**大綱**

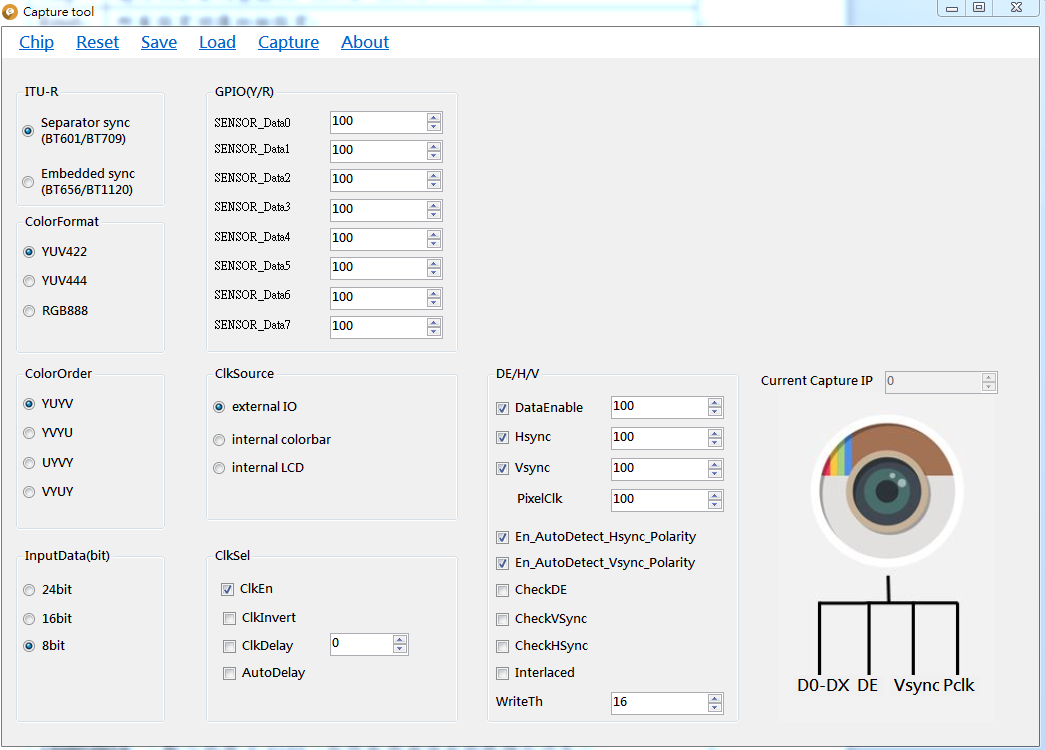
1. **Capture 工具介紹…………………………………………...........2**
2. **如何加入新的感測器………………………………………………3**

**Released rev 0.5 2019/9/12**

1. **Capture 工具介紹**
2. **路徑:**ite\_sdk/Tool/bin/capture.exe
3. **功能:**提供使用者人機介面快速產生capture initialize file
4. **選單說明:**

|  |  |
| --- | --- |
| Chip | 選用ITE系列型號ex IT972 IT976 => IT970 |
| Reset | 所有設定回復初始設定 |
| Save | 儲存當前設定，轉換成initialize file |
| Load | 讀取已存在的initialize file(.txt) |
| Capture | 選擇使用單路CAPTURE or 雙路CAPTURE |
| About | 顯示軟體版本 |

1. **人機介面:**



1. **重要設定選項說明:**

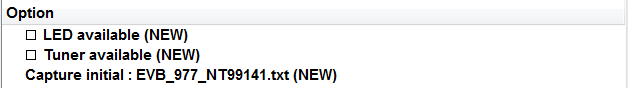
|  |  |
| --- | --- |
| **ITU-R** | Separator sync (BT601、BT709)or Embedded sync (BT656 、BT1120)(依照感測器送出訊號格式決定) |
| **inputData** | 輸入訊號 n pins (依照感測器送出訊號格式決定) |
| **GPIO** | 依照電路圖接線決定 |
| **DataEnable** | separator sync 選擇對應GPIO  embedded sync 無須選擇GPIO ,僅勾選即可 |
| **Hsync** | separator sync依照感測器需求設定  embedded sync 無須選擇GPIO ,僅勾選即可 |
| **Vsync** | separator sync選擇對應GPIO  embedded sync 無須選擇GPIO ,僅勾選即可 |
| **Pixelclk** | 感測器送出的clk，選擇對應GPIO |
| **ColorFormat** | 依照感測器送出的顏色格式 |
| **ColorOrder** | 依照感測器送出的顏色排列順序 |
| **CheckDE** | Capture 會等待勾選的項目穩定後才開始輸出資料，建議開啟 |
| **CheckVSNC** |
| **CheckHSNC** |
| **Interlaced** | 依照感測器輸出Interlaced or progressive  若兩者都有，請在程式碼中做自動偵測的動作 |
| **Clkinvert** | 如果pixel clk pin, data pin 都設對，Capture 依然unstable  可以調整Clkinvert 、Clkdelay來使capture 達到stable |
| **Clkdelay** |

1. **如何加入新的感測器**

**Step 1.** 根據感測器spec 以及PCB電路圖並使用capture工具產生出對應初始化設定檔案。

**Step 2**. RUN ite\_sdk\build\openrtos\test\_cap.cmd

Peripheral-> Capture initial double click 選擇Step 1.生成的檔案。



**Step 3.**

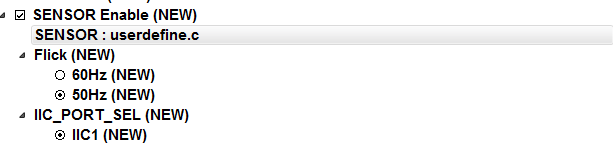
依據感測器類型選擇架構



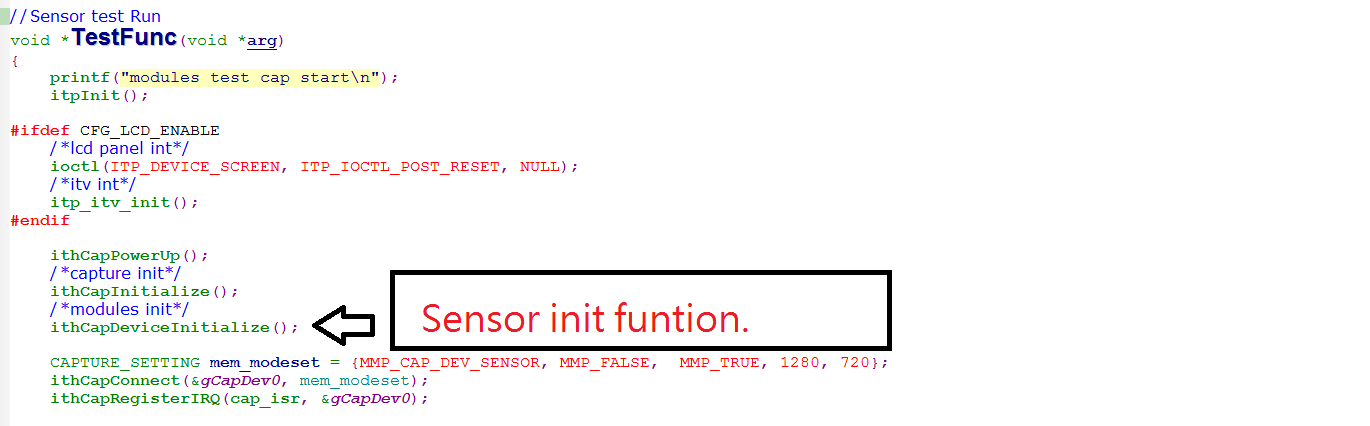
|  |  |  |
| --- | --- | --- |
| mode | Device | .c file |
| CAP\_HDMIRX\_MODE | HDMI\_IC | test\_hdmirx.c |
| CAP\_MODULE\_MODE | Sensor(非HDMI) | test\_modules.c |

**Step 4.**

* 確認I2C是否可以正確與新增設備溝通，範例程式可以參考project/test\_iic.c
* 可以先選擇userdefine.c ,實作IIC comment 至userdefine.c



* ithCapDeviceInitialize(); 會呼叫被選擇到的sensor driver



* 依據感測器輸出的最大寬高做調整

typedef struct CAPTURE\_SETTING\_TAG

{

MMP\_CAP\_DEVICE\_ID inputsource;

MMP\_BOOL OnflyMode\_en;

MMP\_BOOL Interrupt\_en;

MMP\_UINT32 Max\_Width;

MMP\_UINT32 Max\_Height;

} CAPTURE\_SETTING;

**Example code**  width = 1280 height = 720.



**Step 5.**

* 實作感測器driver, 以sdk\driver\sensor\userdefine.c為例
* UserInitialize()

透過IIC command 初始化感測器

* UserTerminate()

透過IIC command 結束感測器

* UserOutputPinTriState()

透過IIC command 使感測器開始輸出訊號，或是不輸出訊號

* UserIsSignalStable()

透過IIC command 讀取感測器訊號是否穩定，若沒有相關資訊則回傳true

* UserGetProperty()

透過IIC command 讀取感測器訊號相關參數

|  |  |
| --- | --- |
| GetHeight | 感測器高度 |
| GetWidth | 感測器寬度 |
| Rate | 每秒顯示帧数  Ex 60fps = 6000 |
| GetModuleIsInterlace | 交錯式掃瞄 = 1 循序式掃瞄 = 0 |

* UserPowerDown()

透過IIC command or gpio 開關感測器電源

* UserGetStatus()

透過IIC command讀取感測器狀態

* UserSetProperty()

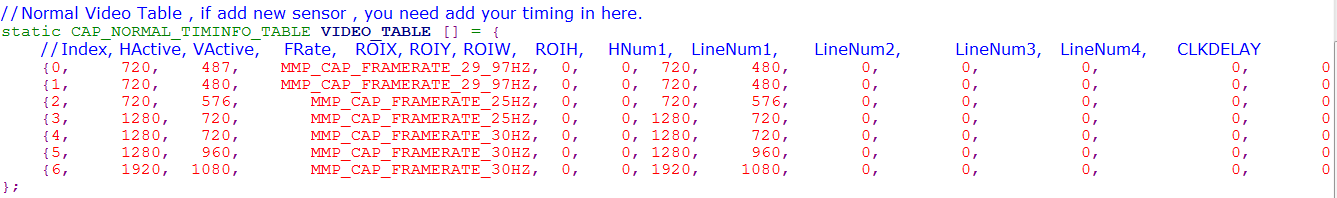
透過IIC command設定感測器狀態

**Step 6.**

**檢查ithCapGetDeviceInfo()是否需要修改**

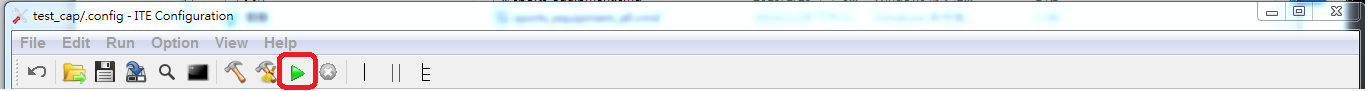
|  |  |
| --- | --- |
| Device | .c file |
| HDMI\_IC | Device\_hdmirx.c |
| Sensor.(not hdmi) | Device\_modules.c |

**檢查Timing table 是否需要新增(video\_device\_table.h)**



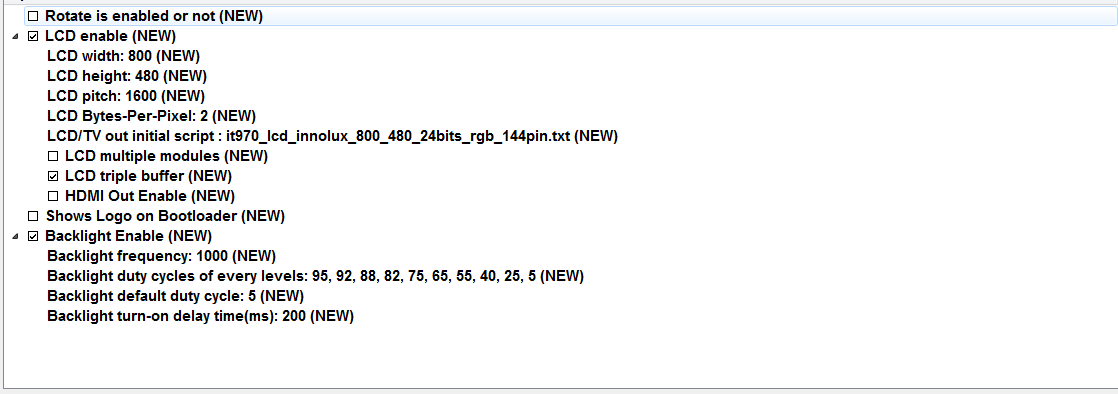
**Step 7.**

**Run your test\_cap.**

****

* 如果有打印Capture Fire!關鍵字表示有成功啟動capture
* 如果只打印Hsync or Vsync not stable!! 而沒有上述fire，表示前端訊號不穩定，請檢查感測器的設定或pin 設定或gpio 設定
* Capture 正確啟動後，希望由LCD直接顯示就要設定正確的LCD script

可以先用project/test\_lcd 確認LCD script是否正確可以點亮LCD



* Dump capture memory

確認capture收到的原始資料

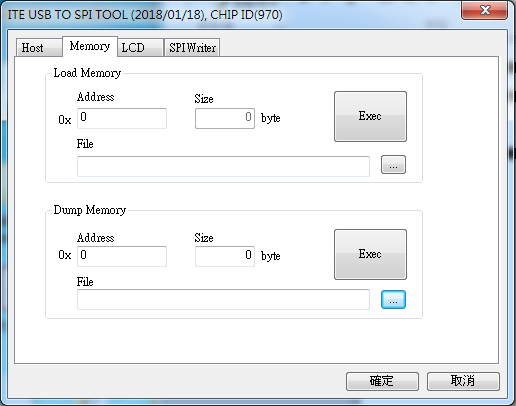
可以使用ITE USB TO SPI TOOL

Address : CapMemBuffer[0] addr = 0x????????

Size: height \* width \* 1.5

File: 選擇存取path

預設顏色格式: YUV420 NV12



**Step 7.**

測試程式一切正常就可以 開始加入project