



NT96660

ADAS Application Note

Table of Content

Table of Content	2
1 Revision History	3
2 目的	4
3 需求與測試環境	5
4 彈性設定與調整	6
4.1 ADAS 鏡頭校正	8
4.2 LDWS 跨越車道靈敏度調整	9
4.3 LDWS Debounce Time 調整	9
4.4 FCWS Debounce Time 調整	10
4.5 FCWS 警示距離調整	10
5 測試與解析	11
5.1 使用標準影片進行測試	11
5.2 確認各功能是否正常開啓	13
5.3 確認自動校正機制正常運作	13
5.4 確認系統效能	15
5.5 確認 LDWS 效果	18
5.6 確認 FCWS 效果	18

1 Revision History

Revision	Date	Author	Changes
1.0	2016/05/05	Pin Su	First formal version

2 目的

- ADAS Turn-key package 包含 LDWS(車道偏移)、FCWS(前車防碰撞)、AutoVP(自動校正)。並輔以靜止偵測與 Stop and Go (停車再開)等應用。
- 說明 ADAS Turn-key Codebase 可調整彈性
- 協助客戶做初步狀況排除。

3 需求與測試環境

- ADAS 須搭配 GPS，於時速 50KM/hr (此為建議值)以上才警示。若未搭載 GPS 系統則在非標準道路例如市區道路，車輛眾多、塞車等狀況會出現警示不穩或錯誤的現象。

Note: ADAS LIB 是針對影像內容作分析，然後將分析的結果回報給上層，上層總和這些結果再輔以額外訊息 (Ex: GPS 車速等) 來決定警示的行為。因此時速未到達建議值時會由上層將 LIB 傳回的警報擋掉，直到時速到達建議值後才正常警示。

- 氣候不佳例如強烈逆光、傾盆大雨；此狀況下，ADAS 偵測率會降低，或出現錯誤警示的狀況。建議先以理想的環境進行測試，確認基本功能具備
 - 理想環境：晴天、標準道路、標準清晰的標線。

4 ADAS 功能開啓/關閉

ADAS Turn-key codebase 只有在 FHD 錄影下才支持 ADAS，但仍需確認系統效能。
系統效能確認請參考 6.4 節。

目前提供以下功能:

- LDWS (Lane Departure Warning System)
- FCWS (Front Collision Warning System)
Note: 此功能需搭配 LDWS 同時開啓，不可單獨啓動。
- SNG (Stop and Go)
- Stop Detection
- AutoVP (Auto-Vanishing point Calibration)

每一功能在 LIB 內部都有對應的開關 API 可使用:

```
INT32 ADAS_AppsSetLdwsEanble(BOOL bkey);  
INT32 ADAS_AppsSetFcwsEanble(BOOL bkey);  
INT32 ADAS_AppsSetSnGEanble(BOOL bkey);  
INT32 ADAS_AppsSetStopDectEanble(BOOL bkey);  
INT32 ADAS_AppsSetAutoVPEanble(BOOL bkey);
```

這些功能的開關目前由 Project 端控制:

- LDWS: 選單控制開關
- FCWS: 選單控制開關
當 FCWS 開啓, LDWS 未開啓時, 會自動開啓 LDWS。
- SNG: 由 compile option 控制開關
`#define _SNG_FUNC_ DISABLE`
- Stop Detection: 預設打開
- AutoVP: 由 compile option 控制開關
`#define _AUTOVP_FUNC_ ENABLE`

5 彈性設定與調整

下列所提到的 API 在 ADAS Turn-key codebase 中已有輸入的設定範例，可參考 Project\DemoKit\SrcCode\UIApp\Movie\ UIAppMovie_Exec.c 。

```
MovieExe_OnOpen()
{
    ....
    #if _ADAS_FUNC_
    // #NT#New ADAS
    // -----
    // *** ADAS parameter setting ***
    // User can set ADAS parameters (FSize/LD sensitivity/FC warning distance/ LD
    // debounce time/FC debounce time)
    // If not set, ADAS lib will use default settings.
    // Important notice:
    // ADAS_SetLdFcFsizeValueByPixel(), ADAS_SetLdwsLeftSensitivity(),
    // ADAS_SetLdwsRightSensitivity() MUST be set before ADAS_Init()
    // Only ADAS_SetFcwsUIWarningDist() / ADAS_SetFcwsDebounceTime /
    // ADAS_SetLdwsDebounceTime can be changed during ADAS detection.
    // Ex. Different car speed has different FC warning distance.
    //
    // FSize : Please refer to APNote to measure FSize
    // LD Sensitivity : 0~50, default is 15. Bigger value means more sentive.
    // FC distance : 1~40, default is 15. The unit is meter.
    // -----
    // Set fsize value. This value CANNOT be changed on-the-fly.
    ADAS_SetLdFcFsizeValueByPixel(188);
    // Set fcws warning distance (unit: m). This value can be changed on-the-fly.
    ADAS_SetFcwsUIWarningDist(15);
    // Set fcws debounce time (unit: ms). This value can be changed on-the-fly.
    ADAS_SetFcwsDebounceTime(10*1000);
    // Set ldws left side sensitivity. This value CANNOT be changed on-the-fly.
    ADAS_SetLdwsLeftSensitivity(15);
    // Set ldws right side sensitivity. This value CANNOT be changed on-the-fly.
```

```

ADAS_SetLdwsRightSensitivity(15);
// Set ldws debounce time (Unit: ms). This value can be changed on-the-fly.
ADAS_SetLdwsDebounceTime(10*1000);
....
}

```

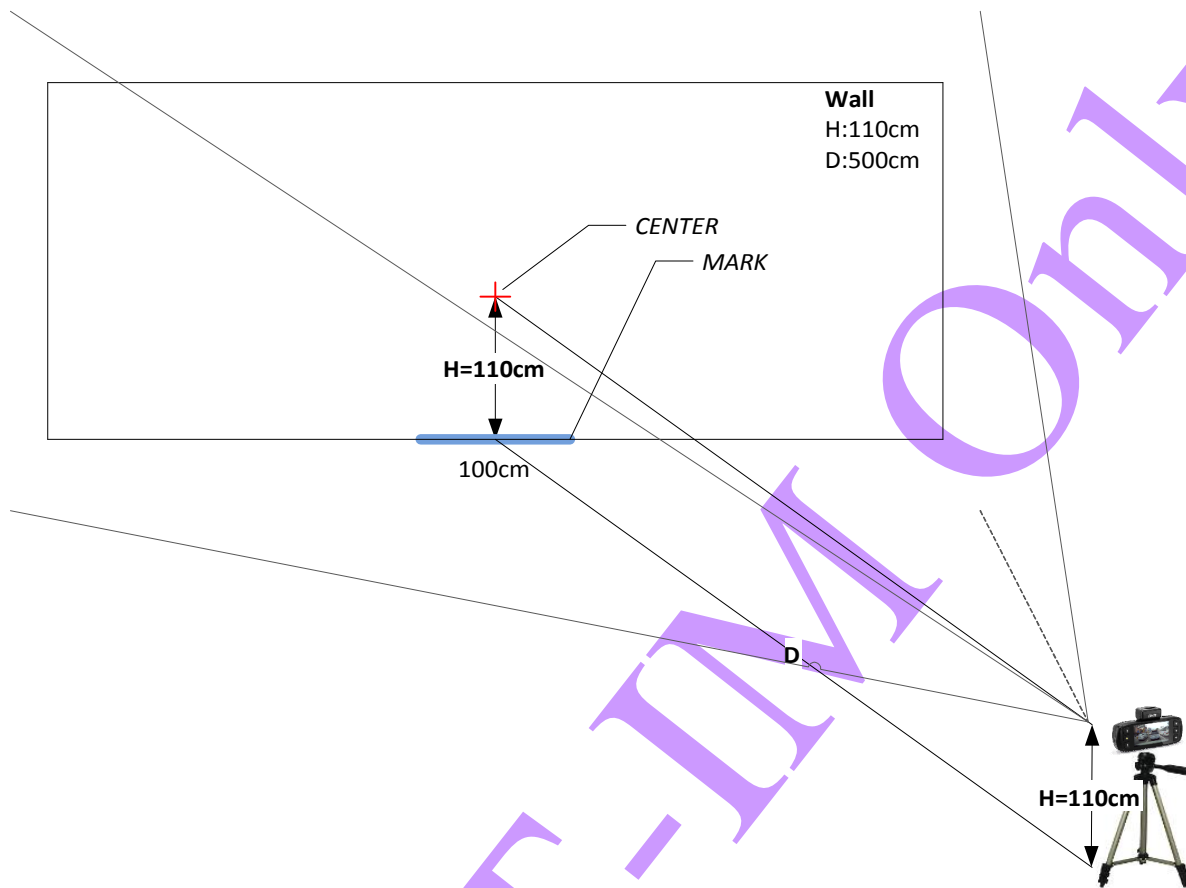
目前系統預設值如下條列:

	Default Value	Be changed on-the-fly?
FsizeValue	100	NO
Fcws UI Warning Distance	15 (m)	YES
Fcws Debounce Time	10*1000 (ms)	YES
Ldws Right Sensitivity	15	NO
Ldws Left Sensitivity	15	NO
Ldws Debounce Time	10*1000 (ms)	YES

5.1 ADAS 鏡頭校正

- 每一款"鏡頭+Sensor"搭配，需要進行一次鏡頭校正。
- 請透過以下方式得到校正值 Fsize
 - 找一面牆，在離地高 **H(110cm)**處標記一"CENTER"，並在地面標記 100cm 長"MARK"
 - 距離 **D(500cm)**遠處架一相機高度 H，相機中心拍攝標記"CENTER"，以確保相機與地面平行。實際架設方式請參考下圖所示。
 - 相機依上述架設好後，請於 **FHD 錄影模式**下錄製一段影片，再從影片中擷取一個畫面，並計算畫面中標記"MARK"寬度的 pixel 數。
 - 將此數值透過以下 API 填入 FW 後進行編譯，即完成校正。

```
void ADAS_SetLdFcFsizeValueByPixel (UINT32 Val)
```

5.2 LDWS 跨越車道靈敏度調整

- LDWS 可設定跨越左右車道時的靈敏程度:

`void ADAS_SetLdwsLeftSensitivity (UINT32 Val);`

`void ADAS_SetLdwsRightSensitivity (UINT32 Val);`

建議值=15, 有效範圍=0~50

設定值越大, 越靈敏, 越早警示

設定值越小, 越不靈敏, 越晚警示

Note: 建議值 15 為校正後影像平面上的 15 pixel, 約等效於距離車道線 10 公分就警示
若設定為 50 則為校正後影像平面上的 50 pixel, 約等效於距離車道線 30 公分就警示

5.3 LDWS Debounce Time 調整

- 為避免 UI 連續警示, LDWS 偵測的結果會做進一步的篩選, 舉例來說 Debounce time = 2s 代表 LDWS 警示之後 2 秒內若再有跨車道的事件發生, 則此次的警示會被忽略。基於上

述的說明，

- 請使用此 API 進行設定，單位為 ms。Debounce Time 建議比警示聲長，且由於一般駕駛跨車道的時間約為 10 秒左右不等，因此建議值為 10s，建議範圍為 10~20s。

`void ADAS_SetLdwsDebounceTime (UINT32 Val); //Unit: ms`

5.4 FCWS Debounce Time 調整

- 為避免 UI 連續警示，FCWS 偵測的結果會做進一步的篩選，舉例來說 Debounce time = 2s 代表 FCWS 警示之後 2 秒內若再有前車過近的事件發生，則此次的警示會被忽略。
- 請使用此 API 進行設定，單位為 ms。Debounce Time 建議比警示聲長，建議值為 10s，建議範圍為 10~20s。

`void ADAS_SetFcwsDebounceTime (UINT32 Val); //Unit: ms`

5.5 FCWS 警示距離調整

- 客戶可自行設定 FCWS 警示距離，建議值為 15~20m，有效範圍為 10~40m。

`void ADAS_SetFcwsUIWarningDist (UINT32 Val); //Unit: m`

6 測試與解析

- ADAS Turn-key package 移交至客戶端後，若使用上有出現問題或者是效果不佳時，可先依照下列介紹的方法進行簡單的問題排除與確認實際效果。
- 若一一確認後仍然不能解決問題，請將客戶實測效果不佳的影片回傳，再由演算法人員進行分析。

6.1 使用標準影片進行測試

- 測試影片 [ADAS_TestVideo.avi](#)
要確認 ADAS 效果最理想的狀況為上路測試，但在這之前可以先在辦公室使用測試影片進行初步效果確認。需注意的是此方法僅能確認 ADAS 有正常運作，實際的效果仍以上路實測為主。
- 但由於演算法內部有一些參數是與實際上路相關的，因此若使用太過複雜的影片進行測試，其效果可能不彰，無法做初步的效果判定。因此我們這邊提供了適合辦公室測試的影片，此影片場景單純，包含了車道偏移與前車防碰撞的事件，利於優先確認效果。
- 使用標準影片進行測試時，請讓影片畫面滿版(如下圖所示)，樣機與螢幕的測試距離建議為 15~20cm，並確認錄製影像清晰。若距離過近會導致影像模糊，嚴重影響測試效果，此時請更換更大的螢幕進行測試。
- 測試時會出現 ADAS 相關警示事件與圖案標記，以下做詳細說明。
 - 剛啟動錄影時 OSD 畫面，十字為自動校正的消失點位置，紅色表示 ADAS 失效，代表尚未找到穩定的車道線。



- 進入正常偵測後會出現車道線與前車位置，十字線也變為藍色。



OSD 圖示說明

圖一

圖二

圖三



前車警示



右邊跨線



左邊跨線

	顏色			意義
	紅	綠	藍	
十字線	ADAS 失效		正常運作	自動校正(AutoVP)的當前位置
左右車道線 (圖二、三)	單邊 偏離	偵測到車道線 但未偏離車道		LDWS 車道線偵測結果
前車位置 (圖一)	達到警 示距離	偵測到前車 尚未進入警示距離		FCWS 前車偵測結果

OSD 警示事件與圖示說明

LDWS (左邊跨線)	LDWS (右邊跨線)	FCWS (與前車距離過近)
		

■ 車距與時速顯示

Format: 與前車車距/GPS 時速資訊 (以此圖為例, 7m/0km)



- 此段影片測試結果請參考 [ADAS_TestVideo_Result.MOV](#)

6.2 確認各功能是否正常開啓

使用 Uart command [[adas getAppStus](#)] 確認 LDWS/FCWS/AutoVP 是否有 enable。

Note: preview 時就可確認

```
> adas getAppStus
ADAS Applications Enable/Disable status:
LDWS = 1
FCWS = 1
AutoVP = 1
SnG = 0
Stop Detection = 1
>
```

靜止偵測(Stop Detection)是一輔助功能，可在車輛靜止時降低 ADAS 誤報，此功能預設為開啓。

輸入[\[adas getStus\]](#) 可確認 ADAS 是否有在執行

```
> adas getStus
ADAS status = 0 (0: stop)
>
```

ADAS 目前的執行流程為錄影時才啟動(ADAS status = 1)，preview 時 ADAS 狀態應為 Stop (ADAS status = 0)。

```
> adas getStus
ADAS status = 1 (0: stop)
```

6.3 確認自動校正機制正常運作

ADAS 採用自動校正機制，且此機制運作在第二顆 CPU (eCos)，因此請先確認自動校正機制有沒有正常運作。

接上 Uart 並將樣機開機，先確認 Uart 上是否出現 eCos 啟動的提醒字串

(藍色提醒框內 Hello, eCos world!!!!!!!!!!!!!!)。若沒有代表 eCos 沒有正確被啟動。

```
=====
NT96660 SDK
Copyright (c) 2014 Novatek Microelectronic Corp.
=====
Hello, World!
> System Boot begin

-----
LD VERISON: LD660
FW --- Daily Build: Mar 31 2016, 13:08:03
-----

Power On Sequence = Safe
ERR:nandfw_setParam() STRG_SET_LOADER_SECTOR_COUNT sector count[000]
WRN:xSysStrg_eCosRun() LDCF_BOOT_CARD is detected.
WRN:FileSys_WaitFinishEx() FileSys_Init not ready
WRN:sdioHost_setBusClk() SDIO host0 : real clock (398340Hz) is not equal to desired (399000Hz)
WRN:DePackCb_eCos() eCos form A:\
WRN:xSysStrg_eCosRun() eCos boot from temp all-in-one bin.
Hello, eCos world!!!!!!!!!!!!!!
==== NOT Start init_inet for ftpd ===
Ready for eCos scheduling...
cpu2 power on ready
uart request
get: uart ipc, addr=A0983938
Run uart main...
```

同時，若 LIB 內部偵測到 eCos 沒有被啟動，也會印出警告訊息。

ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable

```
[0] timeid=0x2b3d48!
[draw19]
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
Flush time=235 ms!
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
[draw24]
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
Flush time=260 ms!
[draw18]
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
[draw20]
[draw19]
ERR:ROI_Probabilistic_Hough() POWERON_FAST_CPU2_BOOT not enable
FS[0] W=0xa770b9c0!s=0x100000,oft=0xa40000,696ms!
[CUT][file0]upHDR ok 106 ms!
ERR:IPL_HeaderRawGetCurInfo_Int() header address error(0, 9, 0)
ERR:Photo_DIS() DIS: Get header error!!!
Call Stop callback 1
```

若看到以上的訊息，請至 PrjCfg_XXX.h 將 eCos 打開，再重新編譯出 FW 進行測試。

#define **POWERON_FAST_CPU2_BOOT** **ENABLE**

目前在 OSD 上會畫出當前的消失點位置，也就是自動校正的位置。請將機器對著影片進行測試，並觀察 OSD 上的十字是否會移動。若 eCos 正常開啓的情況下，十字線移動的速度約為

2 秒更新一次。約 10~15 秒可以校正到正確的消失點位置。



一開始消失點位置錯誤



十字移到了正確消失點位置



ADAS 正常偵測

6.4 確認系統效能

若自動校正的十字線或者是車道線有正常的偵測到，但更新的很緩慢，有可能是系統效能問題，請切換至最低的系統規格，比如說從 FHD P60 切換至 FHD P30 再進行測試。確認效果是否改善。

以下是幾種可以確認系統效能的方法：

錄影模式下在 Uart 上打入下列指令，即可確認當前系統效能：

[usage cpucfg 1000] ←設定計算間隔 interval = 1000
[usage cpustart] ←開始計算 CPU 使用量
[usage cpustop] ←停止計算

```
> usage cpucfg 1000
sterval=1000 m
>
> usage cpustart
> 68
69
72
> usage cpustop
>
>
> usage dmacfg 1000
sterval=1000 m
>
> usage dmastart
> 52
51
52
```

[usage dmacfg 1000]

[usage dmastart]

[usage dmastop]

```
> usage dmastart
> 52
51
52
> usage dmastop
>
>
> #
```

或是使用 ADAS 指令 [adas getSymInfo], interval = 10ms

進一步使用 [adas getProcessTime] 可得知 ADAS Processing time :

```
> adas getProcessTime
High Priority Apps (SnG) processing time = 73 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 67 (ms)
Low Priority Apps (AutoVP) processing time = 1034 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 72 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 78 (ms)
Low Priority Apps (AutoVP) processing time = 1035 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 69 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 81 (ms)
Low Priority Apps (AutoVP) processing time = 1038 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 72 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 82 (ms)
Low Priority Apps (AutoVP) processing time = 1045 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 71 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 82 (ms)
Low Priority Apps (AutoVP) processing time = 1048 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 74 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 82 (ms)
Low Priority Apps (AutoVP) processing time = 1041 (ms)
> adas getProcessTime
High Priority Apps (SnG) processing time = 80 (ms)
Middle Priority Apps (LDWS+FCWS) processing time = 72 (ms)
Low Priority Apps (AutoVP) processing time = 1043 (ms)
```


ADAS 執行時間建議為 10fps，方能得到較佳的效果。

ADAS 所需要的系統效能如下條列:

	CPU1	DMA
ADAS OFF	< 40%	<70%
ADAS ON	<90%	<90%

爲了協助快速確認，另外提供了一支整合性的指令 [\[adas check\]](#)

可同時確認系統效能，並藉由當前的處理時間換算成 fps

```

=== Checking System Resource ===
===== DMA usage =====
Interval=1000 ms
81
Adas_Detection(), AutoVP srcWidth = 0 or srcHeight = 0, please check it!!!
81
81
82
81
81
81
81
81
81
81
===== CPU usage =====
Interval=1000 ms
Adas_Detection(), AutoVP srcWidth = 0 or srcHeight = 0, please check it!!!
90
91
92
90
91
92
89
91
Adas_Detection(), AutoVP srcWidth = 0 or srcHeight = 0, please check it!!!
91
=====
= Checkin System Resource done =

===== ADAS SYSTEM INFO =====
CPU1 : Please check above data to find CPU1 usage
DMA  : Please check above data to find DMA usage
ADAS processing time = 83ms ~= 10fps
Fsize: 188
NOTES:
CPU1 recommended usage < 90 when ADAS on
DMA  recommended usage < 90 when ADAS on
The  recommended performance of ADAS is 10fps
The  default value of Fsize = 188

```

請確認量測出來的系統資源是否符合 ADAS 所需要的條件。若沒有達到條件，請客戶協助改善系統效能問題。

6.5 確認 LDWS 效果

在 ADAS 真正上路測試前可先在辦公室進行初步的效果確認。

- 電腦播放測試影片，拿著機器對著螢幕測試，當自動校正完成後（畫面上的十字線），LDWS 的車道線應會顯示在 OSD 上。若車道線更新速度過慢，請參考 Sec4.4 進行系統效能確認。

6.6 確認 FCWS 效果

在 ADAS 真正上路測試前可先在辦公室進行初步的效果確認。

不過由於 FCWS 有許多參數與鏡頭校正相關，因此辦公室測試的效果會比較不明顯。會發生無法一直追蹤前車的狀況。

- 使用 command `[adas getFsize]` 確認 Fsize 是否有校正。
由於 ADAS Turn-key codebase 會有一組預設值(F size = 188)，因此利用 command 得到的值如果等於 188，代表客戶有很大的機率是沒有校正的。此時請先跟客戶確認是否有依正常流程進行校正過。

6.7 影片分析

若一一確認後仍然不能解決問題，請將客戶實測效果不佳的影片回傳，再由演算法人員進行分析。