AX206 DFP Development Handbook

v0.0.2

Confidential

AppoTech Limited

Address : Unit 705-707, 7/F, IC Development Centre,
No. 6, Science Park West Ave.
Hong Kong Science Park,
Shatin, N.T. Hong Kong

Tel :(852) 2607 4090

Fax :(852) 2607 4096

www.appotech.com

Table of Contents

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK

Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Chapter 1 Introduction

1.1. Objective of this Handbook

This handbook describes the development process of AX206.

1.2. Style and Symbol Conventions

Throughout this document, certain style, format and font conventions are used to signal particular distinctions for the affected text. Table 1-1 lists the symbols, terms and typographic conventions used in this manual.

Table 1-1: Document Conventions

Font Convention	
Convention	**Description**
Arial Font	The standard font used for all text, figures and tables within this manual. Other fonts, as described below, are used to set off mathematical and logical expressions, or device instruction code, from descriptive text.
`Courier New Font`	Within text, this font is used for contrast with the standard text font and specifically denote the following: 1. an instruction set mnemonic or assembler code fragment. 2. the binary value of a bit, range of bits, or a register. 3. the logical state of a digital signal. Within code examples, this font is used exclusively to denote an assembly or high-level language instruction sequence.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Chapter 2 The Development Process

2.1. Pre-development Preparations

The AX206 Development Kit includes a development board (firmware included), buzzer and power supply. To start development, user should separately purchase the compiler tools, Keil C51 Development Tools and UEStudio.

2.2. Program Requirements

AX206 has 3KB memory. There is not enough RAM to load a large scale project. When developing AX206, it is necessary to break down the code into sectors. In any time, there should be less than 3KB program in the RAM. In Keil, this smaller program is called ‘Bank’. The Bank system cannot satisfy our development needs. So in our development process we have not used the ‘Bank’ system. To prevent confusion, we refer to our process as **sector**.

Each Sector has its own compilation and linking process. In view of Keil, the Sectors have no connection among them. The connections are established by tools other than Keil.

It is possible that a module in a large project cannot fit into a sector. A number of sectors are needed to run a function. To ease management, the sectors working for a module can be put into the same folder. This will form a group of sectors working for a module.

2.3. Compilation and Linking Sectors

Keil cannot handle the cross reference between sectors. In the source code, cross reference should be declared using EXTRN. When using Keil compiler to compile and link for the first time, Keil will show a warning but it can still output correct result. The cross-reference value will be 0. This warning can be masked using the linker command DW(1,2).

Using Keil to link the first time can output correct size and code table for cross-reference of each sector. With this information, it can build a cross-reference code table. In the table all variables are defined by EQU and declared PUBLIC. The file is in the compilation source file and compiled independently. Afterwards, each sector is linked together with the cross-reference code table, correct result can be produced.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

2.4.Supporting Tools: Gen and CRC

The code table for cross reference should be generated by supporting tools.

The code table includes function definitions in sectors, address of sector and other resources in Flash.

The M51 file generated by Keil linker has the code table for all PUBLIC variables and its values. Following the naming convention, one can easily find the correct name in M51 file and output to the code table.

After the first linking, the size of codes is fixed. The start address and size of each resource can be directly output to code table.

After the second linking, all codes and resource values are fixed. Gen can take the final position and generate the Flash file.

The Hex file generated by Gen will be translated to bin. This still cannot be burned directly to Flash. There are two missing CRC values. It's possible to fill in the two CRC by Gen. But it leads to a more complicated Gen process.

So we use the CRC tool to generate the values. CRC can first read header, and calculates the first CRC from the program address and size. Then it calculates the CRC of the header. After filling the two CRC in the bin, it can be burned to flash.

2.5.Supporting tool: MAKE

Keil cannot support the above complexity in compiling the code, MAKE is the better tool.

The master Makefile handles the above control flow. The child Makefile handles the compiling of sector groups and the linking henceforth. (In versions after 20090401, all child Makefiles are put into the master Makefile.)

To add a sector, add the record in child Makefiile. To add a sector group, add directory and content in the master Makefile.

2.5.1. Add a Sector Group

1. Create a sector group directory, edit child Makefile and sector process

2. In master Makefile, add the sector name to PROCESS variable.

2.5.2. Makefile Child

The child Makefile should achieve three purposes.

1. Stage 1: Compile all sectors. Use `DW(1,2)` to connect all sectors. Each sector is converted to HEX format.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK

Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

2. Stage 2: Input ..\Res.obj, and link all sectors. Change to the Hex format.
3. Clean: Clean all intermediate files

2.6.Interaction between Sectors

Interaction includes transfer between sectors and data sharing.

2.6.1. Add a Sector

1) Write the sector to implement the target actions. Compile and link by the child Makefile.
2) Register the sector in sector table in ProcTbl.asm.
3) Allocate Flash space to the sector in ResPos.ini.

2.6.2. Standard control between Sectors

Currently the system supports two types of standard control: one sector can call function in another sector, or one sector can jump to another sector.

1. PPUB – sector entrance

To allow a sector to be called, it should be marked with `PPUB`.

Example:

```
PPUB lbl      ; lbl is a tag in sector
```

2. PREF – reference to a sector

To reference to another sector, it should be first declared with `PREF`.

Example:

```
PREF lbl      ; lbl is a tag in sector
```

3. PCALL - Call a sector

To call a function in a sector, use `PCALL` but not `CALL`.

Example:

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

```
PCALL lbl     ; lbl is a tag in sector
```

4. PRET – Code return

After calling PCALL, use PRET to return.

5. PJMP – Jump to a Sector

To jump to a Sector, use PJUMP but not jmp.

Example:

```
PJMP  lbl     ; lbl is a tag in sector
```

It's allowed to jump from code A to code B, and then from code B to code C. When work is completed in code C, call PRET to return to code A directly. It is not necessary to call PJMP to code B and then PRET to code A.

6. DirectLoadProc – Direct load process

This library requires three variables:

1. R5-R7 – Flash address of Sector
2. DPTR – Start address of xData
3. R3:R4 – End address of xData

This library can achieve three purposes:

1. Load a sector, for example, to load the _start sector when loading the common event sector ProcMan. ProcMan is used for jumping between sectors. It cannot load itself. This library is needed to load the common event function.
2. Load a PIC. Functions that are called by various sectors cannot be placed in the same address. The functions may include window opening, calculation functions for flashing screen etc. These functions should be placed in different locations depending on the sectors combination. These functions can be written as a address-irrelevant sector. When it is needed, it can be loaded by DirectLoadPorc and called as local functions.
3. Provide a standard load process. Sector without a lot of free space can utilize space at ProcMan. But before the code exits, it should use DirectLoadProc to place ProcMan in the original position. Then it can use the standard transfer method. Please note that DirectLoadProc function, the function that call

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

DirectLoadProc and the neighbour codes cannot be located in the same address, otherwise there will be abnormalities.

2.6.3. Data sharing between sectors

Currently only EQU data are shared. EQU variables prefixed by '?PPUB?' can be accessed by all sectors. For example, sectors defined in ProcTbl can be prefixed by '?PPUB?'. Other sectors can call the prefixed Sectors.

2.6.4. Other operations between sectors

1. High Efficiency transfer mode

   The public area in standard control amounts to 200 words. In other special conditions 200 words is extensive. It is necessary to use high efficiency transfer mode. It is advised to use this mode only when necessary.

   In variables defined by PPUB, there are several EQU variables.

   - ?FADD?<lbl> - variable values
   - ?FNUM?<lbl> - sector name

   Sector that defines the PPUB variables can call the following variables:

   - ?PADC?<sector name> - sector start address minus 0x800 is the xData start address.
   - ?PADE?<sector name> - sector end address, minus 0x800 is the xData end address.
   - ?PADH?<sector name> - sector Flash address (High two words)
   - ?PADL?<sector name> - sector Flash address (low two words)

   Using these variables, sector can operate in Flash and jump to other sectors.

2. Sharing extensions

Currently the system only supports EQU data sharing. To support other types of data sharing, modify the Gen tools definition and rules.

2.6.5. Some important files for sectors

ProcTbl

ProcTbl is a special sector. It records the code information such as addresses and size.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

The information is used when in referencing between sectors.

ProcTbl can output the sector number. In ProTbl.asm, under ProcTbl each ProCDEF defines the sector. The order number is the sector number. Programmer does not need to memorize the sector number. Each Sector name adding the prefix '?PNUM? is the segtor number. In each sector, code tagged with PPUB adding the ?FNUM? will be the sector number.

ResPos.ini

ResPos.ini explains the position of resources in Flash. It includes sector resources.

In commands defining resources position, resources with the PROCESS command is considered a sector. However, ResPos.ini only defines the sector as a resource in Flash. A sector defined in ResPos.ini only defines the Flash address, and some markings for PPUB in sector. The sector number has not been defined. It is defined by ProcTbl.

ProcTbl can be defined as a special sector that output code, but not as a sector that can be transferred. This sector has all the code information, it can execute the normal transfer mode. If it is required, programmer can utilize this feature.

ProcMan

ProMan handles the transfer between sectors using the sector number and ProcTbl in Flash. The transfer is in standard format.

2.7.Library and Background Process

2.7.1. Library

Some codes should be written as library to be called by other codes. However, the size of these codes are too large, they can be written as sector.

If library codes are packed as sector, caller of library code may not have the correct saved content when return from calling the library code. To avoid this situation, some design rules should be enforced.

The program code can be swapped and the data should be kept. The rules are listed below:

1. Program code should be placed at lower address and data should be placed at higher address.

   This rule is applicable to transferring between different sectors. For example, when decoding photos, Sectors that read table and decode can cooperate that way.

   The library code should be kept minimal so that data will not be corrupted.

2. Data can be kept at xData for retrieval.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Data for function caller can be separated from data for library code. When data memory is not enough, one can consider placing data into xData,

2.7.2. Background Process

Larger background process can also be packed as sector. Same as library sector, one should consider the danger of data corruption.

2.7.3. Code and Data Security

Library code and background process require consideration on data and code overlapping. It is required to separate data area and code area.

The boundary line between data and code area is suggested to be at code address 0x1580, i.e. around 0xD80 xData address. This boundary is approximate. The following rules are suggested:

1. Library and background code should not exceed code address 0x1580.
2. Data code should not be placed less than 0xD80.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Chapter 3 Sector Group and Sector

DFP module functions are separated into sector groups and sectors. In the project, each sector group is a standalone folder. The folder includes source codes of standalone function. The folders that are sector group are listed below:

Folder	Function description
_Start	Initialization and interrupt
Analog	Analog clock display
Calendar	Calendar display
CommonEvent	Common event and response
DigClock	Digital clock, calendar display and clock setup
JudgeMenu	Contrast and backlight interface display and menu
Lcd	LCD initialization, window opening
PhotoDis	Photo display control
PicDecod	Photo decode
TempDetermine	Temperature detection
TimeZone	Time Zone checking
USB	USB module
UserLib	Closed source library files
LibProc	Menu library

Other folders

Folder	Function description
Res	Menu photo, DFPMate resources
Lib	Library files
Tool	Tools

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

3.1._Start

3.1.1. Common.asm

The sector is interrupt service for timer0. The timer0 is set to 10ms interrupt. Interrupt internal events include:

- keys detection,
- .ADC detection and calculation,
- volume control
- software counter (3 sets).

(1) Keys detection can process 8 IOs. The key values generation is listed in the following table:

	Short-press	Long-press	Remarks
P3.0	0x10	0x20	
P3.1	0x11	0x21	‘Up’ Key
P3.2	0x12	0x22	‘Down’ Key
P3.3	0x13	0x23	
P3.4	0x14	0x24	
P3.5	0x15	0x25	
P0.6	0x16	0x26	‘Power’ Key
P0.7			USB detection

Note the clear key values after key detection.

(2) USB detection

USB is detected at P0.7. When USB is inserted, yStableSignal is cleared (=0) and bUSB_Insert_Flag Event0[3] is set. When the USB is removed, bUSB_Desert_Flag Event0[4] is set.

(3) Battery detection

Battery is detected at P0.3 at the 10-bit ADC. It is detected every 2s. Interrupt will detect automatically 8 times and put the values in variables BatteryQ and

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

BatteryQ+1.

(4) Temperature detection

Temperature is detected at P0.2 in ADC. It is detected every 2s. Interrupt will detect automatically 8 times and put the values in variables TemperatuerQ and TemperatureQ+1.

(5) Software counter

The software counter in timer0 includes 3 sets of counters. They can be used for auto shutdown, delay in showing pictures, etc.

3.1.2. System Initialization

The sector for system initialization is Ini.asm. It includes chip's IO initialization, variables initialization, etc.

3.2.Analog

This sector handles the analog clock display and processes the events in this display. The main code is in Analog_Main.asm.

There are two modes available: normal mode and watch mode. To change configuration, change the following setting in the Config.ini:

```
# define WatchVersion 0 ;  - normal mode; 1 – watch mode
```

When it is set to watch mode, program code will switch off backlight after 5s to save power.

3.3.CommonEvent

This Sector includes common functions that are required in all status, such as power down, low power control, etc. The functions included are listed below:

Function	Remarks
BattStepCal	Detect the battery level

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

DisLowPower	Display low power and enter into Sleep mode
KickAlarmEvent	Kick start alarm event
PowerOffEvent	Shut down
LibCommonEvent	Detemine common events for SetUpDown library and Judge_Slideshow

3.4.DigClock

This sector includes clock setup, calendar display and digital clock display.

Function	**Remarks**
Display_Dig_Clock	Display digital clock
Display_Dig_Head Set_Dig_Clock	Set up and display clock
Dis_Clock_setCal_Time	Calculate the current time according to variables

3.5.JudgeMenu

Setup automatic display time, change contrast, and change backlight level utilize similar functions. These functions are included in JudgeMenu.

Function	**Remarks**
Judge_Alarm_Clock	Setup alarm clock
Judge_BackLight	Change backlight level
Judge_ShutDown	Set shut down time
Judge_Slideshow	Set slideshow display time interval

3.6.LCD

The LCD sector includes functions related to the LCD screen. The functions included are:

Function	**Remarks**
Backlight_Fun	Backlight Control

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Lcd_Contrast	Contrast Control
LcdInit	Initilize LCD
OpenWin	Open window to write data
OpenWin_Read	Open window to read data
Read80	Read a byte from the screen (8080)
LcdComm	Setting information such as screen resolution and time

Under the LCD folder, there are files for some supported LCD screens. To add support for other screens, please create new folder.

If normal LCD driver is used (initialization code will be less), user can modify codes using formats in TM125125_Lcd.

If the initialization code for LCD driver is too long, it can be put into a table. An example is available at ST7787_2.4Lcd_Table.

To create LCD library, the LCD sector links to the whole project through the LCD library. In config.ini, the line "`export LCDDIR-TM128128_Lcd`" is used to map to the supported screen. When generating the LCD library, a library file called "LCD.LIB" in the folder after 'equal' sign.

Note:

(1) When the file in LCD folder has been updated, remember to delete the old List, Obj files already in the folder before creating the LCD library.

3.7.PhotoDis

This sector controls the display workflow of photos. The main function is in PhotoDis_Main.asm.

3.8.PicDecod

This sector handles photo decode and change photo effect.

To change photo using PicDecode fuctions, see the following example:

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

```
mov rd,#4
PCALL  PicDecode ;Save the flash address of the photo at R5 R6 R7 (HML)
```

In a DFP project, it is necessary to update part of the photo. For example after the battery logo is displayed, the photo needs to be restored. Using PicDecode can complete the cover function. An example is shown below:

```
mov    r4,#1
PCALL  PicDecode ; save the photo flash address to R5 R6 R7 (HML)


CSEG   AT  0x1600 ; table for cover function
Cover_Num_Tbl:
DB     40  ; There are a total of 40 MCUs need to be covered
DB     0,0
…..
DB     11,15
DB     12,15
DB     13,15
DB     14,15
DB     15,15
```

When the photo is covered, data from Cover_Num_Tbl is used to determine what to cover. For a 128x128 screen, it is divided into a number of 8x8 MCU modules. `DB 1,2` represents the first row in the second column MCU module.

3.9.TimeZone

This sector includes time zone display and event handling. Descriptions on how to add time zone is included in timeZone.asm.

3.10. LibProc (Menu)

This sector controls the menu. The standard version provides menu in high contrast (white

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

words on black background). To show menu in other color, update the Menu.asm file.

The menu is defined in tables in ActSwtsh.asm. A page of complete menu is shown below:

```
(Menu head info)                                    (Buffer for the current menu)
   MENU       MainMenuADR,     0,0,127,15,          MenuItemBuffer
              (Menu head photo)  (Menu head display window position)
                                 (Column_start, Row_start, Column_end, Row_end)
(Menu display range)
   MENUCLIENT MainMenuADR,     0,16,128,112
              (Menu head photo)        (Menu item display position)
                   (Column_start, Row_start, Column_end+1, Row_end+1)
(Menu item)
   MENUITEM   USBConnectADR, USBConnectHADR_idx,       0,0,127,15
   MENUITEM   SlideShowADR,SlideShowHADR_idx,          0,16,127,31
   MENUITEM   ClockADR,     ClockHADR_idx,             0,32,127,47
   MENUITEM   AutoOffADR,   AutoOffADR_idx,            0,48,127,63
   MENUITEM   BackLightADR,Judge_BackLight_idx,        0,64,127,79
   MENUITEM   LcdContrastADR,Judge_Lcd_Contrast_idx,   0,80,127,95
   MENUITEM   DeleteADR,    DeleteHADR_idx,             0,96,127,111
   MENUITEM   ExitADR,Photo_Disp_After_Menu_Exit_idx, 0,112,127,127
              (Menu head photo)  (Menu item call)  (Menu item display position)
                                 (Column_start, Row_start, Column_end, Row_end)
   MENUEND    MainMenuADR
```

This menu has two situations. The first one is to enter the child menu, another is to exit the current menu, and enter the chosen program. As shown in the above menu, USBConnectHADR_idx represents the process to enter the child menu item. The child menu item number is USBConnectHADR. Similarily, Judge_BackLight_idx represents the

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

process to enter the backlight adjustment menu. The item is Judge_BackLight. This item should be stated in ActSwtch.asm using LIST_PROC Judge_BackLight.

3.11. System Configuration - Config.ini

Update the config.ini file can select the screen and main functions in DFP.

3.11.1. Screen setting

- `export LCDDIR = TM128128_Lcd`

  This line defines the screen chosen is `TM128128_Lcd`. When generating LCD library, the tool will select the folder named 'TM128128_Lcd' to compile.

- `export MENUDIR = MENU_128128`

  This line defines the resources files in folder 'MENU_128128' is used.

- `#define WL_128128`

  This line defines the resolution for system startup and photos.

The above three lines should be updated when the screen is changed.

3.11.2. Clock version selection

```
#define          WatchVersion   0
0 - normal mode; 1 - watch mode
```

To use hardware as watch, set the WatchVersion to '1'. The generated program will switch off backlight after displaying digital and analog clock to enforce power control.

3.11.3. Temperature function setting

```
#define    DEBUG_DETECT_TEMPERATURE          0
;0 - No temperature display
;1 - Temperature display
```

If temperature display is chosen, additional hardware is needed to support this function.

3.11.4. Time Zone setting

```
#define          TIMEZONE_CONFIG          1
```


Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

```
;0 – Do not support TimeZone function
;1 – Support Time Zone function
```

The default setting supports Time Zone function. If Time Zone is not needed, use another UserLib.lib.

3.12. Menu Configuration – Menu_Config.ini

Version DFPV1.0_20090401 and later includes menu support for different resolutions. After resolution is defined in the project, if the corresponding menu resolution requires updates, modification to this file is needed. The settings are described in Menu_Config.ini.

The following variables need to be updated:

```
;Digital resources for the current setting
;(Available options: 00_1624、00_816、00_1218、00_3042)
;Digital clock settings
#define          SET_CLOCK_NUM_ADR          00_1218
#define          SET_CLOCK_NUM_COL          12
#define          SET_CLOCK_NUM_ROW          18

;Menu settings
#define          SET_NUM_ADR                00_1218
#define          SET_NUM_COL                12
#define          SET_NUM_ROW                18

;Display settings for minute and seconds in digital clock
#define          DIG_NUM_ADR                00_3042
#define          CLOCK_PIC_ROW               42
#define          CLOCK_PIC_COL               30
;Display settings for year, month and date in digital clock
#define          DIG_YEAR_NUM_ADR           00_1624
#define          DIG_YEAR_NUM_COL           16
#define          DIG_YEAR_NUM_ROW           24
```


Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Currently the system now has the following digital resources: 8x16, 12x18, 16x24,

3.13. Adding Sector

New Sector group and Sector can be added into the project.

Step 1:

Add a folder under project root and register the Sector in the makefile in root.

Example:

```
PROCESSES := _Start PicDecod LibProc Analog PhotoDis USB DigClock
Judge_Menu CommonEvent TimeZone
```

Step 2:

Add the new Sector in ResPos.ini in root.

Step 3:

Register the new Sector in the ProcTbl.asm in _Start folder.

3.14. Resources

DFP project has resource files such as menu photo, start-up screen, DFPMate iso files. They are grouped in the following folders according to type:

Folder / File name	Remarks	Resource Type
Analog_Pin	Analog clock resources	16-bit color image file
Analog_week	Analog clock display for week	Mono-color image file
Battery	Battery icon	24-bit color image file
Flow_clock	Flowing clock resources	
Menu	Menu image	Mono-color image file
Num_8x16	Digital image 8x16	
Num_16x16	Digital image 16x16	
Week	Image display for week	
Picture	Start-up screen, analog clock background file	JPG
DFPMate.ISO	DFPMate software	Iso file

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

To add resources:

1. Add the resources in the corresponding folder
2. Set the resource address in file Res.asm

3.15. Add-on Sectors

Latest functions and sectors are placed in the following folders:

Folder	Function
Audio	Record and play audio
FM	FM radio
Game	Game function – push box
PacMan	Game
Pintu	Game – Puzzle
Snake	Game – Snake
Music	Play wav music file

To play music:

1. Add a wav folder for sector group. The folder name is ‘Music’.
2. Call the music file by PCALL Play_Music.
3. The system supports 8K8bit wav music file. Replace the Music.wav under Res folder with the 8K8bit wav file.
4. Playing music requires common event. Add the record in Music_Main.asm. The following events have been added: exit playing music when key pressed, battery detection, low power shut-down.
5. Use the same circuit as the development board to play music.

3.16. Planning and Distribution of RAM

AX206 has 3K Ram for code and data. They are distributed as following:

Code space:

Range	Usage
0x1000-0x127F	Interrupt

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

0x127F-0x132F	Window opening code
0x1330-0x1923	Other program code
0x1924-0x19FF	System code

Data space:

Description	**Range**	**Usage**
Data	0x00-0x0F	2 sets R0-R7
Direct or indirect accessed	0x10~0x21 0x22.0~0x22.1 0x22.2~0x22.7 0x23~0x74 0x75~0x7F	Global variables Global bit variables Global variables
Data	0x80-0xC3	
Indirectly accessed	0xC4-0xFF	Global variables

Description	**Range**	**Usage**
Low address xData	0x00-0x8F	USB Buffer / Partial variable
Indirectly accessed	0x90-0xFF	Global variables
High address xData Indirectly accessed		Same as code used space. Use cautiously.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Chapter 4 IDE Usage Guide

The current project is built using UEStudio. The small frame digital photo frame project will use the following functions under the 'Advanced' Menu in UEStudio:

❶'Burn': Copy files in bin folder to DFP

❷ 'Create': Link all the codes and resources and copy a binary to bin folder

❸ 'Clear': Clear all files produced by the project, except UserLib.lib

❹ 'Create Resource': Create files from the folder Res. When the resources files are updated, use this function to regenerate the resource files before 'Create'

❺ 'Create UserLib': Generate binary from files in UserLib folder. UserLib.lib is a packed file.

❻ 'Setup LCD Library': Regenerate LCD library according to settings in Config.ini. When LCD is updated, use this function before 'Setup' the project.

❼ 'Create Library': When the Lib files are updated, use this function to regenerate DFP.lib library files from files in Lib folder before 'Create'.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Chapter 5 Common Functions Customization

5.1.LCD Display Resolution

The current standard version supports display resolution in 96 x 64, 128x128, 120x160, 176 x220, 240x320 and the corresponding rotated display. Note the following when modifying the project:

1. Use the following values defined in LcdComm.asm in the LCD folder to change the screen resolution.

LCD_WIDTH	EQU	320
LCD_HEIGHT	EQU	240

2. Update definition in Config.ini. Example:

```
export MENUDIR = MENU_320240

#define WL_320240
```

If 2.4” screen is used, update the following to modify the hand resources used by analog clock.

```
#export ANALOGDIR = Analog_Pin_2.4

export ANALOGDIR = Analog_Pin
```

3. Menu photo can be put into menu resource folder (Firmware206\Res\menu\MENU_320240). Or copy one from the 128128 folder.
4. Picture files with the correct resolution can be put into picture resources folder (Firmware206\Res\picture) following the naming convention.
5. 2.4” inch does not support reading data. In this project an Analog clock sector is provided. (Other resolution screen can also make use of this sector.) When creating the 2.4” project, use Analog_2.4 folder as the Analog sector.
6. Modify Menu_Config.ini as needed.

5.2.Background and Foreground Color in DPF

Update the Global.inc file under project folder.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

```
#define BACK_COLOR_VALUE 0xFFFF

#define FORE_COLOR_VALUE 0x0000
```

5.3. Add in Digital Resources

If the four existing digital resource types are not enough, new digital resource types can be added in the following method:

1. In Res folder new resolution folder, pictures and makefile.
2. Register new folder in makefile in Res folder.
3. Register new digital resources in ResPos.ini.

5.4. Compression rate and Max Photos limit

In AX206, photo compression rate and number of photos can be set. The following describes methods to set limits.

To set limit, make changes at the Head.asm in _Start. In Head.asm, there are 3 definitions:

```
;2008-09-26

DB 0,0 ; Max photo supported is 0, i.e. no limit. First digit is

       ; low byte. Second digit is high byte.

DB 1   ; Photo format in 4:1:1 or 1:1:1

DB 92  ; Compression quality: 1 – 100
```

The first line defines the maximum number of photos supported (max 499 photos). It uses two byte to set control. The first one is low byte, the second is higher byte. Set 0 is not limited.

The second line controls the photo. This byte can be set to 1 or 4, i.e. 1:1:1 compression or 4:1:1 compression. This setting controls the square size when photo reloads.

The third line controls the compression quality at 1 to 99. When max photo is not set, it will compress according to this value.

When the max photo is set to be too large that the Flash memory cannot store all photos, some photos cannot be saved and it cannot store as much photos as stated.

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK
Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com

Appendix I Revision History

Date	Version	Revised items	Revised by
2010-03-15	0.0.1	Draft of English version	Erica Cheong
2010-05-05	0.0.2	Added Section "Pre-development Preparations"	Erica Cheong

Address: Unit 705-707, 7/F, IC Development Centre, No.6, Science Park West Ave., Hong Kong Science Park, Shatin, N.T., HK

Tel: (852) 2607 4090 Fax: (852) 2607 4096 www.appotech.com