

§2-II-3 インターフェースの実験 I

(Experiment of interface I)

1. 実験の狙い

マイコンに代表される組込み用の各種コントローラにはさまざまなペリフェラルが内蔵されている。その中でも、シリアル通信インターフェースはコントローラ間や外部通信モジュール間との通信を実現するだけでなく、センサ出力の取出しやメモリの入出力を実現するためにも広く用いられている。本実験では、代表的なシリアル通信インターフェースである RS-232C の通信フォーマット等の諸特性を観察することにより、シリアル通信の仕組みに関する理解を深めることを目的とする。

2. 解説

通信インターフェースの概要

マイコン等で用いられる通信インターフェースは大きく分けてパラレル通信とシリアル通信に大別できる。

○パラレル通信

パラレルというのは「並列の」という意味で、パラレル通信では 8 本、16 本といった複数の信号線の束を用いて、複数ビットを同時に伝送するためのインターフェースである。このように、パラレル通信においては 1 度に複数ビットを伝送可能なため、比較的低いクロックを用いた場合でも高速なデータ伝送が実現できる点、また送受信間の複数の入出力 (I/O) にそのまま接続すれば良いので構造が簡単であるという利点がある。しかし、信号線の本数の増加に伴い設置面積が大きくなることに加えて、高速な転送クロックを実現しようとすると、各配線の長さのバラツキ等に起因して受信データが正しく受信できないといった欠点がある。

○シリアル通信

シリアル通信では、データ用に 1 本の信号線を用いてデータ伝送を行うインターフェースであり、信号線の本数を少なく抑えることが可能となるため、現在のマイコン等の組込みデバイスで広く用いられている。シリアル通信の利点として、前述の信号線の本数を少なく抑えることが可能であることに起因した設置面積の縮小化が挙げられる一方で、パラレル通信と比べて高速な転送クロックが必要となる欠点がある。特に近年では、数百[Mbps]程度の伝送レートを扱うことも珍しくないため、信号波形が歪まないようなボード設計や信号線配置等の高周波センシングが求められることもある。

シリアル通信の出力インターフェース回路では、MPU 内部のパラレルデータを MSB (most significant bit : 最上位ビット)、または LSB (least significant bit : 最下位ビット) から順に 1 ビットずつに分解して 1 本のデータ線に出力する。この処理をパラレルーシリアル変換と呼ぶ。また、受信側にある入力インターフェース回路では 1 ビットずつ送られてきたデータを入力し、元のパラレルデータへと組立て直す。この処理をシリアルーパラレル変換と呼ぶ。シリアル通信には送受信データの形式等の違いによって、

いくつかの方式に分けられる。以下にその詳細を示す。

調歩同期式シリアル通信方式（または非同期式シリアル通信方式）では、通信フォーマット内にデータの始まりを表すスタートビットや、データの終わりを示すストップビットなどを付加することにより、受信側で送信側とのタイミングを合わせることが可能となり、1本の信号線のみで同期タイミングとデータの両方を送ることができるようにしたものである。信号線を少なくすることができる一方、スタートビット等の挿入により伝送効率は低下する。

一方、同期式シリアル通信方式では、タイミングクロック用の専用線を設け、このクロックで送受信間の同期をとる。調歩同期式シリアル通信方式と比べて、最低でもデータ用とクロック用の2本の信号線を使用しなければならない点はあるものの、スタートビット等は使用しないため伝送効率は良い。

RS-232C

前述の調歩同期式シリアル通信方式の代表例の一つである。この通信では送信部と受信部が独立して構成されていることから、全二重通信を実現することが可能となる。一般的に一度の伝送で8ビットのデータを扱い、伝送レートは数～数百[kbps]程度となっている。また、設定によってデータビットの後にパリティビットを挿入することも可能である。なお、RS-232Cをマイコン等のコントローラで扱うブロックはUART (Universal Asynchronous Receiver Transmitter)と呼ばれる。(一部のコントローラでは、半二重の同期式シリアル通信にも対応して USART (Universal Synchronous Asynchronous Receiver Transmitter)と呼ばれるものもある。)

3. 実験の予習 *予習レポートとして整理して提出すること

予習1： 下記の用語について、意味を調査し理解を進めておく。

黒字は必ず。その他は余裕があれば行うこと。

- a. RS-232C ドライバ(インターフェース) IC
- b. パリティビット
- c. チェックサム
- d. ASCII (アスキー) コード

予習2： RS-232C の通信フォーマットを調査せよ。

4. 実験

4.1 実験の概要

PIC マイコン(PIC18F4553)によって RS-232C のインターフェースを実装し、以下の流れに沿って実験を進める。

- RS-232C を用いたエコーバックの実験
- 誤り発生時の RS-232C の伝送特性
- RS-232C の伝送レート変化に対する特性

以下にこれらの詳細について示す．なお，実験機器の詳細な配線やソフトウェアの使用方法などについては Moodle にアップロードしてある別資料を確認すること．

RS-232C を用いたエコーバックの実験

1. 図 1 に示すように配線する．その後，PC1 で PIC の開発ソフトである MPLAB を起動し，エコーバックに関するプログラムを選択，ビルドした後，実行ファイルを PIC マイコンにダウンロードする．
2. PC2 でシリアル通信用のソフトウェアである Tera Term を起動し，通信ポートの設定をして RS-232C ドライバを介して，PIC マイコンとのシリアル通信路を確立する．
3. PIC マイコン，Tera Term とともに伝送レート(ボーレート)を 9600[bps]，伝送ビット数を 8 ビット，パリティビットをなしと設定した後，Tera Term 上で任意の文字をキーボードで入力した際，同じ文字が返されることを確認するとともに，オシロスコープで送信(Tx)，受信(Rx)の波形を観察，メモリに保存する．
4. 3. においてパリティビットを奇数(Odd)パリティ，偶数(Even)パリティに変更した際の送受信波形も同様に観察，メモリに保存する．

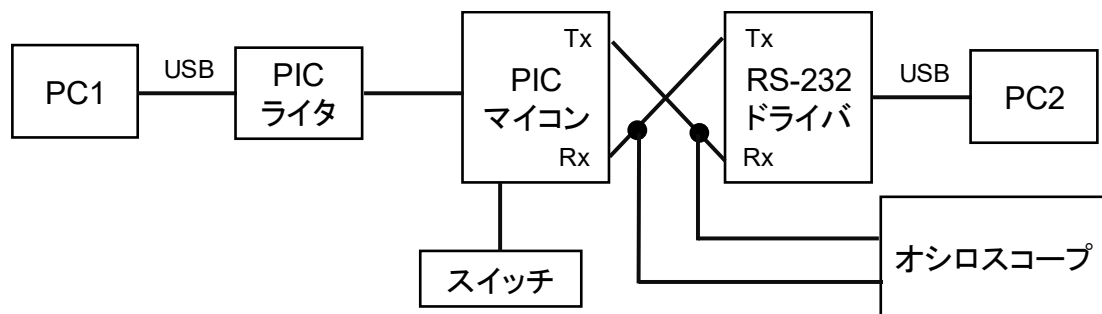


図 1 RS-232C に関する実験の配線図

誤り発生時の RS-232C の伝送特性

1. 前の実験と同様に図 1 に示す配線図を用いる．また，伝送レートを 9600[bps]，伝送ビット数を 8 ビット，パリティビットをなしと設定する．
2. PIC マイコンのプログラム内にある誤り挿入に関する変数によって，1 ビットの伝送誤りが発生するようにセットし，プログラムをビルドした後，実行ファイルを PIC マイコンにダウンロードする．
3. 前の実験と同様に Tera Term 上で任意の文字をキーボードで入力した際，返される文字を確認する．前の実験とは異なり，入力文字と違う文字が返されるはずである．また，オシロスコープで送信(Tx)，受信(Rx)の波形を観察，メモリに保存する．
4. 3. においてパリティビット(奇数か偶数のいずれかで良い)を挿入した際の送受信波形も同様に観察，メモリに保存する．
5. 伝送誤りを 2 ビットに変更して 3. ～ 4. の実験を繰り返す．

RS-232C の伝送レート変化に対する特性

1. 前の実験と同様に図 1 に示す配線図を用いる。また、伝送ビット数を 8 ビット、パリティビットをなしと設定し、以降、これらのパラメータは変更しない。なお、本実験では PIC から PC2 に対して文字列を送信することとする。
2. PC1 の MPLAB で伝送レート変化を測定するためのプログラムを選択、ビルドした後、実行ファイルを PIC マイコンにダウンロードする。
3. PIC マイコン、Tera Term とともに伝送レートを 4800[bps]にセットする。
4. PIC マイコン b-度内に実装されているスイッチ(SW1)を押すことにより、伝送レートが±10%程度変化することを確認する。そして、PIC から送信された文字列が正しく受信できる伝送レートの上限と下限をそれぞれ測定する。
5. 伝送レートを 9600, 19200, 38400[bps]に変更し、4. の実験を繰り返す。

4.2 実験で使用する機材

- ・パソコン 2 台
- ・PIC マイコン(PIC18F4553)
- ・PICKit3 ※PIC ライタとして使用
- ・USB-シリアル変換ケーブル(FT232RL) ※RS-232C ドライバとして使用
- ・ロジックアナライザ(Logic4-Black)

4.3 実験手順

[1] 実験の進め方

~14:30	手順 1：RS-232C を用いたエコーバックの実験 RS-232C の通信フォーマットなどを理解する。
~15:10	手順 2：誤り発生時の RS-232C の伝送特性 パリティビットの役割について理解する。
~16:10	手順 3：RS-232C の伝送レート変化に対する特性 調歩同期式通信の伝送レート誤差の許容範囲について理解する。
~16:25	手順 4：TA による試問

実験室の PC には（各自が持参した）USB を接続しないこと

5. 実験レポート

検討すべき事項（考察事項）

- ① 「RS-232C を用いたエコーバックの実験」において、キーボードで入力した文字と ASCII コードとの対応関係を確認した上で、RS-232C の通信フォーマットがどのようなになっているか考察せよ。また、パリティビットがどのようなルールで生成され、フォーマット中のどの部分に挿入されているか検討せよ。
- ② 「誤り発生時の RS-232C の伝送特性」において、パリティビットが持つ誤り検出能力について考察せよ。
- ③ 「RS-232C の伝送レート変化に対する特性」において、設定した伝送レートの多少の変動に対しても、送受信が可能である理由について考察せよ。また、伝送レートによらず、送受信可能な伝送レート幅が「伝送レート $\pm 5\%$ 」となるのは何故か考察せよ。

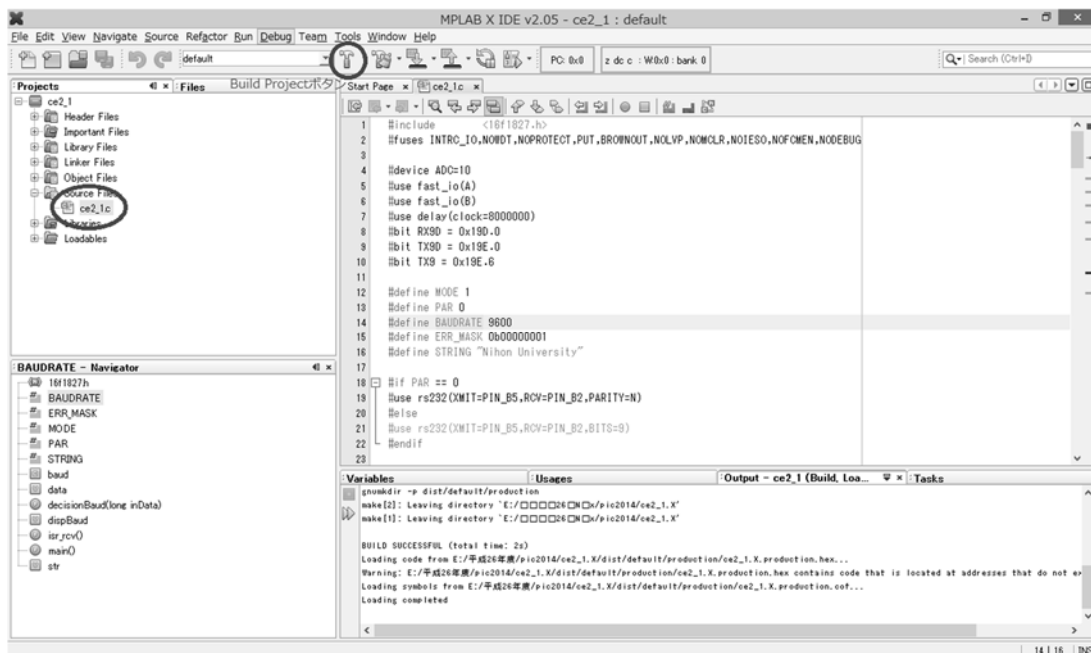
6. 参考文献

- ・日本システムハウス協会エンベデッド技術者育成委員会：“組込みシステム開発のためのエンベデッド技術” 電波新聞社（2006）
- ・中尾司：“マイコンの 1 線 2 線 3 線インターフェース活用入門” CQ 出版社（2008）

補足資料

PIC にプログラムを書き込む流れ

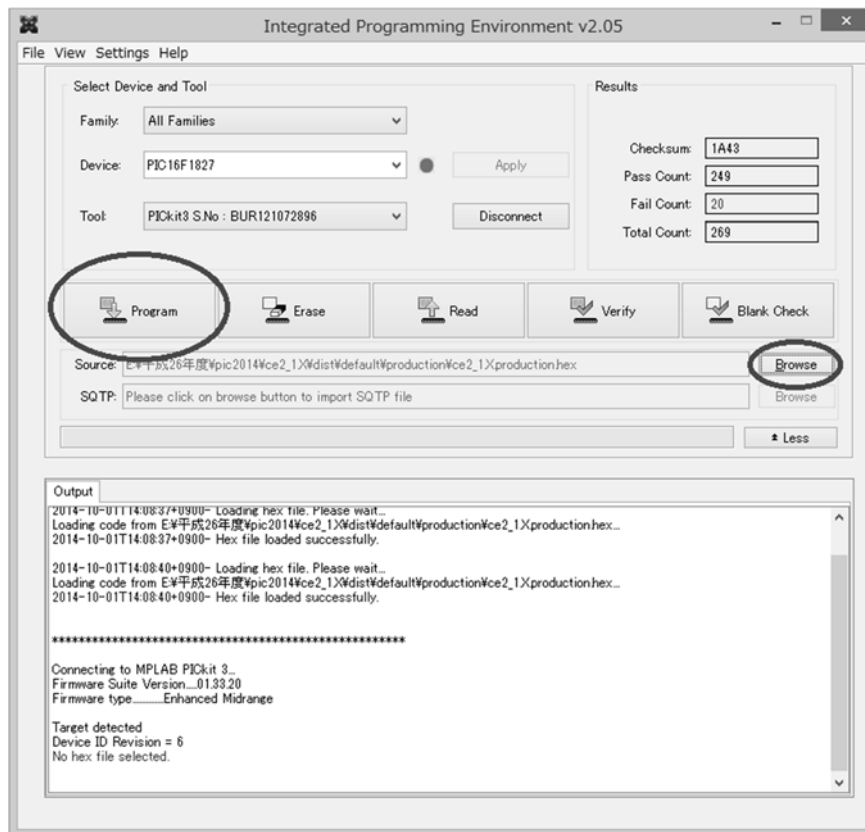
1. PICKit が表向きになるようにして、PIC 基板上に差す。
2. MPLAB X IPE を起動する。
3. Device が PIC18F4553 であることを確認した後、PICKit を USB ケーブルを用いて PC と接続する。
4. MPLAB X IPE 上の Connect ボタンを押して、PIC との接続を確認する。
5. MPLAB X IDE を起動する。
6. [File] -> [Open Project] からプロジェクトフォルダを選択する。
※今回の実験では、「ce2_1.X」(RS-232C 関係) のプロジェクトを使用する。ソフト左上の「Projects」ウインドウにある「Source Files」から C プログラムファイルを選択すると、画面右側にプログラムが表示される。
7. プログラムを変更後、ソフト上の「Build Project」ボタンを押すことで、プログラムをコンパイルできる。もし、プログラムにエラーがなければ、コンパイルは正常に終了する。



8. MPLAB X IPE にソフトを切替, Source の右にある Browse から実行ファイル(HEX ファイル)を選択する。
※このファイルは (プロジェクトフォルダ「ce2_1.X」) ¥dist¥default¥production の中にある。
9. Program ボタンによって PIC に実行ファイルをダウンロードする。
10. プログラムを書き換えた場合は 8.~10.の操作を繰り返す。

実験番号：2-II-3

実験タイトル：インターフェースの実験



Tera Term の設定

○通信路の確立

[ファイル] -> [新しい接続] -> [シリアル] で該当する COM ポートを選択.



○伝送レート，パリティビットなどの設定

[設定] -> [シリアルポート] を選択.

実験番号：2-II-3

実験タイトル：インターフェースの実験



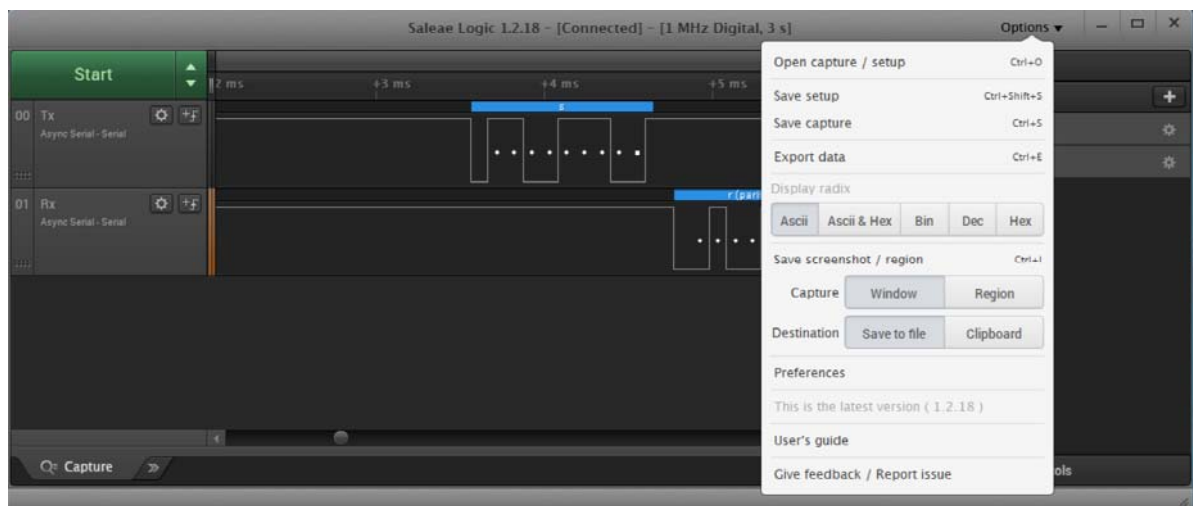
○ログの保存

[ファイル] -> [ログ] を選択後、ファイル名を決定して保存.

ロジックアナライザ (Logic4-Black) の操作

※実験用の設定はあらかじめ保存してあるので、それをロードしてから実験を進めること.

[Options] -> [Open capture/setup] から「rs232cSetup.logicsettings」を選択する.



○信号の計測

左上の **Start** ボタンをクリックする.

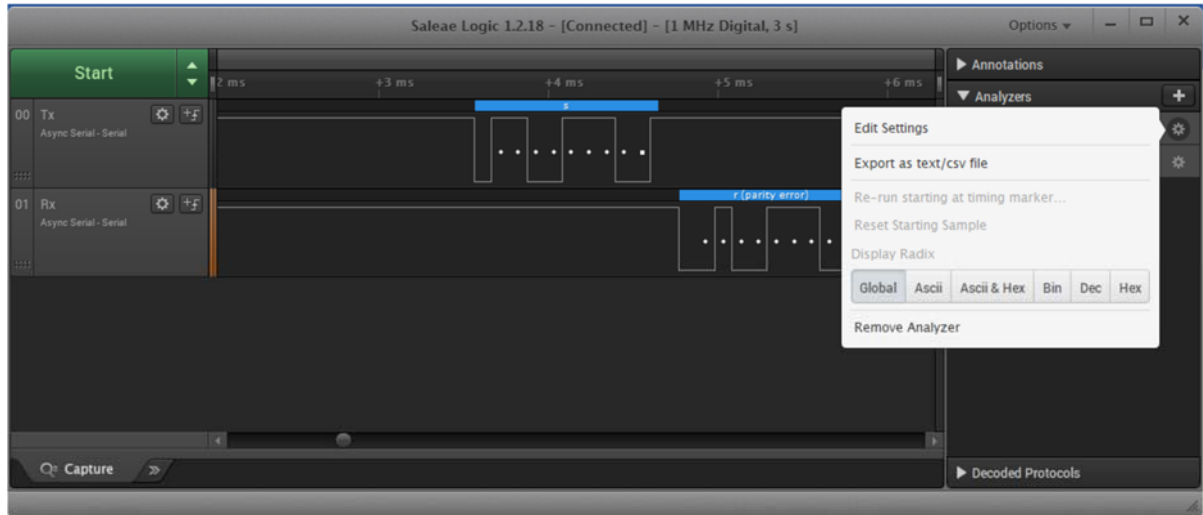
○波形の保存

[Options] -> [Save screenshot / region]を選択し、適当なファイル名で png ファイルを保存する.

○伝送レート，パリティビットなどの設定（左下図参照）

右側にある[Analyzers] -> [Async Serial]にある設定ボタンをクリックした後，[Edit Settings]を選択する．
そして，設定画面上の Bit Rate や Parity Bit を設定する．

※必ず 2 つのチャンネル(Tx, Rx)それぞれで設定すること．



RS-232C を用いたエコーバックの実験

1. 図 1 に示すように配線する．その後，PC1 で PIC の開発ソフトである MPLAB を起動し，エコーバックに関するプログラムを選択，ビルドした後，実行ファイルを PIC マイコンにダウンロードする．
2. PC2 でシリアル通信用のソフトウェアである Tera Term を起動し，通信ポートの設定をして RS-232C ドライバを介して，PIC マイコンとのシリアル通信路を確立する．
3. PIC マイコン，Tera Term とともに伝送レート(ボーレート)を 9600[bps]，伝送ビット数を 8 ビット，パリティビットをなしと設定した後，Tera Term 上で任意の文字をキーボードで入力した際，同じ文字が返されることを確認するとともに，オシロスコープで送信(Tx)，受信(Rx)の波形を観察，メモリに保存する．

※プログラムの設定

```
#define MODE 1
```

```
#define PAR 0
```

```
#define BAUDRATE 9600
```

```
#define ERR_MASK 0b00000000
```

※操作

ロジックアナライザを Run させた後，測定終了時間（2 秒程度）以内にキーボードで文字入力する．

4. 3. においてパリティビットを奇数(Odd)パリティ，偶数(Even)パリティに変更した際の送受信波形も同様に観察，メモリに保存する．

※プログラムの設定（奇数，偶数ともに）「#define PAR 1」に変更する．

誤り発生時の RS-232C の伝送特性

1. 前の実験と同様に図 1 に示す配線図を用いる。また、伝送レートを 9600[bps]、伝送ビット数を 8 ビット、パリティビットをなしと設定する。

※プログラムの設定「#define PAR 0」に戻す。

2. PIC マイコンのプログラム内にある誤り挿入に関する変数によって、1 ビットの伝送誤りが発生するようにセットし、プログラムをビルドした後、実行ファイルを PIC マイコンにダウンロードする。

※プログラムの設定「ERR_MASK のいずれかのビットを 1」とする。

3. 前の実験と同様に Tera Term 上で任意の文字をキーボードで入力した際、返される文字を確認する。前の実験とは異なり、入力文字と違う文字が返されるはずである。また、オシロスコープで送信(Tx)、受信(Rx)の波形を観察、メモリに保存する。

4. 3. においてパリティビット(奇数か偶数のいずれかで良い)を挿入した際の送受信波形も同様に観察、メモリに保存する。

※プログラムの設定（奇数、偶数ともに）「#define PAR 1」に変更する。

5. 伝送誤りを 2 ビットに変更して 3. ～ 4. の実験を繰り返す。

※プログラムの設定「ERR_MASK のいずれかの 2 ビットを 1」とする。

RS-232C の伝送レート変化に対する特性

1. 前の実験と同様に図 1 に示す配線図を用いる。また、伝送ビット数を 8 ビット、パリティビットをなしと設定し、以降、これらのパラメータは変更しない。なお、本実験では PIC から PC2 に対して文字列を送信することとする。

2. PC1 の MPLAB で伝送レート変化を測定するためのプログラムを選択、ビルドした後、実行ファイルを PIC マイコンにダウンロードする。

3. PIC マイコン、Tera Term ともに伝送レートを 4800[bps]にセットする。

※プログラムの設定

```
#define MODE 2
```

```
#define PAR 0
```

```
#define BAUDRATE 4800
```

```
#define ERR_MASK 0b00000000
```

4. PIC マイコンに外付けされたボリュームを変化させることにより、伝送レートが±10%程度変化することを確認する。そして、PIC から送信された文字列が正しく受信できる伝送レートの上限と下限をそれぞれ測定する。

※ボリュームは基板に実装済で、プラスドライバにより調整可能である。

5. 伝送レートを 9600, 19200, 38400[bps]に変更し、4. の実験を繰り返す。

※プログラムの設定「BAUDRATE を上記の伝送レートにそれぞれ設定」する。