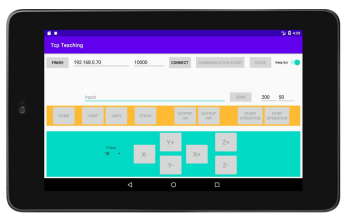
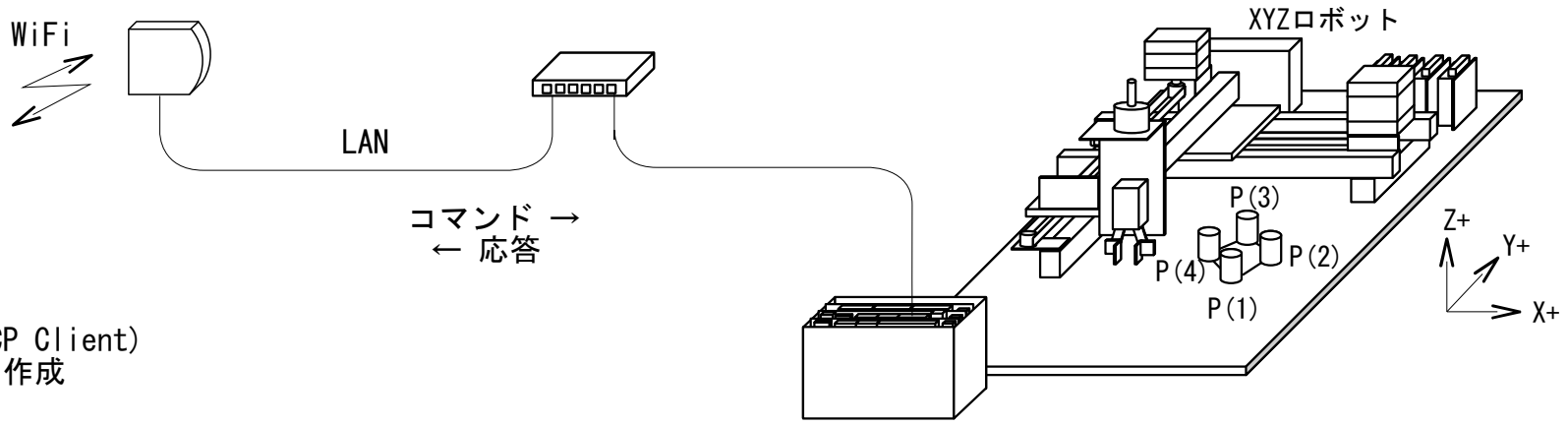
Androidタブレットでティーチング

ACCEL ti2k-200915

AndroidでのTCP/IP通信手法やUIのお勉強を兼ねてティーチングアプリを作ってみました。
Androidタブレットは安価、開発環境は無料です。社内で使用するのならPlayStoreに登録する必要もありません。

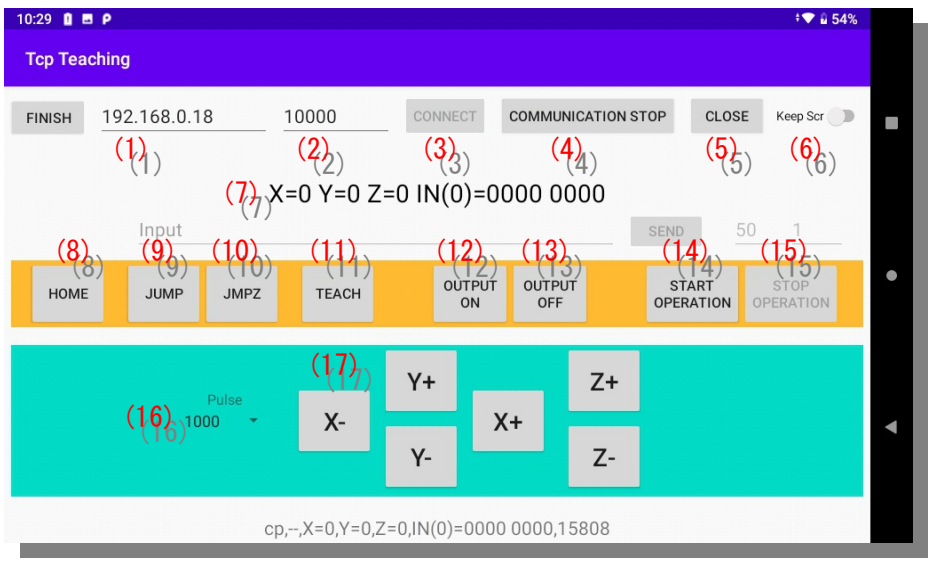


Android9.0タブレット（TCP Client）
Android Studio & Java で作成



MPC-3000（192.168.0.18:10000 TCP PACKET Server）
MPG-2314 または MPG-3514

■ 操作画面



(1) MPCのIPアドレス
(2) MPCのポート番号
(3) Socket Open
(4) MPCとコマンド通信開始･停止
(5) Socket Close
(6) 画面スリープ抑制

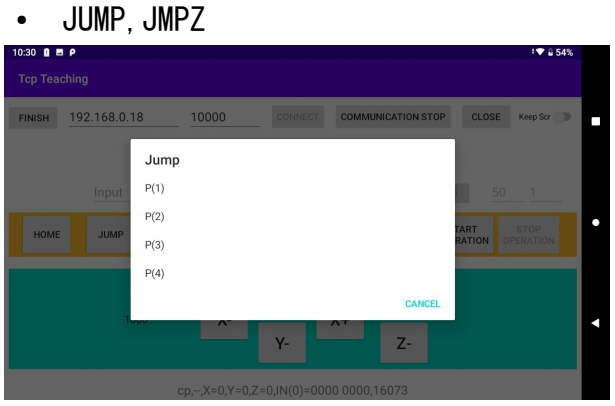
(7) 現在のXYZ座標と入力IN(0)のビット表示
　MPCは出力0~7ビットの状態をIN(0)で読めます

(8) 原点復帰
(9) JUMP移動　※
(10) JMPZ移動　※
(11) 現在点教示　※
(12) 出力ポートON　※
(13) 出力ポートOFF　※
(14) MPC作業開始
(15) MPC作業停止

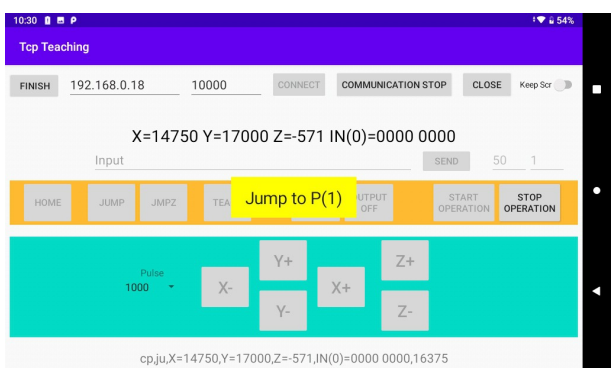
(16) JOG移動量設定
(17) X-～Z- JOG移動

■ ボタンをタップしたとき上記※印以外は即実行します。※印のボタンはAlertDialogで選択、確認を促します。

- JUMP, JMPZ

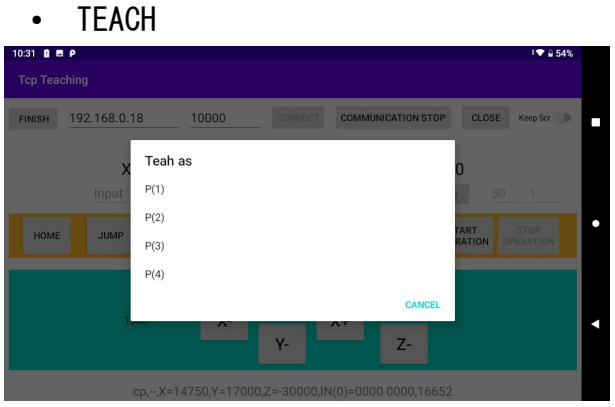


AlertDialogで移動先を選択

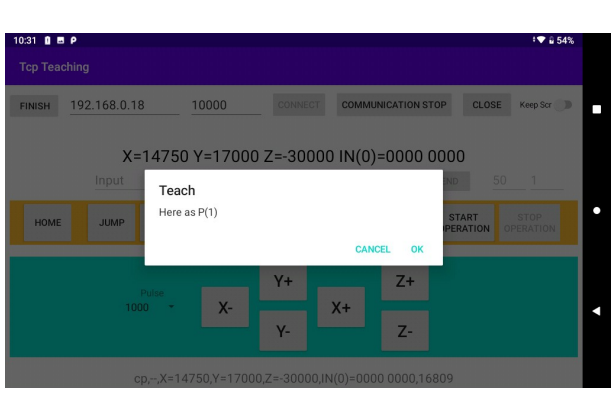


移動中はToast表示

- TEACH

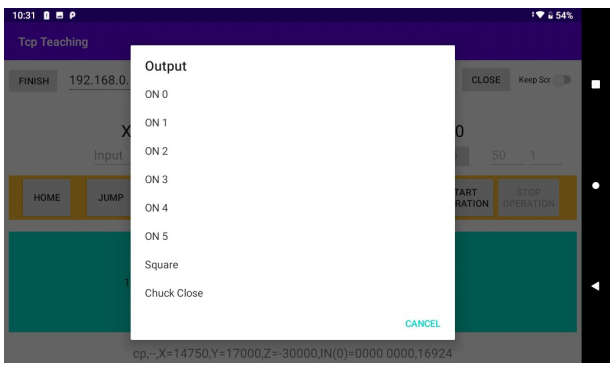


AlertDialogで点番号を選択

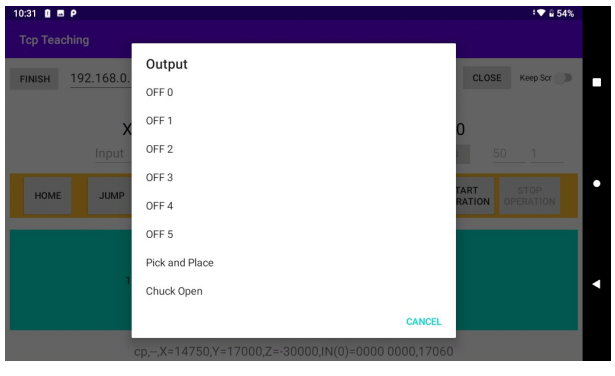


確認

- OUT PUT ON

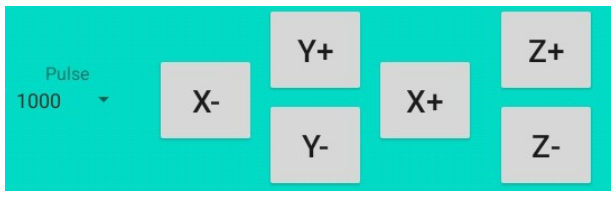


- OUT PUT OFF



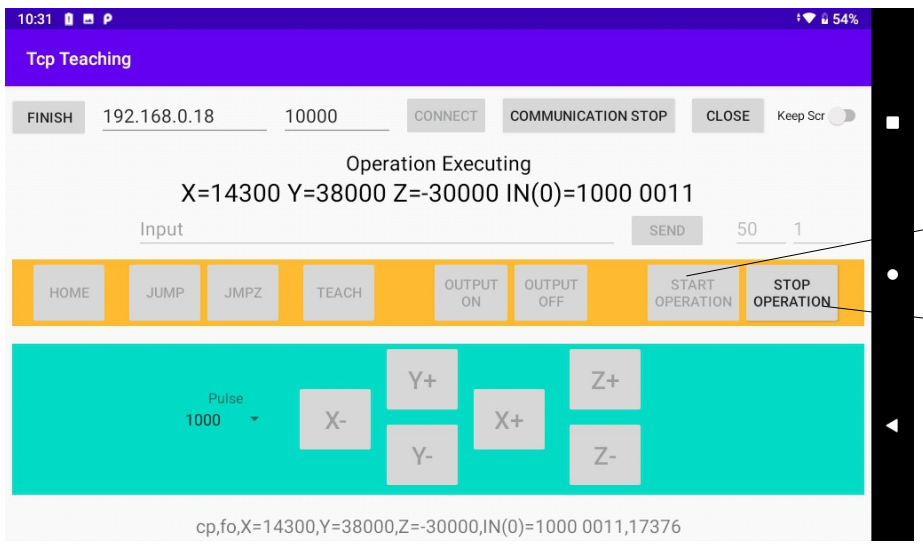
AlertDialogで出力ポートを選択
MPCの作業動作はBit6のON/OFFで切り替わる

- Jog



Spinnerで選択したパルス量を移動します。

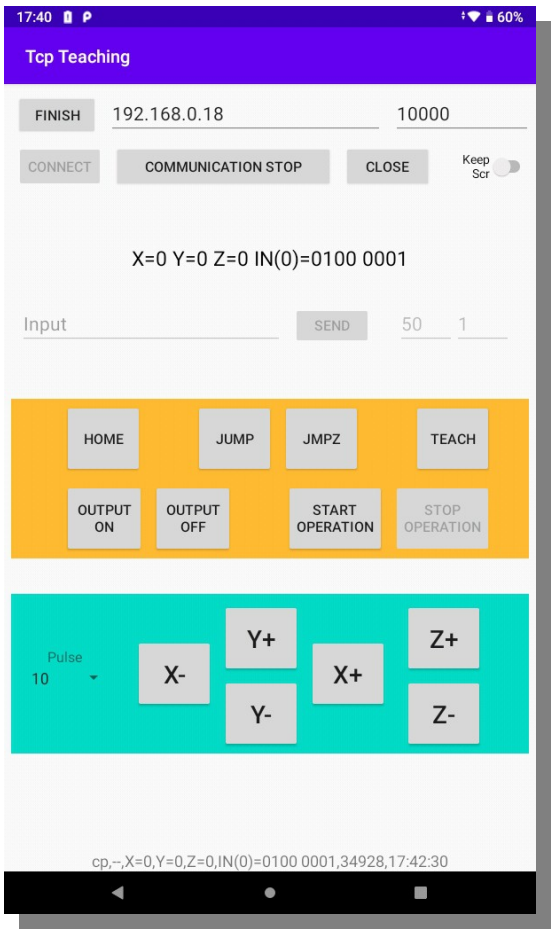
- 作業動作



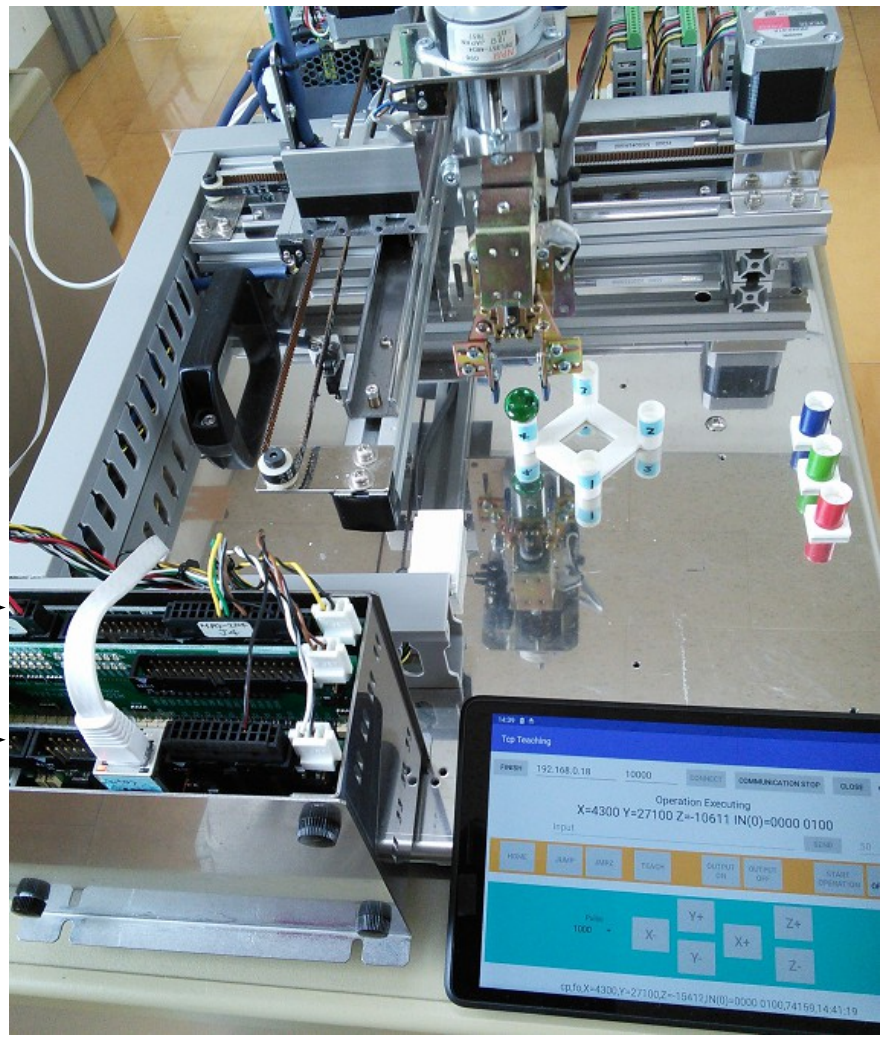
START OPERATION でMPCにプログラムされた
Pick and Place またはデモ動作を開始

STOP OPERATIONで停止

■ 縦画面



■ 実験機



■ MPCプログラム

```
SET_IP 192 168 0 18 255 255 255 0 192 168 0 248
SET_IP 115200                  /* プログラムポート 115200bps
SET_IP TCP PACKET 10000        /* スレーブ設定
TCP_RO$=""
PG 0
CLRPOS
ope$="--"
count_=0
SYSCLK=0
oldsysclk=0
DO
  WAIT TCP(0)==1
  WAIT TCP_RO$<>""                          /* 受信待ち
  RcvString$=TCP_RO$                        /* コマンド+スペース+パラメータ
  ptr_=RcvString$
  RcvCommand$=PTR$(2)                       /* コマンド(2キャラ)
  RcvParameter=VAL(RcvString$)              /* パラメータ(数値)

  /* コマンド解析
  SELECT_CASE VOID
    CASE  RcvCommand$=="cp" : GOSUB *ResCp  /* 現在位置,IN(0)など
    CASE  RcvCommand$=="xr" : GOSUB *ResXr  /* X軸JOG
    CASE  RcvCommand$=="yr" : GOSUB *ResYr  /* Y軸JOG
    CASE  RcvCommand$=="zr" : GOSUB *ResZr  /* Z軸JOG
    CASE  RcvCommand$=="ho" : GOSUB *ResHo  /* 原点復帰
    CASE  RcvCommand$=="te" : GOSUB *ResTe  /* 現在点教示
    CASE  RcvCommand$=="ju" : GOSUB *ResJu  /* JUMP移動
    CASE  RcvCommand$=="jz" : GOSUB *ResJz  /* JUMPZ移動
    CASE  RcvCommand$=="on" : GOSUB *ResOn  /* 出力ポートON
    CASE  RcvCommand$=="of" : GOSUB *ResOf  /* 出力ポートOFF
    CASE  RcvCommand$=="ex" : GOSUB *ResEx  /* 動作実行
    CASE  RcvCommand$=="qt" : GOSUB *ResQt  /* 動作停止
    CASE_ELSE
  END_SELECT
  count_=count_+1
  elapse=SYSCLK-oldsysclk
  oldsysclk=SYSCLK
  PR "elapse" elapse
LOOP


*ResCp
  FORMAT "00"
  in$=""
  FOR i_=7 TO 0 STEP -1
    IF SW(i_)==0 THEN
      in$=in$+"0"
    ELSE
      in$=in$+"1"
    END_IF
    IF i_==4 THEN : in$=in$+" " : END_IF
  NEXT
  FORMAT ""
  ResString$=RcvCommand$+","+ope$
  ResString$=ResString$+",X="+STR$(X(0))+",Y="+STR$(Y(0))+",Z="+STR$(Z(0))
  ResString$=ResString$+",IN(0)="+in$+","+STR$(count_)+","+TIME$(1)

  GOSUB *ResTX0
  RETURN

*ResXr
  ope$="xr"
  axis=X_A
  QUIT_FORK 1 *Rmov
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResYr
  ope$="yr"
  axis=Y_A
  QUIT_FORK 1 *Rmov
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResZr
  ope$="zr"
  axis=Z_A
  QUIT_FORK 1 *Rmov
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*Rmov
  ACCEL axis 5000 500 500
  FEED axis 100
  RMVS axis RcvParameter
  WAIT RR(axis)==0
  GOSUB *SetSpeed
  ope$="--"
  END
```

右上へ続く

```
*ResHo
  ope$="ho"
  QUIT_FORK 1 *HomeTask
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*HomeTask
  GOSUB *Home
  ope$="--"
  END

*Home
  PG 0
  OFF 7
  ACCEL X_A|Y_A 5000 500 500
  ACCEL Z_A 10000 500 500

  IF HPT(ZIN0)==1 THEN
    RMVS Z_A -5000                  /* 退避移動
    WAIT RR(Z_A)==0
  END_IF
  SHOM Z_A IN0_ON
  HOME Z_A POS_L
  WAIT RR(Z_A)==0

  IF HPT(XIN0)==1 THEN
    RMVS X_A 5000                   /* 退避移動
  END_IF
  IF HPT(YIN0)==1 THEN
    RMVS Y_A 5000                   /* 退避移動
  END_IF
  WAIT RR(X_A|Y_A)==0

  SHOM X_A|Y_A IN0_ON
  HOME X_A|Y_A NEG_L
  WAIT RR(ALL_A)==0

  MOVS 3000 3000 VOID -5000
  WAIT RR(ALL_A)=0
  CLRPOS
  GOSUB *SetSpeed
  RETURN

*SetSpeed
  ACCEL X_A|Y_A 60000 2000 1000
  ACCEL Z_A 60000 5000 500
  FEED X_A|Y_A|Z_A 100
  RETURN

*ResEx
  ope$="fo"
  QUIT_FORK 1 *PgExec
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*PgExec
  PG 0
  OUT (IN(0)&&HF0)|0 0
  GOSUB *Home
  TIME 1000

  IF SW(6)==1 THEN                  /* デモ動作
    FEED X_A|Y_A|Z_A 20
    DO
      OUT (IN(0)&&HF0)|1 0
      MOVS 50000 VOID VOID -30000
      WAIT RR(ALL_A)==0
      TIME 100
      OUT (IN(0)&&HF0)|2 0
      MOVS Y_A 50000
      WAIT RR(ALL_A)==0
      TIME 100
      OUT (IN(0)&&HF0)|4 0
      MOVS 0 VOID VOID 0
      WAIT RR(ALL_A)==0
      TIME 100
      OUT (IN(0)&&HF0)|8 0
      MOVS Y_A 0
      WAIT RR(ALL_A)==0
      TIME 100
    LOOP
  ELSE                              /* Pick and Place
    OFF 7                           /* ハンドチャック開
    DO
      FOR p_=1 TO 4
        OUT (IN(0)&&HF0)|p_ 0
        JUMP P(p_)
        OFF 7
        JMPZ P(p_)
        JUMP P(p_)
        ON 7                        /* ハンドチャック閉
      NEXT
  LOOP
  END_IF
```

次ページへ続く

```
*ResTe
  SETP RcvParameter X(0) Y(0) U(0) Z(0)
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResJu
  ope$="ju"
  QUIT_FORK 1 *Jump
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*Jump
  PG 0
  GOSUB *SetSpeed
  FEED X_A|Y_A|Z_A 50
  JUMP P(RcvParameter)
  WAIT RR(ALL_A)==0
  ope$="--"
  END

*ResJz
  ope$="jz"
  QUIT_FORK 1 *Jmpz
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*Jmpz
  PG 0
  GOSUB *SetSpeed
  FEED X_A|Y_A|Z_A 50
  JMPZ P(RcvParameter)
  WAIT RR(ALL_A)==0
  ope$="--"
  END


*ResOn
  ON RcvParameter
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResOf
  OFF RcvParameter
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResQt
  STOP ALL_A STP_D
  WAIT RR(ALL_A)==0
  QUIT 1
  ope$="--"
  ResString$=RcvCommand$+",ok"
  GOSUB *ResTX0
  RETURN

*ResTX0                                   /* 応答送信
  PR ResString$
  TCP_RO$=""
  TCP_XO$=ResString$+"¥n"
  RETURN
```

■ Android TCP/IP 関係コード(抜粋)
Androidに関する情報はネット上に星の数ほどありますのでそちらをご参照下さい。
下記は、「こんな感じでやってみました」的なもので、あくまで自己流です。
梃子摺ったのは、
　　ボタンのEnable/Disable処理と通信が衝突すると、通信の具合が悪くなる(?)
　　WiFiとBluetoothが干渉する(2.4GHz帯)(これはタブレットのBluetoothを無効にする)

- Socket Open と 受信ループ

```
address = editTextIp.getText().toString();
port =  Integer.parseInt(editTextPort.getText().toString());

Runnable sender = new Runnable() {
    @Override
    public void run() {

        socket = new Socket();
        InetSocketAddress endpoint = new InetSocketAddress(address, port);
        int sdt = Integer.parseInt( editText6.getText().toString());

        try {

            socket.connect(endpoint, 5000);         // Connection
            textView.setText("Connected");

            socket.setTcpNoDelay(true);
            reader = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            writer = new BufferedWriter(new OutputStreamWriter(socket.getOutputStream()));

            // 受信ループ
            while (true){
                if(socket == null)break;
                if (reader.ready()) {
                    ResString = reader.readLine();   // 受信
                    rcvFlg = false;                  // 送信Threadとのインターロック
                    textViewRes.setText(ResString);
                    Thread.sleep(sdt);
                }
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        } catch (IOException e) {
            socket = null;
            textView.setText(e.toString());
            e.printStackTrace();
        }
    }
};
Thread th = new Thread(sender);
th.start();
```

- 送信と受信した文字列の処理

```
Runnable sender = new Runnable() {
    String st="";
    @Override
    public void run() {
        while (true){
            if(socket == null) break;
            rcvFlg = true;          // 受信インターロック 送受信が衝突すると具合が悪い
            if(MpcCommand.equals("")) {
                st = "cp 0";        // 基本command
            }else{
                st = MpcCommand;    // ボタンタップ時
             }
           MpcCommand = "";

            try {
                writer.write(st + "¥n");   //送信
                writer.flush();
            } catch (IOException e) {
                e.printStackTrace();
            }

            while (true) {
                if(!rcvFlg){
                   // 受信処理
                   break;
                }
            }
        }
    }
};
Thread th = new Thread(sender);
th.start();
```

- X+ボタンをタップしたとき

```
if( view == button15){
    MpcCommand = "xr " + spinner.getSelectedItem().toString().trim();
```