



IHRCO-Based MCU

MG82G5Exx

IEC-60730 範例程式 使用手冊

Version: 1.00

List of Contents

1. IEC-60730 規範概述.....	4
2. IEC-60730 測試項目說明.....	5
2.1. CPU Register Test.....	5
2.2. CPU PC Test.....	5
2.3. Interrupt Test.....	5
2.4. Clock Test	5
2.5. Invariable Memory Test(ROM)	6
2.6. Variable Memory Test(RAM)	8
2.7. IO Test.....	8
2.8. AD Test	8
3. 注意事項	8
4. 版本历史	9
5. 免责声明	9

1. IEC-60730 規範概述

在美國和歐洲的監管機構紛紛提出符合法規要求的家電安全設計。國際電工委員會推出了 IEC60730 家用電器的安全發展標準。在歐洲，這些要求是由 IEC 60730 定義的。2007 年 10 月，出售到歐洲市場上的所有家電產品須符合 IEC 60730 標準。其目的是避免失敗或至少保證在設備的任何故障不存在為害使用者的風險。微控制器開發這些設備時，半導體供應商必須考慮這些標準對家電廠商的影響。IEC60730 的附錄 H 描述了三種軟件 Class A~C 分類。使用微控制器 (MCU) 的情況下。在發展之初，設計者必須考慮其硬體和軟體功能的 MCU 需要符合 IEC60730 標準之要求。此範例程式是以 Class B 來進行。

2. IEC-60730 測試項目說明

IEC-60730 Class B 主要測試項目如下：

2.1. CPU Register Test

此測試是針對 CPU 內核相關暫存器去測試是否可以正常寫與讀，且此類暫存器較為核心，故使用 Assembly 來執行測試，若測試結果是錯誤，將會執行無窮迴圈。

測試暫存器名稱：R0 ~ R7、A、B、DPH、DPL、SP。

函式名稱：Test_Reg

2.2. CPU PC Test

此測試是測試 Program Counter，使用者預先定義 15 個 subroutines 的程式位址(program address)，呼叫這 15 個 subroutines 並比對 PC 是否與預先定義的程式位址(program address)相同，此測試是 destructive，當系統重置(reset)時應該呼叫執行，若測試結果是錯誤，將會執行無窮迴圈。

函式名稱：Test_PC

2.3. Interrupt Test

此測試是由兩個不同時間、固定會發中斷的中斷來執行，主要測試中斷 A 數中斷 B 發生的中斷次數是否在預期範圍內，這兩個中斷使用同一個 clock source，以笙泉提供的範例程式來說，Timer0 使用 IHRCO clock source，設定每 1ms 產生中斷一次，Timer1 也使用 IHRCO clock source，設定每 100us 產生中斷一次，在 Timer0 中斷裡去比對 Timer1 的中斷次數是否在預先定義的範圍內，若超出範圍，則必須回傳一個錯誤。

函式名稱：unsigned char IntTest(void)

回傳：若比對正確則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

2.4. Clock Test

此測試是由兩個不同時間、固定會發中斷且來自不同 clock source 的中斷來執行，主要測試中斷 C 數中斷 A 發生的中斷次數是否在預期範圍內，以笙泉提供的範例程式來說，Timer0 使用 IHRCO clock source，設定每 1ms 產生中斷一次，RTC 使用 ILRCO clock source，設定大約每 0.5s 產生中斷一次，在 RTC 中斷裡去比對 Timer0 的中斷次數是否在預先定義的範圍內，若超出範圍，則必須回傳一個錯誤。

函式名稱：unsigned char ClockTest(void)

回傳：若比對正確則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

2.5. Invariable Memory Test(ROM)

此測項是檢查 ROM 是否有 single bit fault，用 CRC16 來執行測試，此測項會把 MCU 硬體 CRC Engine 產生的 CRC code 記錄在 IAP 的 0x79FC ~ 0x79FF(若此位址是空白 0xFF，將會記錄在此)，每次 boot 會用 CRC Engine 檢查 flash 並產生 CRC code，再與 IAP 記錄的 CRC code 做比對看是否正確，若 CRC code 比對錯誤將會執行無窮迴圈。

函式名稱：unsigned int MakeCRCCode(unsigned char Addr_H, unsigned char page)

參數 Addr_H：檢查 flash 的起始位址。

參數 page：要檢查的 page 數量，1 page = 512bytes。

回傳：CRC Engine 產生的 CRC code。

函式名稱：unsigned char CRCTest(unsigned char Addr_H, unsigned int crc_original)

參數 Addr_H：檢查 flash 的起始位址。

參數 crc_original：要和 IAP 比對的 CRC code。

回傳：若比對正確則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

當 CPU 跑 12MHz，使用 CRC Engine 檢查 1K 的 ROM Size，花費時間為 88us。

當 CPU 跑 12MHz，使用 CRC Engine 檢查 2K 的 ROM Size，花費時間為 188us。

當 CPU 跑 12MHz，使用 CRC Engine 檢查 4K 的 ROM Size，花費時間為 344us。

當 CPU 跑 12MHz，使用 CRC Engine 檢查 8K 的 ROM Size，花費時間為 697us。

當 CPU 跑 12MHz，使用 CRC Engine 檢查 16K 的 ROM Size，花費時間為 1.38ms。

當 CPU 跑 16MHz，使用 CRC Engine 檢查 1K 的 ROM Size，花費時間為 66us。

當 CPU 跑 16MHz，使用 CRC Engine 檢查 2K 的 ROM Size，花費時間為 130us。

當 CPU 跑 16MHz，使用 CRC Engine 檢查 4K 的 ROM Size，花費時間為 268us。

當 CPU 跑 16MHz，使用 CRC Engine 檢查 8K 的 ROM Size，花費時間為 522us。

當 CPU 跑 16MHz，使用 CRC Engine 檢查 16K 的 ROM Size，花費時間為 1.037ms。

當 CPU 跑 24MHz，使用 CRC Engine 檢查 1K 的 ROM Size，花費時間為 44us。

當 CPU 跑 24MHz，使用 CRC Engine 檢查 2K 的 ROM Size，花費時間為 87us。

當 CPU 跑 24MHz，使用 CRC Engine 檢查 4K 的 ROM Size，花費時間為 172us。

當 CPU 跑 24MHz，使用 CRC Engine 檢查 8K 的 ROM Size，花費時間為 350us。

當 CPU 跑 24MHz，使用 CRC Engine 檢查 16K 的 ROM Size，花費時間為 689us。

當 CPU 跑 32MHz，使用 CRC Engine 檢查 1K 的 ROM Size，花費時間為 33us。

當 CPU 跑 32MHz，使用 CRC Engine 檢查 2K 的 ROM Size，花費時間為 65us。

當 CPU 跑 32MHz，使用 CRC Engine 檢查 4K 的 ROM Size，花費時間為 129us。

當 CPU 跑 32MHz，使用 CRC Engine 檢查 8K 的 ROM Size，花費時間為 262us。

當 CPU 跑 32MHz，使用 CRC Engine 檢查 16K 的 ROM Size，花費時間為 517us。

2.6. Variable Memory Test(RAM)

此測項是檢查 RAM 是否可以正常寫讀，每次 boot 會對 IC 上所有的 RAM 做寫讀測試，測試完並清除為 0x00，若測試結果是錯誤，將會執行無窮迴圈。

函式名稱：Test_RAM

2.7. IO Test

此測項是測試 IO 的輸入 & 輸出功能，當在測試輸出功能時，有可能因為使用者的電路關係而無法輸出“使用者要測試的準位”。

函式名稱：unsigned char Pn_Input_Test(unsigned char value)

參數 value：填入 Pn 預期要輸入的值

回傳：若比對正確則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

函式名稱：unsigned char Pn_Output_Test(unsigned char value)

參數 value：填入 Pn 預期要輸出的值，看輸出是否正確。

回傳：若輸出正確則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

2.8. AD Test

此測項是測試 ADC 功能，測試選擇的 AD 輸入通道所轉換出來的 AD 值是否有在預期的範圍之內，若結果不在範圍內，程式要執行無窮迴圈。

函式名稱：unsigned char AD_Test(unsigned char AIN_num, unsigned int value_low, unsigned int value_upper)

參數 AIN_num：AD 通道

參數 value_low：預期範圍的最小值

參數 value_upper：預期範圍的最大值

回傳：若得到的 AD 值在範圍內則回傳“TEST_NORMAL”，若錯誤則回傳“TEST_FUNC_ERROR”。

3. 注意事項

此程式屬於 Demo Code，使用者需依照實際應用來調整測試時間(interval & timing)及 IO 腳。

4. 版本历史

版本	描述	日期
V1.00	(1) 首次发布版本	2018/03/29

5. 免责声明

在此，笙泉（Megawin）代表“*Megawin Technology Co., Ltd.*”

生命支援—此产品并不是为医疗、救生或维持生命而设计的，并且当设备系统出现故障时，并不能合理地预示是

否会对人身造成伤害。因此，当客户使用或出售用于上述应用的产品时，需要客户自己承担这样做的风险，笙泉公司并不会对不当地使用或出售我公司的产品而造成的任何损害进行赔偿。

更改权—笙泉保留产品的如下更改权，其中包括电路、标准单元、与/或软件 – 在此为提高设计的与/或性能的描述或内容。当产品在大批量生产时，有关变动将通过工程变更通知（ECN）进行通知。