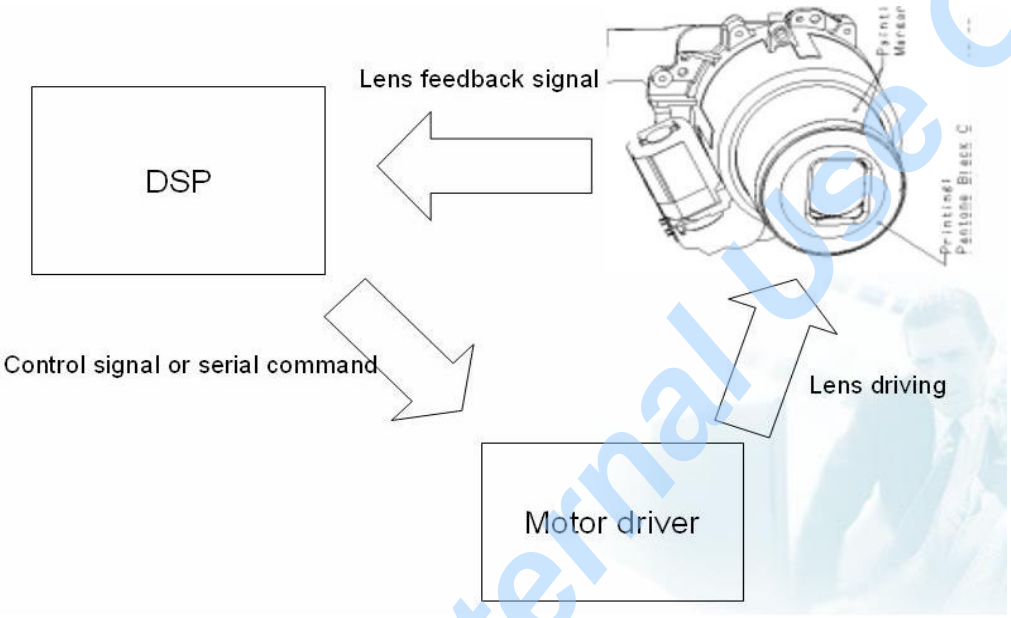
Introduction to Lens Control

Content

Revision History

Revision	Date	Author	Changes
V1.0	2011/03/28	Chris Chung	Draft
V1.1	2011/06/10	Chris Chung	Add 5.3 Miss Step Test.
V1.2	2011/07/13	Chris Chung	Refine lens driver for DRV_NT96220.

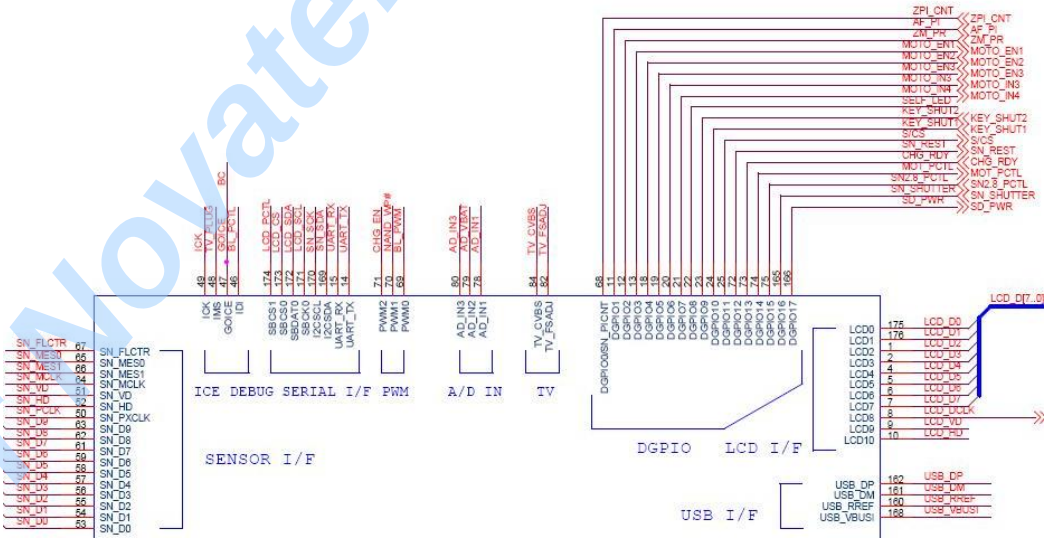
1. Introduction to the components of zoom lens system



1.1. DSP

- **NT96211**

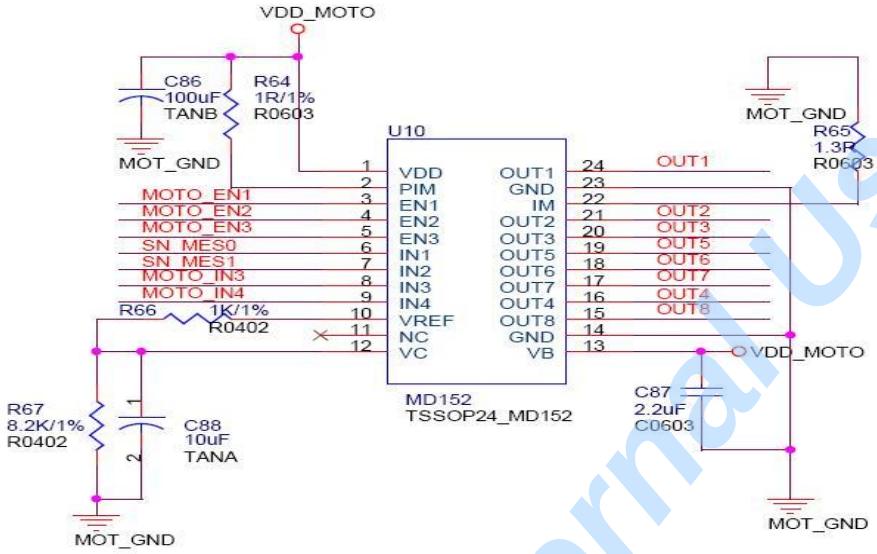
DSP輸出GPIO, SIF, or I2C的指令去控制Motor driver。



1.2. Motor Driver

- **MD152, WT7026**

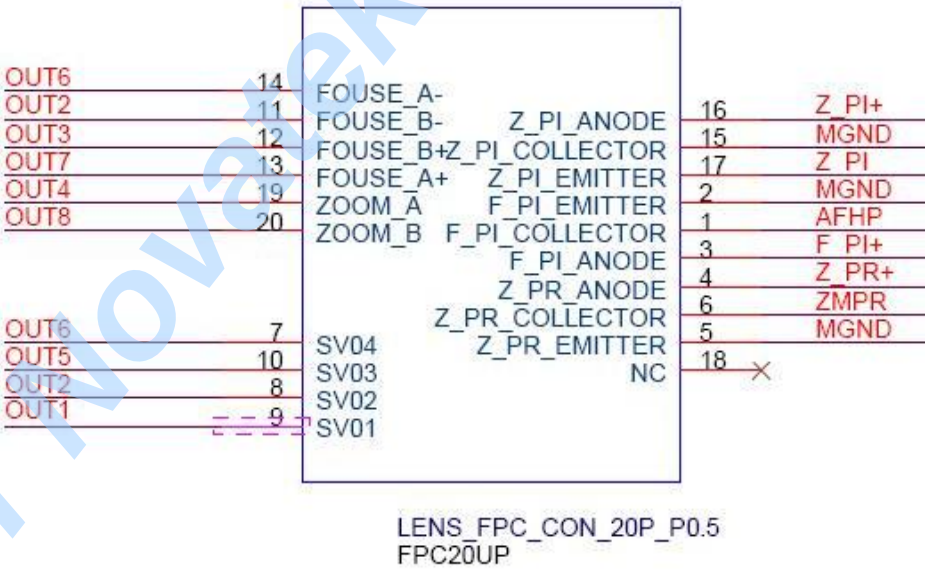
Motor driver收到DSP指令後，輸出較大電流的訊號去驅動Lens。



1.3. Lens

- **ST8002, ST8003**

Lens 回饋 Zoom PI, Zoom PR, Focus PI..等訊號，通知 DSP 目前所在的位置



2. Introduction to the key parts of Lens

2.1. Zoom

- **DC Motor**

DSC Zoom 常使用DC Motor來控制，因爲它的成本比Stepping Motor較低。

- **Stepping Motor**

DV Zoom 常使用Stepping Motor來控制，因爲它容易調整速度及精準定位。

2.2. Focus

- **DC Motor**

DSC Focus 使用DC Motor來控制較少。爲了達到較低成本的目的，聯合光電有出使用DC Motor來做Focus的2x光學變焦鏡頭ST8005 (T2085)。它只需要一顆DC Motor就可以分別去控制Zoom及Focus，因此成本上非常低。它的缺點是不易精準定位及對焦時間慢，加上小步移動需要對DC Motor做瞬間的Fwd, Brake, Off動作，馬達容易因爲晃動產生雜波造成計數失步。

- **Stepping Motor**

DV/DSC Focus 常使用Stepping Motor來控制，因爲它容易調整速度及精準定位。

- **Voice Coil Motor (VCM)**

手機 Focus 常使用 VCM 來控制，它具有尺寸小及成本低的優勢。它的缺點是如果要固定在某個對焦位置，需要保持通電(耗電)。

2.3. Shutter & Iris

- **VCM**

DV/DSC Shutter及Iris (光圈) 常使用VCM來控制，它的控制上簡單。

- **Stepping Motor**

亞光有出使用 Stepping Motor 來控制 Shutter & Iris 的 3x 光學變焦鏡頭 ST8002，它將孔徑全開, 小開, 全關當作三個狀態，透過類似 power on sequence 的方式來定位。

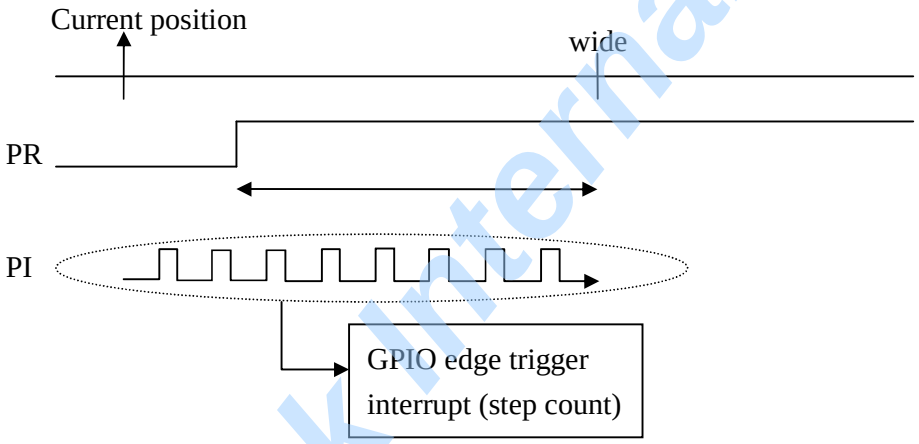
3. Lens driving fundamental

3.1. DC Motor

	Fwd (CW)	Bwd (CCW)	Brake	Off
Z+	H	L	H	L
Z-	L	H	H	L

- **DC Motor**

DC Motor使用一個Channel兩個IO (Z+, Z-)訊號，控制馬達的Fwd, Bwd, Brake,及Off動作，如上表所示。



- **PR and PI signals**

Lens採用DC Motor會搭配PR及PI兩個訊號，回饋給DSP做同步定位使用。PR signal由Low轉High的反轉點，表示DC Motor的Home點。PI signal是馬達每走一步，就會發出一個Pulse來給DSP計數使用。在應用上，如果DSC想把Zoom推到Wide端，若PR signal初使位置爲Low，我們會Fwd馬達到PR signal反轉，此時會把內部計數Count歸零。接著會繼續Fwd，每次收到一個PI signal就會對Count加一，直到Count到達Wide所在的步數後，Brake停止後Off。

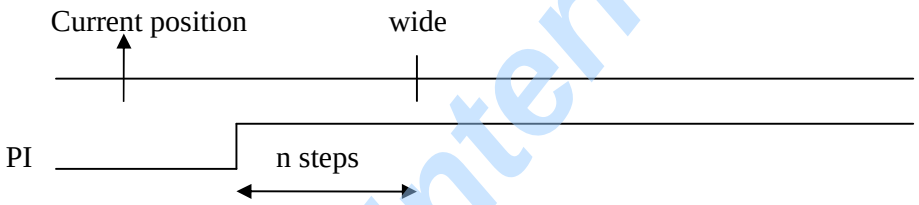
3.2. Stepping Motor

	1	2	3	4	5	6	7	8	Off
FA+	H	H	H	L	L	L	L	L	L
FA-	L	L	L	L	H	H	H	L	L
FB+	L	L	H	H	H	L	L	L	L
FB-	H	L	L	L	L	L	H	H	L

CCW ←→ CW

- **Stepping Motor**

Stepping Motor使用兩個Channel四個IO (FA+, FA-, FB+, FB-)訊號，控制馬達的Fwd, Bwd, 及Off動作。上表表示採用1-2 phase的Stepping Motor共有八個相位。兩個Channel輸出控制訊號，由左往右相位1依序往相位8變化 (1->2->3…->8)，可控制Stepping Motor Fwd (CW) 7步。由右往左相位8依序往相位1變化 (8->7->6…->1)，則表示Bwd (CCW) 7步。



- **PI signal**

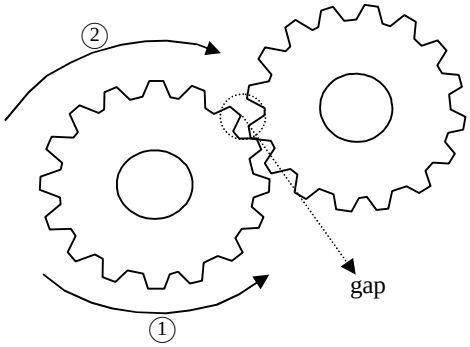
Lens採用Stepping Motor會搭配一個PI訊號，回饋給DSP做同步定位使用。在應用上，如果DSC想把Focus推到Far端，若PI signal初使位置爲Low，我們會Fwd馬達到PI signal反轉，此時會把內部計數Count歸零。接著會繼續Fwd，每次變化一個1-2 phase，就會對Count加一，直到Count到達Far所在的步數後Off。

3.3. VCM

	Open	Close	Off
S+	H	L	L
S-	L	H	L

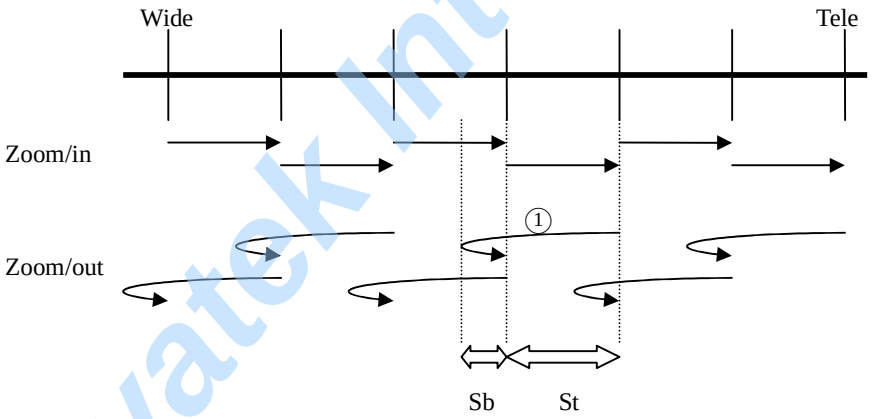
VCM是使用一個Channel兩個IO (S+, S-)訊號，來控制馬達的Open, Close, 及Off動作，如上表所示。在應用上，如果DSC想打開Shutter，則會輸出Open訊號打開Shutter。經過一段短暫的時間後(ex: 30ms)，需將輸出轉爲Off，以免Shutter線圈燒毀。

3.4. BackLash



- **Backlash Problem**

Backlash (背隙, BL)是兩個齒輪之間的間隙，如上圖所示的gap。不論是DC Motor或Stepping Motor，只要使用到齒輪，都會存在這個間隙。若是左下的齒輪往逆時針方向轉(方向1)，由於兩個齒輪間是處於頂住的狀態，因此左下的齒輪走N步，就可以帶動右上的齒輪走N步。若是左下的齒輪往順時針方向轉(方向2)，則左下的齒輪需先填滿gap步數（假設是M步），因此左下的齒輪要讓右上的齒輪走N步，共需走N+M步。

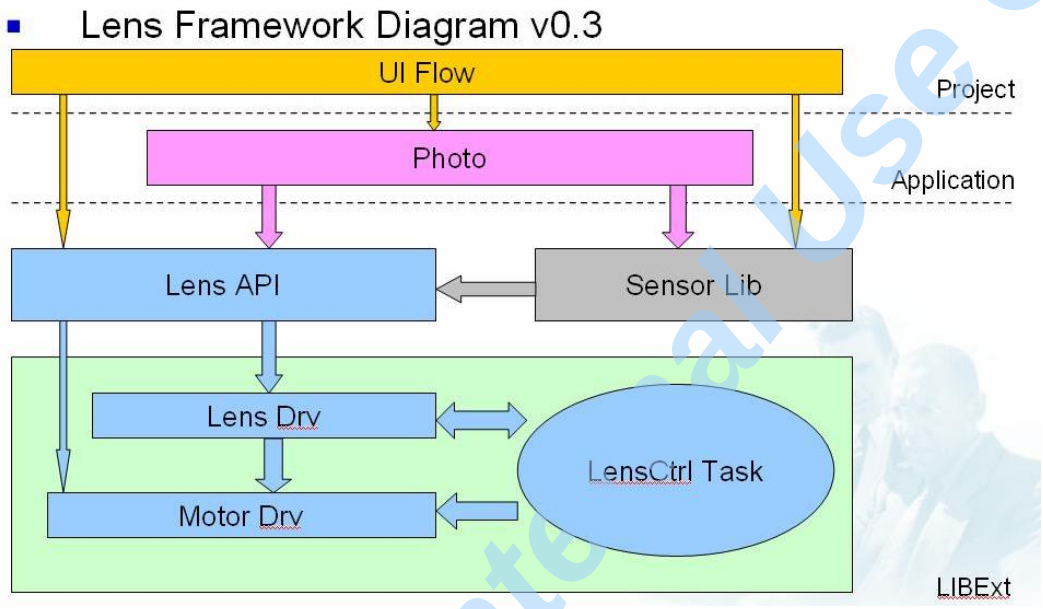


- **Backlash Elimination**

爲了消除Backlash問題，在應用上會統一由一個方向去補BL步數，若Zoom in的方向無需補BL，則Zoom out方向需要補BL。如上圖所示，St表示Zoom out想移動的步數，Sb表示補BL的步數，Sb只要大於等於齒輪BL的步數即可。

4. Lens framework

4.1. Lens Framework Diagram



- Lens API
  - This part provides public API of lens framework.
  - One static object (LENS_MODULE_OBJ) records lens module state and function pointers to the tables of lens and motor drivers.
  - Lens module initialization
  - Lens power on/off
  - Lens zoom/focus initialization and retraction
  - General zoom operations
  - General focus operations
  - General aperture operations
  - General shutter operations
- Lens Driver
  - This part provides static and specific functions of lens driver.
  - One static function table of lens driver is declared here.
  - Four static objects (ZOOM_Struct, FOCUS_Struct, IRIS_Struct and SHUTTER_Struct) are used to records current information of lens.
  - Specific lens information is defined in private header file.
  - Interrupt service routine (ISR) is used to handle the PI signal of DC motor.

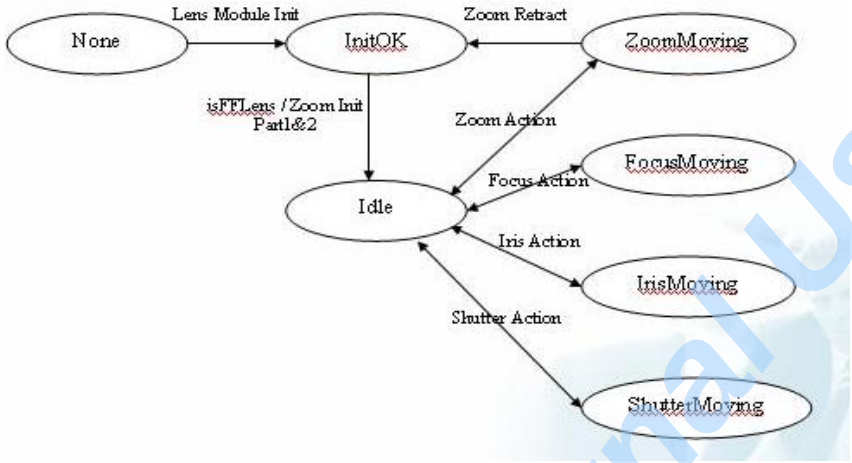
- Motor Driver
  - This part provides static and specific functions of motor driver.
  - One static function table of motor driver is declared here.
  - One static object (MOTOR_PVT_OBJ) is used to records current information of motor driver.
  - GPIO/SIF/I2C is used to configure and control motor driver.
- Lens Control Task
  - This part provides high priority task to handle critical lens control.

4.2. File Tree for DRV_NT96220

@\Include\Subsystem\Lens\
    Lens.h
    LensCtrlTsk.h
@\LibExt\LibExt_Src\Subsystem\Lens\
    Makefile
    LensInt.h
    LensCmd.c/.h
    LensAPI.c/.h
    LensCtrlTsk.c
    \LensDrv\LensDrv_ID1.c/.h (ex: ID1=ST8003)
    \LensMotor\LensMotor_ID2.c/.h (ex: ID2=WT7026)
@\Project\DemoKit\SrcCode\Calibration\Lens\
    CalLens.c/.h
    CalLens_ID1.c (ex: ID1=ST8003)
@\Project\DemoKit\SrcCode\UIApp\Lens\
    UILensObj.c/.h
    UILensObj_ID1.c (ex: ID1=ST8003)

4.3. Introduction to the API of LensAPI.c

- **Lens State Machine Diagram v0.3**



- **LensAPI.c**
  - UINT32 **Lens_Module_GetState**(void)
    This function is used to get the state of lens module.
  - static void **Lens_Module_SetState**(LENS_MODULE_STATE state)
    This function is used to set the state of lens module.
  - void **Lens_Module_Init**(PLENS_DEVICE_OBJ pLensDev, PLENS_TAB pLensTab, PMOTOR_TAB pMotorTab)
    This function is used by SysInit_InitLens( ) to initialize I/O parameters and task for lens module.
  - void **Lens_OnOff**(MOTOR_POWER ON_OFF, UINT32 param1)
    This function is used to power on/off lens.
  - INT32 **Lens_Init**(LENS_INIT_CATEGORY category)
    This function is used to initialize lens to three categories. The first category, LENS_INIT_ZOOM_PART1, can be executed in CopyRest_RunPartOne( ) for boot speed. The LENS_INIT_ZOOM_PART2 and LENS_INIT_FOCUS are usually executed in DevCtrl_ModePhoto( ) or DevCtrl_ModeMovie( ).
  - INT32 **Lens_Retract**(UINT32 param)
    This function is used to retract zoom and focus to garage.
  - INT32 **Lens_Zoom_Goto**(UINT8 section)
    This function is used to move zoom to any section.
  - INT32 **Lens_Zoom_In**(void)
    This function is used by Zoom In key press to move zoom forward to TELE

side until Zoom In key is released.

- INT32 **Lens_Zoom_Out**(void)
  This function is used by Zoom Out key press to move zoom backward to WIDE side until Zoom Out key is released.
- INT32 **Lens_Zoom_Stop**(void)
  This function is used to stop zoom move immediately when Zoom In/Out key is released.
- void **Lens_Zoom_EnableIsr**(BOOL enable)
  This function is used to enable/disable zoom PI interrupt.
- INT32 **Lens_Focus_Goto**(INT16 position)
  This function is used to move focus to any position.
- INT32 **Lens_Focus_DoAction**(FOCUS_CATEGORY category, INT32 position)
  This function is an enhanced version of Lens_Focus_Goto and is used to move focus with three categories: FOCUS_PREEXC, FOCUS_RUN, and FOCUS_POSTEXC.
- INT32 **Lens_Focus_GotoHome**(void)
  This function is used to search and move focus to home position.
- INT32 **Lens_Aperture_Move**(unsigned short NewPosition)
  This function is used to move Aperture(IRIS) to big, small or other position.
- INT32 **Lens_Shutter_Move**(unsigned char open)
  This function is used to open/close shutter.

4.4. Introduction to the API of LensCtrlTsk.c

- **LensCtrlTsk.c**
  - ER **LensCtrl_Open**(PLENSCTRL_APPOBJ pLensCtrlObj)
    This function is used to open Lens Control task.
  - ER **LensCtrl_Close**(void)
    This function is used to close Lens Control task.
  - PLENSCTRL_APPOBJ **LensCtrl_GetObject**(void)
    This function is used to get the pointer of LENSCTRL_APPOBJ that records the information for Lens Control task.
  - void **LensCtrl_GetParam**(PLENSCTRL_PARAMOBJ pLensCtrlParamObj)
    This function is used to get/copy all the data of LENSCTRL_PARAMOBJ that records the information for lens and motor driver settings.
  - ER **LensCtrl_SetParam**(PLENSCTRL_PARAMOBJ pLensCtrlParamObj)
    This function is used to set/store all the data of LENSCTRL_PARAMOBJ.
  - ER **LensCtrl_WaitCmdFinish**(UINT32 TimeOut)
    This function is used to wait Lens Control task to FLGLENS_IDLE state.
  - ER **LensCtrl_WaitCmdTypeFinish**(UINT32 TimeOut,UINT32 cmd_type)
    This function is used to wait Lens Control task to cmd_type state.

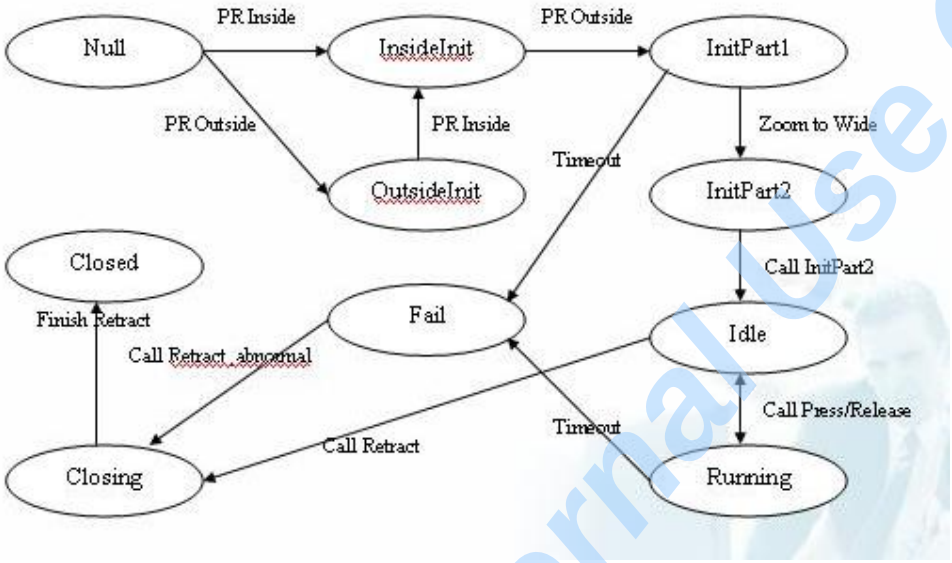
4.5. Add new lens driver

- Lens Driver – ST8003
  1. ST8003 與 ST8002 是很相似的 3x lens，可以先從 LensDrv_ST8002.c/.h 複製成 LensDrv_ST8003.c/.h 後開始修改。
  2. 依照 Lens 的規格，先改寫 LensDrv_ST8003.h 裡的 definition，Ex: ZOOM_POS_WIDE, ZOOM_POS_TELE, ZOOM_SECTION_TELE..等。接著再改寫 LensDrv_ST8003.c 裡的其他定義，Ex: Zoom Section Steps, Fno, FL, or Focus Phase..等。
  3. 檢查每個基本控制 function 的內容，Ex: Zoom Fwd/Bwd/Brake/Off, Focus Fwd/Bwd/Off, Shutter On/Off, and Iris Big/Small..等，是否符合該馬達種類控制方式。例如，ST8002 的 Shutter 是 Stepping Motor，ST8003 的 Shutter 是 VCM，因此需要修改 ST8003 Shutter 的控制程序。
- Motor Driver – WT7026
  1. 雖然 WT7026 與 MD152 的控制方式差別很大，但是直接拿 LensMotor_MD152.c./h 來複製一份 LensMotor_WT7026.c/.h，這樣開始改比較快。
  2. 依照實際 Application Circuit 的內容，改寫 motorWT7026_init( )，將 IO Mapping 的變數一一填對應。
  3. 檢查一些基本控制 function 的內容，Ex: Zoom Fwd/Bwd/Brake/Off, Focus Fwd/Bwd/Off, Shutter On/Off, and Iris Big/Small..等，依照該 Motor driver 的控制方式，做對應的修改。例如:，MD152 的 Zoom 是透過 GPIO 設定做控制，而 WT7026 是透過 SIF Command 來控制。
- LensCmd.c
  1. 若新的 Lens driver 及 Motor driver 已經加好，接著可以到 LensCmd.c 填好實際的 IO 設定，然後在串口下 Command 做基本測試。
  2. 一般常用的測試 Command 如下:

     lens init --> lens 初始化，一定要先執行這個才能執行其他 cmd
     lens info --> 查看一些 IO 資訊
     lens zfw --> lens zoom forward
     lens zbw --> lens zoom backward
     lens ffw --> lens focus forward
     lens fbw --> lens focus backward
     lens ab --> lens aperture big
     lens as --> lens aperture small
     lens so --> lens shutter open
     lens sc --> lens shutter close

4.6. Zoom initialization

- **Zoom State Machine Diagram v0.2**



- **Zoom initialization flow**

由於Zoom的初始化有較多狀態上的轉換，我們以State Machine的方式來表現，如上圖所示。Zoom的初使化流程如下:

1. 當DSC應用層執行Zoom Init Part1，會呼叫lensXX_zoom_initPart1( )進而呼叫lensXX_zoom_changeState( )來啓動State Machine的狀態轉換。
2. lensXX_zoom_changeState( )會檢查Zoom PR訊號是在Inside(Low) or Outside(High)，此時將Zoom的狀態由Null轉換成InsideInit or OutsideInit。
3. 如果是OutsideInit狀態，則會執行Zoom Bwd到Inside後，再將狀態轉換成InsideInit。
4. 如果是InsideInit狀態，則會執行Zoom Fwd到Outside。當Zoom PR訊號由Low轉換成High時，將Zoom Count歸零，接著會將Zoom的轉態轉換成InitPart1。此時Zoom Fwd仍然是保持前進，Zoom Count持續收到PI pulse做加一。當它接近Wide端時提前(Over-run 步數)煞車停止，最後將Zoom的狀態轉換成InitPart2。
5. DSC應用層接著執行Zoom Init Part2，如果Zoom的狀態已經跑到InitPart2，則將它在轉換成Idle狀態。如果Zoom的狀態一直停在其他狀態Timeout以後，則將它轉換成Fail狀態。
6. 如果Zoom的狀態成功進入Idle後，DSC應用層即可對它操作Zoom In/Out/Stop/Goto/Retract..等動作。

5. Lens Calibration

5.1. Focus Adjust

- **Far Focus Adjust**

Far Focus Adjust 建議是採用模擬無窮遠的治具，將 Lens 各 Zoom section 都校正出無窮遠最清晰的步數，保存到 DSC 的內存。若是沒有模擬無窮遠的治具，可以採用校正 2 or 3m 的方式，然後依據 Lens 規格補償到無窮遠的步數。

- **Near Focus Adjust**

Near Focus Adjust 是採用校正 0.5m 的方式，將 Lens 各 Zoom section 都校正出 0.5m 最清晰的步數，保存到 DSC 的內存。

5.2. Backlash Adjust

- **Backlash Adjust**

一般 Lens 規格都會提供 Zoom 及 Focus 的 Backlash 步數。若是 Lens 沒有提供此規格時，需要採用 Backlash 校正的方式取得。Backlash 校正的方式如下:

1. Zoom or Focus 馬達縮回到 Inside 位置。
2. Fwd Zoom or Focus 馬達，當經過 Home 點，將 Count1 歸零。
3. 繼續 Fwd 並且 Count1++ 直到 Count1 = N 煞車停止。(Ex: N = 200)
4. 將 Count2 歸零。
5. Bwd Zoom or Focus 馬達並且 Count2++。當經過 Home 點後，即不再對 Count2++，然後煞車停止。
6. Backlash 等於 (Count2 – Count1)。

5.3. Miss Step Test

- **Miss Step Test**

爲了確認 Lens 馬達的品質，我們會在工程模式增加馬達的失步測試程序，確認是否會有失步現象。測試的方法如下:

1. 馬達大步數 Fwd 到 Outside位置 (PI = 1)。
2. 馬達小步數(N) Bwd 直到 PI 訊號反轉 (PI = 0)，將 Count 歸 0。
3. 馬達移動到Outside位置，然後做任意且多次的來回移動。
4. 馬達再次小步數(N) Bwd 直到 PI 訊號反轉 (PI = 0)，檢查 Count 如果在正負(N)步左右，表示沒有失步。