MST070M-AUT 활용 방법 참고, {<t>, !t! /터치 좌표 응용 시험}

RS232-TTL 과 RP2040 Pico W 를 활용
Arduino IDE 에서 버튼 영역의 비교및 연동기능 시험.

>> 준비물.
1. PC Arduino IDE , RP2040 pico 패키지 설치.
2. RP2040 Pico W
3. MST070M-AUT
4. RS232-TO-TTL 변환기
5. 전기 연결선 및 하우징. USB 케이블 5V 전원등.
6. 가로세로 배열 구성된 이미지 BMP 화일.

<주의> 준비된 배선이 없는 경우 전선 간의 납땜 연결(인두기)이 필요할 수 있다.

>> 목적 : MST070-AUT의 Touch 좌표를 획득하고, 연동기능을 아두이노에서 시험한다.

1. Arduion IDE : RP2040 Pico W 빌드환경 구성하기.
   <FILE>-<환경설정>-<추가적인 보드 메니저의 URL> 에 아래 내용을 추가 기입하고,
   RP2040 PICO W 보드 빌드 환경을 구성한다. (보드메니저, RP2040 검색 후 툴설치):

https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

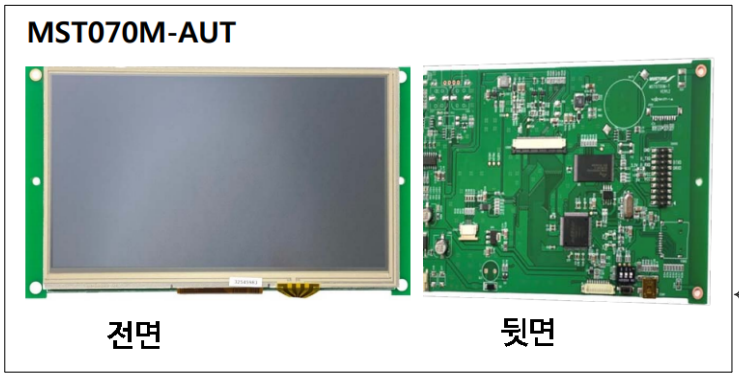
2. RP2040 Pico or Pico W 를 준비, USB 연결케이블도 함께 준비한다.

라즈베리파이 피코 W
(Raspberry Pi Pico W) 를 부품 마켓에서 구매한다.

micro-USB 5P cable

< 이런 모습이다>

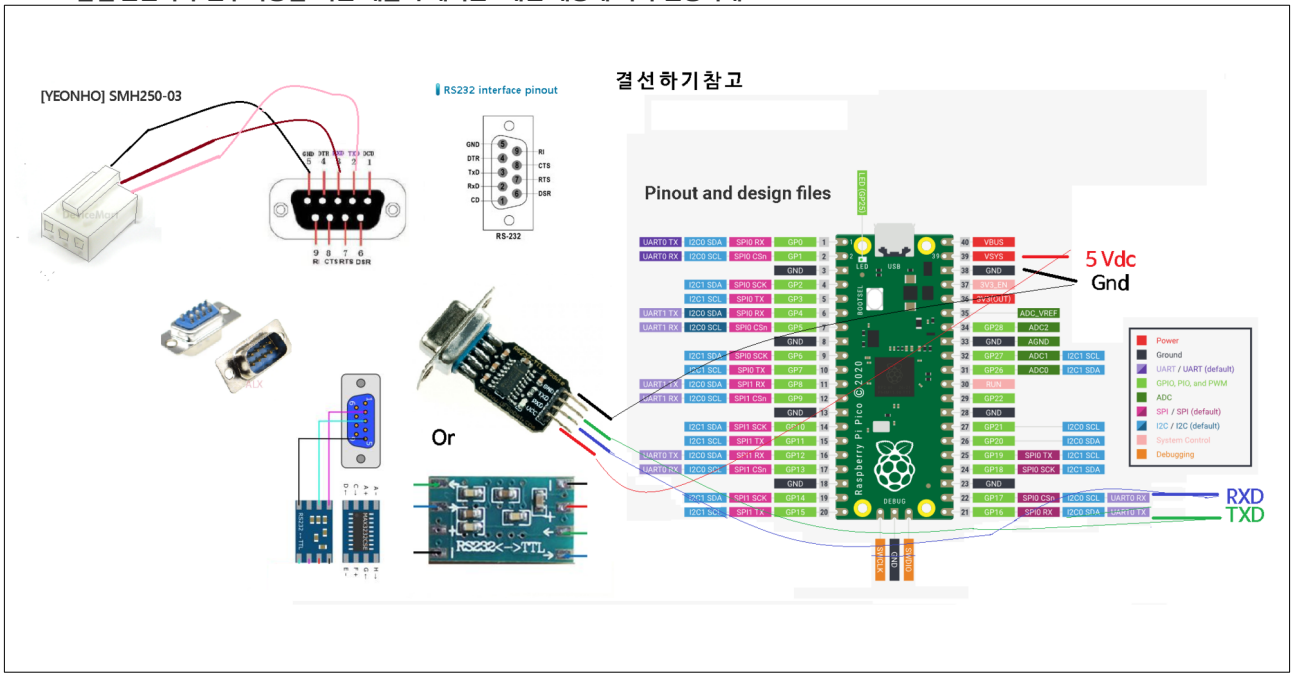
3. MST070M-AUT 를 준비 한다.



← <MST070M-AUT 본체>

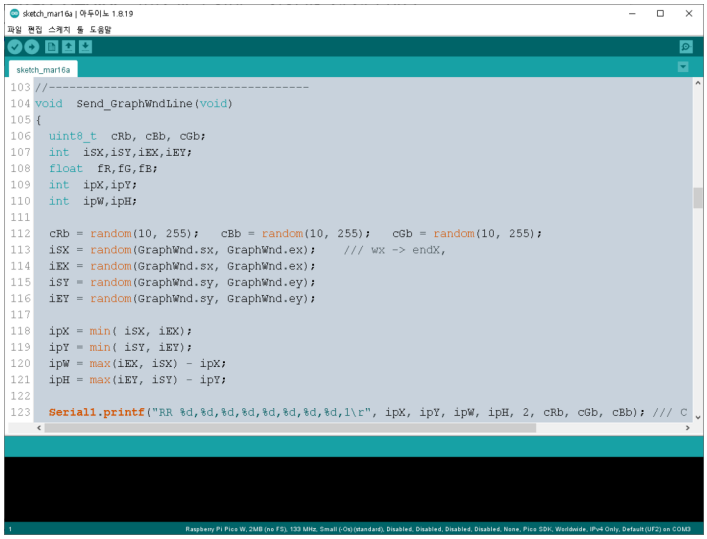
4.-5. RS232-TTL 변환기 구하기및 결선하기.

RS232 to TTL 변환기는 시중에 몇 종류가 있다, 결선은 아래의 그림을 참조하여 결선 한다.
납땜 연결이나 인두사용법, 혹은 제품 구매처는 , 개인 재량에 따라 활용하세요.



6. 가로세로 배열된 BMP image – USB 모드상태로 MST070M-AUT 저장하기.

아래 페이지에 말 움직임의 배열그림을, MST070에 USB 저장 모드 상태로 하여
"horse_run.bmp" 로 루트 폴더에 저장 한다.
이때 루트폴더에서 config.txt 를 Notepad로 열어,
"CMD UART0 115200", "CMD HT_1" 이 동일한지 확인한다.
<< **통신속도와, 터치좌표 출력형식은 동일해야한다. >>**
배열 그림의 크기가 달라지면 좌표값이 예제보다 더 많이 어긋나므로, 참조용 그림을
예제 제공시 함께 압축하여 포함한다.
모든 경우 Arduino IDE 에서 Serial1.begin(_##) 과 동일한 수치일 때 잘 작동한다.
BMP image 크기는 480x480 이하를 권장한다.
boot mode S/W 를 원상 복귀하여 정상 부팅 시킨다.



7. 이제 아두이노 를 연결하고 아래 코드를 기록하고 동작을 확인한다. (첨부화일).

```
/*===== RP2040 PICO-W ====**
* RS232-TTL with RP2040 Pico W
* Demo , Random/EGL Graphic Draw Demo
*        Touch Get IN, [STOP] / [Redraw]
* PIN 21, (GP16), TXD RS-TTL-OUT
* PIN 22, (GP17), RXD RS-TTL-IN
**/
#include <Arduino.h>

#define   HR_IMAGE_BOX_WIDTH      (158)
#define   HR_IMAGE_BOX_HEIGHT     (158)
#define   HR_IMAGE_LOAD_MSG       "ia 1,horse_run.bmp₩r"
#define   HR_IMAGE_CUT_SHOW_MSG   "ico 1,100,100,%d,%d,%d,%d₩r"
#define   HR_IMAGE_UNLOAD_MSG     "ix 1₩r"

typedef unsigned char  UBYTE ;

typedef struct {
    int  nTouchID;
    int  sx, sy;
    int  ex, ey;
    char  czStrTitle[80];
}IDWrect;

/***
 *  Random Seed,
 *  DrawBar_Graphc  (Building-Shape-box-Fill-poly)
 *  Random-build-BOX, Window Rect chart,
 *     /---/|
 *     |   ||
 *     |   ||
 *     -----
 *  MainButton Center Touch    [STOP]/[ReDraw]
 *  [RUN] [STOP] [CLEAR]
 *****/

IDWrect  GraphWnd = {
   88, 15,15, 785,350, "Graph Window"        //// graphic draw area.
};

IDWrect BtnBoxList[4] =
{
  { 1,  80,390,  220, 430, " RUN " },   //// RUN-BOX,
  { 2, 290,390,  420, 430, "STOP" },    //// STOP-BOX,
  { 3, 520,390,  680, 430, "CLEAR" },   //// CLEAR-BOX,
  { 0, 0  ,  0,    0,   0, "" }   //// empty-final-data
};

int CenX,CenY, WidX, WidY;
uint8_t nRGB_r, nRGB_g, nRGB_b;
int  nWndLeft, nWndRight, nWndTop, nWndBottom;

/** RS232-String buffer */
char rs232_czStrBuf[250]={""};
char Xsbuf[512];
String  strTouchRecv;
//=========================================================

int  GridX_Pos[4] = { 0, HR_IMAGE_BOX_WIDTH  ,
                      HR_IMAGE_BOX_WIDTH*2, HR_IMAGE_BOX_WIDTH*3 };  //// Image Box-Width
int  GridY_Pos[4] = { 0, HR_IMAGE_BOX_HEIGHT ,
                     HR_IMAGE_BOX_HEIGHT*2, HR_IMAGE_BOX_HEIGHT*3 };  //// Image Box-Height

//=========================================================
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);

  Serial1.setRX(17);
  Serial1.setTX(16);
  Serial1.begin(115200);  //// or 9600 etc
  randomSeed(analogRead(0));  //// random useing...
}
//---------------------------------
int is_RectWnd(int newX, int newY, IDWrect *pWndRect )
{
                if ( (newX >= (pWndRect->sx)) && ( newX <= (pWndRect->ex)) ){
                if ( (newY >= (pWndRect->sy)) && (newY <= (pWndRect->ey))) {
                        return 1;  ///// TRUE;
                } else return 0; //// FALSE
                } else {
                return  0; // FALSE
                };
}
//---------------------------------
int  getid_TouchWndPos( int tpX, int tpY, IDWrect *pWndRect )
{
 if ( ( tpX >= (pWndRect->sx)) && ( tpX <= (pWndRect->ex)) ){
   if ( ( tpY >= (pWndRect->sy)) && ( tpY <= (pWndRect->ey))) {
      return pWndRect->nTouchID;  ///// TRUE window area BOX-ID
   }else return 0; //// Where Pos ?
 } else {
   return  0; //// Where is Pos
 };
}
```

```
//---------------------------------------
void  Send_Buzz(void)
{
  ////sprintf(Xsbuf, "b 1₩r");
  Serial1.printf("b 1₩r" );
}

//---------------------------------------
void  Send_GraphWndLine(void)
{
                uint8_t  cRb, cBb, cGb;
                int  iSX,iSY,iEX,iEY;
                float  fR,fG,fB;
                int  ipX,ipY;
                int  ipW,ipH;

                cRb = random(10, 255);   cBb = random(10, 255);   cGb = random(10, 255);
                iSX = random(GraphWnd.sx, GraphWnd.ex);    /// wx -> endX,
                iEX = random(GraphWnd.sx, GraphWnd.ex);
                iSY = random(GraphWnd.sy, GraphWnd.ey);
                iEY = random(GraphWnd.sy, GraphWnd.ey);

                ipX = min( iSX, iEX);
                ipY = min( iSY, iEY);
                ipW = max(iEX, iSX) - ipX;
                ipH = max(iEY, iSY) - ipY;

                Serial1.printf("RR %d,%d,%d,%d,%d,%d,%d,%d,1₩r", ipX, ipY, ipW, ipH, 2, cRb, cGb, cBb); /// C x,y,r,R,G,B
                fR = cRb * 1.2f;              fG = cGb * 1.2f;              fB = cBb * 1.2f;
                Serial1.printf("C %d,%d,%d,%d,%d,%d,1₩r", ipX+ 3, ipY+ 3, 2, min(fR,255), min(fG,255), min(fB,255) ); /// C x,y,r,R,G,B
}
//---------------------------------------------------------------------
void  Send_DrawCircle(int  CenX, int CenY, int WidX, int WidY, UBYTE bR, UBYTE bG, UBYTE bB)
{
 sprintf( Xsbuf,  "C %d,%d,%d,%d,%d,%d₩r", CenX, CenY, WidX, WidY, nRGB_r, nRGB_g, nRGB_b);
}
//---------------------------------------------------------------------
void  Send_DrawPie(int  StaX, int StaY, int EndX, int EndY, UBYTE bR, UBYTE bG, UBYTE bB)
{
 sprintf( Xsbuf,  "l %d,%d,%d,%d,%d,%d₩r", StaX, StaY, EndX, EndY, nRGB_r, nRGB_g, nRGB_b);
}
//---------------------------------------------------------------------
void  Send_DrawLeftArrow(int  StaX, int StaY, int EndX, int EndY, UBYTE bR, UBYTE bG, UBYTE bB)
{
 // rr box +  ellipse +  ellipse +  rr box
 int xOfs, yOfs;
 xOfs = (EndX- StaX ) / 3;
 yOfs = (EndY - StaY) / 4;
 Serial1.printf("RR %d,%d,%d,%d,2,%d,%d,%d,0₩r", StaX+ xOfs , StaY+ yOfs , xOfs*2, yOfs*2, bR,bG,bB);
 Serial1.printf("L %d,%d,%d,%d,%d,%d,%d₩r", StaX+ 1, StaY+ yOfs*2, StaX+ xOfs,          StaY+ 1, bR,bG,bB);  // '/'
 Serial1.printf("L %d,%d,%d,%d,%d,%d,%d₩r", StaX+ 1, StaY+ yOfs*2, StaX+ xOfs,  StaY+ yOfs*4-1, bR,bG,bB);  //  '₩'  back-slask
 Serial1.printf("L %d,%d,%d,%d,%d,%d,%d₩r", StaX+ xOfs, StaY+ 1, StaX+ xOfs,  StaY+ yOfs*4-1, bR,bG,bB);  //  '₩'  horiz-line
}

//---------------------------------------------
void  Send_DrawButton(  IDWrect *pWndRect, UBYTE bR, UBYTE bG, UBYTE bB)
{
 int   nlen, nPixel;
 int   pStrX, pStrY;
 char  *prMsg;
 int   StaX, StaY, EndX, EndY, WidX, WidY;
 // fill round-box
 prMsg = pWndRect->czStrTitle;
 StaX = pWndRect->sx; EndX = pWndRect->ex - StaX;
 StaY = pWndRect->sy; EndY = pWndRect->ey - StaY;
 Serial1.printf("RR %d,%d,%d,%d,5,%d,%d,%d,1₩r", StaX, StaY, EndX, EndY, bR, bG, bB);
 WidX = pWndRect->ex - StaX; WidY = pWndRect->ey - StaY;

 nlen   = strlen( prMsg);
 nPixel = nlen * 16;  // FONT : 9 pixel width char. default::
 pStrX = (StaX +  WidX/2) -  (nPixel / 2);
 pStrY = (StaY +  WidY/2) - 16 +  3;  /// Font:: 16 pixel-Height:

 if (nlen != 0){
    Serial1.printf("flW₩r");    /// 24x24, 16x24 , font large
    Serial1.printf("fc 9,9,9₩r");         //// Drak-black-Color
    Serial1.printf("f %s,%d,%d₩r", prMsg, pStrX+ 1, pStrY+ 2);
    Serial1.printf("fc 255,255,255₩r");  //// White Text
    Serial1.printf("f %s,%d,%d₩r", prMsg, pStrX, pStrY);
    Serial1.printf("fs₩r");    /// 16x16, 8x16 , font small
 } else {
   //// Serial1.printf("f L0!!,%d,%d₩r", pStrX, pStrY);
  return;
 }
}
```

```
//--------------------------------------

void  Send_DrawfillBox(  IDWrect *pWndRect, UBYTE bR, UBYTE bG, UBYTE bB)
{
 int    pStrX, pStrY;
 int    StaX, StaY, EndX, EndY, WidX, WidY;

 StaX = pWndRect->sx; EndX = pWndRect->ex - StaX;
 StaY = pWndRect->sy; EndY = pWndRect->ey - StaY;
 Serial1.printf("RR %d,%d,%d,%d,2,%d,%d,%d,1₩r", StaX, StaY, EndX, EndY, bR, bG, bB);
 ////    Serial1.printf("RR 15,15,770,335,2,63,63,63,1₩r" );    //// back-gl-box-gray
}

////--------------------------------------
int get_CheckTouchKey(void)
{
                char rdByte;
                int32_t  tlastpos;
                String   tStrPoint="";
                String   xStrDigit="";
                long  tRiX, tRiY;
                int    j;
                int    nIDbtn=(0); /// non-key

                /// Serial get <t>...!t!  and scanf("%d,%d", &posX, &posY);
                while( Serial1.available() )
                {
                rdByte = Serial1.read();
                strTouchRecv.concat (rdByte);
                if (rdByte == '!')        //// Touch-up,release-check
                {
                                tlastpos = strTouchRecv.lastIndexOf("t(");
                                ///if (strTouchRecv.substring("<t>") ==
                                if ( tlastpos > 0) {
                                                tStrPoint = strTouchRecv.substring(tlastpos);
                                    Serial.print(tStrPoint);

                                                xStrDigit = tStrPoint.substring(2); /// "t(" removing...
                                                tRiX = xStrDigit.toInt();
            tlastpos =      tStrPoint.indexOf(',');
                                                // Serial.printf("Is=%d", tlastpos);
            if (tlastpos <=0) return (-1); /// error find ',' not..
                                                xStrDigit = tStrPoint.substring(tStrPoint.indexOf(',')+ 1);
                                                tRiY = xStrDigit.toInt();
                                                strTouchRecv = "";
                                                Serial.printf("POS(%d,%d)|", tRiX, tRiY);

                                    //=== check-buttons- show-Windows
                                                for(j=0; j < 4; j++ ){
                                                                nIDbtn =  getid_TouchWndPos( tRiX, tRiY, &BtnBoxList[j] );
                                                                if ( nIDbtn > 0) break;
                                                };
                                                return nIDbtn;
                                };
                    return (-1);                /// error-key
                } else {
                                // if ( recv over size) clear !!
                };
                };

                return (0);   //// not load-key
}
//-------------------------------------------------
void play_horse_loop(void)
{
                static int nEntryCnt = 0;
                static int eki=0, ekj=0;
                int  i,j;

                if (++nEntryCnt % 10 != 0) return;

   if (eki < 4)
   {
    ekj = (ekj < 4) ?  1+ekj : 0 ;
                //------------------------------
    sprintf( rs232_czStrBuf, HR_IMAGE_CUT_SHOW_MSG, //
            GridX_Pos[eki] , GridY_Pos[ekj],
            HR_IMAGE_BOX_WIDTH - 3, HR_IMAGE_BOX_HEIGHT- 2 );  // X-grid line thickness .3, // Y-grid line thickness .2
    ////Serial.printf( "Out>>₩"%s₩" !! ₩n₩r" , rs232_czStrBuf );
    Serial1.printf( rs232_czStrBuf );
                //------------------------------
    eki++;
   } else {
    eki = 0;
   };
}
```

```
//============================================================

///================================================
void loop() {
  // put your main code here, to run repeatedly:
   int  i, j,k,tmp, m;
   int  nKeyCode;
   int  nWorkMode=1;

   reLoad_Main_SCREEN: ;

   Serial1.printf("i test3.bmp,0,0₩r" );
   Serial.printf(" Control Text Sending ₩'Horse Load Image₩' ");
   Serial1.printf( HR_IMAGE_LOAD_MSG );
   delay(250);

   Send_DrawfillBox( &GraphWnd, 63,63,63);  //// back-gl-box-gray
   Send_DrawButton(  &BtnBoxList[0], 255, 127, 39);  /// Run,red-yello color
   Send_DrawButton(  &BtnBoxList[1],  34, 177, 76);  /// Stop,green color
   Send_DrawButton(  &BtnBoxList[2],   0, 162, 232); /// Clear,blue color

   ////=== thin x3 => thick(3) arrow
   Send_DrawLeftArrow( 480,390, 510, 430, 0,162,232);
   Send_DrawLeftArrow( 481,392, 508, 428, 0,162,232);
   Send_DrawLeftArrow( 482,394, 509, 429, 0,162,232);

   ///// Touch get-key-control LOOP:::
   while(1)
   {
                ///-- get touch read. Serial1 , convert Touch IDCode
                nKeyCode = get_CheckTouchKey();
                if (nKeyCode != (-1) )   /// ! not key code or error get.
                {
                if (nKeyCode != 0) nWorkMode = nKeyCode;
                };

                //-------------------------------------------------------------
                switch(nWorkMode)
                {
                case 0:   break;  /// idle-state
                case 1:          /// RUN
                          nWorkMode = 1; /// Next to run .. run...

                   Send_GraphWndLine();     /// box_fill-play
                          play_horse_loop();       /// horse-play
                break;

                case 2:          /// STOP
                          nWorkMode = 0; /// Next to idle State
                break;

                case 3:          /// CLEAR- reload
                          Serial1.printf("RR 15,15,770,335,2,63,63,63,1₩r" );   //// back-gl-box-gray
                          nWorkMode = 0; /// Next to idle State
                          goto reLoad_Main_SCREEN;
                break;

                defalut:   break;
                };

                delay(10);
   };
   ////==================================

   Serial.printf(">> conterol UnLoad Image. ");
   Serial1.printf(HR_IMAGE_UNLOAD_MSG);
}
```

8. 동작 결과는 사진과 같다.

<MST070M-AUT> 의 터치 좌표 해석및 일반형태의 egl- 그림그리기 기능으로
버튼을 그리는 fill-box 기능과 터치 좌표의 해석예제임.

[RUN] 그림그리기,
[STOP] 정지.
[CLEAR] 처음상태로 화면 청소.

> 사용된 사진 배열의 눈금 간격 보기