คู่มือการใช้งาน
User's Manual
ET-BASE PX32 V1.0
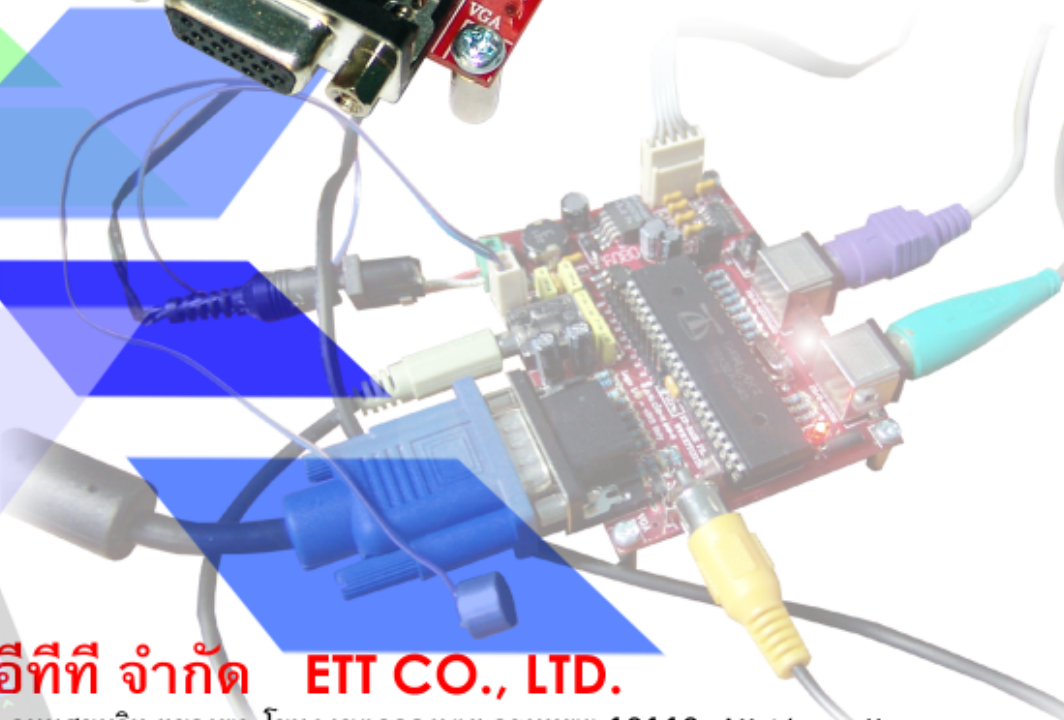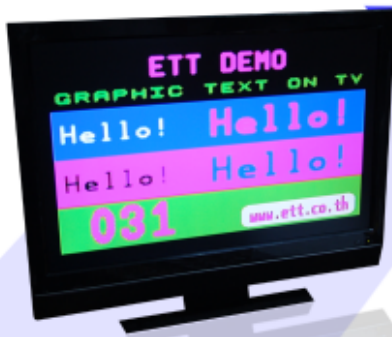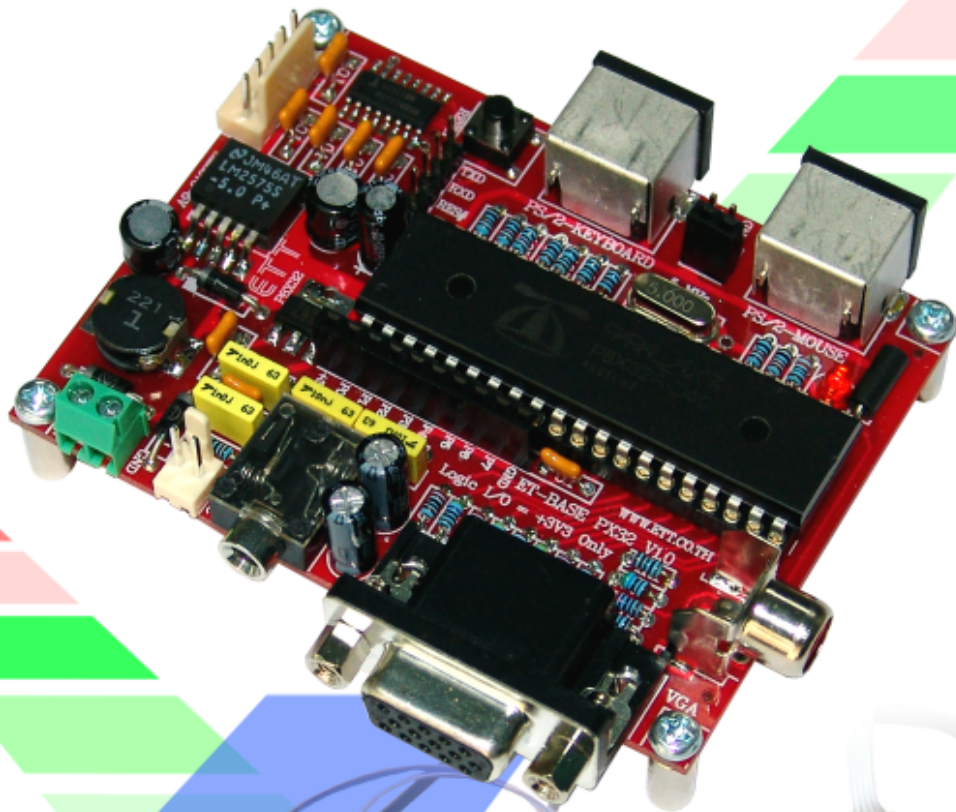
Contents

ET- BASE PX32 V1.0

Board ET-BASE PX32 is the Training Board that uses high speed MCU No.P8X32A-D40 from PARALLAX that is a 32 BIT 8 Cog Multiprocessor. The structure of MCU is DIP 40 PIN that can run with maximum frequency of 80 MHz and run at the Voltage 2.7-3.6 VDC. It uses Software Tool "Propeller V1.06" to develop program; this Software can be used to write program, Compile Code and Download Code through RS232 instantly (can not Debug, see the operation step by step). It uses SPIN Language to write program that makes user can develop program easily and quickly because this Program Propeller has already had Library to connect between peripherals and MCU P8X32A, so user can use it instantly; such as Library RS232, Library VGA, Library TV, Library Keyboard, and Mouse.

1. Specifications of MCU P8X32A

1.1)   MCU 32 BIT 8 Cog Multiprocessor (8 CPU in one Chip)

1.2)   DIP 40 PIN has Port I/O 32 Pin

1.3)   MCU runs at Voltage 2.7-3.6 VDC and I/O Port can drive current Source/Sink as 40 mA at 3.3 VDC

1.4)   Run with maximum frequency of 80 MHz; it can be selected to use as External Clock or Internal Clock; and it has PLL insides.

1.5)   Has internal RAM to store 32 Kbyte Code when removing Power Supply for MCU, it makes Code deleted. When user actually uses, must externally interfaces I2C EEPROM to store Code.

1.6)   Pin that is configured to be Input function can receive Voltage Input not higher than VDD (2.7-3.6) only.

1.7)   Speed operation internal Chip is 20 MIPS/cog

2. Specifications of Board ET-BASE PX32 V1.0

2.1)   Use MCU P8X32A DIP 40 PIN

2.2)   I2C EEPROM #24LC256 (32Kb) to store Code Program

2.3)   Board can receive Voltage Input 6-12 VDC; there is a Regulator on board to adjust voltage to be 5V and 3.3V

2.4)   Use Crystal 5.00 MHz (External) can use PLL internal Chip to multiply frequency up to 80 MHz

2.5)   Use SPIN Language to develop program by using Tool "Propeller" (Freeware); this Tool is used to write program, Compile and Download insides. Moreover, there is Library that is used to interface between MCU and peripherals on board. User can download further Library from website http://www.parallax.com/ and user can see Help of Program Propeller to learn how to use SPIN Language.

2.6)   Download Code through RS232 and use Propeller to select to download and store Code in EEPROM (External) or RAM inside MCU.

2.7)   Board arranges fixed I/O to interface with peripherals as follows;
- Port Key Board(PS2),
- Port Mouse (PS2),
- Port RS232,
- Port VGA,
- Port TV(AV)
- Port MIC,
- Port Headphone,
- Port I/O 8 PIN for other applications

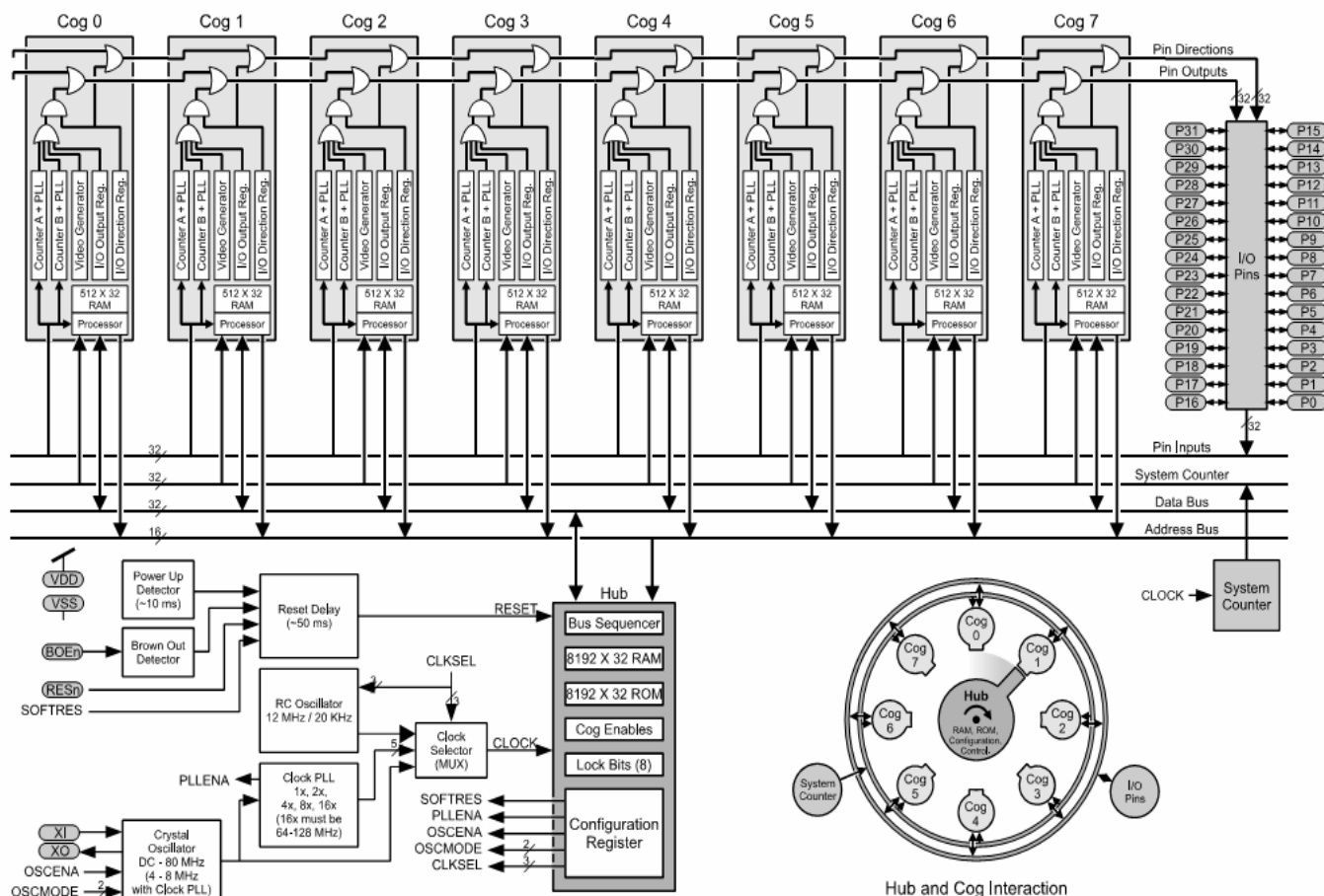# 3. Structure and Block Diagram of MCU P8X32A

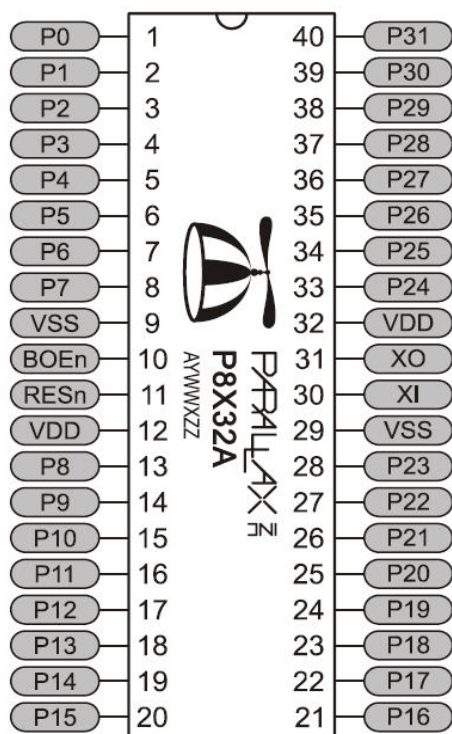Figure 3.1 displays block diagram of Chip P8X32A.

Figure 3.2 displays structure of Chip P8X32A (DIP 40 PIN).

Refer to figure 3.1; it displays internal structure of Chip that consists of 8 Processors called "Cog". This device is designed to be high speed operation, low energy consumption, small dip, highly flexible to run through all 8 Processors; moreover it can run simultaneously but it is independent to each other. This Chip can share resources through HUB Center, so it makes each Cog can work and use resources together. In the part of Clock system, it will be shared to each Cog and refer to the same Time Base. Interrupt is not be used with this Chip but it is used to configure position for jumping to run of Cog to directly.

Figure 3.2 is structure of Chip as DIP 40 PIN; we can classify Pin as follows;

Table 3.1: Pin Descriptions

PIN Name	Direction	Description
P0-P31	I/O	It is Port I/O that has a half Logic of VDD or 1.6 VDC at Voltage 3.3 VDC. There are 32 PIN but their 4 PIN are configured to be a special function after Power-Up or RESET as follows; **P28-I2C SCL** is connected to external EEPROM. **P29-I2C SDA** is connected to external EEPROM. **P30-Serial TX** is connected to Download Code and send data through RS232 **P31-Serial RX** is connected to Download Code and receive data through RS232
VDD	---	3.3 V Power (2.7-3.6 VDC)
VSS	---	Ground
BOEn	I	Brown Out Enable (Active Low): It is connected to VDD or VSS. If this Pin is Low, Pin RESn will be Output function but it can Drive Low to reset Chip; on the other hand, if this Pin is High, it will be Input function.
RESn	I/O	Reset (Active Low): When it is Low, Chip and all Cog will be disabled; Pin I/O will be free; and Chip will be restarted within 50ms after Logic at RESn is converted from Low to High.
XI	I	Crystal Input
XO	O	Crystal Output

4.Structure of Board ET-BASE PX32 V1.0

Board ET-BASE PX32 V1.0 has already configured Port to interface with particular peripherals, except 8 PIN in the part of Port I/O (P0-P7) only because it is provided for user to further interface with other devices such as LED, SW, SD Card, and LCD.

Table 4.1: Interfacing Port to use on Board ET-BASE PX32

| Port Number	Function
P0-P7	It is configured to be I/O that can be further interfaced with other applications.
P8-P9	It is configured to interface with Microphone.
P10-P11	It is configured to be AUDIO Out Stereo that can be interfaced with Amplifier or Headphone.
P12-P15	It is configured to be Port Video Out (NTSC/PAL) to interface with AV Channel of TV.
P16-P23	It is configured to be Port VGA to interface with PC Screen or LCD.
P24-P25	It is configured to interface with PS/2 Mouse.
P26-P27	It is configured to interface with PS/2 Key Board.
P28-P29[System]	It is interfaced with I2C-EEPROM to store Code Program (P28:Clock; P29:Data).
P30-P31[System]	It is configures to be Port RS232 to download program and receive-transmit data through RS232(P30:Rx; P31:Tx).
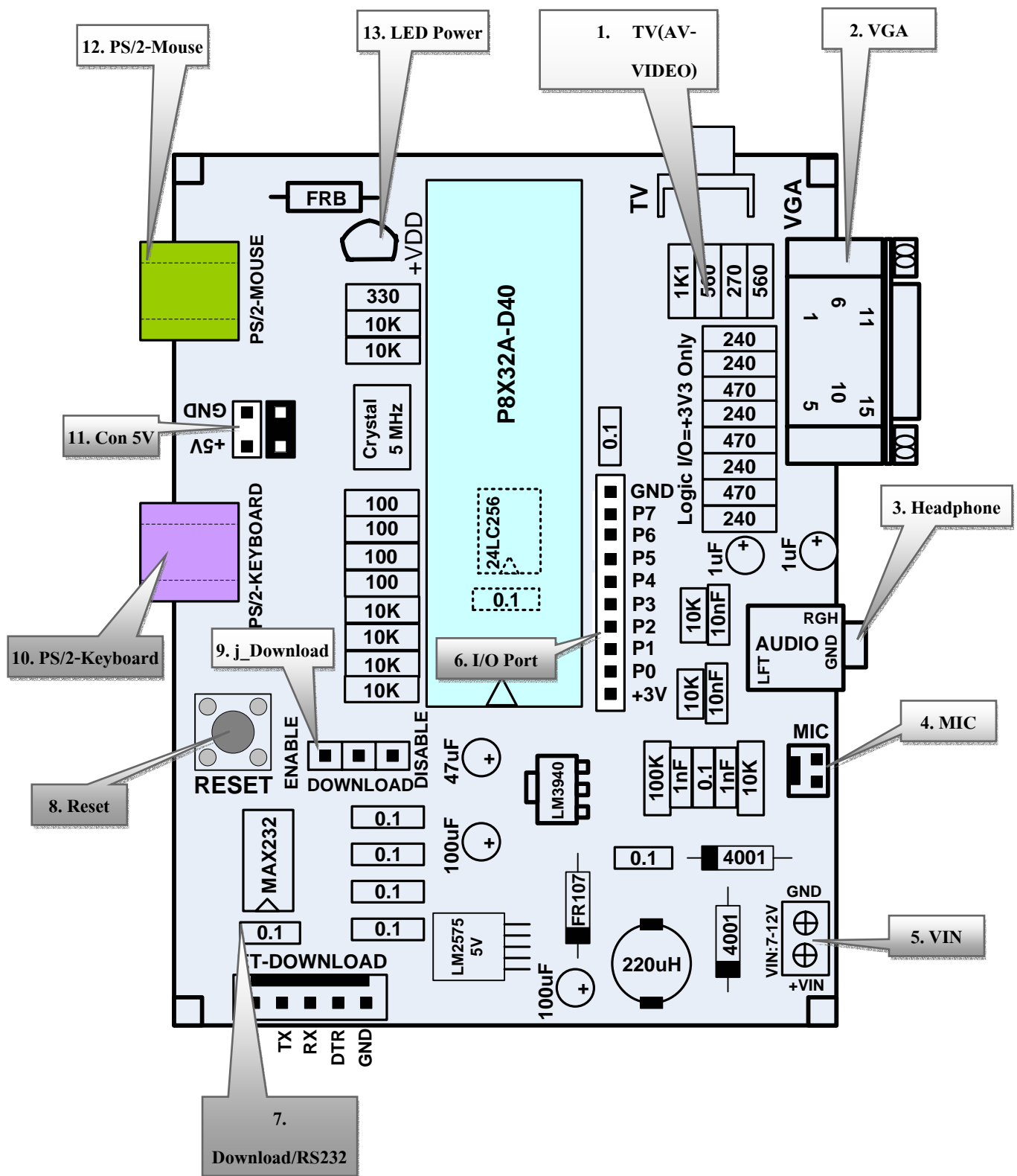
Figure 4.1 displays structure of Board ET-BASE PX32 V1.0.

No.1: TV (AV-VIDEO) is VIDEO OUT Connector to interface with TV; in this case, it must be interfaced at VIDEO IN Connector of TV.

No.2: VGA is VGA OUT Connector (DB15) to interface with Monitor of PC such as LCD.

No.3: Headphone is AUDIO OUT Connector to interface with Headphone or Amplifier.

No.4: MIC is Microphone Connector.

No.5: VIN is DC 7V-12V Connector for board; please carefully interface cathode and anode correctly.

No.6: I/O Port is Port I/O 8 PIN that is provided to further interface with I/O such as LCD, Key, LED or SD Card.
Be Careful: For I/O Port, if using it as Input function, it can receive voltage into Port not higher than VDD or 3.3V. Don't directly interface with 5V Voltage because it makes Port Pin damaged, should use R to interface with Divider to reduce it to 3.3V first.

No.7: Download/RS232 is connector to download program and receive-transmit data through RS232; in this case, there is Line Driver Max232 to adjust signal level of Rx, Tx from 3.3V to ±12V, so it can be interfaced with PC.

No.8: Reset is SW. to reset operation of MCU and it will run at Logic 0.

No.9: Jumper Download when user needs to download Program, must set Jumper to Enable side because this Jumper will interface RES PIN of MCU with DTR PIN of Download Connector; on the other hand, if Set Jumper on Disable side, it will remove RES PIN of MCU from DTR PIN of Download Connector to protect signal Reset from PC after it has already downloaded completely but not removed Download Cable yet.

No.10: PS/2-Keyboard is PS/2 Connector to interface with keyboard

No.11: Con 5V is DC 5 V Output Connector to interface with 5 V Power Supply from board to external.

No.12: PS/2-Mouse is PS/2 Connector to interface with Mouse.

No.13: LED Power displays operating status of Power Supply.

5. Initial Applications of Program Propeller

5.1) Install Program Propeller V1.06 into the computer first.
5.2) Then, open Program Propeller and it will display window as shown in picture 5.2 below
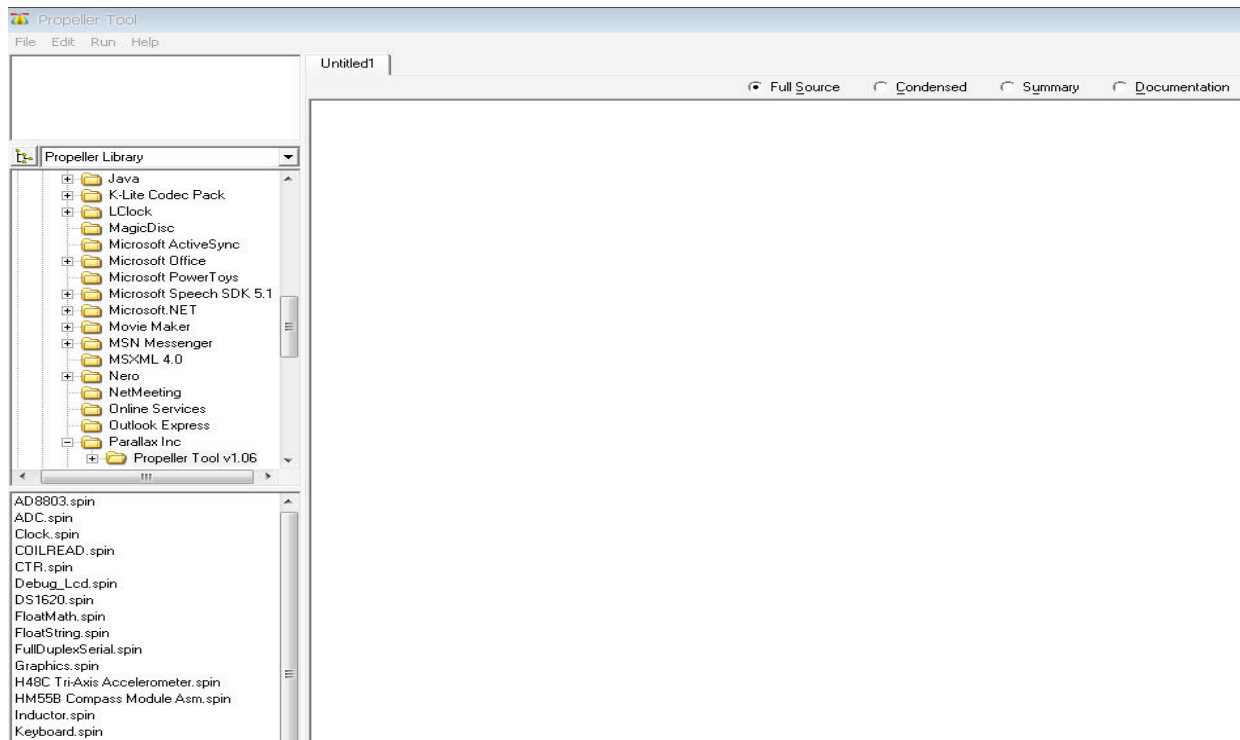
Picture 5.2

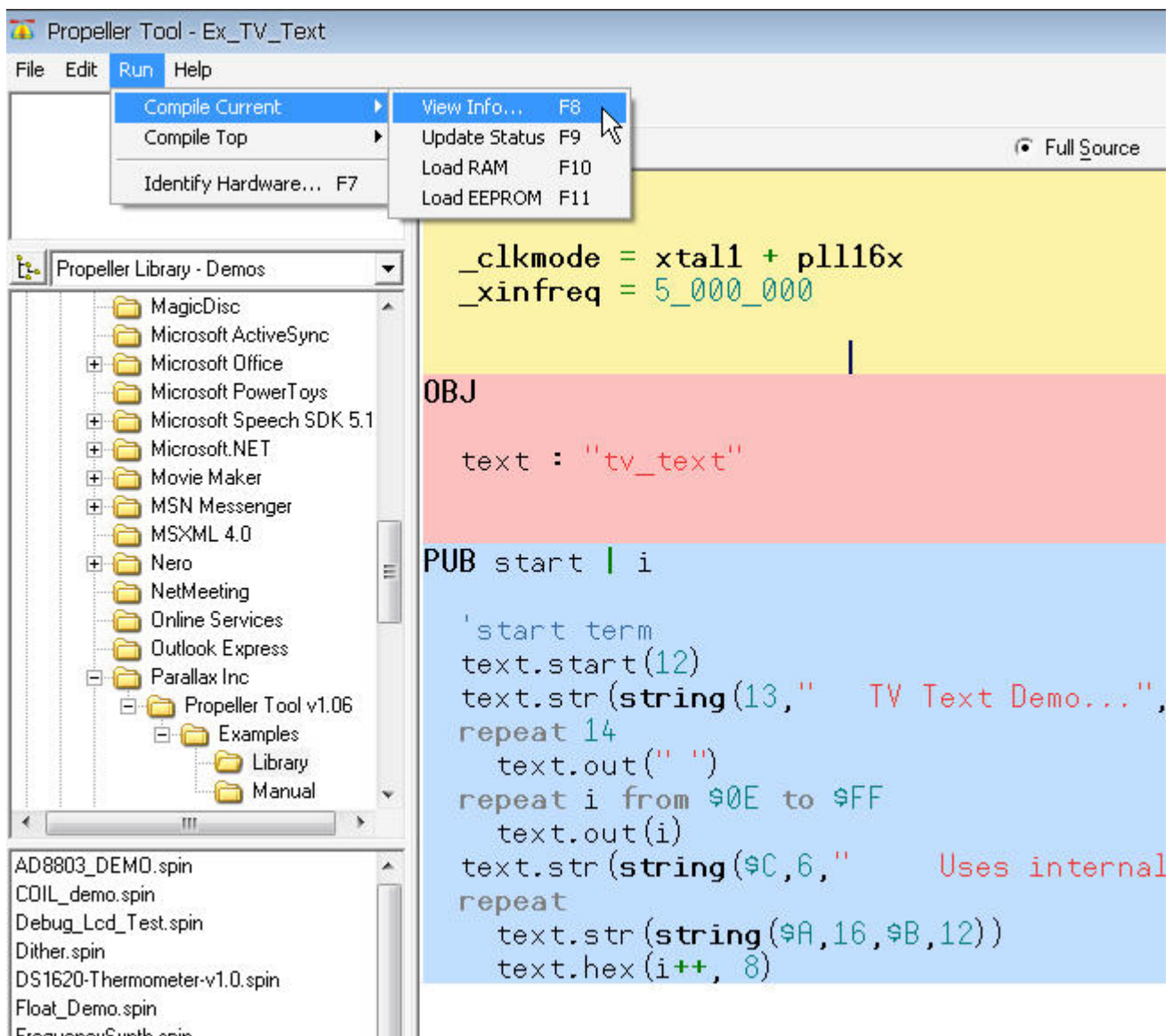From the picture in the window on the right hand (Tab Untitled1), it is available area that can be written program.

5.3) After has already written program, click **Menu File**, select **Save As** to save file with file sure name **.spin**.
5.4) When has already saved file, click **Menu RUN,** select **Compile Current**, then it will display menu to select value as shown in the picture 5.3. If select…
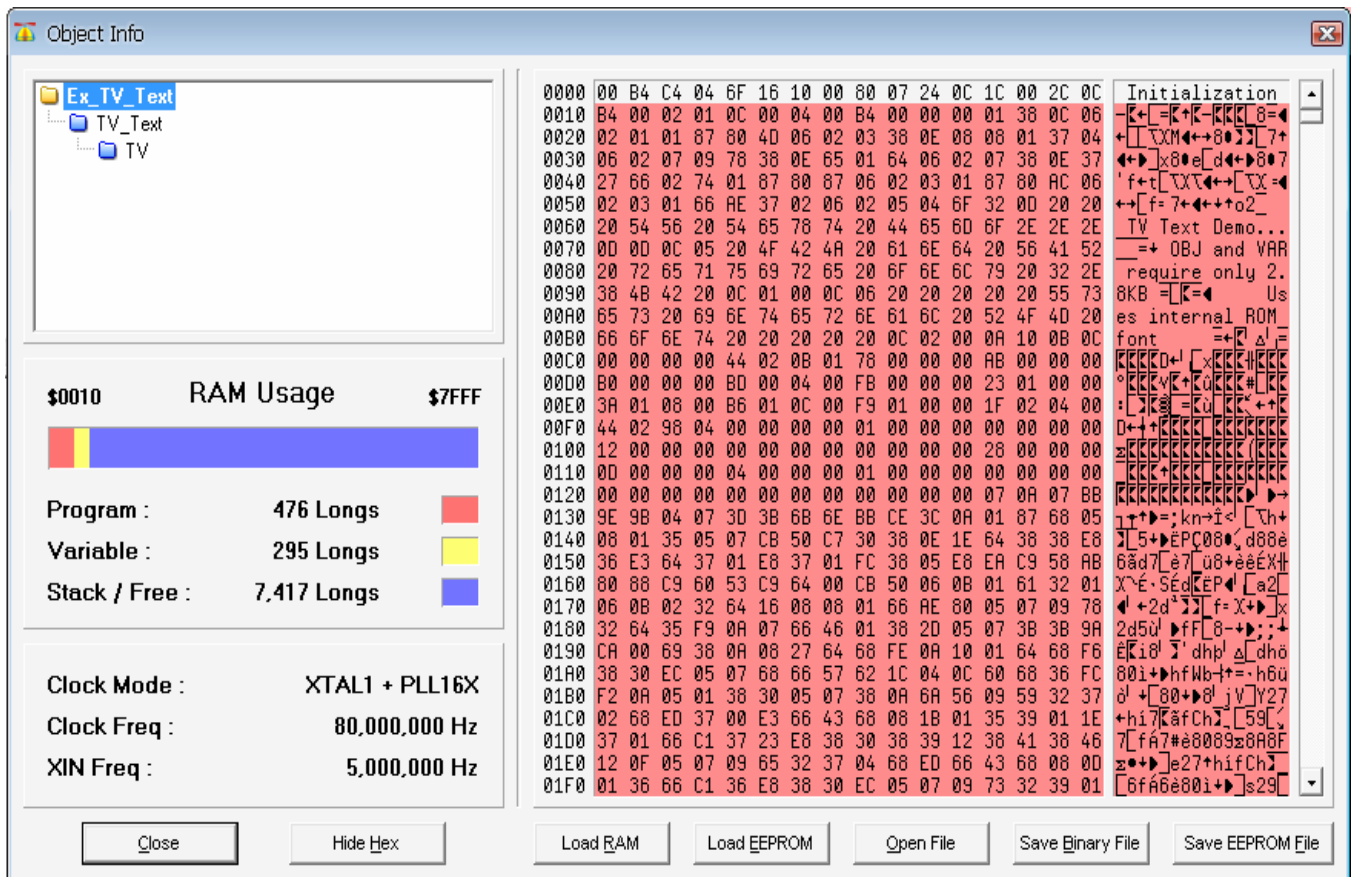
Picture 5.2

View Info…= It only is Compile Program and there is window
Object Info to report data of the written program as
shown in the picture 5.3; it displays the used RAM
area, the selected Clock Frequency. The data of the
example program that is shown in the picture 5.3 can
be described as follows;

- It is data of the Filename Ex_TV_Text. Internal
  this file uses Function in File Library called
  TV_Text and internal File library TV_Text uses
  Function in File Library called TV.

- This File Ex_TV_Text uses RAM area as 476 Long in
  the part of Program and 295 Long in the part of
  variable, so there is an available area as 7,417
  Long. In this case, 1 Long equals 32 Bit or 4
  Byte.

- This program uses Clock Mode in XTAL 1+PLL16X and actual Clock Frequency that is used equals 80MHz; moreover, it uses external Crystal 5MHz (5MHz x PLL16 = 80MHz). After the window Object Info is displayed, click Button **Load Ram** or Button **Load EEPROM** to download program. If select Button **Load RAM**, program will be loaded and stored in RAM of MCU directly; however, if select this method, program will be deleted when pressing SW.RESET or removing Power Supply of board. On the other hand, if select Button **Load EEPROM**, program will be loaded and stored in external EEPROM that is interfaced first and then MCU will load Code from EEPROM to run in RAM automatically; if select this method, program is not be deleted when pressing SW.RESET or removing Power Supply of board.

Picture 5.3

Load RAM = It is Compile and Download program into RAM directly without displaying the window Object Info. Moreover, program will be deleted if pressing SW.RESET or removing Power Supply.

Load EEPROM = It is Compile and Download program into EEPROM that is externally interfaced without displaying the window Object Info. In this case, the program is remaining although Reset or removing Power Supply.

5.5) The method to Set Baud Rate or Set COM Port for downloading Code is unnecessary because Program Propeller will set Baud Rate and set COM Port automatically. If user wants to set COM Port by self, click **Menu Edit** and select **Preference**. Then, it will display window, click **Tab Operation** in the blank of **Serial Port Search:** and then user can select the desired COM Port (normally = Auto).

5.6) This Program Propeller is used to develop program with SPIN Language; in this case, user can see more detail and learn how to use the commands, click **Menu Help** and then select **Propeller Manual (pdf).**

6. Test for Running Example Program

1) Install Program Propeller Tool V1.06.exe; copy example of ETT from CD ROM and then paste it in PC.
2) Interface Download Cable from COM Port of PC to Download Connector of board and then set Jumper DOWNLOAD (No.9) on Enable side.
3) In this case, it tests the display with TV, please follow theses instructions. First, interface signal cable from TV OUT of board with VIDEO IN Connector of TV, turn on TV in the AV Channel and finally supply power into board.
4) From example of ETT; enter Folder **"Ex1_TV"**, click Folder **"Ex1.2_TvText_Graphic",** double click filename **"Ex1.2_TV_TextGPH.spin",** and Program Propeller will open this file.
5) Click **Menu RUN,** select **Compile Current** and then select **Load RAM** or **Load EEPROM**, program will be compiled and downloaded into MCU.
6) After it has already downloaded program, on the TV Screen will display message "Hello!".
7) If user wants to test other examples, should follow the instructions above and then prepare the unnecessary devices to interface with Port to test the desired example.

Beside the written example of ETT, user can test for running example that is in Program Propeller on ETT board as follows;

1) Open Program Propeller; see the picture 6.1 below in the left blank at the position that is pointed by mouse, click **Propeller Library-Demos** and user will see the example file

with file sure name **.SPIN** will be displayed on the left-bottom side.

2) Double click the desired example file to run, should select file that supports interfaces that are provided on ETT board. Example file that is interfaced through Port VGA is VGA_DEMO.spin; example file that is interfaced through Port Mic, Audio such as Microphone_to_Headphones.spin, SingingDemoSeven.spin.

3) Click **Menu RUN**, select **Compile Current** and then select **Load RAM** or **Load EEPROM**, program will be compiled and downloaded into MCU.

4) See the operating program from the devices that are interfaced.

If user opens program simultaneously and there are many windows repeatedly, Program Propeller will only compile and download program in the part of file that is active on the screen.
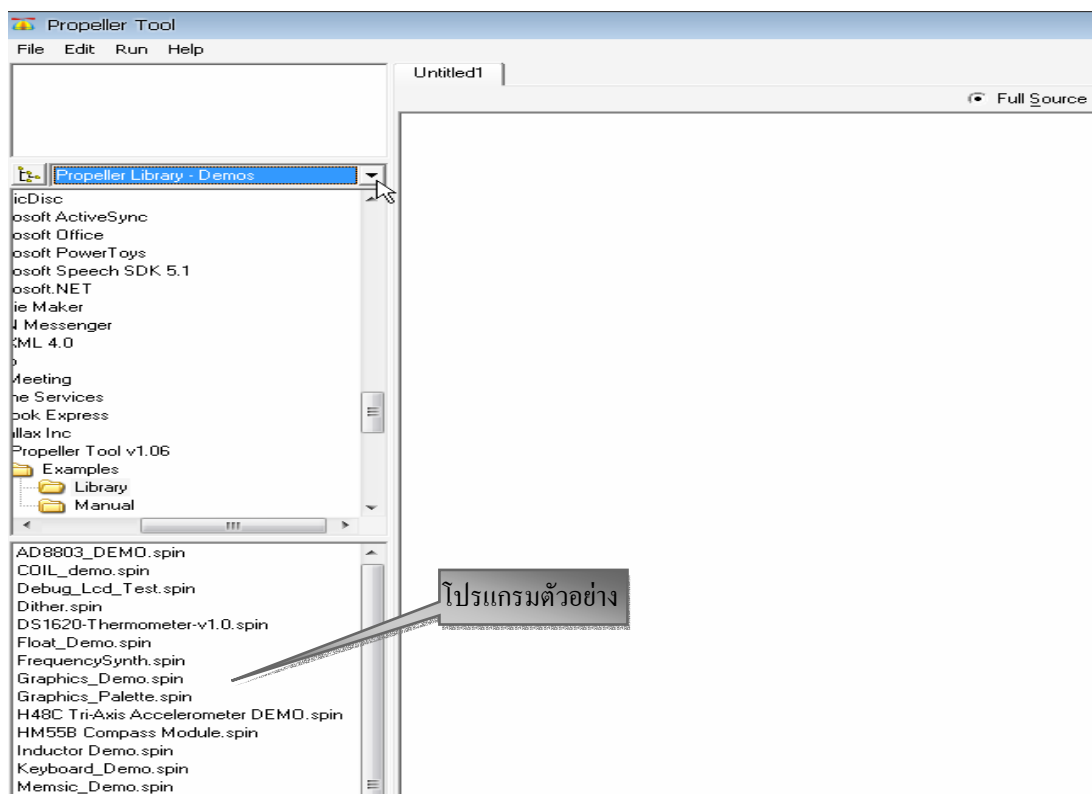
Picture 6.2

7. Interfacing Download Cable and RS232 Cable

- For the cable for downloading program and the cable for transmitting-receiving data through RS232 between Board PX32 and PC, it uses the same cable and the method to interface is shown in the picture 7.1 below.
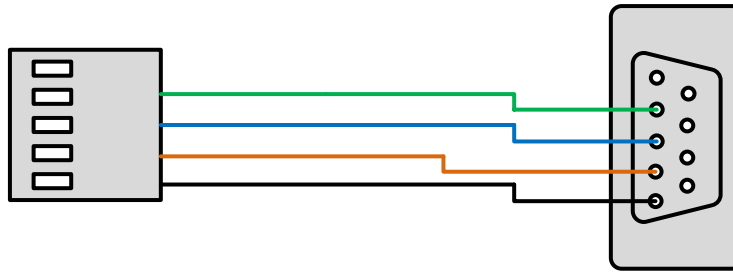
Picture 7.1 displays the method to interface Download Cable & RS232 Cable between Board PX32 and PC.

If receiving-transmitting data through RS232 between Board PX32 and other Controller Boards, user can interface the cable as shown in the picture 7.2 below;
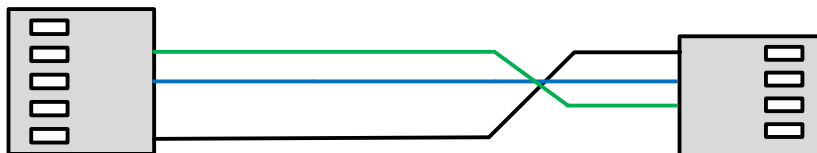
Figure 7.2 displays the method to interface RS232 Cable for transmitting-receiving data between Board PX32 and other Controller Boards.

- When downloading program into Board PX32; should interface Download Cable on Connector 5 PIN with connector of Board PX32 and on the side Connector DB9 must be interfaced with COM Port of PC as shown in the picture 7.3. Then, set Jumper Download (No.9) on Board PX32 on Enable side when user wants to receive-transmit data through RS232 between Board PX32 and PC or other Controller Boards. After it has already downloaded program, should set Jumper Download (No.9) on board on Disable side.

Picture 7.3 displays the method to directly interface Download Cable with Com-Port of PC.
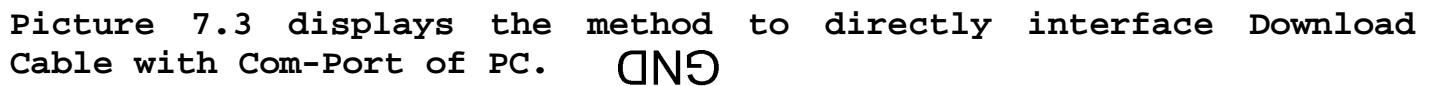
If using PC or Note Book that has not any Com-Port Connector, user must use converter to converse USB to RS232 first. If using converter from ETT version ET-USB/RS232 Mini, it will be interfaced as shown in the picture 7.4.
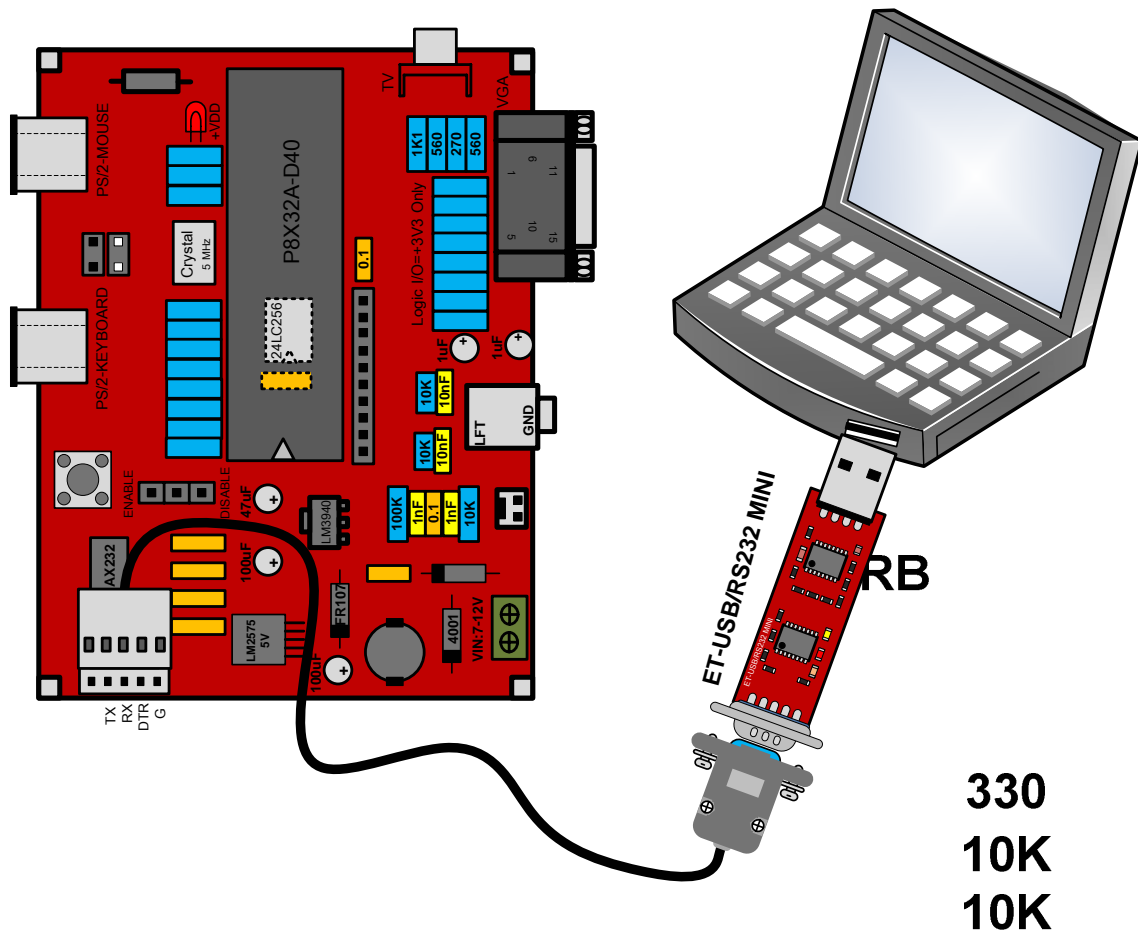
Figure 7.4 displays the method to interface Download Cable through Converter version ET-USB/RS232 Mini.

8. Restrictions of Development Program with SPIN Language on Tool Propeller

8.1) When the SPIN Language is used, we can see it as Block that has 6 blocks. If we want to write value, we must type the particular name of each block and must always start typing at the end left Column of the window that is used to write program. The name of all 6 Blocks is listed below;

1.) **CON** – This Block is used to configure the constant value for variable that will be used. The variable that is declared in this Block can be called to use in the same file; as example,

```
CON

        _clkmode        = xtal1 + pll16x
        _xinfreq        = 5_000_000
             tt         = 20
```

2.) **VAR** – This Block is used to declare variable that will be used to write program. The variable that is declared in this Block can be called to use in the same Project File; as example,

```
VAR
        long    testerror
        long    vlong
        word    vword
        byte    vbyte
```

3.) **OBJ** – This Block is used to configure filename from external, so it can call to use the function from external file (this external file is not File Library of Propeller; if user want to use the function in that file, must copy file and then paste it in the same Project file that is written by user). The method to see function that is in File Library of Propeller is shown in the figure 6.1. In the blank that is pointed by mouse, select **"Propeller Library";** then in the bottom window, it will display filename **.spin** that is File Library of Propeller. When double click File Library, the file will be opened and there are the written functions. User can use these functions in the program as required; as example;

```
OBJ
        term    : "tv_terminal"
```

When using function that is in file **Library " tv_terminal",** must be replaced by term.*function's name* such as term.out(12)

4.) **PUB** – This Block is used to write the desired program and use the written functions in this block both in the same Project File and different Project File. Every Project file that is generated, there is Block PUB at least 1 Block and Propeller will consider the first Block PUB as the main Block of program and the next Block PUB that is next to the first Block PUB will be considered as the sub program that will be used by the main Block. When type word "PUB" in this Block, user must specify name for the Block PUB, this name is the same as function's name; as example

```
PUB stop

  vga.stop
```

When user wants to declare the variable internal PUB, must add sign "|" after the name and then follow by the variable's name. The variable that is declared is only used internal PUB and the variable size that is declared is 32 Bit.

If user wants to receive the external value to be used with the function, must add **(the first variable's name, the second variable' name,…)** after the function's name. Note,

When user wants to send value from function to the external, must add sign ":" after the name and then follow by the variable's name that need to send out as example;

```
PUB Ticks(Pin) : Microseconds | cnt1, cnt2

  outa[Pin]~
  dira[Pin]~~
  outa[Pin]~~
  outa[Pin]~
  dira[Pin]~

  waitpne(0, |< Pin, 0)
  cnt1 := cnt
  waitpeq(0, |< Pin, 0)
  cnt2 := cnt
  Microseconds := (||(cnt1 - cnt2) / (clkfreq / 1_000_000)) >> 1
```

This example, the PUB's name is **Ticks**; for **(Pin)** is the variable declaration to receive the external value and then store it in the variable Pin; for **: Microsecond** is the variable declaration to send out from the function; and finally, **| cnt1, cnt2** is variable declaration for using internal Block PUB Tricks. When it is actually used, user can declare the necessary value for application without declaration totally.

5.) **PRI** – This Block is used to write the required program the same as Block PUB; it only is different in the part of Function that is written in the Block. This Block will be considered as the sub program to be used in the same Project File only; it can not be used by the different Project File. For the method to declare variable is the same as Block PUB.

```
PRI bound(i, delta) : b | d

    d := bx_div[i]
    b := bx_min[i] + (bx_acc[i] := bx_acc[i] + delta
```

6.) **DAT** – This Block is used to configure data table or write Assembly Code as example;

```
DAT

'********************************
'* Assembly language VGA driver *
'********************************

                        org

entry                   mov       taskptr,#tasks

                        mov       x,#8
:init                   jmpret    taskret,taskptr
                        djnz      x,#:init


vgacolors               long      $C000C000       'red
                        long      $C0C00000
                        long      $08A808A8        'green
                        long      $0808A8A8
                        long      $50005000        'blue
                        long      $50500000
```

When these 6 Blocks are actually used, user can declare only the used Block, it is unnecessary to declare all Blocks. Moreover, each Block can be declared repeatedly more than 1 Block. Remember, the Block PUB that is declared repeatedly, its function's name that is typed after PUB must be different.

8.2) The character that is typed as command or variable's name can be typed with capital letters or small letter as desired. Propeller considers it as the same variable unlike C Language.

8.3) When replacing the decimal number that is more than 3 digits must always separate it with sign "_" such as x := 1_200.

8.4) The method to use sign "=" is listed as follows; if using in Block CON, it is =, if using in Block OBJ, it is :, if using in Block PUB, it is :=

8.5) Using the numeral system in Propeller: $ precedes the constant value such as $56 to represent the hexadecimal number; % precedes the constant value (1 and 0) such as %1001 to represent the binary number, %% precedes the constant value (0, 1, 2, 3) such as %%2130 that will be considered as Quaternary Number. It means that it considers that 1 digit
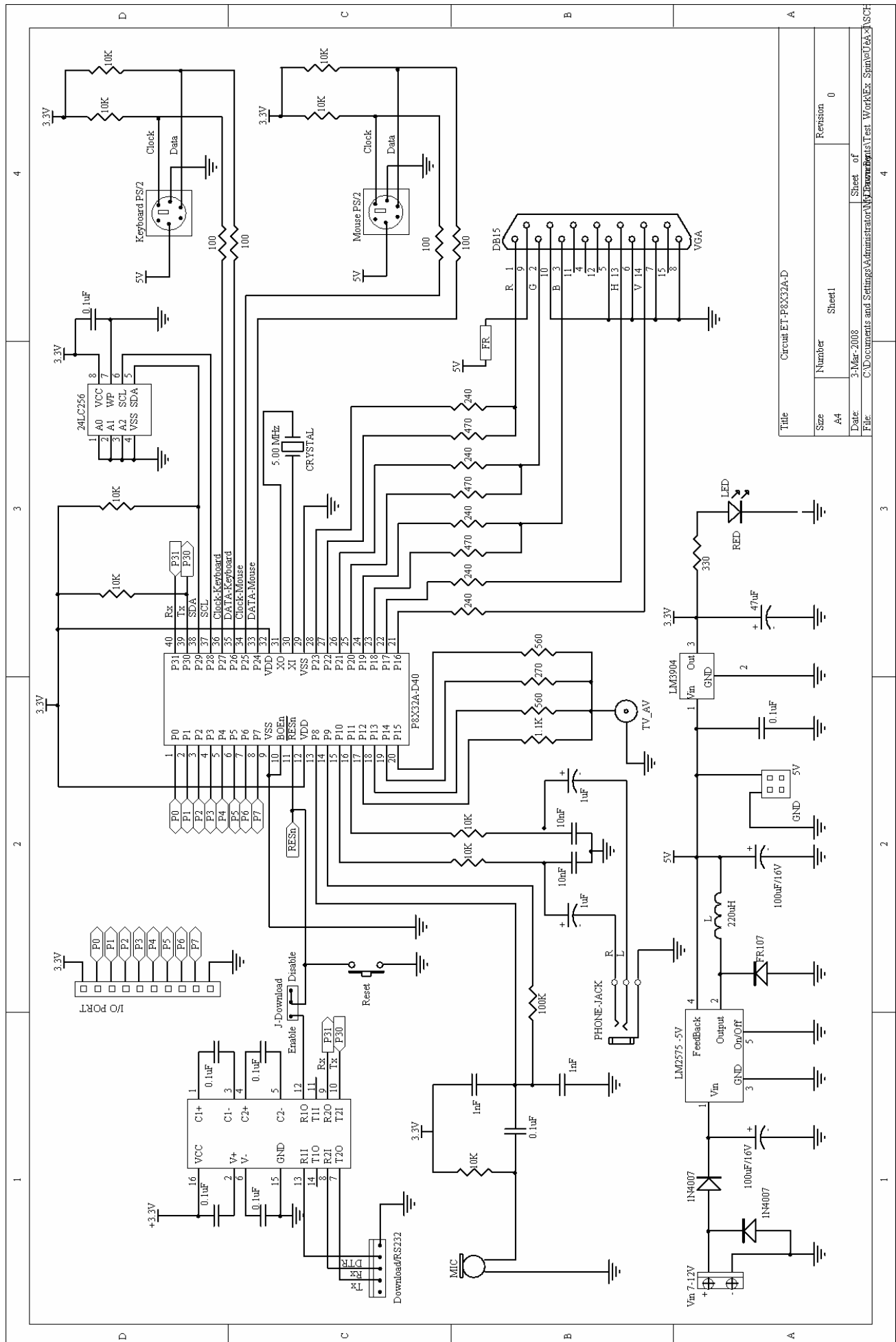
has 2 Bit; refer to example above, %%2130 = 10 01 11 00 (binary number) = 9C (Hexadecimal number).

8.6) Using Loop command such as *repeat* or conditional command such as *if*; after user has already typed command, user must type and set program that is in the next line as right-diagonally at least 1 gap because this program is configured to run follow the command *repeat* or *if* above. If user type program in the same level as command or set it as left-diagonally, Propeller considers that the program in the line is not under command *repeat* or *if*; for example,

```
PRI BlinkingLED
  Pin := 07
  DirA[Pin] := Out
  Repeat
    OutA[Pin] := High          'LED ON
    WaitCnt(40_000_000 + Cnt)  'ONE-HALF SECOND WAIT
    OutA[Pin] := Low           'LED OFF
    WaitCnt(40_000_000 + Cnt)  'ONE-HALF SECOND WAIT
```

8.7) The method to write Comment to describe program is to use sign ' or " and then follow by Comment or write it in the brace {comment} or {{comment}}

8.8) The Block PUB that is used to be main Block of program; if the written program in the Block is Open Loop and it does not run repeatedly in any loop, it must ends the program with Repeat command to protect MCU to reset program.

For more information about applications of commands of SPIN Language, see Help of Program Propeller; click **Menu Help** and then click **Propeller Manual (pdf).**

Figure of ET-BASE PX32 V1.0

ET- BASE PX32
V1.0