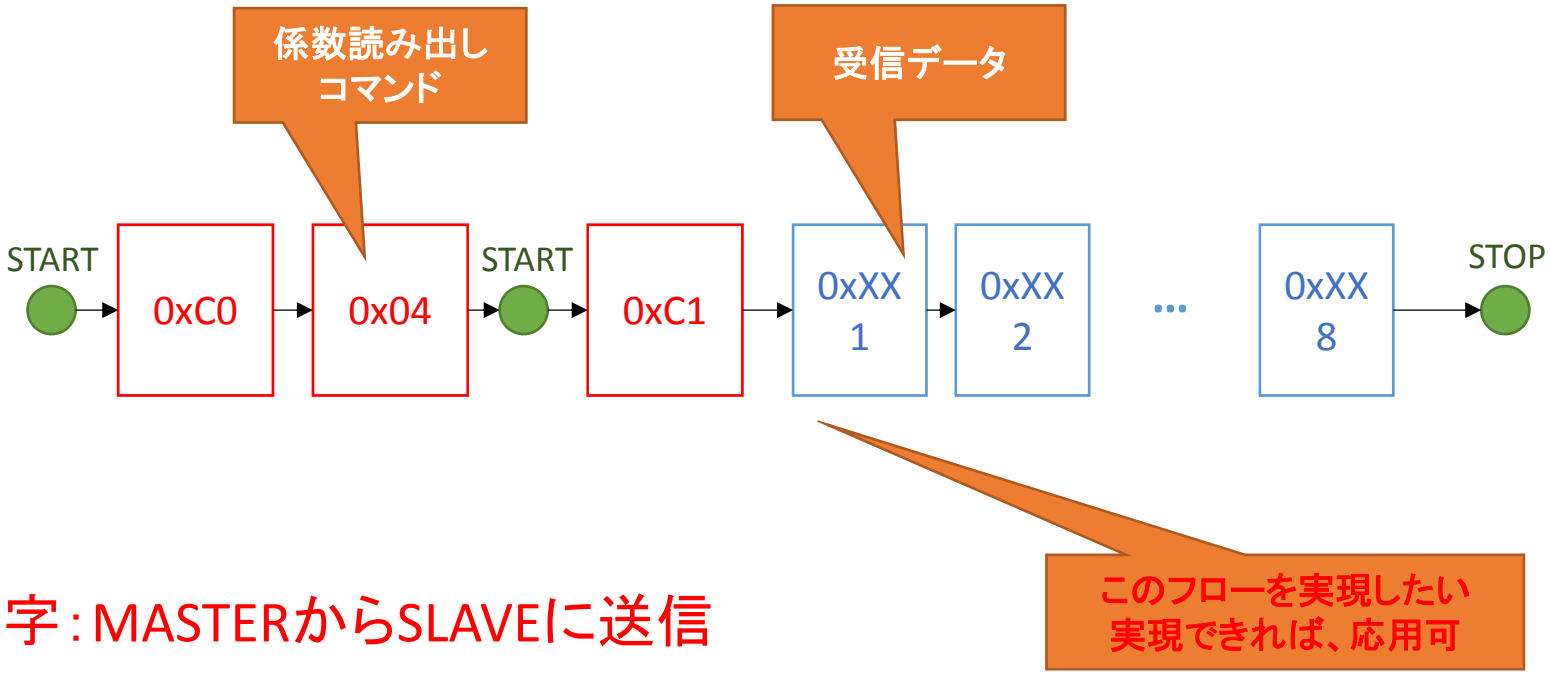
- 検証に使用したエンドデバイスの簡易回路図を示す。



・センサには、MPL115A2（気圧センサ）を使用した。
・電源は、ACアダプタ+レギュレータ（3.3V）を使用した。

4. フロー

- I2C通信実現の流れ（イメージ）
  MPL115A2には、係数読み出しというシーケンスがあり、今回はそのシーケンスを実施してみた（用いるセンサによって異なる）。



赤字：MASTERからSLAVEに送信

青字：SLAVEからMASTERに送信

緑字：スタートコンディションまたはストップコンディション

5. APIフレーム構成（C→E）

- フレーム構造

このフレームをXbeeに送信することで様々なデータ通信が可能になる。

フレームの長さ 3～18		フレームID	16bitアドレス 不明なら0xFFFE	ATコマンド	チェックサム
7E	0010	17	01	XXXXXXXXXXXXXXXX	FFFE	02	5031	05	XX

フレームの先頭文字

フレームタイプ リモート制御の場合は、0x17

リモートXbeeのアドレス64bit

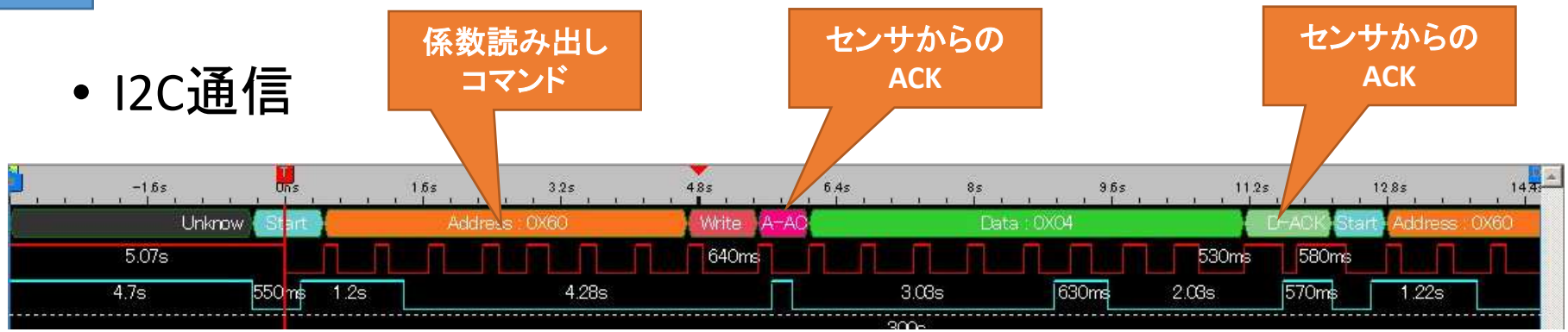
02はコマンドの結果を直ちに反映する

ATコマンドパラメータ
03：Input
04：Output L
05：Output H

・上記はすべてHEX値
・チェックサムは、0xFF－（フレームIDからの和）
・1フレームで、1つのピンだけをL/Hできる。

6. I2C（コマンド送信）

- I2C通信



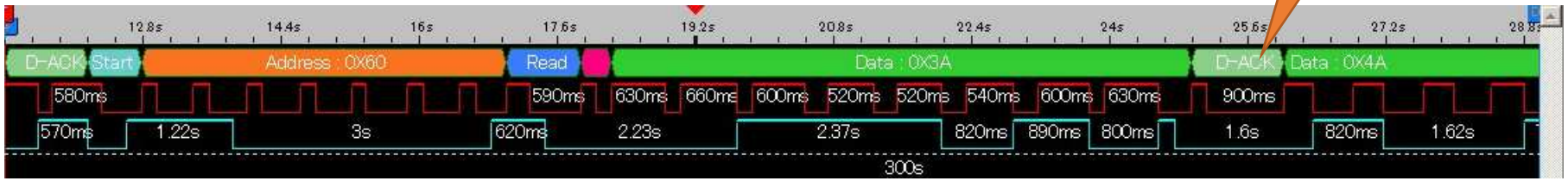

```
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
# 0xC0
#################################################################
# START
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,32,04,9D
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,04,9E
# 1bit 1
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,32,05,9C
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,04,9E
# 2bit 1
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,32,05,9C
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
…
```

上記の波形を実現するために、数十～数百個のコマンド群を送信している。

7. I2C（係数受信）①

MASTERからの
ACK

- I2C通信




```
# 0xZZ 1
##########################################################
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,04,9E
# 1bit Z
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,32,03,9E
7E,00,0F,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,49,53,87
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,04,9E
# 2bit Z
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,05,9D
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,32,03,9E
7E,00,0F,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,49,53,87
7E,00,10,17,01,00,13,A2,00,40,6C,B0,B4,FF,FE,02,50,31,04,9E
…
```

・ポートを入力に設定
・ポートの状態を取得する
ATコマンド「IS」
にて、センサの出力を取得する。

・SCLがHighの時に、SDAポートを読み取る（コマンドを送信する）。
タイミングが合わないと正確に読み取れない（200ms置きぐらいに送信すると良い、）。

7. I2C（係数受信）②

- 係数の受信内容の確認

ロジアナ取得

3A	4A	BA	6F	C3	16	34	18

このロジアナで取得した値と合っていればOK

Xbee受信フレーム

00111010	01001010	10111010	01101111	11000011	00010110	00110100	00011000
3A	4A	BA	6F	C3	16	34	18

Bit単位で取得し、あとでByteに結合する。

・ロジアナで取得した係数値と一致！
（Xbee単体でもI2C通信が実現できた）

8. メリット＆デメリット

- Xbee単体のI2Cのメリット＆デメリット

メリット	デメリット
・エンドデバイスの回路構成を簡素化できる。 ・多様なセンサと接続することは可能。	・1回のI2C通信に時間がかかる。大体1～2分はかかる。 ・I2C通信するために、無線通信をするため電力を消費する。