

一、程式檔案架構

● Aetina_SW_TestingTool

- bin/
 - ◆ config.ini
 - ◆ curlftpfs
 - ◆ lib/
- config.ini
- include/
 - ◆ common.h
 - ◆ doTests.h
 - ◆ utils/
 - dictionary.h
 - iniparser.h
 - libTestOfBoard.h
- Makefile
- Readme.md
- src/
 - ◆ common.c
 - ◆ doTests.c
 - ◆ utils/
 - dictionary.c
 - iniparser.c
 - libTestOfBoard.c
 - lm75_tmp100.c
 - serial_port_test.c
 - spidev_test.c

程式根目錄

執行檔,執行檔參數,函式庫,相關程式目錄

執行檔參數

FTP Tool

函式庫目錄

執行檔參數(備份用)

程式標頭檔目錄

主程式標頭

主程式標頭

函式庫 libTestOfBoard 標頭目錄

用於讀寫 config.ini

用於讀寫 config.ini

libTestOfBoard 標頭

Make 檔(編譯參數及程序,安裝程序)

程式目錄

主程式程式

主程式程式

函式庫 libTestOfBoart 程式目錄

用於讀寫 config.ini

用於讀寫 config.ini

libTestOfBoard 來源檔

AN310-溫度感測 IC 測試函式

UART,RS232 測試函式

SPI 測試函式

二、程式流程

三、程式變數及函式說明

測試程序及報告相關：

- **define:**
 - **RESET_TIMES** 測試限制次數，預設 3，超出時重設 config
 - **REPORT_CONTENT_LENGTH** 報告行數長度
 - **GONDAM_NUMBER_LENGTH** 工單字數長度標準，用於檢查工單輸入
 - **BOARD_NUMBER_LENGTH** 底板字數長度標準，用於檢查底板輸入
- **enum:**
 - **ItemKind** 定義測項種類
 - **TestMode** 定義測試模式
- **Struct:**
 - **Items** 存放測項容器
 - **TestConfigs** 測項輸入參數
 - **TestResults** 測項輸出結果
 - **TestReport** 測試報告用參數
 - **TestCore** 測試核心(含測項,輸入,輸出,報告參數)
- **Function:**
 - config**
 - **int checkConfigItem(char*, int*)** 讀取測項測試狀態
 - **int setConfigItem(char*, int*)** 設定測項測試狀態
 - **resetConfigItem(void)** 重設所有測項狀態
 - **char* checkConfigInfo(char*,char*)** 讀取報告用資訊(工單、品號、底板號碼)
 - **int setConfigInfo(char*, char*)** 設定工單/品號/底板號碼
 - **resetConfigInfo(void)** 重設工單、品號、底板號碼
 - Items**
 - **Items* ItemsInit(void)** 初始化測項變數
 - **Items* addItem(Items*, ItemKind)** 增加測項
 - **Items* findItem(Items*, ItemKind)** 搜尋測項
 - **void showAddedItems(Items*)** 顯示已加入測項
 - **void showNonFinishItems(Items*)** 顯示未完成測項
 - TestConfigs**
 - **TestConfigs TestConfigInit(void)** 初始化測項輸入變數
 - TestResults**
 - **TestResults TestResultInit(void)** 初始化測項結果變數
 - TestReport**
 - **TestReport* TestReportInit(void)** 初始化結果報告變數
 - **void writeInfoSparateLine(FILE*)** 撰寫分割線(工單,底板,品號,MAC 資訊用)

■ void writeItemSparateLine(FILE*)	撰寫分割線(測項用)
■ void writeReportHead(FILE*)	撰寫主要資訊(工單,品號,底板,MAC,Module,)
■ void writeInfoContent(FILE*, int, ...)	撰寫資訊內容
■ void writeItemContent(FILE*, int, ...)	撰寫測項內容
■ void writeResultContents(FILE*, Item*, TestConfigs*, TestResult*)	撰寫所有測項結果
■ TestReport* generateReport(TestReport*, DeviceInfo*, Item*, TestConfigs*, TestResult*, bool)	生成報告
■ void showReport(TestReport*)	顯示報告
■ uploadReport(TestReport*)	上傳報告
■ void deleteReport(void)	刪除報告

TestCore

■ TestCore* TestCoreInit(void)	初始化測試核心變數
■ void showDebugMenu(void)	顯示測項選單
■ TestReport* scanBoardNumber(TestReport*)	等待掃描底板序號
■ TestReport* scanGondamAndPartNumber(TestReport*)	等待掃描工單及品號序號
■ waitForShutdown(void)	等待輸入 Enter 後自動關機
■ autoRunTests(TestCore*)	執行自動測試
■ configureAutoRunItems(TestCore*)	設置自動測試的測項
■ runTest(TestCore*, char)	執行測試
■ configureDebugItems(TestCore*)	設置 Debug 測試的測項
■ setupCfgAndRst(Items*, TestConfigs*, TestResults*)	初始化測項輸出入變數
■ setCfgNRst_TX2(const ProductPortFolio, Items*, TestConfigs*, TestResults*)	設定輸出入變數參數(TX2)
■ setCfgNRst_TX2_4GB(const ProductPortFolio, Items*, TestConfigs*, TestResults*)	設定輸出入變數參數(TX2-4GB)
■ setCfgNRst_Xavier(const ProductPortFolio, Items*, TestConfigs*, TestResults*)	設定輸出入變數參數(Xavier)
■ setCfgNRst_Nano(const ProductPortFolio, Items*, TestConfigs*, TestResults*)	設定輸出入變數參數(Nano)

libTestOfBoard

- define:
 - MAC_SIZE
 - IP_SIZE
 - MAX_PATH
 - MAX_CMD
- enum:
 - JetsonModule
 - ProductPortFolio
 - ERROR_CODE
- Struct:
 - DeviceInfo
 - LAN_CFG
 - LAN_RESULT
 - WIFI_RESULT
 - BT_RESULT
 - SDCARD_CFG
 - SDCARD_RESULT
 - MSATA_CFG
 - MSATA_RESULT
 - MPCIE_CFG
 - MPCIE_RESULT
 - RS232_CFG
 - RS232_RESULT
 - GPIO_CFG
 - GPIO_RESULT
 - UART_CFG
 - UART_RESULT
 - FAN_CFG
 - FAN_RESULT
 - CAN_CFG
 - CAN_RESULT
 - I2C_CFG
 - I2C_RESULT
 - SPI_CFG
 - SPI_RESULT
 - TEMPERATURE_CFG
 - TEMPERATURE_RESULT

libTestOfBoard 測項函式庫相關定義

網卡 MAC 字串長度

網路 IP 字串長度

用於宣告

用於宣告命令字串長度

定義 Module Chip ID

定義產品組合

定義錯誤情況

存放 Module,LAN 數量,底板,S/N,MAC,BSP,DTS 結構

LAN 輸入參數結構

LAN 輸出結果結構

WIFI 輸出結果結構

BlueTooth 輸出結果結構

SDCard 輸入參數結構

SDCard 輸出結果結構

m-SATA 輸入參數結構

m-SATA 輸出結果結構

m-PCIE 輸入參數結構

m-PCIE 輸出結果結構

RS232 輸入參數結構

RS232 輸出結果結構

GPIO 輸入參數結構

GPIO 輸出結果結構

UART 輸入參數結構

UART 輸出結果結構

FAN 輸入參數結構

FAN 輸出結果結構

CAN Bus 輸入參數結構

CAN Bus 輸出結果結構

i2c 輸入參數結構

i2c 輸出結果結構

spi 輸入參數結構

spi 輸出結果結構

AN310-溫度感測器輸入參數結構

AN310-溫度感測器輸出結果結構

四、程式修改程序

修改情況分為新增測項、新增底板、新增 Module、新增產品組合

1. 新增測項

- A. 了解測項及測試方法，額外撰寫獨立測項程式並測試
- B. 將測項程式依照產測程式框架進行整合

需更動檔案：

● libTestOfBoard.h	宣告測項所需輸出入結構、參數、函式
● libTestOfBoard.c	定義測項函式程序
● doTest.h	宣告測項執行函式
● doTest.c	定義函式與結果判斷
● common.h	新增測項相關變數
I. ItemKind	增加測項種類
II. TestConfigs	增加測項輸入變數
III. TestResults	增加測項輸出變數
● common.c	測項所需變數及修改部分函式
I. items_name	新增測項項目名稱
II. TestConfigsInit	測項輸入變數初始化
III. TestResultsInit	測項輸出變數初始化
IV. setFailedResults	定義測項 Failed
V. writeResultContents	新增撰寫測項敘述
VI. showDebugMenu	新增 Debug 模式測項選單敘述
VII. configureAutoRunItems	定義 Auto 模式加入測項情況
VIII. configureDebugItems	定義 Debug 模式加入測項情況
IX. autoRunTests	定義 Auto 模式測試測項情況
X. runTests	定義 Debug 模式測試測項情況
XI. setupCfgAndRst	定義測項輸出入初始化
XII. setCfgNRst_ "Module"	定義測項輸出入變數在不同 Module
● config.ini	新增測項

2. 新增底板

- A. 釐清新底板的所需測項:
是否需新增測項，是->1.新增測項->繼續 2-B，否，繼續 2-B
- B. 確認所需測項的相關參數:
 - I. 確認 BSP 的底板腳位定義
 - II. 參考 libTestOfBoard 函式庫測試方法，透過終端機測試並記錄結果
- C. 依照需求，增加新底板至產測程式

需更動檔案：

I. libTestOfBoard.h	II. libTestOfBoard.c
	III. common.c
● ProductPortFolio	● setCfgNRst_ "Module"

- configureTestCore

新增底板

依照底板所需，加入參數
加入底板判斷

3. 新增 Module

- 是否搭配新底板，是->2.新增底板->繼續 3-B，否，繼續 3-B
- 依照需求，增加 Module 至產測程式
需更動檔案：

I. common.h

新增 setCfgNRst_ "Module" 函式

II. common.c

- setCfgNRst_ "Module"
- configureTestCore

定義 setCfgNRst_ "Module"
加入 Module 判斷

III. libTestOfBoard.h

- JETSON_MODULE_ID
- JetsonModule

加入 Module Chip ID
增加 Module 種類

IV. libTestOfBoard.c

- TOB_GetDeviceInfo

定義 Module 判斷

4. 新增產品組合

- 搭配周邊是否為新導入
 - 是->評估是否已有相同種類周邊
 - 1.修改測項
 - 否->2.新增測項
- 依照需求，增加 Module 至產測程式
需更動檔案：

i. libTestOfBoard.h

- ProductPortFolio

新增產品組合

ii. common.c

- setCfgNRst_ "Module"
- configureTestCore

根據產品組合設定輸入參數
加入新產品組合判斷