シリアル通信 実習

目的
マイコン内蔵機能のUART0を使いシリアル通信について,
マイコンとパソコンでRS-232C 通信を行い学習する。

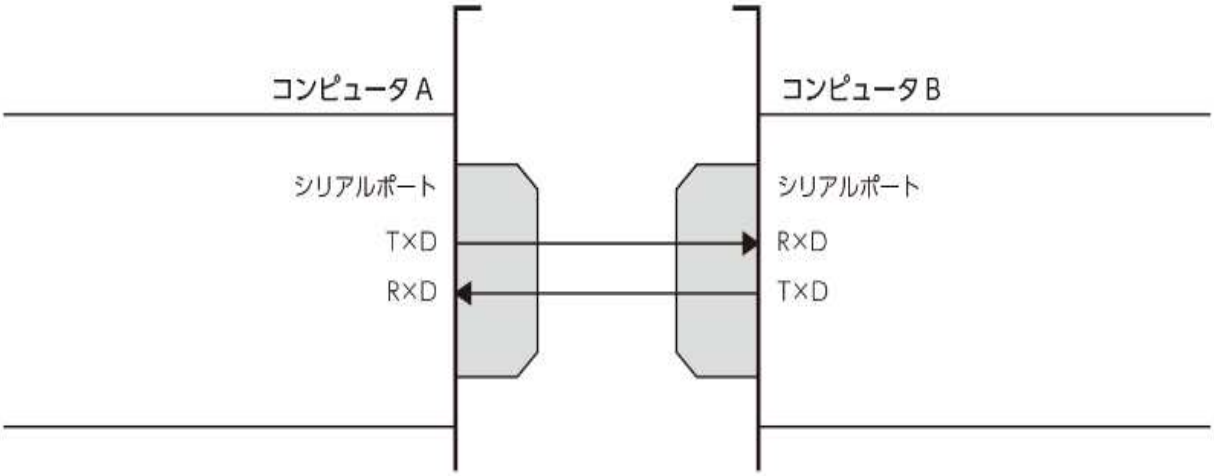
関係知識

シリアル通信の基本

- 多数の機器が通信を行うネットワークの仕組みを見る前に、もっとも基本となる2台の機器の間でのシリアル通信について考えてみましょう。
- コンピュータなど、通信を行える機器を相互に接続してデータのやり取りを行いたい場合、もっとも簡単なやり方は、2台の機器を1本の通信ケーブルで接続するというものです。例えばパソコンであれば、シリアルポートを使ってこのような接続を行うことができます

関係知識

- 2台のパソコンの接続



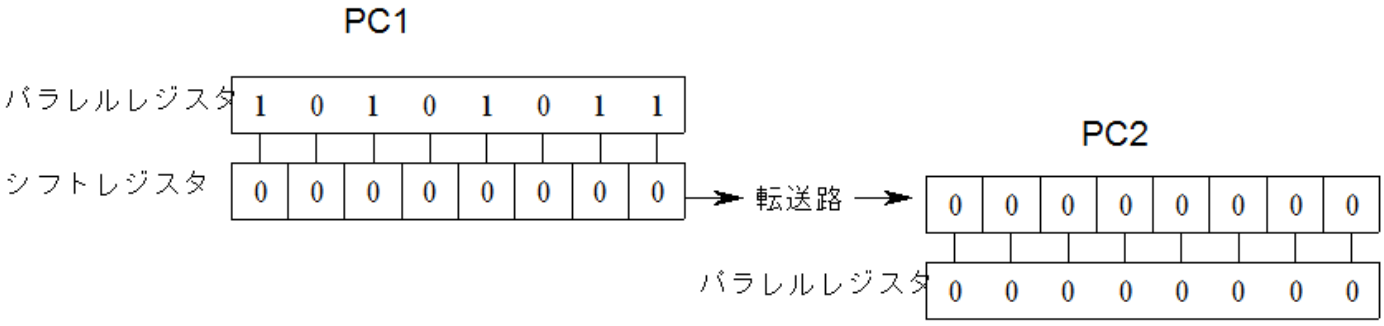
関係知識

- **シリアル通信の基本**
  - シリアルポートは、もともとはモデム（アナログ電話回線を使ってデータ通信を行うためのアダプタ）などの通信機器を接続するためのインターフェイス ですが、ネットワークの普及でモデムをほとんど使わなくなったこと、より汎用的なUSBが一般化したことなどで、最近のパソコンの多くはシリアルポートを 備えていません（シリアルポートを使いたい場合は、USB−シリアル変換アダプタなどを使います）。しかしもっとも基本的なシリアル通信インターフェイス ということで、ここではシリアルポートを例に説明していきます。

関係知識

シリアル通信とは

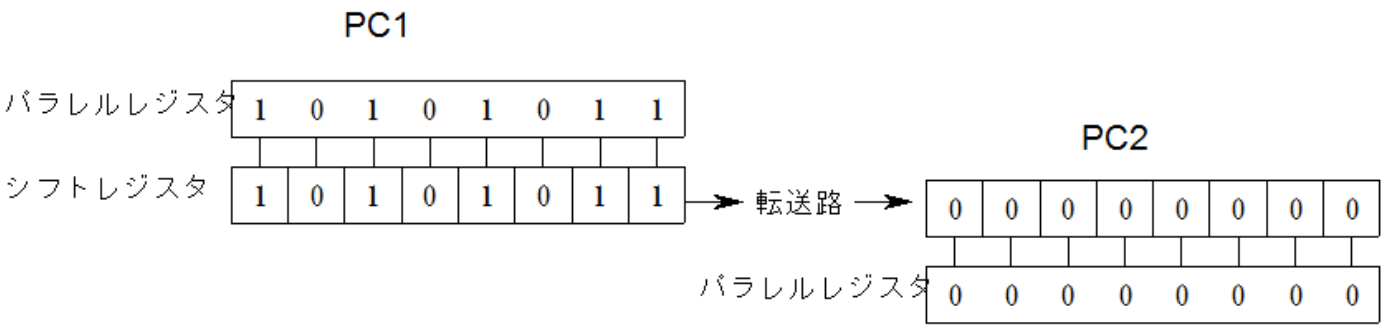
- PC1のパラレルレジスタからシフトレジスタへ



関係知識

シリアル通信とは

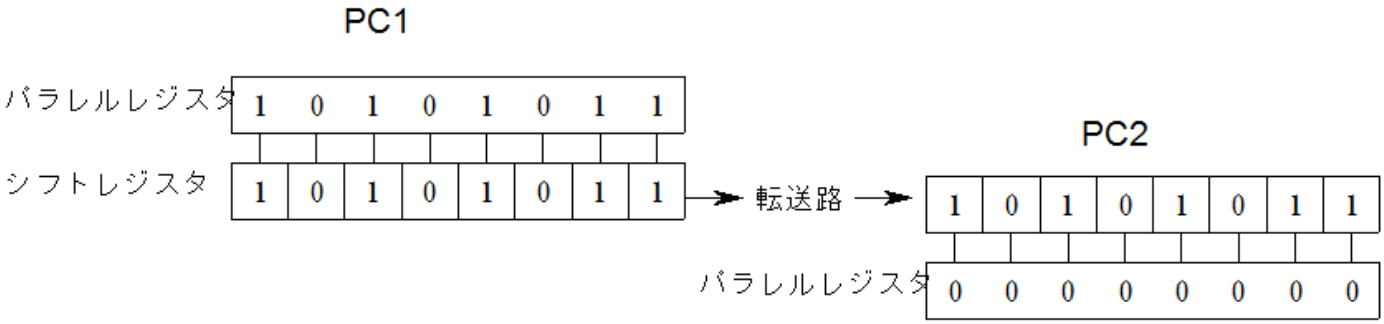
- PC1のパラレルレジスタからシフトレジスタへ



関係知識

シリアル通信とは

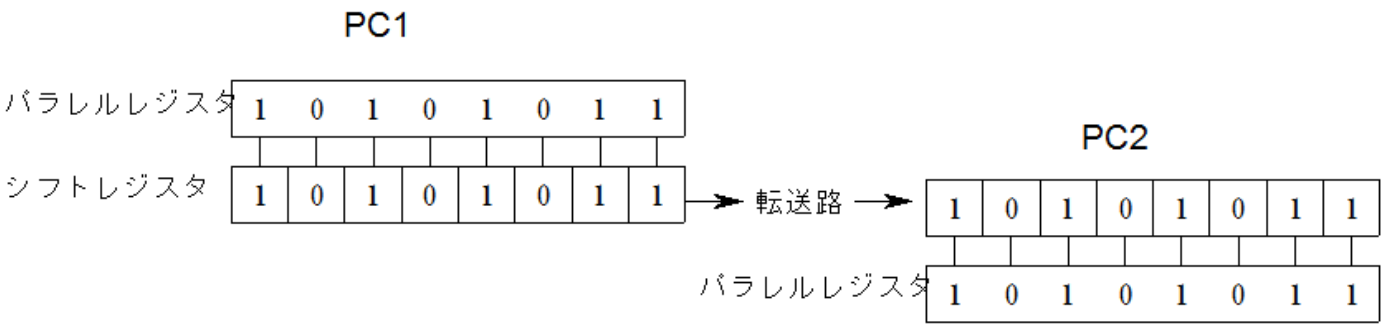
- クロックが送られPC2のシフトレジスタへ転送



関係知識

シリアル通信とは

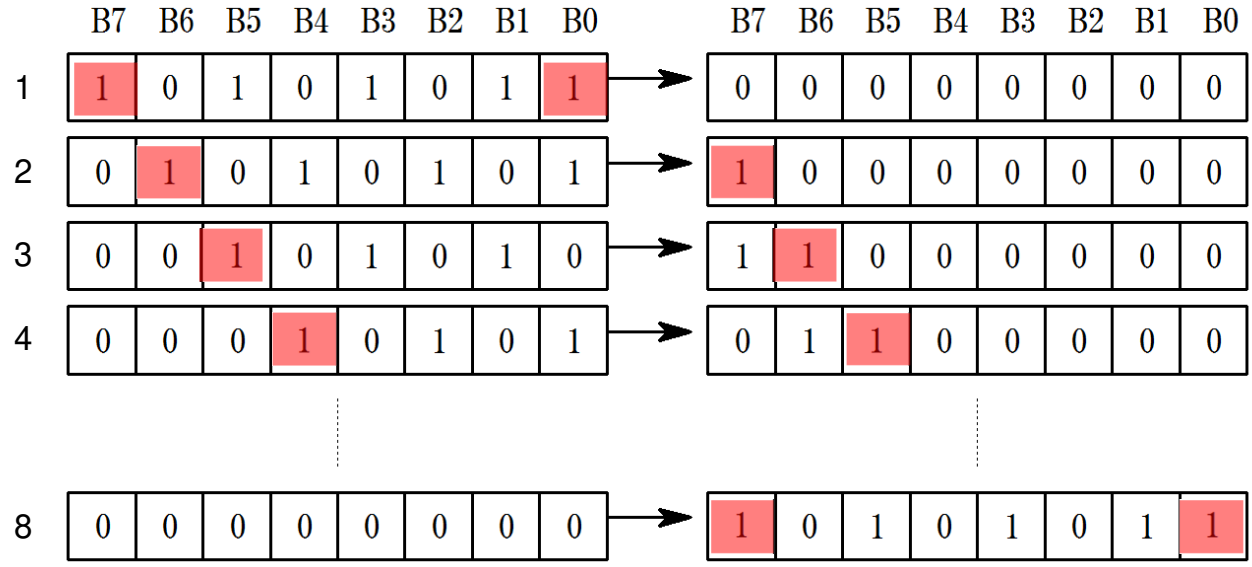
- PC2のシフトレジスタからパラレルレジスタへ転送終了



関係知識

シリアル通信とは

- クロックが入力される度に1ビットシフトと転送される



関係知識

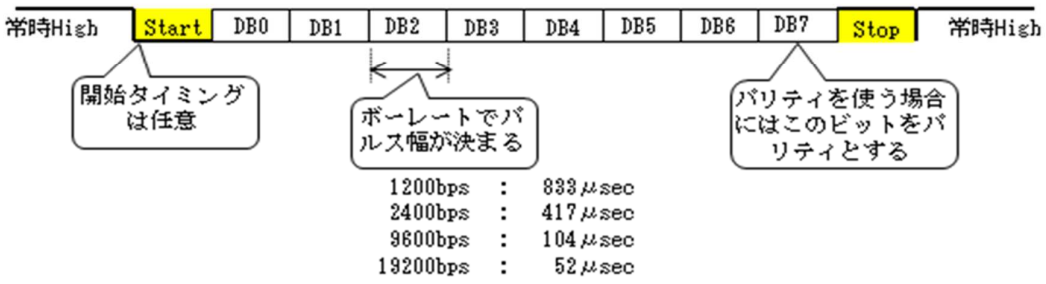
シリアル通信の約束

- シリアル伝送を行う際は、送信側と受信側で、いくつかの点について合意していなければなりません。送信側が送りだした信号を、受信側が正しく解釈できなければ、正しいデータ通信を行えません。いくつか例を挙げましょう。
  - **電気信号や光信号の形式** 論理形式　信号形式
  - **データのサイズや送信の順序** データのビット構成
  - **データを送る速度** 通信速度　（bps）
  - **送受信の同期** 非・同期（クロック信号）
  - **伝送エラーの検出　フロー制御**

関係知識

USART通信フォーマット

《通信データフォーマット》
調歩同期方式の基本のデータ転送はバイト単位で行われ、下図のフォーマットで1ビットずつが順番に送信されます。



1200bps : 833μsec
2400bps : 417μsec
9600bps : 104μsec
19200bps : 52μsec

10ビット分の時間の誤差が許容範囲内であれば正常に通信できることになります。
許容誤差は以下のようになります。
1ビットの読み取りは通常ビットの中央で行われますのでこのより込位置が1/3ビット
つまり30%のずれまでは許容するとすれば時間誤差は３%と言うことになります。
送信側と受信側で逆方向にずれている最悪の場合を考えて±1.5%と言うことになります。

関係知識

UARTについて

- シリアル通信の複雑な信号処理をCPUに替わって処理を行うのがUART（Universal Asynchronous Receiver Transmitter）です。古くから使われている基本のシリアル通信方式をサポートするモジュールです。
- クロック信号をデータと別に用意するのが、同期式でUSART（Universal Synchronous Asynchronous Receiver Transmitter）と言います。

  両方をサポートしているので単にURATと言っています

- 非同期といっても、送信側・受信側が正確な時計を持ってタイミングよく通信を行っています。

関係知識

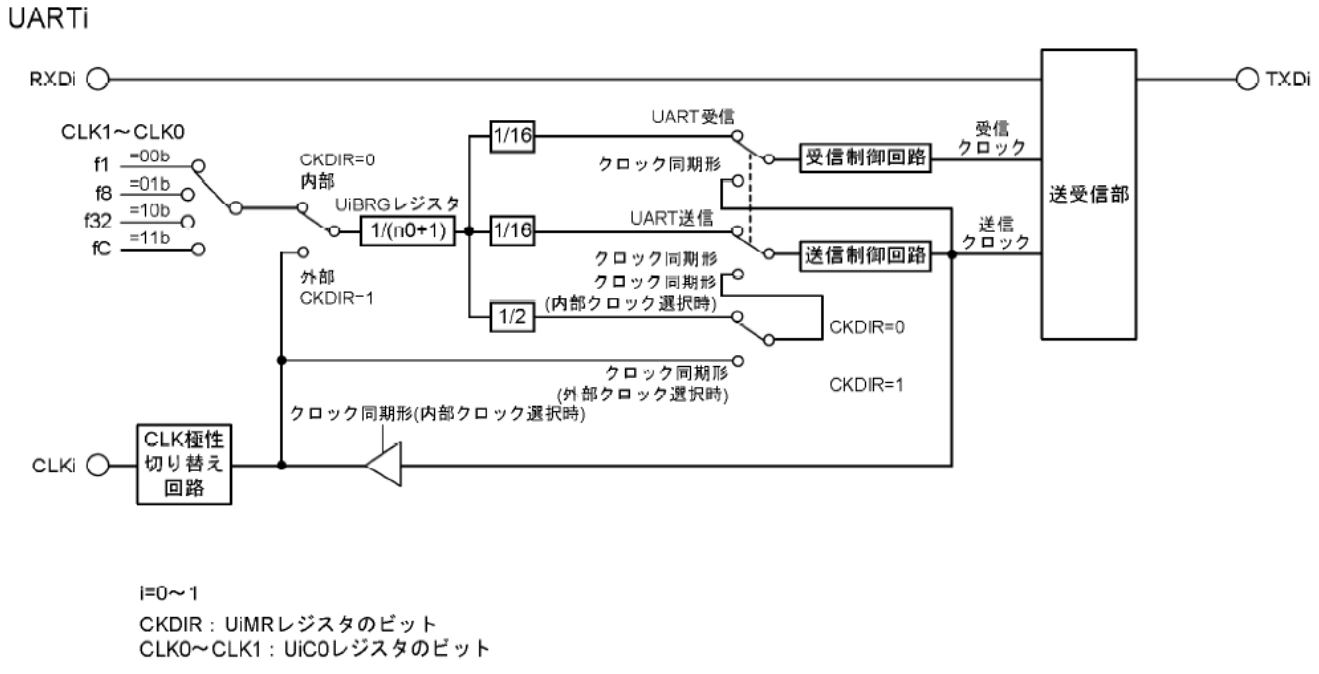
UARTについて

- シリアル通信の複雑な信号処理をCPUに替わって処理を行うのがUART（Universal **Asynchronous** Receiver Transmitter）です。古くから使われている基本のシリアル通信方式をサポートするモジュールです。
- クロック信号をデータと別に用意するのが、同期式でUSART（Universal **Synchronous** *Asynchronous* Receiver Transmitter）と言います。

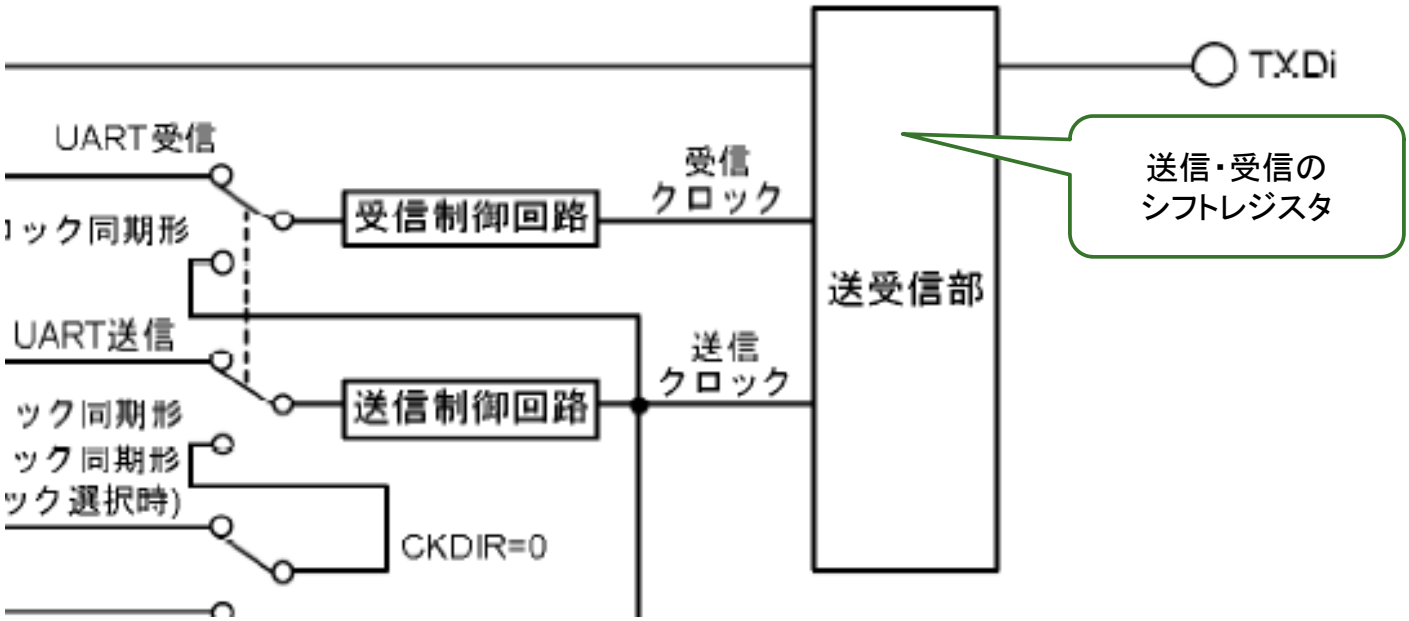
  両方をサポートしているので単にURATと言っています

- 非同期といっても、送信側・受信側が正確な時計を持ってタイミングよく通信を行っています。

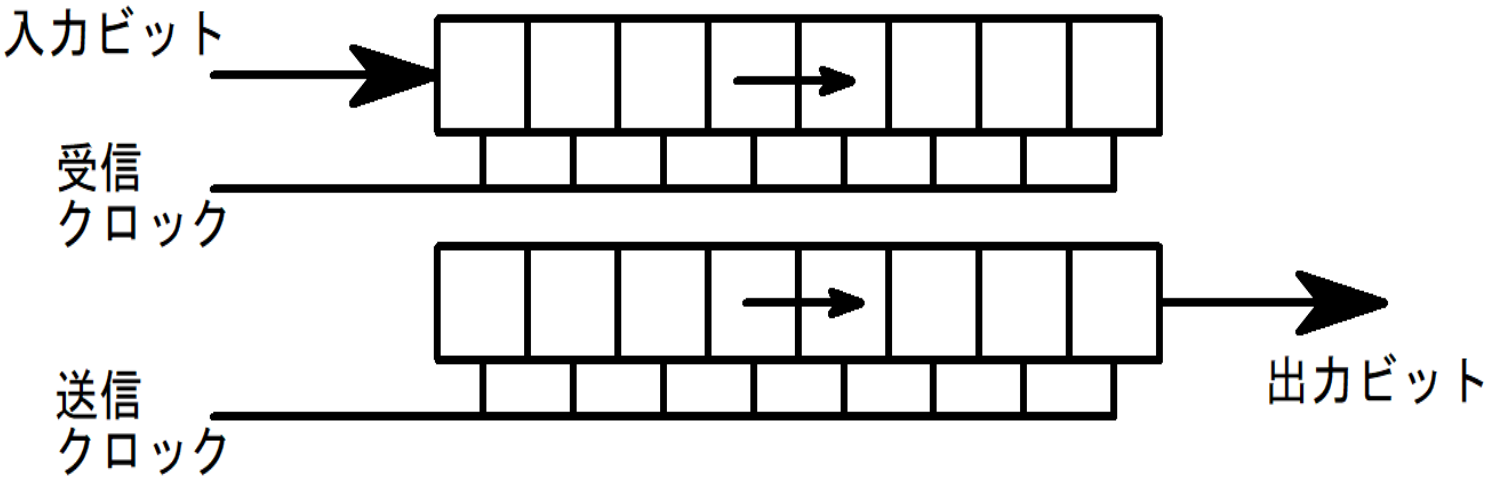
R8のUART構成図



R8のUART構成図

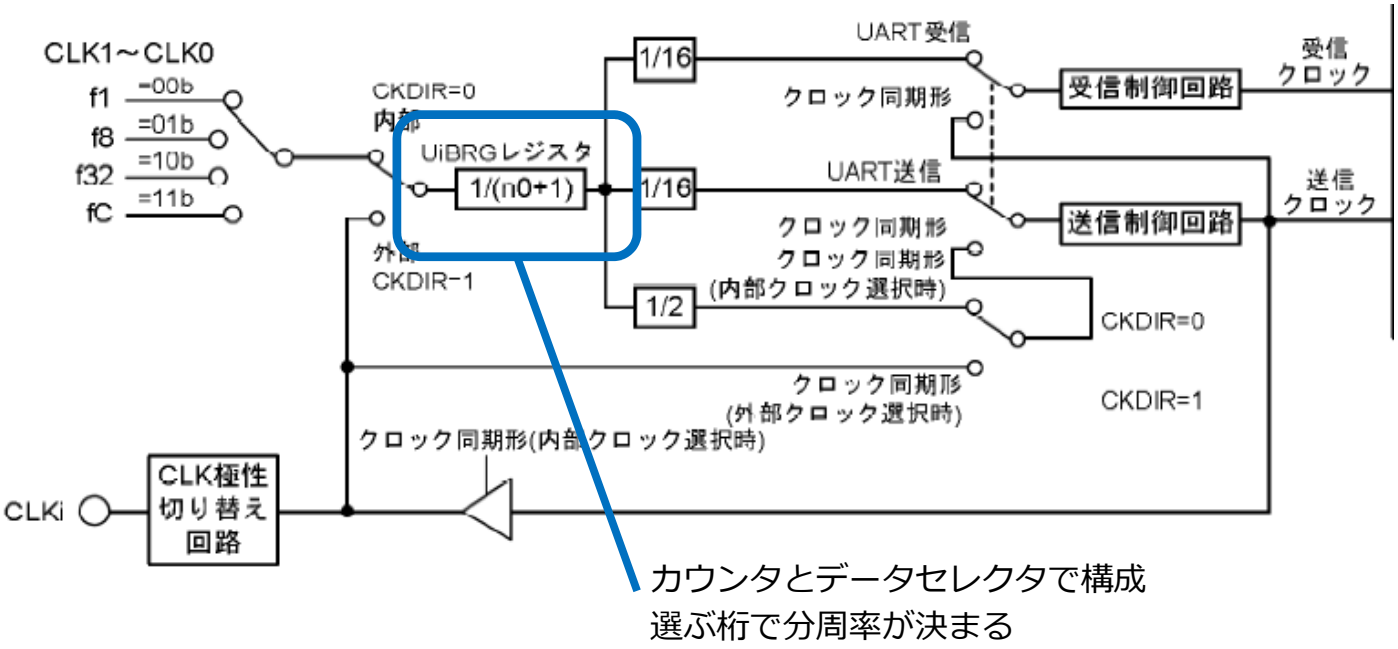


R8のUART構成図　送受信部



同一のクロックを送受信クロックとすると同時にシフトする

R8のUART構成図　クロック生成



R8のUART使用端子

端子名	割り当てる端子	入出力	機能
TXD0	P1_4	出力	シリアルデータ出力
RXD0	P1_5	入力	シリアルデータ入力
CLK0	P1_6	入出力	転送クロック入出力
TXD1	P0_1またはP6_3	出力	シリアルデータ出力
RXD1	P0_2またはP6_4	入力	シリアルデータ入力
CLK1	P0_3、P6_2またはP6_5	入出力	転送クロック入出力

関係知識

UARTの初期化

```
/* UART0の設定 */
  /* U0BRG = ｶｳﾝﾄｿｰｽの周波数/(設定したいbps*16)-1
        = 20MHz        /(    9600    *16)-1
        = 129.208 = 129
  */
  u0sr = 0x05;                  /* P14=TXD0,P15=RXD0に設定      */
  u0c0 = 0x00;                  /* カウントソースなどの設定     */
  u0c1 = 0x05;                  /* 送信、受信許可               */
  u0brg = 129;                  /* 通信速度=9600pbs             */
  u0mr = 0x05;                  /* UART0 ﾃﾞｰﾀ長8bit 1ｽﾄｯﾌﾟﾋﾞｯﾄ  */
```

関係知識

printfの導入

printf文を導入するためにマイコンの初期化後uart・print文の初期化をinit_uart0_printfで行います。割り込みを使用するので割り込み許可を掛けます。これでprintf・scanfが使用できます。

```
init();                         /* SFRの初期化                  */
init_uart0_printf( SPEED_9600 );    /* UART0とprintf関連の初期化    *
asm(" fset I ");                /* 全体の割り込み許可           */

printf( "Hello World!¥n" );
```

関係知識

scanfの注意事項

```
while( 1 ) {
      printf( "Input data : " );
      ret = scanf( "%d", &i );
      if( ret == 1 ) {
          printf( "Get data : %d¥n", i );
          p6 = i;
      } else {
          printf( "Data Error!!¥n" );
          scanf( "%*[^¥n]" );
      }
  }
```

scanfには致命的な欠点があります。
入力型と書式指定が違うと永久ループに入ってしまいます。
そこで、読み込み終わってもバッファにデータが残っているときはデータエラーと判断してすべての型を読み込みバッファをクリアします。

課題

① urat0.cを使って、キーボの1文字を拡張ボード8ビットにアスキーコードで表示しなさい。

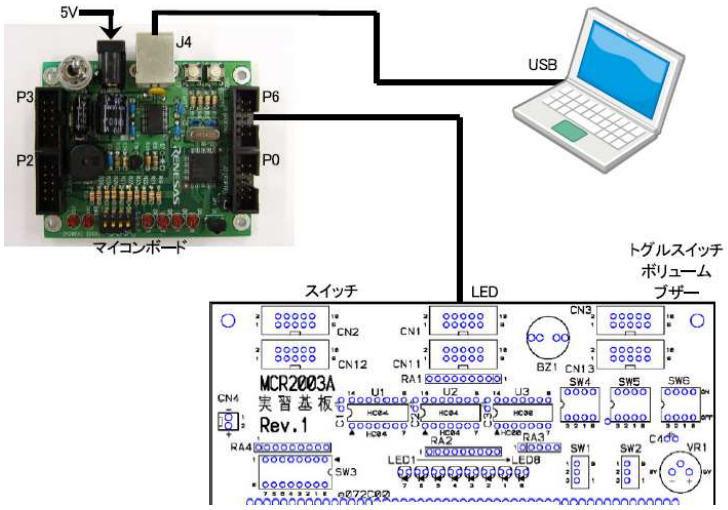
② 拡張ボードスイッチのデータを１０進数でＰＣに表示させなさい。

③ Ａ／ＤコンバータのデータをＬＥＤに8ビットでＰＣに１０進数（０～１０２４）で表示させなさい。完全に０又は１０２４にはならなくても良い。

④ 全二重・半二重通信とエコーバック・ローカルエコーバックについて調べなさい。またどのような場面で有効か考えなさい。

①課題　接続と実習

- 図のように結線し
- 書き込み後、PCのTeraTeramで確認します。



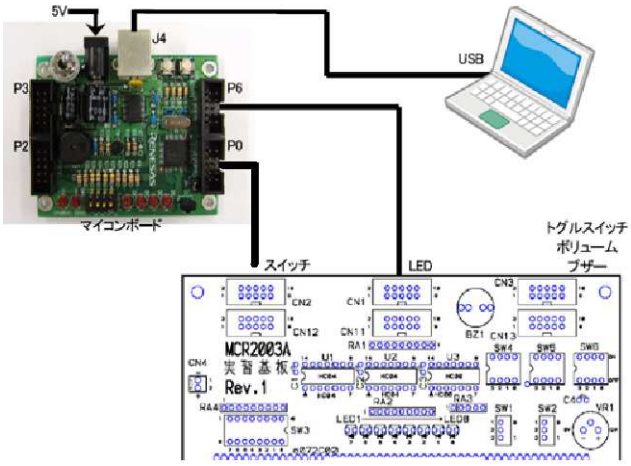
①課題　接続と実習

urat0.cを使って、キーボの1文字を拡張ボード8ビットにアスキーコードで表示しなさい。

I. ポート６を出力に設定
II. UART０から1バイトデータを取得
III. ポート６へデータの出力
IV. 以上のプログラムをR8マイコンボードで準備し、ダウンロードする。
V. TeraTeramの起動 R8マイコンをRUNモードで再起動する。キーボードを打つと拡張ボードのLEDが表示される。

②課題　接続と実習

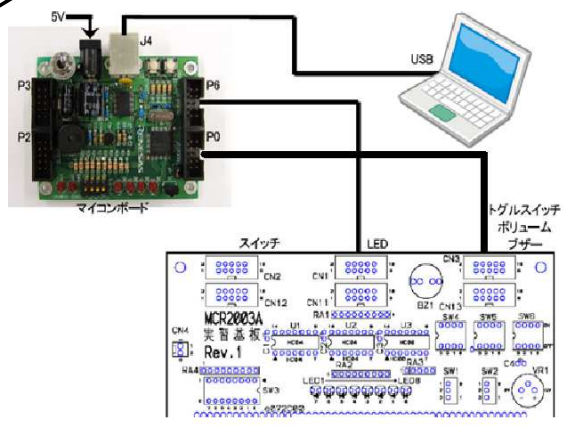
- 図のように結線し
- 書き込み後、拡張ボードのディップスイッチの値をPCで表示し、LEDにも表示する。



③課題　接続と実習

- 図のように結線する.
- A/Dが使えるように切替え用ディップスイッチをONにする.
- ad.cのプロジェクトからget_ad7(void);サブ プログラムに追加LEDに表示できる事を確認する.
- PrintfでPCに転送し、課題１・２同様TeraTeramで確認します.

ad.cにprintfを移すより簡単.



レポートについて

タイトル　R8実習4-UART

目的　マイコン内蔵機能のUART0を使いシリアル通信について学びマイコンとパソコンで
RS-232C 通信を行い理解を深める。

関連知識
- シリアル通信の基本
- シリアル通信の約束
- USART通信フォーマット
- R8のUART構成図（簡単な図　切り張り可）

課題1

接続と実験

① urat0.cを使って、キーボードの1文字を拡張ボード8ビットにアスキーコードで表示しなさい。
（ Key→uart→ポート出力）

② 拡張ボードスイッチのデータを10進数でPCに表示させなさい。
次のどちらかで
- prinntf関数を利用（低難易度）
- コード変換を行う（高難易度）
(数値を10進数各桁に分解(BCD化）各桁の数値（0~9）に‘1’を加算しUARTに送る）

③A／Dコンバータのデータを LEDに8ビットでPCに10進数（0~1024）で表示させなさい。
（ 電源の問題で、完全に0又は1024にはならない可能性あり、トリマーの回転と数値が
比例対応すること。一定に動かした場合同じようにAD値が増加していればよい。）

レポートについて

考察

UARTの有用性、実習中の問題点等考察しなさい。

紙レポートには、接続図等どのように実習したかきっちりと書き込むこと

メール課題は、課題③ ソースのみ添付する。

プロジェクトを添付すのは不可

メール課題の提出先　n-masuya@hamako-ths.ed.jp

件名　R8実習4-UART実習(ei3〇〇)　※〇〇は出席番号