

展示用波長多重光通信システムの設計と構築

Design and Trial Manufacture of a Wavelength Division Multiplexed Optical Transmission System for Exposition

袴田 吉朗*

Yoshiro HAKAMATA

Abstract: The paper summarizes the design of a Wavelength Division Multiplexed Optical Transmission System for Exposition. It consists of a plastic optical fiber, a personal computer, PIC micro computers and optical filters. A Green and a Red LED are used for optical transmitters. The system hardware configuration and software configuration are precisely described in the paper.

1. はじめに

多くの高校生に電気電子に対して興味を持って貰うためには、"動く", "光る", "音が出る"などの要素を取り入れた展示物が効果的ではないかと考えている。この観点から昨年度には文字が"光り"ながら"動く", という2つの要素を取り込んだ漢字表示電光掲示板(以下漢字ディスプレイ)を設計, 試作し [1], いろいろな所で使用してきて効果がありそうな感触を得た。

一方, 3年前にプラスチック光ファイバ(POF)を用いた低速の光通信システムを作り, オープンキャンパスの展示において使用してきた。

今回両者を組み合わせることにより, オープンキャンパスに来場された高校生に「高校名」を教えて貰い, その「高校名」を光通信システムにより伝送し, 漢字ディスプレイに表示するようにすればよりインパクトがあり, 電気電子に対して今までよりも興味を持ってもらえるようになるのではないかと考えたのが, 本システムを構築することにしたきっかけである。

幸いにも所望の波長フィルタを入手することができたので, 従来の光通信システム(以下緑色通信系)をそのまま生かして, 新たに赤色通信系を製作し波長多重光通信システムとして構築することにした。本論文では今回試作した波長多重光通信システムの設計および試作内容をとりまとめた結果を示す。

2. システムの概要

2.1 概要

図2.1にシステム全体の構成を示す。本システムは, 緑色通信系(green)と赤色通信系(red)を2つの波長フィルタで合成・分離し, 1本のプラスチック光ファイバ(POF)を用いて伝送する波長多重光通信システムである。送信器(以下TX), 受信器(以

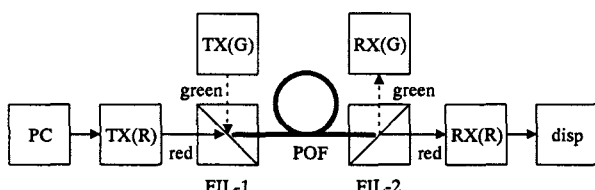


図2.1 システム全体の構成

下RX), 漢字ディスプレイ(disp)などを構成する主要部品にはPICマイコンを使用している。

2.2 緑色通信系

TX, RXともPIC16F84Aにより構成している。TX内部に格納した下記のデータをSW切り替えにより選択して送信する。

- ① Welcome to DENKIDENSHI KOUGAKUKA (メモリに格納)
- ② QUICK FOX JUMPS OVER THE LASY DOG (EEPROMに格納)

図2.2に示すように, 8ビットのデータを1ビットのスタートビットおよび1ビットのストップビットで挟み込んで伝送する調歩同期方式を用いて送受信を行っている。1ビット時間は100ms, すなわち伝送速度は10baudであり, 伝送速度を取って遅くしてLEDの点滅がよく分かるように配慮した。TXでは, データの送信に先立ちフレーム同期パターンとして5個の* (あるいはSW切り替えにより7FH)を送信するようにし, 受信データを液晶ディスプレイに表示するときにフレーム同期が取りやすいようにした。

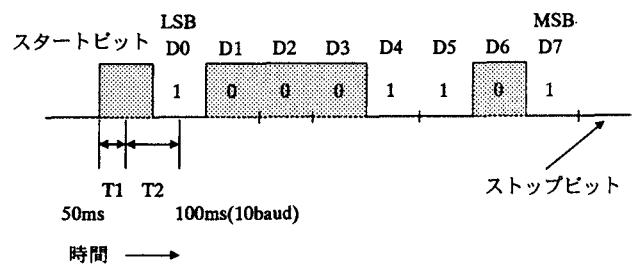


図2.2 緑色通信系における伝送路フォーマット

RXにおいて1バイトのデータを受信する手順は以下の通りである。なおプログラムはCCS社のCコンパイラを用いて作成し, RS232Cの送受信はC言語の命令を使用してソフトウェア的に実現した。

- (a) ポートAのRA4にフォトトランジスタTPS601Aを接続しており, RA4出力をセンスしてスタートビットがHになるのを待つ。PICマイコンのクロック周波数は20MHzである。
- (b) RA4=Hを検出したら半ビット時間である50ms(T1)だけ待ち, その後1ビット時間100ms(T2)待つ毎に8ビットのデータをLSBから順次受信し, 変数rxdに格納する。

2010年2月5日受理

*理工学部 電気電子工学科

- (c) 8ビットのデータ受信が終了したら、1ビット時間(100ms)待ち、RA4=Lによってストップビットを確認し1バイトの受信を終了する。
- (d) 受信したデータを20×2行のLCDキャラクタディスプレイに表示させる。
- (e) なおRXにおけるPICをリセットするとフレーム同期パターンの受信に移行し、5個のフレーム同期パターンを受信後データの受信が可能になる。

図2.3は、使用した波長フィルタの波長-透過率特性を示している。遷移帯域は560nm~600nmであり、緑色(525nm)における透過率はほぼ0である。また赤色(660nm)における透過率は0.98である。したがって緑色のLED光が反射、赤色のLED光が透過するように、POFに対して波長フィルタを45°の角度をなして固定する治具を作製して使用した。

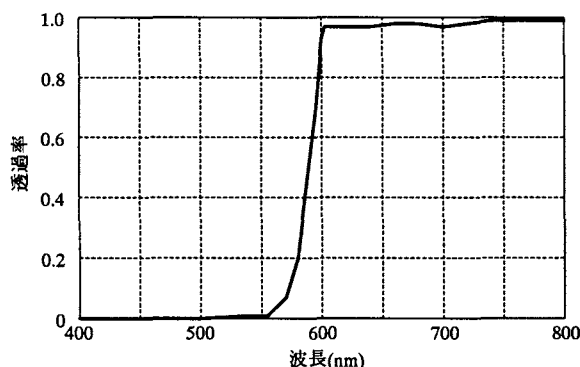


図2.3 使用した波長フィルタの波長-透過率特性

2.3 赤色通信系

PC, TX, RX および漢字ディスプレイ, 赤色 LED, フォトトランジスタ, 波長フィルタおよびPOFなどからなる。

通信には、PICマイコン(PIC16F648A)およびPCにインプリメントされているUART(調歩同期通信)機能を使用した。またPCで使用したRS232C通信は、インターネットからダウンロードした「C++におけるRS232C通信クラス」を基にして一部変更を加えて実現した。

(1) PC

以下の2つのプログラムを作成して使用した。

- 漢字データ作成プログラム kanji_input.exe
- 漢字データ送受信プログラム hterm.exe

hterm.exeを起動してマウスを右クリックすると、ShellExecute命令を用いてkanji_input.exeを実行し、ファイルkanji_data.iniに格納した漢字データを読み込むことができる。

(2) TX, RX および漢字ディスプレイ

最初は、PCから送信した高校名のデータをPIC内部のEEPROMに書き込んだ後、このデータを順次バケツリレー的に受信側に伝送する方式を検討した。漢字ディスプレイでは、EEPROMに漢字データを書き込んだ後、漢字個数の値を変数NAME_MAXに格納し、データの表示に移行する。

伝送速度は以下のようにになっている。

- PC-TX間 19.2kbaud

- TX-RX間 19.2kbaud
- RX-漢字ディスプレイ間 19.2kbaud

しかし、この方式ではPCにおいて高校名を送信してから、漢字ディスプレイが表示を開始するまでに数10秒の時間がかかった。このため最終的にはRXにおいて受信データをEEPROMに書き込むのを止め、受信データを単にトランスバレイメントに漢字ディスプレイに送信する方式に変更した。

3. ハードウェア構成

3.1 TX

図3.1にTXの回路図を示す。ポートBにおけるRB3の出力によりNANDゲート(74HC00)を用いた2-1セレクタを制御し、RB3=Lの間は、RB2出力(RS232C出力)をPCに返送する。一方TXのEEPROMに漢字データが書き込まれるとRB3=Hになり、RB2出力(RS232C出力)によって赤色LEDを点滅させ、POFを介してRXに向けて光信号を送信する。

光信号を送信中に新しい高校名を受信すると、RS232Cの受信割り込みPIRがアクティブになる。この結果図4.4のフローチャートに示すように変数write_flag=1となる。このwrite_flag=1の結果を受けてRB4がLとなる。この信号を、100pFと10kΩから成る微分回路を介してNANDゲートを用いたアースクランプ型の単安定マルチバイブレータに入力し、入力信号の立下りエッジを用いてトリガをかけ、PICのリセット信号(9ms幅)を生成する。このリセット信号によってPICをリセットし、新しい高校名の受信に移行するようにしている。

MAX232AはRS232Cのレベル変換用のICであり、PCから送信される±15Vの信号をTTLレベル(+5V)に変換している。2番ピン(RxD)にT線を、3番ピン(TxD)にR線を接続することによりクロス接続を実現している。またRTS線はCTS線に、DTR線をDCD線およびDSR線に折り返している。

3.2 RX

図3.2にRXの回路図を示す。フォトトランジスタTPS601Aに1kΩの負荷抵抗を直列接続し、全体に5Vの逆バイアスをかけて広帯域化を図り受光系を構成している。負荷抵抗の値は19.2kbaudの伝送速度に対して十分な帯域が得られるように、カットアンドトライによって調整して1kΩに決定した。後置したコンパレータLM311Pによって論理振幅を5Vにまで増幅しているが、オペアンプを用いればしきい値の調整が不要となるためベターであったかも知れない。

当初は、漢字データに先立って伝送する図4.5に示すヘッダーフレーム(58個の*, number(2B), データ個数(2B)および終結デリミタ7Eh)において、フィールドnumの値が更新されたときに「新しい高校名」を検出できたとするプログラムを検討した。しかし、この方法では理由は不明であるが、numの差分を再現性良く検出することができず断念した。

そこで「新しい高校名」を送出するのに先立ち、TXからRestart信号を送信するようにし、この信号を検出したときに「新しい

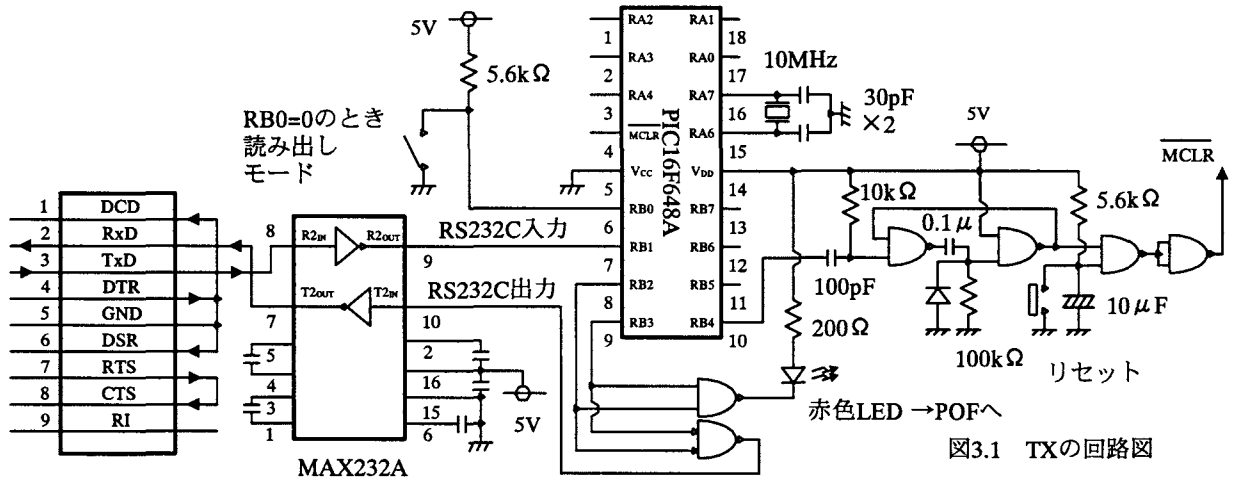


図3.1 TXの回路図

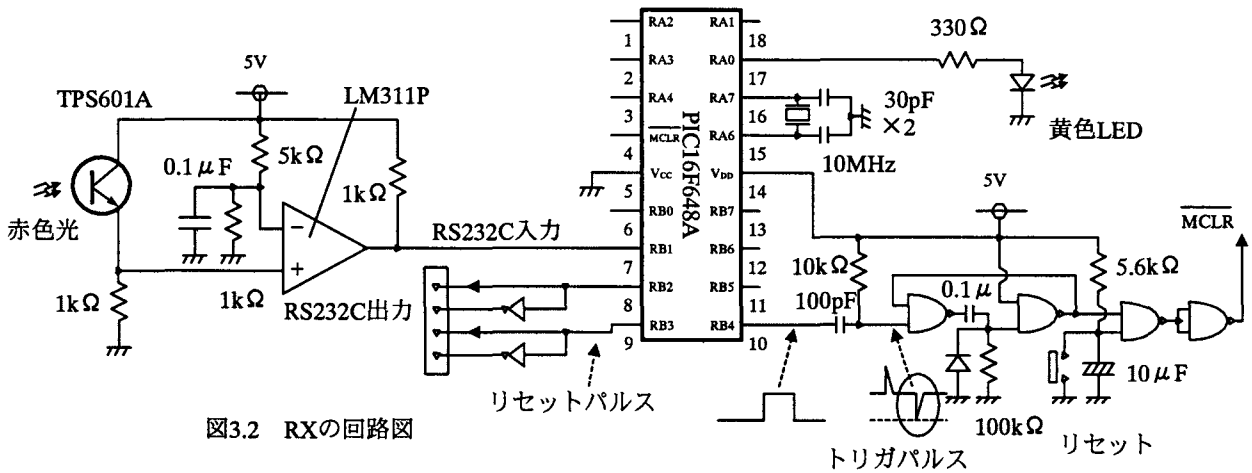


図3.2 RXの回路図

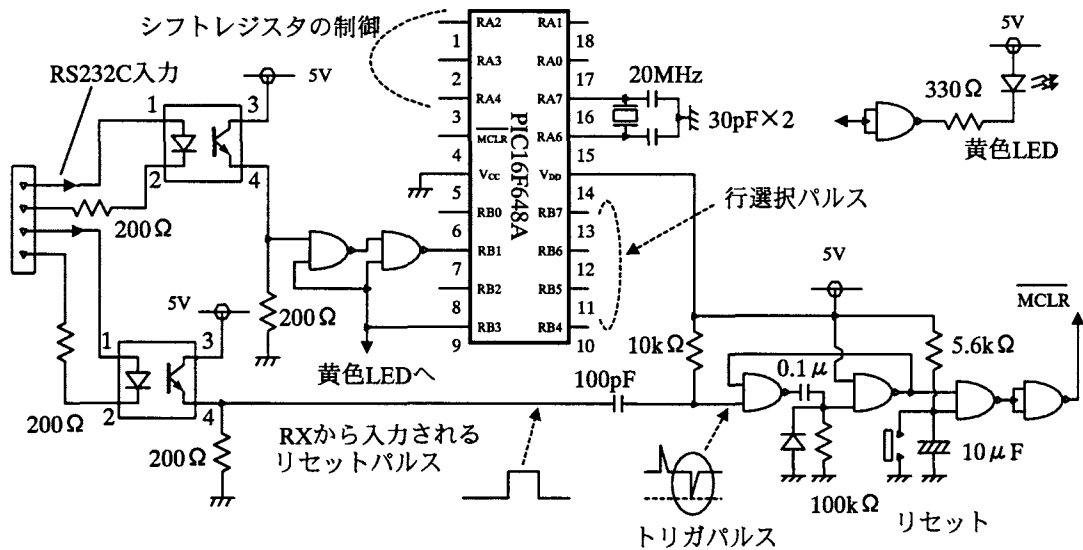


図3.3 漢字ディスプレイの回路図 (RS232Cおよびリセット関連のみ)

高校名」をEEPROMに書き込むようにする方法を検討した。

まずTXからのRestart信号を検出したときに処理をプログラムの冒頭にジャンプさせるようにしてみたが、うまく動作しなかった。このためRestart信号を検出したときにRB4を10msだけHとし、この信号を微分して外付けの単安定マルチバイブレータにトリガ信号を加えPICをリセットするようにした。リセットをかけると処理がプログラムの最初から実行され、「新しい高校名」をEEPROMに書き込むことができるようになった。

RA0に接続した黄色LEDは、高校名を構成する一つの漢字データ(アスキー変換により64B)が漢字ディスプレイに転送される度に10msだけ点灯する。

RB3は、RXにおいてEEPROMへの漢字データの書き込みが全て終了した後に10msだけHとなり、漢字ディスプレイに転送されて漢字ディスプレイをリセットする。この結果、漢字ディスプレイはEEPROMへの読み込みモードに移行する。当初は負極性のパルスとしていたが、この場合にはRestart信号を検

出してPICにリセットがかけるとRB3もLとなり、このとき漢字ディスプレイが一旦リセットされる。したがってリセットパルスが2回出ることになり、この不都合を回避するために正極性パルスに変更した。

RS232C 出力およびリセット出力を差動出力としているのは、RX と漢字ディスプレイの電源を分離するためにフォトカップラーを用いており、ダイオードとの間でカレントループを作るためである。

なお最終的には、転送時間を短縮するためにRXにおいては漢字データをEEPROMに書き込まず、入力データを単にトランスペアレントに出力するプログラムに変更したが、回路の変更は特に必要なかった。

3.3 漢字ディスプレイ

図 3.3 に漢字ディスプレイの回路図を示す。行選択パルスおよびシフトレジスタの制御に関する回路は省略してある。RX から、RS232C 経由のデータと「新しい高校名」検出時のリセット信号（いずれも TTL レベルの信号）が差動信号として入力される。

これらの信号をフォトカップラーで受け、電源を分離している。

なお RS232C のデータについては、フォトカップラーの出力を 2 段の NAND ゲートを介して RB1 (RS232C 入力) に入力している。このゲートを RB3 出力によって EEPROM へのデータ書き込み時のみ RB1 にデータが入力されるように制御し、漢字を表示している間には RB1 にデータが入力されないように制御している。このゲートがないと漢字を表示しているときに、RB1 にデータが入力される度に表示が誤る現象が生じたのでこれを回避するために取った方策である。

4. ソフトウェア構成

4.1 PC

図 4.1 に漢字データ送受信プログラム hterm.exe における OnDraw() 関数の概略フローチャートを示す。RS232C

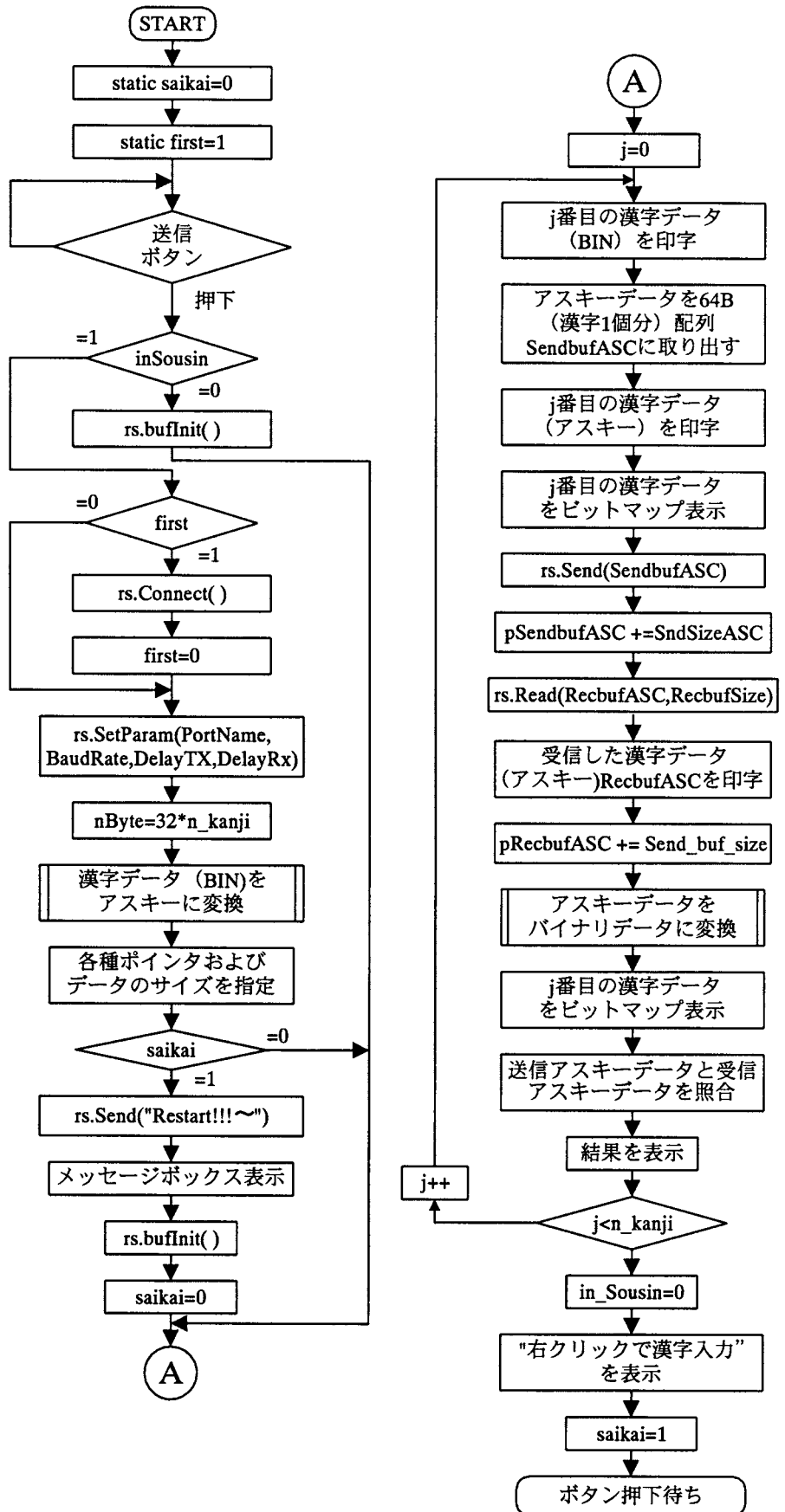


図4.1 hterm.exe における OnDraw() 関数のフローチャート

を使用した通信は、インターネットからダウンロードした「C++におけるRS232C通信クラス」を使用させて頂いた。なお追加、修正した箇所は以下の通りである。

- ① 図 4.1 には明示されていないが、OnDraw()関数の外側でRS232c rs; とクラスRS232cのオブジェクトrsをグローバル宣言している。
- ② メンバ関数 SetParam()を追加し、使用するポート (Portname)、ボーレート (BaudRate)、送信遅延 (Delay_TX)、受信遅延 (Delay_RX) を変更できるようにした。切断時に、メニューから通信設定、インタフェース設定を選択すると設定用のダイアログが起動して先のパラメータを変更できる。
- ③ データ長 (8 ビット)、パリティ (なし)、ストップビット長 (1 ビット)、フロー制御 (なし) については固定とした。

プログラムが起動し「送信」ボタンを押すとドキュメントクラスの変数 inSousin が1になる。このとき静的変数 first は1 であり、命令 rs.Connect() が1 回だけ実行されRS232C インタフェースにおけるデフォルトの設定が行われる。その後で命令 rs.SetParam() を用いて②で示した4つのパラメータの設定変更が行われる。

変数 n_kanji は「漢字データの個数+1」である。変数 saikai は静的変数であり、プログラム起動時にはリスタート信号が送信されないようにしている。

図 4.1 における右側の部分が実際にデータを送受信する部分である。漢字データはバイナリデータであるため以下のようにアスキーデータに変換して配列 SendbufASC()に一旦格納し、命令 rs.Send(SendbufASC) 用いて送信している。

1B のバイナリデータを 0xab とするとき、 2B のアスキーデータ 0x3a 0x3b として送信する

TX から返送された漢字データを 命令 rs.Read(RecdbufASC,RecbufSize) により受信している。

漢字の個数データおよび漢字データを送信し終わると「切断 (inSousin=0)」に移行し、「右クリックで漢字入力を表示」を画面に表示する。

また変数 saikai=1 とするので、次に送信ボタンを押して inSousin=1 になったとき、命令 rs.Send("Restart!!!Restart!!!~") によってTX に対して1 回だけリスタート信号を送信する。

4.2 TX, RX および漢字ディスプレイにおける共通事項

(1) リングバッファによる受信 [3]

TX, RX および漢字ディスプレイにおいてデータを受信するときに、リングバッファを用いてデータの取りこぼしのないようにしている。図 4.2 にリングバッファの構成を示す。2つのポインタ s_p および e_p を使用し、データを受信するたびにポインタを制御しながらサイクリックにデータをバッファに格納する。バッファのアドレス範囲は 0x30~0x6F であり、漢字 1 文字分のアスキーデータを格納できる 64B とした。漢字 1 文字分のデータがバッファに貯まると、そのデータを EEPROM に書

き込む。

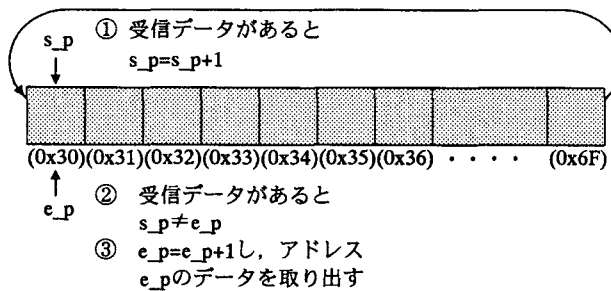


図4.2 受信リングバッファ

(2) EEPROM への書き込み時間の確保

今回使用した PIC16F648A の EEPROM にデータを書き込むためには、書き込むための時間の余裕が必要である。定量的な値が分からなかったため以下のように対処した。

・TX・・・PC から 64B のデータを受信する度に EEPROM に書き込んでいる。次のデータが書き込まれるまでには、EEPROM に書き込んだデータを再度読み出して PC に返送し、正しく送信されたか否かの検査後に、次のデータが送受信されるまでの時間が経過する。このやり方で問題はなかった。

・RX, disp・・・最初 TX から 19.2kbaud の速度で連続的に送信して RX の EEPROM に書き込もうとしたところうまく書き込めなかった。試行錯誤した結果、TXから図4.3に示すように、64B 送信するたびに1s のギャップを入れるようにした。また1 バイト送信するたびに0.5ms のギャップを入れるようにした。

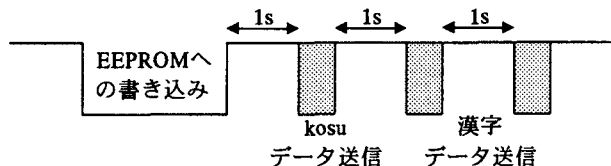


図4.3 TX-RX間におけるタイミングチャート

(3) RX, disp の UART におけるオーバーランエラー、フレーミングエラーの検出とリセット

TX では上記の検出を行っていないが、特に問題は生じていない。RX でも最初は検出を行っていなかったが、書き込みの再現性が良くなかった。そのため LED を点灯させることによりオーバーランエラーおよびフレーミングエラーが発生しているか否かを検査したところ、時々生ずることが分かった。そこで単なるデータの受信だけでなく、オーバーランエラー、フレーミングエラーの検出とエラーのリセット処理を追加したところ安定して受信でき EEPROM に書き込めるようになった (図 4.4 における破線で囲った部分)。

4.3 TX

図 4.4 に TX におけるプログラムの概略フローチャートを示す。大まかな流れは以下のようなものである。

- ① RS232C インタフェースや各種変数の初期化を行う。
- ② ポート B における RB0 は、RB0=0 のとき TX に書き込ま

れた漢字データをRXに対して連続的に送信するためのデバッグ用である。

- ③ 通常動作ではRB0=1であり、このときポインタ s_p および e_p を 0x30 に、BUFF_FULL=0に初期化する。
- ④ 関数 GET_RD は割り込み処理ルーチンにおける処理と呼応して、先に述べたリングバッファによってPCから送信されたデータの受信を行う。64Bのデータを受信し終わると、変数 BUFF_FULL=1となる。
PC→TX 間のフレーム構成は、図 4.5 に示す TX からRXに送信されるフレームとは、漢字個数データの送信フレームが「漢字個数データ×64 個」になっている点が異なるだけである。
- ⑤ リセット時には変数 kosu_set=1 である。関数 capture_kanji_kosu_new において、最初にPCから送信される漢字個数を調べ変数 kanji_kosu_cnt に代入する。また変数 kosu_set=0として次に送信される漢字データの取得に備える。
- ⑥ 関数 data_send_back では、漢字個数データをPCに返送する。このときRB3=Lになっている。
- ⑦ 次にPCから漢字データ(64B)が送信されてくると、関数 write_into_eeprom においてそのデータを順次PICにおけるEEPROMに書き込む。
- ⑧ 関数 read_from_eeprom において、書き込んだデータをEEPROMから読み出し、関数 data_send_back によって漢字データをPCに返送する。⑦⑧の処理を変数 kanji_kosu_cnt の回数だけ実行する。
- ⑨ 図 4.4 における真ん中の部分は、赤色のLEDを駆動してRXに対してデータを送信する部分である。変数 write_flag はRS232C受信割り込み(PIR)によって「新しい高校名」を検出したときに1となる。したがって初期値を0にしておく必要がある。
- ⑩ ポートBにおいてRB3=Hとし、図3.1の回路図における2-1セレクタを制御し、送信データがLEDを駆動できるようにする。

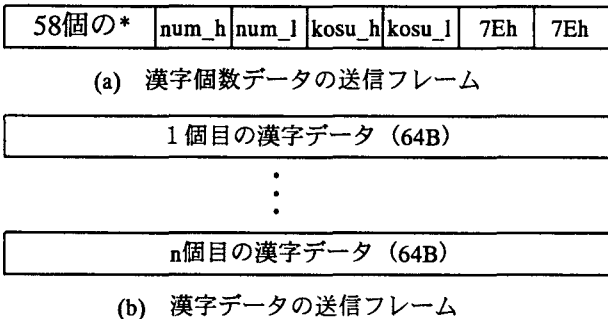


図4.5 TXからRXに送信される信号のフレーム

- ⑪ 関数 send_kosu_new_64B によって、漢字データの個数を光信号に乗せて送信する。TX→RX間のフレーム構成は

図 4.5 に示すフレーム構成とほぼ同じである。

- ⑫ 引き続き関数 read_from_eeprom によりEEPROMから漢字データを読み出し、関数 data_send_forward によりLEDを駆動して送信する。PICのEEPROMにデータを書き込むためにはある程度の待ち時間が必要である。そのため図4.3に示したように64B送信するたびに1sのギャップを入れている。
- ⑬ 変数 write_flag が1になるのは、「新しい高校名」の転送に先立ちTXから送信されるRestart信号をISRにおいて受信した時である。このとき kosu_set=1としてヘッダーフレームの受信を可能にさせ、RB3=Lとして光信号の送信を停止し、RB4=Lとして単安定マルチバイブレータにリセット信号を送出してTXにおけるPICをリセットする。

4.4 RX

(1) バケツリレー方式の場合

図 4.6 にバケツリレー方式の場合におけるRXの概略フローチャートを示す。大まかな流れは以下のようである。

- ① RS232Cインタフェースや各種変数の初期化を行う。
- ② 図 4.5 に示したフレームがTXから連続して送信されてくるので、漢字個数データの送信フレームにおける*を検出しヘッダーフレームを受信する。
- ③ *の検出ができれば、引き続き漢字個数(kosu)を取得する。TXと異なりデータの返送はない。
- ④ 次に漢字データを取得しEEPROMに書き込む。
図 4.6 における右側の部分は、漢字データをEEPROMに書き込んだ後漢字ディスプレイに対してデータを送信する部分である。まずRB3を10msだけHとして漢字ディスプレイにリセット信号を送り、漢字ディスプレイをEEPROMへの書き込みモードにセットする。
- ⑥ その後、データを1回だけ送信する。
- ⑦ データ送信後は割り込みを可とし、「新しい高校名」が送られたか否かのセンス(検査)に入る。TXから送信されるRestart信号を検出するとISRにおいて変数 r_detect が1になるので、これを検出する。
- ⑧ r_detect=1になるとRB4を100msだけHとし、RXをリセットすることによって「新しい高校名」の受信モードに移行させる。

(2) トランスペアレント伝送の場合

図 4.7 にフローチャートを示す。

- ① まずRS232Cの初期化、RB3の初期化を行う。
- ② ポインタの初期化を行った後、関数 GET_RD で受信する。
- ③ 関数 GET_RD を実行すると受信データがなく s_p=e_p のときはZフラグ=1となる。一方受信データがあると s_p≠e_p となり、Zフラグ=0となる。
- ④ 受信データがありZフラグ=0となった場合には、まずRestart信号の先頭の値“R”であるかどうかを確認する(“R”=0x52)

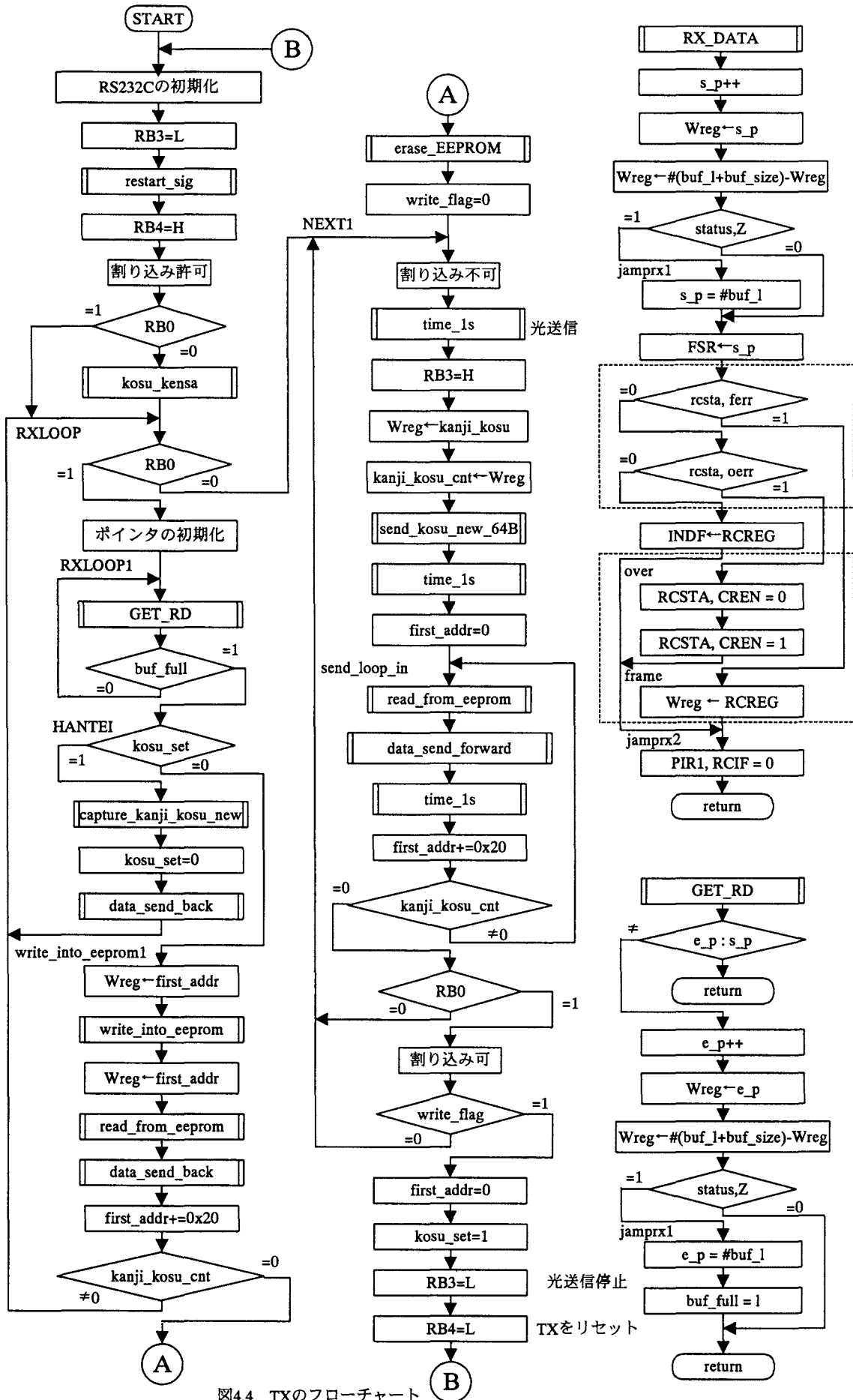


図4.4 TXのフローチャート

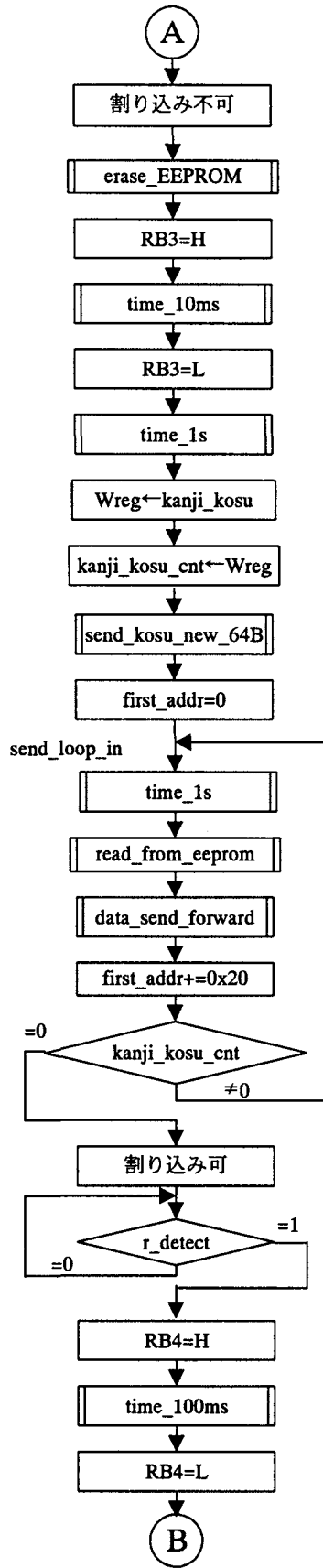
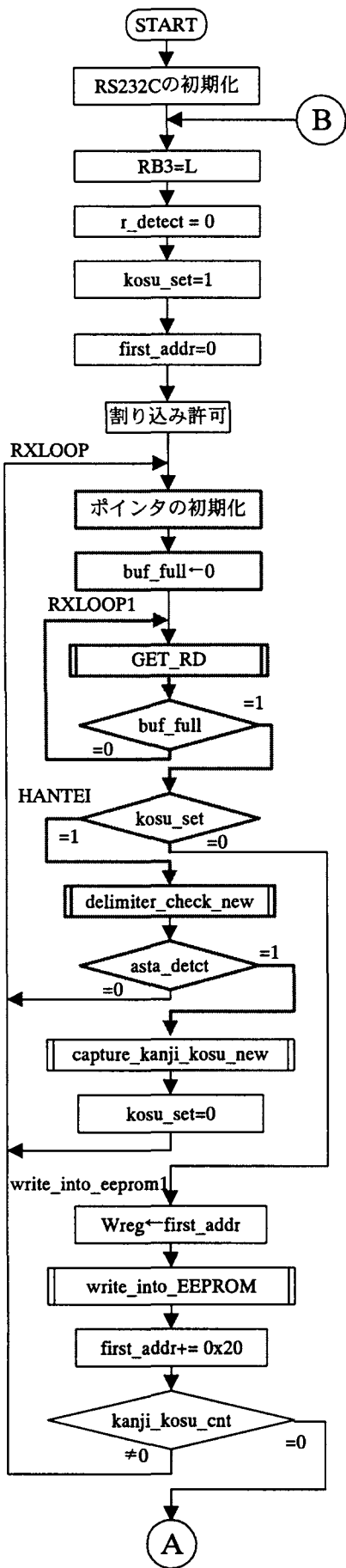


図4.6 RXのフローチャート (パケットリレー方式の場合)

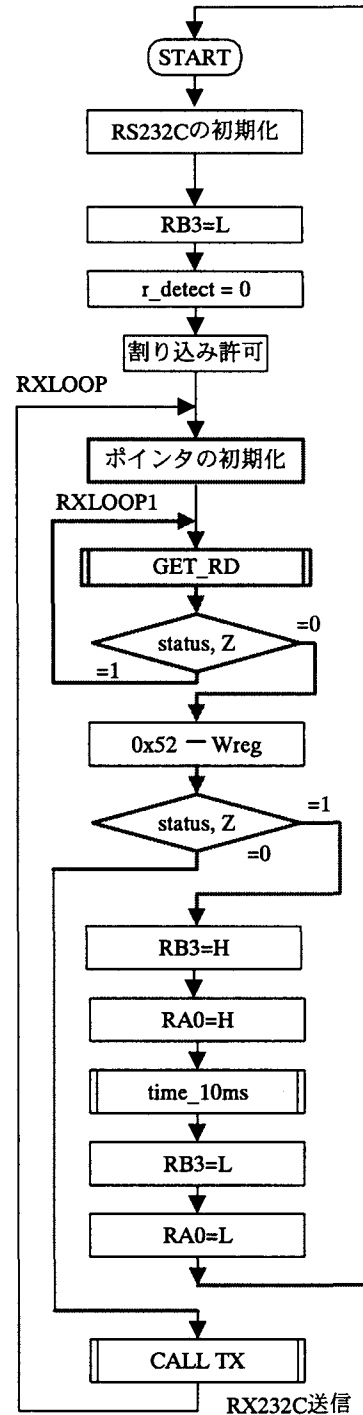


図4.7 RXのフローチャート (トランスペアレント伝送の場合)

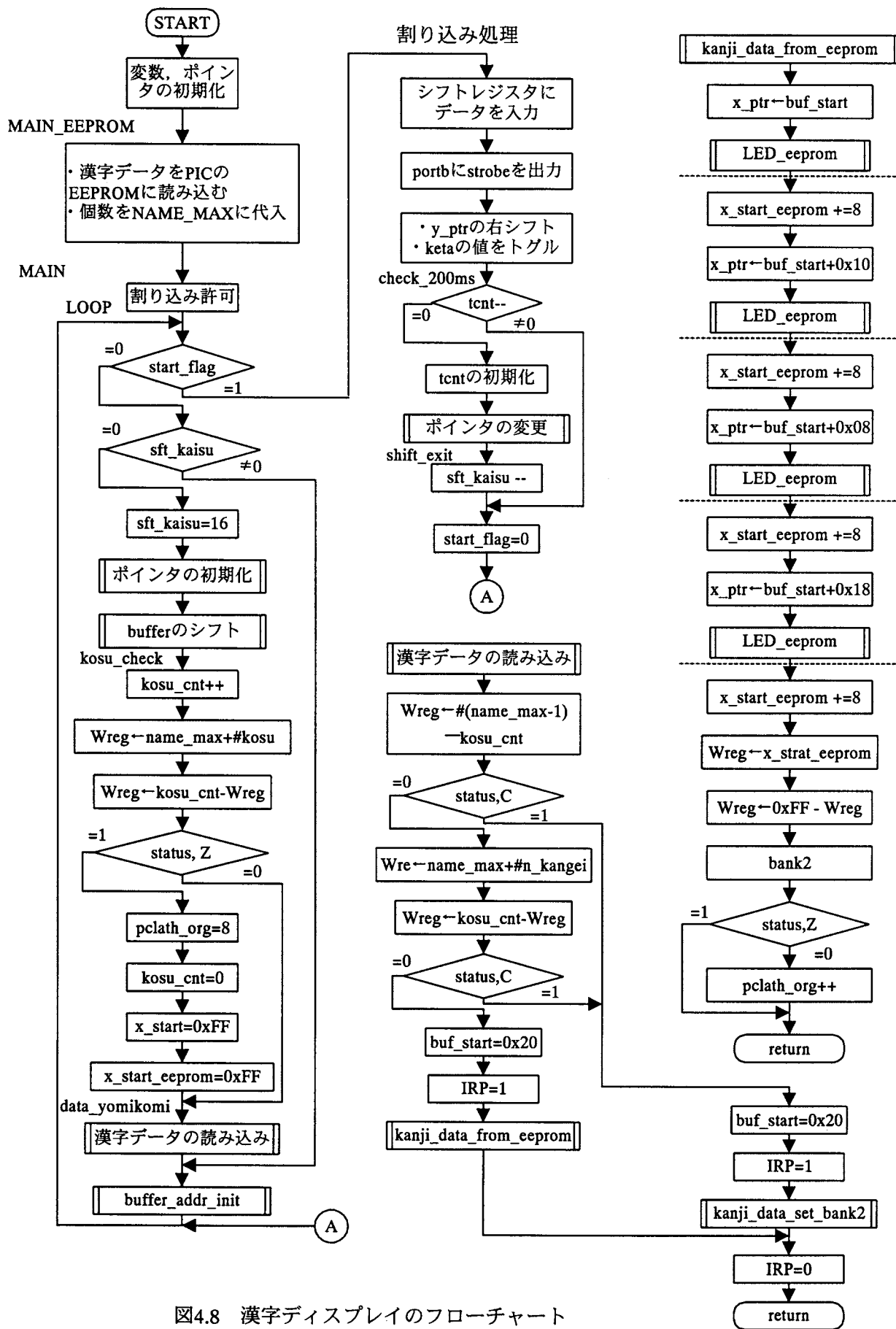


図4.8 漢字ディスプレイのフローチャート

- ⑤ 受信データが“R”でない場合には関数 TX によって漢字ディスプレイに対して受信したデータを即時に送信する。
- ⑥ 一方受信データが“R”の場合には 10ms 間 RB3=H とし、漢字ディスプレイにリセット信号を送信する。また同じ時間だけ RA0=H とし、黄色の LED を点灯させる。

4.5 漢字ディスプレイ

漢字ディスプレイのプログラム構造は

「RX のプログラム (バケツリレー方式) + 漢字ディスプレイのプログラム」のようになっている。

- ① PIC をリセットすると図 4.6 に示した RX のフローチャートに従って漢字データを EEPROM に書き込む。
- ② 漢字データの個数を変数 NAME_MAX に格納後、処理が図 4.8 に示す漢字ディスプレイの処理に移る。
- ③ 漢字ディスプレイでは、PC から送信した高校名を漢字ディスプレイに格納してある文章にはめ込んで表示する。高校名が「静岡理工科」高校とすれば、以下ようになる。
「!! 歓迎!! 静岡理工科高等学校の皆さん。電気電子工学科へようこそ」
この文章を変更することができる。
- ④ なお、プログラムを工夫して高校名の箇所が反転表示されるようにしてある。

5. 実行結果

(1) オープンキャンパスにおける実行結果

当初考えていたよりも動作が思うようにいかず試作に多くの時間を取られた。平成 21 年度における第 1 回目のオープンキャンパスは 7 月 26 日 (日) であったが、漢字ディスプレイのプログラムの修正に着手したのは 1 日前の 7 月 25 日 (土) であった。それまでに、差分検出により新規データを検出できないことに対する対策として単安定マルチバイブレータを使用する方策を取って解決してきたが、この時点でさらに以下の問題が生じた。

- ① 高校名の先頭の文字だけが正しく表示され、2 番目以降はすべての LED が点灯していた。
- ② 書き込み時間が大きい。
①についてはアセンブラの命令 DECFSZ がうまく動作していないことが原因と判明し、INCFNZ に変更して解決できた。
- ②については、RX をトランスベアレント伝送に変更した。さらにゲートを追加して、割り込みが入る度に表示が誤る不具合を解決した。
- ③ さらに以後の 3 回のオープンキャンパスにおける使用において、特定の文字が誤りやすいという蓋然性が認められた。

(2) その後の検討結果

(1) で述べた実行結果の不具合、特に③は極めて不可解であるが影響が重大であり、是非解決しなければならぬ課題であった。そこで更に以下の点について調査を行ってみた。

- ① RX のコンパレータの出力波形は方形波になっているか
- ② コンパレータの識別レベルは入力振幅の 1/2 か
これらについては十分な特性であり、問題なかった。そこで、

TX-RX-漢字ディスプレイ間のボーレートを 2400baud にまで下げてみたが、何ら改善が見られなかった。

一方 RX としてバケツリレー方式を用いると、データを EEPROM に正しく書き込めることから、漢字ディスプレイと RX (バケツリレー方式) の相違点を冷静に考えてみた。漢字ディスプレイでは一部の変数をバンク 2 に移動している点異なっており、それらの変数の取り扱いにバグがあるのではないかと見当をつけてみた。そこで RX (バケツリレー方式) の EEPROM への書き込み部分を取り出し、変数をバンク 2 に移動することを止めてほぼそのまま適用し、そこに漢字ディスプレイのプログラムを付け加えてみた。これにより、バグのない動作が首尾良く得られた。DECFSZ の問題も同時に解決された。

なお、表示速度を 1/2 にしてダイナミック点灯の原理がよく分かるようにするスイッチを付加し、また「高校名」を強調するためにその部分の表示速度を落として表示できるようにした。

6. まとめ

高校生に電気電子に対して興味を持って貰うことを狙いとして、3 年前からオープンキャンパスで展示用に使用してきた低速の光通信システムと、昨年度に試作した漢字表示電光掲示板 (漢字ディスプレイ) を組み合わせ波長多重光通信システムを設計、試作した。来場した高校生に自分の「高校名」を言って貰い、その「高校名」を光ファイバによって伝送することでインパクトを与えられるのではないかと観点で試作したものである。オープンキャンパスにおいて使用して一定の成果が得られたものとする。このシステムの概要を以下に示す。

- ・緑色の LED を使用した緑色通信系と赤色の LED を使用した赤色通信系からなる波長多重光通信システムである。
- ・伝送媒体は、プラスチック光ファイバである。
- ・赤色通信系を用いて、PC から入力した「高校名」を漢字ディスプレイに伝送し、「!! 歓迎!! 「高校名」 高等学校の皆さん。電気電子工学科へようこそ」と表示できる。

謝辞

波長多重光通信システムを構成する上で必要不可欠な波長フィルタを快く御提供頂いた野村龍男教授に深謝致します。また当研究室の卒研究生にはオープンキャンパスにおいて説明、案内等のご協力を頂いた。ここに深く感謝します。

[参考文献]

- [1] 袴田吉朗, “PIC マイコンと 16×16LED を用いた漢字表示電光掲示板の設計と試作”, 静岡理科大学紀要, Vol.17, pp.133-142 (2009)
- [2] PIC16F627A/628A/648A Data Sheet Flash-Based 8-Bit CMOS Micro controllers with nano Watt Technology, Microchip
- [3] 神崎康宏, “作りながら学ぶ PIC マイコン入門”, CQ 出版社, 2007 年