

三菱電機定位板 SSCNETIII/H 介面

MR-MC200/MR-MC300系列定位板 用戶手冊 (API庫篇)



- MR-MC210
- MR-MC211
- MR-MC220U3
- MR-MC220U6
- MR-MC240
- MR-MC241
- MR-MC240N
- MR-MC241N
- MR-MC341



安全注意事項


(使用之前請務必閱讀)

在使用本產品之前，請仔細閱讀本手冊以及本手冊中介紹的關聯手冊，同時在充分注意安全的前提下正確地操作。

本手冊中的注意事項僅記載了與本產品有關的內容。

在“安全注意事項”中，安全注意事項被分為“警告”和“注意”這二個等級。

 警告	表示錯誤操作可能造成災難性後果，引起死亡或重傷事故。
 注意	表示錯誤操作可能造成危險的後果，引起人員中等傷害或輕傷，還可能使設備損壞。

此外，注意根據情況不同，即使“注意”這一級別的事項也有可能引發嚴重後果。

對兩級注意事項都須遵照執行，因為它們對於操作人員安全是至關重要的。

請妥善保管本手冊以備需要時閱讀，並應將本手冊交給最終用戶。

[設計注意事項]

警告

- 應在個人電腦的外部設置安全電路，確保在外部電源異常及個人電腦、定位板本體故障時，整個系統始終都會安全運行。未在個人電腦的外部設置安全電路的情況下，誤輸出或誤動作可能導致事故。
 - (1) 應在定位板的外部組態緊急停止電路、保護電路、正轉/反轉等相反動作的互鎖電路、定位的上限/下限等防止機械損壞的互鎖電路。
- 應組態在定位板本體的電源啟動後再接通外部供應電源的電路。如果先啟動外部供應電源，誤輸出或誤動作可能引發事故。
- 關於網路通信異常時各站的動作狀態，請從三菱電機全球網站下載使用的設備的手冊並進行參考。誤輸出或誤動作可能引發事故。
- 將外部設備連接到定位板上，對運行中的定位板進行控制(資料更改)時，應在程式中組態互鎖，以確保整個系統始終都會安全運行。此外，對運行中的定位板進行其它控制(程式更改、參數更改、運行狀態更改(狀態控制))時，應仔細閱讀手冊並充分確認安全之後再進行操作。如果疏於確認，則操作錯誤可能導致機械損壞或事故。
- 從外部設備對遠端的定位板進行控制時，由於資料通信異常可能無法立即對定位板側的故障進行處理。應在程式中組態互鎖的同時，在外部設備與定位板之間確定發生資料通信異常時系統方面的處理方法。
- 在定位板的雙埠記憶體中，請勿對廠商設置用的區域進行資料寫入。此外，在從定位板對各模組的輸出信號中，請勿對禁止使用的信號進行輸出(ON)操作。如果對廠商設置用的區域進行資料寫入，或對禁止使用的信號進行輸出，有可能導致定位板系統誤動作。關於廠商設置用的區域、禁止使用的信號有關內容，請參閱各模組的用戶手冊及關聯手冊。

[設計注意事項]

警告

- 通信電纜斷線的情況下，線路將變得不穩定，可能導致多個站網路通信異常。應在程式中組態互鎖，以便即使發生通信異常也能保證系統安全運行。誤輸出或誤動作可能引發事故。
 - 應在個人電腦的外部設置安全電路，確保在外部電源異常及個人電腦、定位板本體故障時，整個系統始終都會安全運行。誤輸出或誤動作可能引發事故。
 - (1) 原點復位控制通過原點復位方向及原點復位速度這2個資料進行控制，通過近點狗ON開始減速。因此，如果原點復位方向設置錯誤，有可能不減速而繼續運行，因此應在定位板的外部組態防止機械損壞的互鎖電路。
 - (2) 定位板檢測出錯誤時，將進行通常的減速停止或急停止。參數應符合定位系統的規格。此外，原點復位用參數及定位資料應設置在參數的設置值以內。
 - (3) 由於定位板無法檢測的輸出電路的絕緣元件及電晶體等部件的故障，輸出有可能保持為ON狀態或OFF狀態，或變得不穩定。在可能引發重大事故的系統中，應組態監視輸出信號的電路。
 - 對於使用了定位板、模組、伺服驅動器、伺服馬達的具有安全標準(例如機器人等的安全通則等)的系統，應滿足安全標準。
 - 定位板、模組、伺服驅動器的異常時動作與作為系統的安全方向動作不相同的情況下，應在定位板・模組・伺服驅動器的外部構建相應防範電路。
-

[設計注意事項]

注意

- 請勿將控制線及通信電纜與主電路或動力線捆紮在一起，或使其相互靠得過近。電磁干擾可能導致誤動作。對於控制線及通信電纜，應該彼此相距100mm及以上。
 - 控制燈負載、加熱器、螺線管閥等的感應性負載時，輸出OFF → ON時有可能有較大電流(通常的10倍左右)流過，因此應使用額定電流留有餘裕的模組。
 - 定位板的電源OFF → ON或軟體重啟時，定位板變為系統啟動中狀態所需的時間根據系統組態、參數設置等而變動。在設計上應做到即使變為系統啟動中狀態所需的時間變動，也能確保整個系統安全運行。
 - 在登錄各種設置過程中，請勿進行定位板的電源OFF及軟體重啟。如果在登錄過程中進行定位板的電源OFF及軟體重啟，快閃記憶體內的資料內容將變得不穩定，需要將設置值重新設置到雙埠記憶體並重新登錄到快閃記憶體中。此外，還可能導致定位板的故障及誤動作。
-

[安全注意事項]

警告

- 對於經由網路的來自於外部設備的非法訪問、DoS攻擊、電腦病毒及其它網路攻擊，為了保護定位板及系統的安全(可用性、完整性、機密性)，應採取安裝防火牆及VPN、將殺毒軟體導入到電腦等的措施。
-

[安裝注意事項]

警告

- 在拆裝定位板時，必須先將系統使用的外部供應電源全部斷開後再進行操作。如果未全部斷開，有可能導致觸電、定位板的故障及誤動作。
-

[安裝注意事項]

注意

- 應在符合MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)中記載的一般規格的環境下使用定位板。在不符合一般規格的環境下使用定位板時，有可能導致觸電、火災、誤動作、產品損壞或性能變差。
 - 定位板應通過板固定螺栓可靠擰緊。應在規定的扭矩範圍內擰緊板固定螺栓。如果螺栓擰得過鬆，可能導致部件及配線脫落、短路或誤動作。如果螺栓擰得過緊，可能會損壞螺栓及模組而導致脫落、短路或誤動作。關於規定扭矩範圍，請參閱隨個人電腦本體附帶的使用說明書。
 - 通電中及電源斷開之後，定位板本體及散熱器可能會處於高溫狀態，因此應加以注意。
 - 請勿直接觸碰定位板或連接器的導電部分。應握住前面嵌板或印刷電路板的邊緣部分。如果直接觸碰，有可能導致定位板的故障及誤動作。
 - 請勿拆開或改造定位板。否則有可能導致故障、誤動作、人員傷害或火災。
 - 在接觸定位板之前，必須先接觸已接地的金屬等的導電性物體，釋放掉人體等所攜帶的靜電。如果不釋放掉靜電，有可能導致定位板的故障及誤動作。
 - 定位板應安裝到符合支持定位板的PCI/CompactPCI[®]/PCI Express[®]標準的個人電腦上。如果安裝到不符合標準的個人電腦上，有可能導致誤動作、故障。
 - 定位板應按照要安裝的個人電腦的板安裝方法，將其可靠地安裝到插槽上。如果定位板未正確安裝，有可能導致誤動作、故障或脫落。
 - 應注意在安裝定位板時，請勿與其它板接觸。
 - 應注意在安裝定位板時，請勿被實際安裝部件及周圍的部件等損傷。
 - 應在沒有靜電的地方操作定位板。否則有可能導致故障及誤動作。
 - 定位板應放入防靜電用袋中。進行保管的情況下或搬運時，必須將其放入防靜電用袋中。否則有可能導致故障及誤動作。
 - 請勿使定位板掉落或對其施加強烈衝擊。否則有可能導致故障及誤動作。
-

[配線注意事項]

警告

- 安裝或配線作業時，必須先將系統使用的外部供應電源全部斷開後再進行操作。如果未全部斷開，有可能導致觸電、定位板的故障及誤動作。
 - 在安裝或配線作業後，進行通電或運行的情況下，必須安裝定位板安裝設備的本體蓋板。如果未安裝本體蓋板，有可能導致觸電。
-

[配線注意事項]

注意

- 必須對FG端子、裝有定位板的控制器、伺服驅動器及伺服馬達採用可程式控制器專用接地(接地電阻小於或等於100Ω)。否則可能導致觸電或誤動作。此外，請勿與其他機器共用接地。
 - 應使用合適的壓裝端子，並按規定的扭矩擰緊。如果使用Y型壓裝端子，端子螺栓鬆動的情況下有可能導致脫落、故障。
 - 對定位板配線時，應確認產品的額定電壓及信號排列後正確地操作。如果連接了與額定不相符的電源或配線錯誤，有可能導致火災或故障。
 - 對於外部設備連接用連接器，應使用生產廠商指定的工具進行壓裝、壓接或正確地焊接。連接不良的情況下，有可能導致短路、火災或誤動作。
 - 連接器應可靠安裝到定位板上。否則接觸不良可能導致誤動作。
 - 請勿將控制線及通信電纜與主電路或動力線捆紮在一起，或使其相互靠得過近。否則雜訊可能導致誤動作。對於控制線及通信電纜，應該彼此相距100mm及以上。
 - 定位板上連接的電線及電纜必須納入導管中，或通過夾具進行固定處理。如果未將電線及電纜納入導管中或未通過夾具進行固定處理，由於電纜的晃動或移動、不經意的拉扯等可能導致誤動作或定位板及電纜的破損。
尤其是在振動、衝擊較大的場所中使用的情況下，電線及電纜的重量可能會給定位板帶來負載。
 - 連接電纜時，應在確認連接的介面類別型的基礎上，正確地操作。如果連接了不同類型的介面或者配線錯誤，有可能導致定位板或外部設備故障。
 - 應在規定的扭矩範圍內擰緊端子螺栓及連接器安裝螺栓。如果螺栓擰得過鬆，可能導致脫落、短路、火災或誤動作。如果螺栓擰得過緊，可能會損壞螺栓及定位板而導致脫落、短路、火災或誤動作。
 - 拆卸定位板上連接的電纜時，請勿拉扯電纜部分。對於帶有連接器的電纜，應握住定位板連接部分的連接器進行拆卸。對於端子排連接的電纜，應將端子排端子螺栓鬆開後進行拆卸。如果在與定位板相連接的狀態下拉扯電纜，有可能導致誤動作或定位板及電纜破損。
 - 應注意防止切屑及配線頭等異物掉入裝有定位板的個人電腦內。否則有可能導致火災、故障或誤動作。
 - 系統中使用的電纜應符合MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)記載的規格。不符合規格的配線時，將無法保證正常的資料傳送。
-

[啟動・維護注意事項]

警告

- 請勿在通電狀態下觸碰端子。如果觸碰端子，有可能導致觸電或誤動作。
 - 應正確連接電池連接器。請勿對電池進行充電、拆開、加熱、置入火中、短路、焊接、附著液體、強烈衝擊等動作。如果電池處理不當，由於發熱、破裂、著火、漏液可能導致人身傷害或火災。
 - 擰緊端子螺栓、連接器安裝螺栓或模組固定螺栓以及清潔模組時，必須先將系統使用的外部供應電源全部斷開後再進行操作。如果未全部斷開，有可能導致觸電。
-

[啟動・維護注意事項]

注意

- 將外部設備連接到定位板上或智慧功能模組上，對運行中的定位板進行控制(資料更改)時，應在程式中組態互鎖，以確保整個系統始終都會安全運行。此外，對運行中的定位板進行其它控制(程式更改、參數更改、強制輸出、運行狀態更改(狀態控制))時，應仔細閱讀手冊並充分確認安全之後再進行操作。如果疏於確認，則操作錯誤可能導致機械損壞或事故。
 - 從外部設備對遠端的定位板進行控制時，由於資料通信異常，可能無法立即對定位板側的故障進行處理。應在程式中組態互鎖的同時，在外部設備與定位板之間確定發生資料通信異常時系統方面的處理方法。
 - 請勿拆開或改造模組。如果進行模組的拆開或改造，有可能導致故障、誤動作、人員傷害或火災。
 - 在使用行動電話及PHS等無線通信設備時，應在全方向與定位板本體保持25cm及以上的距離。如果從定位板本體的全方向到無線通信設備為止的距離小於25cm，有可能導致誤動作。
 - 在拆裝模組時，必須先將系統使用的外部供應電源全部斷開後再進行操作。如果未全部斷開，有可能導致模組故障及誤動作。
 - 應在規定的扭矩範圍內擰緊螺栓。如果螺栓擰得過鬆，可能導致部件及配線脫落、短路或誤動作。如果螺栓擰得過緊，可能會損壞螺栓及模組而導致脫落、短路或誤動作。
 - 產品投入使用後，定位板的拆裝次數不應超過50次。如果超過了50次，有可能導致誤動作。
 - 執行控制盤內的啟動・維護作業時，應由在觸電保護方面受到過良好培訓的維護作業人員操作。此外，控制盤應配鎖，以便只有維護作業人員才能操作控制盤。
 - 在接觸定位板之前，必須先接觸已接地的金屬等的導電性物體，釋放掉人體等所攜帶的靜電。此外，建議佩戴已接地的防靜電腕帶。如果不釋放掉靜電，有可能導致模組故障及誤動作。
 - 對於定位板上附著的污垢，應用清潔且乾燥的布擦去。
 - 試運行時，應將參數的速度限制值設置為較慢的速度，做好發生危險狀態時能立即停止的準備之後再進行動作確認。
 - 運行前應進程式及各參數的確認・調整。根據機械情況可能發生意外動作。
-

[啟動・維護注意事項]

注意

- 在使用絕對位置系統功能的情況下，首次啟動時，或更換了定位板、絕對位置對應電機等時，必須進行原點復位。
 - 應確認制動功能之後再投入運行。
 - 點檢時請勿進行兆歐測試(絕緣電阻測定)。
 - 維護・點檢結束時，應確認絕對位置檢測功能的位置檢測是否正確。
 - 控制盤應配鎖，以便只有受過電氣設備相關培訓，具有充分知識的人員才能打開控制盤。
 - 定位板應放入防靜電用袋中。進行保管的情況下或搬運時，必須將其放入防靜電用袋中。否則有可能導致故障或誤動作。
 - 定位板中內置的微處理器在動作時處於高溫狀態，因此應注意在更換定位板等時，請勿直接用手觸碰。否則可能會導致燙傷。
-

[運行注意事項]

注意

- 將個人電腦等外部設備連接到智慧功能模組上對運行中的定位板進行控制(尤其是資料更改、程式更改、運行狀態更改(狀態控制))時，應在仔細閱讀用戶手冊並充分確認安全之後再進行操作。如果資料更改、程式更改、狀態控制錯誤，有可能導致系統誤動作、機械損壞及事故。
 - 將雙埠記憶體의設置值登錄到定位板內的快閃記憶體中使用的情況下，在登錄過程中請勿進行定位板的電源OFF及軟體重啟。如果在登錄過程中進行定位板的電源OFF及軟體重啟，快閃記憶體內的資料內容將變得不穩定，需要將設置值重新設置到雙埠記憶體並重新登錄到快閃記憶體中。此外，還可能導致定位板的故障及誤動作。
 - 插補運行的基準軸速度指定時，應注意物件軸(第2軸、第3軸、第4軸)的速度有可能大於設置速度(超過速度限制值)。
 - 試運行及示教等的運行過程中請勿靠近機械。靠近機械時，可能造成人員傷害。
-

[個人電腦連接注意事項]

⚠注意

- 系統中使用的乙太網路電纜應符合MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)記載的規格。不符合規格的配線時，將無法保證正常的資料傳送。
 - 連接個人電腦與配備USB介面的模組的情況下，應在按照個人電腦的使用說明書進行操作的同時，使用時還應遵守下述(1)、(2)的注意事項。如果使用時不遵守注意事項，則可能會導致模組故障。
 - (1) 以AC電源使用個人電腦的情況下
使用電源插頭為三芯或電源插頭上有接地線的個人電腦時，應使用帶接地的插座，或接地線必須接地。此外，對於個人電腦與模組，必須採用可程式控制器專用接地(接地電阻小於或等於100Ω)。
使用電源插頭為二芯且無接地線的個人電腦時，應按照下述1.~3.的步驟連接個人電腦與模組。
此外，對於個人電腦與模組，建議從同一電源系統供電。
 1. 應從AC插座上拔下個人電腦的電源插頭。
 2. 應在確認個人電腦的電源插頭已從AC插座上拔下的基礎上，連接USB電纜。
 3. 應將個人電腦的電源插頭插入到AC插座上。
 - (2) 以電池驅動使用個人電腦的情況下
可以原樣不變地使用。
-

[廢棄注意事項]

⚠注意

- 產品廢棄時，應將其作為工業廢棄物處理。
-

[運輸注意事項]

⚠注意

- 如果木製包裝材料的消毒及防蟲措施的薰蒸劑中包含的鹵素物質(氟、氯、溴、碘等)進入到三菱電機產品中有可能導致故障。應注意防止殘留的薰蒸成分進入到三菱電機產品，或採用薰蒸以外的方法(熱處理等)進行處理。此外，消毒及防蟲措施應在包裝前的木材階段實施。
 - 定位板是精密機械，因此請勿使其掉落或對其施加強烈衝擊。
-

前言

在此感謝您選擇三菱電機定位板產品。

本手冊是用於讓用戶瞭解進行程式時所需的API函數、結構有關內容的手冊。

在使用之前應熟讀本手冊及關聯手冊，在充分瞭解定位板的功能・性能的基礎上正確地使用本產品。

應將本手冊交給最終用戶。

物件板

MR-MC210、MR-MC211、MR-MC220U3、MR-MC220U6、MR-MC240、MR-MC241、MR-MC240N、MR-MC241N、MR-MC341

要點

本手冊中使用的符號如下所示。

- [MC200]：表示僅對應於MR-MC210/MR-MC211/MR-MC220U3/MR-MC220U6/MR-MC240/MR-MC241/MR-MC240N/MR-MC241N的符號
 - [MC300]：表示僅對應於MR-MC341的符號
-

目錄

安全注意事項	1
前言	9
關聯手冊	15
術語	16
總稱/簡稱	16
第1章 概要	17
第2章 使用條件	19
第3章 API函數一覽	22
第4章 API函數詳細	28
4.1 輔助函數群	28
sscGetLastError	28
sscGetMountChannel	29
4.2 軟元件函數群	30
sscOpen	31
sscClose	32
4.3 資訊函數群	33
sscGetControlCycle	33
sscGetBoardVersion	34
sscGetDriverVersion	35
sscGetOperationCycleMonitor	36
sscClearOperationCycleMonitor	37
sscGetBoardSerialNumber [MC300]	38
4.4 參數函數群	39
sscResetAllParameter	40
sscChangeParameter	41
sscChange2Parameter	43
sscCheckParameter	45
sscCheck2Parameter	47
sscLoadAllParameterFromFlashROM	49
sscSaveAllParameterToFlashROM	50
sscCheckSvPrmChangeNumEx	51
sscChangeParameterEx	52
sscChange2ParameterEx	54
sscCheckParameterEx	56
sscCheck2ParameterEx	58
4.5 系統函數群	60
sscReboot	61
sscSystemStart	62
sscGetSystemStatusCode	63
sscReconnectSSCNET	64
sscReconnectSSCNETEx [MC300]	65
sscDisconnectSSCNET	66
sscDisconnectSSCNETEx [MC300]	67

	sscGetControllingAxis [MC300].	68
4.6	陳述式/狀態函數群.	69
	sscSetCommandBitSignalEx	70
	sscGetStatusBitSignalEx.	71
	sscWaitStatusBitSignalEx	72
4.7	點位元表函數群	73
	sscSetPointDataEx.	73
	sscCheckPointDataEx.	74
	sscSetPointOffset.	75
	sscCheckPointOffset.	76
	sscGetDrivingPointNumber	77
	sscSetLatestPointNumber.	78
4.8	擋塊控制資料函數群	79
	sscSetPressData.	79
	sscGetPressData.	80
4.9	運行函數群	81
	sscJogStart.	81
	sscJogStop	83
	sscJogStopNoWait	84
	sscIncStart.	85
	sscAutoStart	87
	sscHomeReturnStart	88
	sscLinearStart	89
	sscInterpolationStart [MC300].	91
	sscDataSetStart.	93
	sscDriveStop	94
	sscDriveStopNoWait	95
	sscDriveRapidStop.	96
	sscDriveRapidStopNoWait.	97
	sscSetDriveMode.	98
	sscGetDriveMode.	100
	sscGetDriveFinStatus	101
	sscChangeControlMode	103
4.10	更改函數群	104
	sscChangeManualPosition.	104
	sscChangeAutoPosition.	105
	sscChangeLinearPosition.	106
	sscChangeManualSpeed	108
	sscChangeAutoSpeed	109
	sscChangeManualAccTime	110
	sscChangeAutoAccTime	111
	sscChangeManualDecTime	112
	sscChangeAutoDecTime	113
4.11	報警函數群	114
	sscGetAlarm.	114
	sscResetAlarm.	115
4.12	通用監視函數群	116
	sscSetMonitor.	118
	sscStopMonitor	119
	sscGetMonitor.	120
4.13	高速監視函數群	121

	sscGetCurrentCmdPositionFast	121
	sscGetCurrentFbPositionFast	122
	sscGetIoStatusFast	123
	sscGetCurrentCmdSpeedFast	124
	sscGetCurrentFbSpeedFast	125
	sscGetCurrentFbFast	126
	sscGetPositionDroopFast	127
4.14	用戶看門狗函數群	128
	sscWdEnable	128
	sscWdDisable	129
	sscChangeWdCounter	130
4.15	其他軸啟動函數群	131
	sscSetOtherAxisStartData	131
	sscGetOtherAxisStartData	132
	sscOtherAxisStartAbortOn	133
	sscOtherAxisStartAbortOff	134
	sscGetOtherAxisStartStatus	135
4.16	通過位置中斷函數群	136
	sscSetIntPassPositionData	136
	sscCheckIntPassPositionData	137
	sscSetStartingPassNumber	138
	sscGetExecutingPassNumber	139
4.17	採樣函數群	140
	sscStartSampling	140
	sscStopSampling	141
	sscSetSamplingParameter	142
	sscGetSamplingParameter	143
	sscGetSamplingError	144
	sscGetSamplingStatus	145
	sscGetSamplingData	146
4.18	日誌函數群	147
	sscStartLog	147
	sscStopLog	148
	sscCheckLogStatus	149
	sscCheckLogEventNum	150
	sscReadLogData	151
	sscClearLogData	152
	sscGetAlarmHistoryData	153
	sscCheckAlarmHistoryEventNum	154
	sscClearAlarmHistoryData	155
4.19	數位輸入輸出函數群	156
	sscGetDigitalInputDataBit	156
	sscGetDigitalInputDataWord	157
	sscSetDigitalOutputDataBit	158
	sscSetDigitalOutputDataWord	159
	sscGetDigitalOutputDataBit	160
	sscGetDigitalOutputDataWord	161
4.20	標記檢測函數群	162
	sscGetMarkDetectionData	162
	sscGetMarkDetectionCounter	164
	sscClearMarkDetectionData	165

4.21	介面模式函數群	166
	sscIfmGetReadErrorCount	166
	sscIfmSetHomePosition	167
	sscIfmGetMaximumBufferNumber	168
	sscIfmGetMaximumBufferNumberEx	169
	sscIfmRenewLatestBuffer	170
	sscIfmRenewLatestBufferEx	171
	sscIfmCheckLatestBuffer	172
	sscIfmCheckLatestBufferEx	173
	sscIfmGetTransmitBuffer	174
	sscIfmGetTransmitBufferEx	175
	sscIfmTrqSetSpeedLimit	176
	sscIfmSetControlMode	177
	sscIfmGetControlMode	178
	sscIfmGetEventStatusBits	179
	sscIfmGetEventStatusBitsEx [MC300]	180
4.22	中斷函數群	181
	sscIntStart	183
	sscIntEnd	184
	sscIntEnable	185
	sscIntDisable	186
	sscRegisterIntCallback	187
	sscUnregisterIntCallback	188
	sscResetIntEvent	189
	sscSetIntEvent	190
	sscWaitIntEvent	191
	sscResetIntEventMulti	193
	sscSetIntEventMulti	194
	sscWaitIntEventMulti	195
	sscResetIntOasEvent	197
	sscSetIntOasEvent	198
	sscWaitIntOasEvent	199
	sscResetIntPassPosition	201
	sscSetIntPassPosition	202
	sscWaitIntPassPosition	203
	sscResetIntDriveFin	204
	sscSetIntDriveFin	205
	sscWaitIntDriveFin	206
4.23	輸入輸出軟元件函數群	208
	sscGetInputDeviceBit	208
	sscGetInputDeviceWord	209
	sscSetOutputDeviceBit	210
	sscSetOutputDeviceWord	211
	sscGetOutputDeviceBit	212
	sscGetOutputDeviceWord	213
4.24	暫態傳送函數群	214
	sscSendReceiveTransientData	214

第5章 結構一覽 215

5.1	PNT_DATA_EX結構	215
-----	---------------	-----

PNT_DATA_EX結構[MC200]	215
PNT_DATA_EX結構[MC300]	217
5.2 OAS_DATA結構	219
OAS_DATA結構[MC200]	219
OAS_DATA結構[MC300]	221
5.3 PRESS_DATA結構	223
5.4 SMP_ERR結構	224
SMP_ERR結構[MC200]	224
SMP_ERR結構[MC300]	225
5.5 SMP_DATA結構	226
SMP_DATA結構[MC200]	226
SMP_DATA結構[MC300]	227
5.6 LOG_DATA結構	228
5.7 ALH_DATA結構	229
ALH_DATA結構[MC200]	229
ALH_DATA結構[MC300]	230
5.8 TRANSIENT_CMD結構	231
5.9 TRANSIENT_STS結構	232
5.10 INT_CB_DATA結構	233
INT_CB_DATA結構[MC200]	233
INT_CB_DATA結構[MC300]	235
5.11 SLAVE_INFO結構[MC300]	237
第6章 位定義一覽	238
<hr/>	
6.1 系統陳述式位	238
6.2 系統狀態位元	248
6.3 軸陳述式位	258
6.4 軸狀態位元	268
6.5 站陳述式位	278
6.6 站狀態位元	282
第7章 中斷事件原因一覽	286
<hr/>	
第8章 詳細出錯代碼一覽	289
<hr/>	
附錄	295
<hr/>	
附1 MR-MC2_ 與MR-MC3_ 的不同點	295
API庫的不同點	295
MR-MC3_ 用API庫中添加的API函數	295
從MR-MC2_ 到MR-MC3_ 的API函數替換	296
索引	298
<hr/>	
修訂記錄	301
保固	303
商標	304

關聯手冊

與本產品相關的手冊如下所示。

關於最新的手冊，請向當地三菱電機代理店諮詢。

手冊名稱[手冊編號]	內容
MR-MC200/MR-MC300系列定位板用戶手冊 (API庫篇) [IB-0300305CHT] (本手冊)	說明了用於從主機個人電腦控制定位板的庫函數群等有關內容。
MR-MC200/MR-MC300系列定位板用戶手冊 (詳細篇) [IB-0300303CHT]	說明了定位板的規格、系統構建所需的知識、維護點檢、故障排除、與定位板的定位控制相關的功能、程式、雙埠記憶體等有關內容。

術語

在本手冊中，除了特別標明的情況外，將使用下述術語進行說明。

術語	內容
API版本	是定位板用API庫的軟體版本。
Board版本	是定位板的系統版本。
MR Configurator2	是伺服設置軟體的產品名。
MR-J3(W)-_B	是MR-J3-_B_(-RJ)/MR-J3W-_B型伺服驅動器。
MR-J4(W)-_B	是MR-J4-_B_(-RJ)/MR-J4W-_B型伺服驅動器。
MR-J5(W)-_B	是MR-J5-_B_(-RJ)/MR-J5W-_B型伺服驅動器。
MR-MC2_ _	是支持PCI匯流排的定位板MR-MC210/MR-MC211、支持CompactPCI匯流排的定位板MR-MC220U3/MR-MC220U6、支持PCI Express匯流排的定位板MR-MC240/MR-MC241/MR-MC240N/MR-MC241N的別稱。
MR-MC3_ _	是支持PCI Express匯流排的定位板MR-MC341的別稱。
SSCNETⅢ*1	是定位板 ↔ 伺服驅動器之間高速同步網路。
SSCNETⅢ/H*1	
智慧功能模組	是A/D、D/A轉換模組等，具有輸入輸出以外的功能的模組。
站編號	是定位板上的站編號。
系統程式	是控制定位板的內部程式。
控制週期	是定位板進行陳述式的獲取、位置控制、狀態的輸出、與伺服驅動器的通信等的週期。
暫態傳送	是在網路的節點(站)之間非定期進行資料通信的功能。
緩衝記憶體	是用於儲存設置值、監視值等的資料的定位板及智慧功能模組的記憶體。
用戶程式	是在主機個人電腦上運行的用戶創建的程式。
遠端暫存器(RWr)	是從傳感模組、SSCNETⅢ/H起始模組向定位板以16位(1字)為單位輸入的資訊。
遠端暫存器(RWw)	是從定位板向傳感模組、SSCNETⅢ/H起始模組以16位(1字)為單位輸出的資訊。
遠端輸入(RX)	是從傳感模組、SSCNETⅢ/H起始模組向定位板以位元單位輸入的資訊。
遠端輸出(RY)	是從定位板向傳感模組、SSCNETⅢ/H起始模組以位元單位輸出的資訊。
連結軟元件	是定位板、傳感模組、SSCNETⅢ/H起始模組的內部軟元件(RX/RY/RWr/RWw)。

*1 SSCNET: Servo System Controller NETwork

總稱/簡稱

在本手冊中，除了特別標明的情況外，將使用下述總稱/簡稱進行說明。

總稱/簡稱	內容
雙埠記憶體	是在緩衝記憶體內，尤其是為了進行運動控制而進行通信的區域的總稱。
API庫	是用於從主機個人電腦控制定位板的定位控制用庫函數群的總稱。
SSCNETⅢ/H起始模組	是MELSEC-L系列SSCNETⅢ/H起始模組(LJ72MS15)的簡稱。
伺服驅動器	是SSCNETⅢ/H對應伺服驅動器的簡稱。
傳感模組	是SSCNETⅢ/H對應傳感模組MR-MT2000系列的總稱。
通道(CH)	是SSCNET控制通道的簡稱。
測試工具	是定位板用的啟動・調查輔助工具的簡稱。
輸入輸出軟元件	是雙埠記憶體上的連結軟元件的總稱。
定位板	是MR-MC2_ _與MR-MC3_ _的總稱。
主機個人電腦	是安裝定位板，用戶程式運行的個人電腦的總稱。
實用軟體	是包括定位板API庫及定位板測試工具、定位板軟元件驅動器在內的定位板實用程式2(MR2JW3-MC2-UTL)的總稱。
遠端I/O模組	是用於將輸入輸出模組及智慧功能模組連接到SSCNETⅢ/H上的模組的總稱，包括傳感模組、SSCNETⅢ/H起始模組。

1 概要

本API庫是用於創建控制下述三菱電機生產定位板的主機個人電腦上的應用程式的API函數群。

- 支持PCI匯流排的定位板 (MR-MC210/MR-MC211)
- 支持CompactPCI匯流排的定位板 (MR-MC220U3/MR-MC220U6)
- 支持PCI Express匯流排的定位板 (MR-MC240/MR-MC241/MR-MC240N/MR-MC241N/MR-MC341)

通過使用本API函數，可以進行與定位板的通信的打開、關閉以及與伺服驅動器的通信初始化、參數的更改、各運行模式下的運行的啟動、監視等。


使用本API庫的情況下，最多可以同時使用4塊定位板。

根據軟體版本的功能限制

根據本API庫的軟體版本，可使用的功能有所限制。

一：無版本的限制。

功能・項目名稱	MR-MC2_ _		MR-MC3_ _		更改內容
	API版本	Board版本	API版本	Board版本	
數位輸入輸出	1.02	A1	—	—	函數添加
其他軸啟動・數位輸出信號控制	1.02	A1	—	—	添加結構成員
通過位置中斷	1.02	A1	—	—	函數添加
中斷回檔	1.02	—	—	—	函數添加
中斷事件通知	1.02	—	—	—	函數添加
支持64位OS	1.10	—	—	—	添加支持OS
報警履歷功能	1.40	A3	—	—	函數添加
介面模式	1.40	A3	—	—	函數添加
支援速度・轉矩控制 (僅限介面模式時)	1.60	A4	—	—	函數添加
在高速監視中添加位置下降 (僅限介面模式時)	1.60	A4	—	—	函數添加
支援標記檢測功能	1.60	A5	—	—	函數添加
支援擋塊運行控制 (僅限標準模式時的自動運行)	1.60	A5	—	—	函數添加
支持Windows®8	1.60	—	—	—	添加支持OS
支持定位板MR-MC240N/MR-MC241N	1.60	B6	不支持	不支持	支持機型添加
支持點位表循環方式	1.80	A6	—	—	函數添加
支持遠端I/O模組	1.80	A8	—	—	擴展軸編號
支援輸入輸出軟元件	1.80	A8	—	—	函數添加
支持暫態傳送	1.80	A8	—	—	函數添加
支援插補組可變	1.90	A9	—	—	函數更改
支持傳感模組	1.90	B1	—	—	—
支持Windows®10	2.00	—	—	—	添加支持OS
支持定位板MR-MC341	不支持	不支持	—	—	支持機型添加
支持圓弧插補	不支持	不支持	1.10	A1	添加結構成員
支持MR-J5(W)-_B	2.40	B7	1.30	A6	函數添加
支持Windows®11	2.40	—	1.40	—	添加支持OS
Windows®XP、Windows®Vista不支持	2.40	—	—	—	刪除支持OS
Windows®7、Windows®8不支持	2.40	—	1.40	—	刪除支持OS

-
- 對於API版本，可以通過下述檔案的屬性對話方塊確認檔案版本。
MR-MC2_ _: mc2xxstd.dll或mc2xxstd_x64.dll
MR-MC3_ _: mc3xxstd.dll或mc3xxstd_x64.dll
〈例〉API版本1.00的情況下，檔案版本將被顯示為“1.0.0.0”。
 - 關於Board版本的確認方法，請參閱下述手冊的“製造編號與軟體版本的確認方法”。
MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)
-

2 使用條件

使用本API函數時的條件如下所示。

- 本API函數的前提條件是在如下所示的OS上運行的編譯器中使用。

定位板	OS	編譯器
MR-MC2_ _	<ul style="list-style-type: none"> • Windows[®]10 (32位/64位) • Windows[®]11 Enterprise (64位) • Windows[®]11 Pro (64位) • Windows[®]11 IoT Enterprise (64位) 	<ul style="list-style-type: none"> • Microsoft[®] Visual C++[®] 2022/2019/2017/2015 • Microsoft[®] Visual C#[®] 2022/2019/2017/2015 • Microsoft[®] Visual Basic[®] 2022/2019/2017/2015
MR-MC3_ _	<ul style="list-style-type: none"> • Windows[®]10 (32位/64位) • Windows[®]11 Enterprise (64位) • Windows[®]11 Pro (64位) • Windows[®]11 IoT Enterprise (64位) 	<ul style="list-style-type: none"> • Microsoft[®] Visual C++[®] 2022/2019/2017/2015 • Microsoft[®] Visual C#[®] 2022/2019/2017/2015

- 本API函數提供以下庫與頭檔案。

定位板	支持32位OS的庫	支持64位OS的庫	頭檔案
MR-MC2_ _	<ul style="list-style-type: none"> • mc2xxstd.dll • mc2xxstd.lib (COFF格式) 	<ul style="list-style-type: none"> • mc2xxstd_x64.dll • mc2xxstd_x64.lib (COFF格式) 	<ul style="list-style-type: none"> • mc2xxstd.h (C/C++用頭檔案) • mc2xxstd.cs (C#用頭檔案) • mc2xxstd.vb (支持32位OS的VB用頭檔案) • mc2xxstd_x64.vb (支持64位OS的VB用頭檔案)
MR-MC3_ _	<ul style="list-style-type: none"> • mc3xxstd.dll • mc3xxstd.lib (COFF格式) 	<ul style="list-style-type: none"> • mc3xxstd_x64.dll • mc3xxstd_x64.lib (COFF格式) 	<ul style="list-style-type: none"> • mc3xxstd.h (C/C++用頭檔案) • mc3xxstd.cs (C#用頭檔案)

- 本API函數支援下述內容。

定位板	匯流排規格	型號
MR-MC2_ _	支持PCI匯流排的定位板	<ul style="list-style-type: none"> • MR-MC210 • MR-MC211
	支持CompactPCI匯流排的定位板	<ul style="list-style-type: none"> • MR-MC220U3 • MR-MC220U6
	支持PCI Express匯流排的定位板	<ul style="list-style-type: none"> • MR-MC240 • MR-MC241 • MR-MC240N • MR-MC241N
MR-MC3_ _	支持PCI Express匯流排的定位板	MR-MC341

- 在Visual C#或Visual Basic的工程中使用本API庫的情況下，應執行下述操作。
 - 應在與使用API庫的用戶程式(.exe)相同的檔案夾中儲存API庫(.dll)。(由於本API庫不是.NET Framework類庫或COM組件，因此無法將對dll的引用添加到工程中。)
 - 應將API庫用頭檔案添加到工程中。
 - Visual C#工程的情況下，應在使用前調用LoadLibraryDll函數後再載入庫。此外，應在使用後調用FreeLibraryDll函數後再卸載。關於詳細內容，請參閱C#用的API示例。
- 可同時打開的用戶程式最多為1個。請勿通過2個及以上用戶程式打開相同的定位板。

要點

- 對於本API函式呼叫規定，應使用__stdcall。
- 不支援以下OS與庫的組合。
 - 32位OS + 支持64位OS的庫(mc2xxstd_x64.dll、mc3xxstd_x64.dll)
 - 64位OS + 支持32位OS的庫(mc2xxstd.dll、mc3xxstd.dll)
- 裝入用戶裝置時，請對用戶程式進行充分評估。
- 無法在1個用戶程式中同時使用MR-MC2_ _與MR-MC3_ _的庫。

Visual C#工程

本API通過mc2xxstd [MC200]/mc3xxstd [MC300]命名空間的SscApi類被定義。

在Visual C#的工程中使用本API庫的情況下，應執行下述操作。

- 應按照下表所示替換讀取各API的參數及結構成員的類型。

C++的API的參數類型		C#的API的參數類型
unsigned char		byte
unsigned char*	in	ref byte
	out	out byte
unsigned char* (陣列的起始指標)		byte[]
char		sbyte
char*	in	ref sbyte
	out	out sbyte
char* (陣列的起始指標)		sbyte[]
unsigned short		ushort
unsigned short*	in	ref ushort
	out	out ushort
unsigned short* (陣列的起始指標)		ushort[]
short		short
short*	in	ref short
	out	out short
short* (陣列的起始指標)		short[]
unsigned int		uint
unsigned int*	in	ref uint
	out	out uint
unsigned int* (陣列的起始指標)		uint[]
int		int
int*	in	ref int
	out	out int
int* (陣列的起始指標)		int[]
unsigned long		ulong
unsigned long*	in	ref ulong
	out	out ulong
unsigned long* (陣列的起始指標)		ulong[]
long		int
long*	in	ref int
	out	out int
long* (陣列的起始指標)		int[]
unsigned long long		ulong
unsigned long long*	in	ref ulong
	out	out ulong
unsigned long long* (陣列的起始指標)		ulong[]
long long		long
long long*	in	ref long
	out	out long
long long* (陣列的起始指標)		long[]
wchar_t陣列		string

- 各API的參數為結構指標/結構陣列的情況下，應按下表所示替換讀取參數類型。

C++的API的參數類型	C#的API的參數類型	
結構指標 例: PNT_DATA_EX*	in	<pre>ref PNT_DATA_EX 使用示例: PNT_DATA_EX pnt_data_ex = new PNT_DATA_EX(); pnt_data_ex.Initialize(); SscApi.sscSetPointDataEx(board_id, channel, axnum, pntnum, ref pnt_data_ex);</pre>
	out	<pre>out PNT_DATA_EX 使用示例: PNT_DATA_EX pnt_data_ex = new PNT_DATA_EX(); pnt_data_ex.Initialize(); SscApi.sscSetPointDataEx(board_id, channel, axnum, pntnum, out pnt_data_ex);</pre>
結構陣列 例: SMP_DATA* (陣列的起始指標)	<pre>SMP_DATA[] 使用示例: SMP_DATA[] smp_data = new SMP_DATA[128]; for (int i = 0; i < 128; i++) { smp_data[i].Initialize(); } SscApi.sscGetSamplingData(board_id, channel, page_num, out valid_num, smp_data);</pre>	

3 API函數一覽

API函數一覽如下所示。

○：可以使用，△：替換推薦函數，×：不能使用

分類	函數名稱	內容	可否使用		參照
			MR-MC2_ _	MR-MC3_ _	
輔助函數群	sscGetLastError	獲取詳細出錯代碼。	○	○	28頁 sscGetLastError
	sscGetMountChannel	獲取安裝通道資訊。	○	○	29頁 sscGetMountChannel
軟元件函數群	sscOpen	打開記憶體訪問埠。	○	○	31頁 sscOpen
	sscClose	關閉記憶體訪問埠。	○	○	32頁 sscClose
資訊函數群	sscGetControlCycle	獲取控制週期狀態。	○	○	33頁 sscGetControlCycle
	sscGetBoardVersion	獲取定位板的系統版本資訊。	○	○	34頁 sscGetBoardVersion
	sscGetDriverVersion	獲取驅動器版本資訊。	○	○	35頁 sscGetDriverVersion
	sscGetOperationCycleMonitor	獲取運算週期監視資料。	○	○	36頁 sscGetOperationCycleMonitor
	sscClearOperationCycleMonitor	清除(初始化)運算週期監視資料。	○	○	37頁 sscClearOperationCycleMonitor
	sscGetBoardSerialNumber	獲取定位板的製造編號。	×	○	38頁 sscGetBoardSerialNumber [MC300]
參數函數群	sscResetAllParameter	在系統啟動前設置所有參數初始值。	○	○	40頁 sscResetAllParameter
	sscChangeParameter	寫入參數。 (支援系統參數、伺服參數(MR-J4(W)-_B/ MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入)	○	○	41頁 sscChangeParameter
	sscChange2Parameter	寫入參數(2點)。 (支援系統參數、伺服參數(MR-J4(W)-_B/ MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入)	○	○	43頁 sscChange2Parameter
	sscCheckParameter	讀取參數設置值。 (支援系統參數、伺服參數(MR-J4(W)-_B/ MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取)	○	○	45頁 sscCheckParameter
	sscCheck2Parameter	讀取參數設置值(2點)。 (支援系統參數、伺服參數(MR-J4(W)-_B/ MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取)	○	○	47頁 sscCheck2Parameter
	sscLoadAllParameterFromFlashROM	在系統啟動前從快閃記憶體載入所有參數。	○	○	49頁 sscLoadAllParameterFromFlashROM
	sscSaveAllParameterToFlashROM	在系統啟動前將所有參數儲存到快閃記憶體。	○	○	50頁 sscSaveAllParameterToFlashROM
	sscCheckSvPrmChangeNumEx	獲取伺服參數更改編號。	○	○	51頁 sscCheckSvPrmChangeNumEx
	sscChangeParameterEx	寫入參數。 (支援系統參數、伺服參數(MR-J5(W)-_B/ MR-J4(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入)	○	○	52頁 sscChangeParameterEx

分類	函數名稱	內容	可否使用		參照
			MR- MC2_ _	MR- MC3_ _	
參數函數群	sscChange2ParameterEx	寫入參數(2點)。 (支援系統參數、伺服參數(MR-J5(W_)_B/ MR-J4(W_)_B、傳感模組)、控制參數、RIO 模組參數、RIO控制參數的寫入)	○	○	54頁 sscChange2ParameterEx
	sscCheckParameterEx	讀取參數設置值。 (支援系統參數、伺服參數(MR-J5(W_)_B/ MR-J4(W_)_B、傳感模組)、控制參數、RIO 模組參數、RIO控制參數的讀取)	○	○	56頁 sscCheckParameterEx
	sscCheck2ParameterEx	讀取參數設置值(2點)。 (支援系統參數、伺服參數(MR-J5(W_)_B/ MR-J4(W_)_B、傳感模組)、控制參數、RIO 模組參數、RIO控制參數的讀取)	○	○	58頁 sscCheck2ParameterEx
系統函數群	sscReboot	重啟系統。	○	○	61頁 sscReboot
	sscSystemStart	啟動系統。	○	○	62頁 sscSystemStart
	sscGetSystemStatusCode	獲取系統狀態碼。	○	○	63頁 sscGetSystemStatusCode
	sscReconnectSSCNET	重新連接SSCNET通信。	○	△	64頁 sscReconnectSSCNET
	sscReconnectSSCNETEx		×	○	65頁 sscReconnectSSCNETEx [MC300]
	sscDisconnectSSCNET	斷開SSCNET通信。	○	△	66頁 sscDisconnectSSCNET
	sscDisconnectSSCNETEx		×	○	67頁 sscDisconnectSSCNETEx [MC300]
	sscGetControllingAxis	獲取控制中軸資訊及控制中站資訊。	×	○	68頁 sscGetControllingAxis [MC300]
陳述式/狀態 函數群	sscSetCommandBitSignalEx	任意設置陳述式位。	○	○	70頁 sscSetCommandBitSignalEx
	sscGetStatusBitSignalEx	任意獲取狀態位元。	○	○	71頁 sscGetStatusBitSignalEx
	sscWaitStatusBitSignalEx	等待直至指定的狀態位元變為ON/OFF。	○	○	72頁 sscWaitStatusBitSignalEx
點位元表函數 群	sscSetPointDataEx	設置點資料。	○	○	73頁 sscSetPointDataEx
	sscCheckPointDataEx	獲取點資料。	○	○	74頁 sscCheckPointDataEx
	sscSetPointOffset	設置點編號偏置。	○	○	75頁 sscSetPointOffset
	sscCheckPointOffset	獲取點編號偏置。	○	○	76頁 sscCheckPointOffset
	sscGetDrivingPointNumber	獲取運行中的點編號。	○	○	77頁 sscGetDrivingPointNumber
	sscSetLatestPointNumber	設置最新陳述式點編號。	○	○	78頁 sscSetLatestPointNumber
擋塊控制資料 函數群	sscSetPressData	設置擋塊控制資料。	○	○	79頁 sscSetPressData
	sscGetPressData	獲取擋塊控制資料。	○	○	80頁 sscGetPressData

分類	函數名稱	內容	可否使用		參照
			MR- MC2_ _	MR- MC3_ _	
運行函數群	sscJogStart	啟動JOG運行。	○	○	81頁 sscJogStart
	sscJogStop	停止JOG運行。	○	○	83頁 sscJogStop
	sscJogStopNoWait	停止JOG運行。(非等待函數)	○	○	84頁 sscJogStopNoWait
	sscIncStart	啟動增量進給。	○	○	85頁 sscIncStart
	sscAutoStart	啟動自動運行。	○	○	87頁 sscAutoStart
	sscHomeReturnStart	啟動原點復位。	○	○	88頁 sscHomeReturnStart
	sscLinearStart	啟動直線插補。	○	△	89頁 sscLinearStart
	sscInterpolationStart	啟動插補運行。	×	○	91頁 sscInterpolationStart [MC300]
	sscDataSetStart	啟動原點重新設置(資料設置)。	○	○	93頁 sscDataSetStart
	sscDriveStop	停止運行。	○	○	94頁 sscDriveStop
	sscDriveStopNoWait	停止運行。(非等待函數)	○	○	95頁 sscDriveStopNoWait
	sscDriveRapidStop	緊急停止運行。	○	○	96頁 sscDriveRapidStop
	sscDriveRapidStopNoWait	緊急停止運行。(非等待函數)	○	○	97頁 sscDriveRapidStopNoWait
	sscSetDriveMode	更改運行模式。	○	○	98頁 sscSetDriveMode
	sscGetDriveMode	獲取運行模式。	○	○	100頁 sscGetDriveMode
	sscGetDriveFinStatus	獲取運行完成狀態。	○	○	101頁 sscGetDriveFinStatus
sscChangeControlMode	切換伺服驅動器的控制模式。	○	○	103頁 sscChangeControlMode	
更改函數群	sscChangeManualPosition	增量進給時更改位置。	○	○	104頁 sscChangeManualPosition
	sscChangeAutoPosition	自動運行時更改位置。	○	○	105頁 sscChangeAutoPosition
	sscChangeLinearPosition	直線插補時更改位置。	○	○	106頁 sscChangeLinearPosition
	sscChangeManualSpeed	更改JOG運行或增量進給時的速度。	○	○	108頁 sscChangeManualSpeed
	sscChangeAutoSpeed	更改自動運行或直線插補時[MC200]/插補運行時[MC300]的速度。	○	○	109頁 sscChangeAutoSpeed
	sscChangeManualAccTime	更改JOG運行或增量進給時的加速時間常數。	○	○	110頁 sscChangeManualAccTime
	sscChangeAutoAccTime	更改自動運行或直線插補時[MC200]/插補運行時[MC300]的加速時間常數。	○	○	111頁 sscChangeAutoAccTime
	sscChangeManualDecTime	更改JOG運行或增量進給時的減速時間常數。	○	○	112頁 sscChangeManualDecTime
	sscChangeAutoDecTime	更改自動運行或直線插補時[MC200]/插補運行時[MC300]的減速時間常數。	○	○	113頁 sscChangeAutoDecTime
報警函數群	sscGetAlarm	獲取報警編號及報警詳細編號。	○	○	114頁 sscGetAlarm
	sscResetAlarm	復位報警。	○	○	115頁 sscResetAlarm
通用監視函數群	sscSetMonitor	開始監視。	○	○	118頁 sscSetMonitor
	sscStopMonitor	停止監視。	○	○	119頁 sscStopMonitor
	sscGetMonitor	獲取監視資料。	○	○	120頁 sscGetMonitor

分類	函數名稱	內容	可否使用		參照
			MR- MC2_ _	MR- MC3_ _	
高速監視函數群	sscGetCurrentCmdPositionFast	獲取當前位置。(高速監視功能)	○	○	121頁 sscGetCurrentCmdPositionFast
	sscGetCurrentFbPositionFast	獲取F/B位置。(高速監視功能)	○	○	122頁 sscGetCurrentFbPositionFast
	sscGetIoStatusFast	獲取外部信號狀態。(高速監視功能)	○	○	123頁 sscGetIoStatusFast
	sscGetCmdSpeedFast	獲取移動速度。(高速監視功能)	○	○	124頁 sscGetCmdSpeedFast
	sscGetFbSpeedFast	獲取F/B移動速度。(高速監視功能)	○	○	125頁 sscGetFbSpeedFast
	sscGetCurrentFbFast	獲取電流回饋。(高速監視功能)	○	○	126頁 sscGetCurrentFbFast
	sscGetPositionDroopFast	獲取位置下降。(高速監視功能)	○	○	127頁 sscGetPositionDroopFast
用戶看門狗函數群	sscWdEnable	啟用用戶看門狗功能。	○	○	128頁 sscWdEnable
	sscWdDisable	禁用用戶看門狗功能。	○	○	129頁 sscWdDisable
	sscChangeWdCounter	更新看門狗檢查計數器。	○	○	130頁 sscChangeWdCounter
其他軸啟動函數群	sscSetOtherAxisStartData	設置其他軸啟動用資料。	○	○	131頁 sscSetOtherAxisStartData
	sscGetOtherAxisStartData	獲取其他軸啟動用資料。	○	○	132頁 sscGetOtherAxisStartData
	sscOtherAxisStartAbortOn	將其他軸啟動中斷信號(OSSTP□)置為ON。	○	○	133頁 sscOtherAxisStartAbortOn
	sscOtherAxisStartAbortOff	將其他軸啟動中斷信號(OSSTP□)置為OFF。	○	○	134頁 sscOtherAxisStartAbortOff
	sscGetOtherAxisStartStatus	獲取其他軸啟動狀態。	○	○	135頁 sscGetOtherAxisStartStatus
通過位置中斷函數群	sscSetIntPassPositionData	設置通過位置中斷條件資料。	○	○	136頁 sscSetIntPassPositionData
	sscCheckIntPassPositionData	獲取通過位置中斷條件資料。	○	○	137頁 sscCheckIntPassPositionData
	sscSetStartingPassNumber	設置通過位置條件開始編號/結束編號。	○	○	138頁 sscSetStartingPassNumber
	sscGetExecutingPassNumber	獲取執行中的通過位置條件編號。	○	○	139頁 sscGetExecutingPassNumber
採樣函數群	sscStartSampling	開始採樣。	○	○	140頁 sscStartSampling
	sscStopSampling	停止採樣。	○	○	141頁 sscStopSampling
	sscSetSamplingParameter	寫入採樣參數。	○	○	142頁 sscSetSamplingParameter
	sscGetSamplingParameter	讀取採樣參數。	○	○	143頁 sscGetSamplingParameter
	sscGetSamplingError	獲取採樣出錯資訊。	○	○	144頁 sscGetSamplingError
	sscGetSamplingStatus	獲取採樣的執行狀態。	○	○	145頁 sscGetSamplingStatus
	sscGetSamplingData	獲取採樣資料。	○	○	146頁 sscGetSamplingData
日誌函數群	sscStartLog	開始日誌。	○	○	147頁 sscStartLog
	sscStopLog	停止日誌。	○	○	148頁 sscStopLog
	sscCheckLogStatus	獲取日誌的執行狀態。	○	○	149頁 sscCheckLogStatus
	sscCheckLogEventNum	獲取日誌資料有效事件數。	○	○	150頁 sscCheckLogEventNum
	sscReadLogData	讀取日誌資料。	○	○	151頁 sscReadLogData
	sscClearLogData	清除(初始化)日誌資料。	○	○	152頁 sscClearLogData
	sscGetAlarmHistoryData	獲取報警履歷資料。	○	○	153頁 sscGetAlarmHistoryData
	sscCheckAlarmHistoryEventNum	獲取報警履歷資料有效事件數。	○	○	154頁 sscCheckAlarmHistoryEventNum
	sscClearAlarmHistoryData	清除(初始化)報警履歷資料。	○	○	155頁 sscClearAlarmHistoryData

分類	函數名稱	內容	可否使用		參照
			MR- MC2_ _	MR- MC3_ _	
數位輸入輸出 函數群	sscGetDigitalInputDataBit	以1點為單位獲取指定的數位輸入DI資料。	○	○	156頁 sscGetDigitalInputDataBit
	sscGetDigitalInputDataWord	以16點為單位獲取指定的數位輸入DI資料。	○	○	157頁 sscGetDigitalInputDataWord
	sscSetDigitalOutputDataBit	以1點為單位設置指定的數位輸出DO資料。	○	○	158頁 sscSetDigitalOutputDataBit
	sscSetDigitalOutputDataWord	以16點為單位設置指定的數位輸出DO資料。	○	○	159頁 sscSetDigitalOutputDataWord
	sscGetDigitalOutputDataBit	以1點為單位獲取指定的數位輸出DO資料。	○	○	160頁 sscGetDigitalOutputDataBit
	sscGetDigitalOutputDataWord	以16點為單位獲取指定的數位輸出DO資料。	○	○	161頁 sscGetDigitalOutputDataWord
標記檢測函數 群	sscGetMarkDetectionData	獲取標記檢測資料。	○	○	162頁 sscGetMarkDetectionData
	sscGetMarkDetectionCounter	獲取標記檢測次數。	○	○	164頁 sscGetMarkDetectionCounter
	sscClearMarkDetectionData	清除(初始化)標記檢測資料。	○	○	165頁 sscClearMarkDetectionData
介面模式函數 群	sscIfmGetReadErrorCount	獲取讀取錯誤計數器。	○	○	166頁 sscIfmGetReadErrorCount
	sscIfmSetHomePosition	進行原點設置。	○	○	167頁 sscIfmSetHomePosition
	sscIfmGetMaximumBufferNumber	獲取最大緩衝編號。(位置控制模式專用)	○	△	168頁 sscIfmGetMaximumBufferNumber
	sscIfmGetMaximumBufferNumberEx	獲取指定控制模式的最大緩衝編號。	○	○	169頁 sscIfmGetMaximumBufferNumberEx
	sscIfmRenewLatestBuffer	更新最新陳述式緩衝編號及資料。(位置控制模式專用)	○	△	170頁 sscIfmRenewLatestBuffer
	sscIfmRenewLatestBufferEx	更新指定控制模式的最新陳述式緩衝編號及資料。	○	○	171頁 sscIfmRenewLatestBufferEx
	sscIfmCheckLatestBuffer	獲取最新陳述式緩衝編號及資料。(位置控制模式專用)	○	△	172頁 sscIfmCheckLatestBuffer
	sscIfmCheckLatestBufferEx	獲取指定控制模式的最新陳述式緩衝編號及資料。	○	○	173頁 sscIfmCheckLatestBufferEx
	sscIfmGetTransmitBuffer	獲取傳送中緩衝編號及資料。(位置控制模式專用)	○	△	174頁 sscIfmGetTransmitBuffer
	sscIfmGetTransmitBufferEx	獲取指定控制模式的傳送中緩衝編號及資料。	○	○	175頁 sscIfmGetTransmitBufferEx
	sscIfmTrqSetSpeedLimit	設置轉矩控制用速度限制值。	○	○	176頁 sscIfmTrqSetSpeedLimit
	sscIfmSetControlMode	設置控制模式。	○	○	177頁 sscIfmSetControlMode
	sscIfmGetControlMode	獲取控制模式。	○	○	178頁 sscIfmGetControlMode
	sscIfmGetEventStatusBits	使用事件檢測功能，獲取對指定狀態信號的所有軸的狀態位元資訊。	○	△	179頁 sscIfmGetEventStatusBits
	sscIfmGetEventStatusBitsEx		×	○	180頁 sscIfmGetEventStatusBitsEx [MC300]

分類	函數名稱	內容	可否使用		參照
			MR- MC2_ _	MR- MC3_ _	
中斷函數群	sscIntStart	啟動中斷驅動器。	○	○	183頁 sscIntStart
	sscIntEnd	退出中斷驅動器。	○	○	184頁 sscIntEnd
	sscIntEnable	啟用中斷輸出。	○	○	185頁 sscIntEnable
	sscIntDisable	禁用中斷輸出。	○	○	186頁 sscIntDisable
	sscRegisterIntCallback	登記中斷回呼函數。	○	○	187頁 sscRegisterIntCallback
	sscUnregisterIntCallback	解除中斷回呼函數。	○	○	188頁 sscUnregisterIntCallback
	sscResetIntEvent	將中斷事件設成非信號狀態。	○	○	189頁 sscResetIntEvent
	sscSetIntEvent	將中斷事件設成信號狀態。	○	○	190頁 sscSetIntEvent
	sscWaitIntEvent	待機至中斷事件變為信號狀態。	○	○	191頁 sscWaitIntEvent
	sscResetIntEventMulti	將多個中斷事件設成非信號狀態。	○	○	193頁 sscResetIntEventMulti
	sscSetIntEventMulti	將多個中斷事件設成信號狀態。	○	○	194頁 sscSetIntEventMulti
	sscWaitIntEventMulti	待機至多個中斷事件變為信號狀態。	○	○	195頁 sscWaitIntEventMulti
	sscResetIntOasEvent	將其他軸啟動的中斷事件設成非信號狀態。	○	○	197頁 sscResetIntOasEvent
	sscSetIntOasEvent	將其他軸啟動的中斷事件設成信號狀態。	○	○	198頁 sscSetIntOasEvent
	sscWaitIntOasEvent	待機至其他軸啟動的中斷事件變為信號狀態。	○	○	199頁 sscWaitIntOasEvent
	sscResetIntPassPosition	將通過位置中斷事件設成非信號狀態。	○	○	201頁 sscResetIntPassPosition
	sscSetIntPassPosition	將通過位置中斷事件設成信號狀態。	○	○	202頁 sscSetIntPassPosition
	sscWaitIntPassPosition	待機至通過位置中斷事件變為信號狀態。	○	○	203頁 sscWaitIntPassPosition
	sscResetIntDriveFin	將運行完成的中斷事件設成非信號狀態。	○	○	204頁 sscResetIntDriveFin
	sscSetIntDriveFin	將運行完成的中斷事件設成信號狀態。	○	○	205頁 sscSetIntDriveFin
sscWaitIntDriveFin	待機至運行完成的中斷事件變為信號狀態。	○	○	206頁 sscWaitIntDriveFin	
輸入輸出軟元件函數群	sscGetInputDeviceBit	以1點為單位獲取指定的輸入位軟元件。	○	○	208頁 sscGetInputDeviceBit
	sscGetInputDeviceWord	以字單位獲取指定的輸入字軟元件。	○	○	209頁 sscGetInputDeviceWord
	sscSetOutputDeviceBit	使用定位板的雙埠記憶體排他控制功能，以1點為單位設置指定的輸出位軟元件。	○	○	210頁 sscSetOutputDeviceBit
	sscSetOutputDeviceWord	不使用定位板的雙埠記憶體排他控制功能，以字單位設置指定的輸出字軟元件。	○	○	211頁 sscSetOutputDeviceWord
	sscGetOutputDeviceBit	以1點為單位獲取指定的輸出位軟元件。	○	○	212頁 sscGetOutputDeviceBit
	sscGetOutputDeviceWord	以字單位獲取指定的輸出字軟元件。	○	○	213頁 sscGetOutputDeviceWord
暫態傳送函數群	sscSendReceiveTransientData	對連接至SSCNET的軸或站進行指定的暫態傳送資料的發送接收。	○	○	214頁 sscSendReceiveTransientData

4 API函數詳細

4.1 輔助函數群

sscGetLastError

在各API函數中出錯(返回值“SSC_NG”)的情況下，通過調用本API函數獲取詳細出錯代碼。

```
int sscGetLastError (  
    void  
);
```

詳細說明

■參數

無。

■返回值

最新出錯代碼

■詳細出錯代碼

無。

■要點

返回值為“SSC_UNOPEN”的情況下，無法設置詳細出錯代碼。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetMountChannel

獲取安裝通道資訊。

```
int sscGetMountChannel (  
    int board_id,  
    short *mountch  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
mountch [out]	指向儲存安裝通道資訊的2位元組變數 (2位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

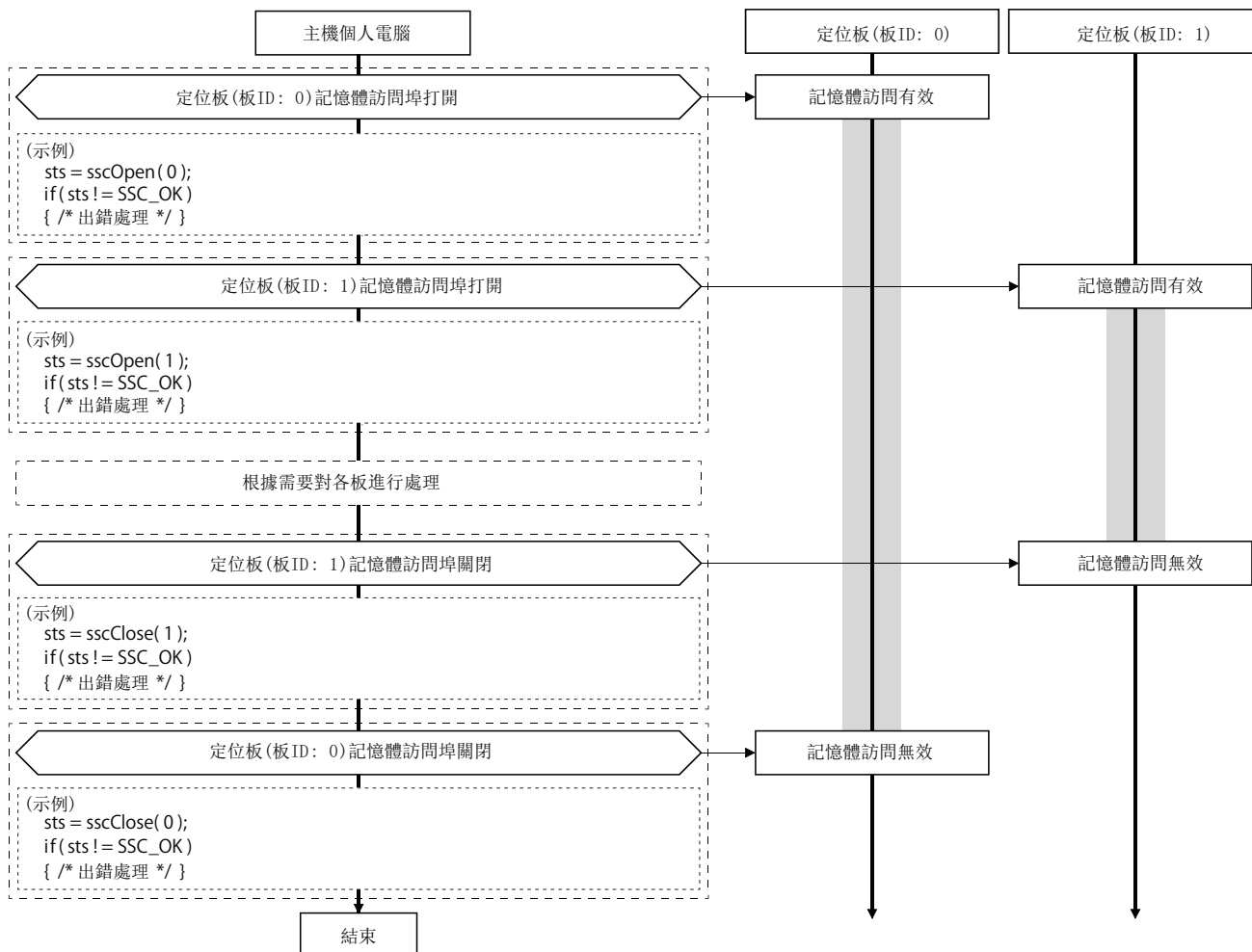
無。

4.2 軟元件函數群

處理步驟

用於記憶體訪問的軟元件處理步驟如下所示。

■在主機個人電腦上連接了2個定位板(板ID: 0與板ID: 1)的情況下



要點

- 不是同一軟元件(板ID)的情況下，可以同時打開記憶體訪問埠。
- 請勿連續調用sscOpen函數/sscClose函數。
- 通過將記憶體訪問埠的打開/關閉集中到用戶程式處理的最初與最後，可以防止因API函式呼叫時的記憶體訪問埠未打開而導致發生出錯。

<例>用戶程式處理示例

電源接通後，在用戶程式初始處理內打開記憶體訪問埠。



通電時任意處理



電源斷開前，在用戶程式結束處理內關閉記憶體訪問埠。

sscOpen

打開記憶體訪問埠。

```
int sscOpen (  
    int board_id  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號 (0~3)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_REOPEN	已調用sscOpen函數。
SSC_FUNC_ERR_DEVICE_DRIVER	軟元件驅動器的調用時發生了出錯。 應確認已經安裝了軟元件驅動器。
SSC_FUNC_ERR_GET_CHANNEL_NUM	無法獲取安裝通道資訊。 有時OS不能正常識別定位板。 應通過軟元件管理器確認定位板已正確安裝。
SSC_FUNC_ERR_CREATE_SEMAPHORE	在CreateSemaphore函數 (Windows API) 中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
SSC_FUNC_ERR_NOT_FOUND_BOARD	<ul style="list-style-type: none">找不到指定板ID的定位板。應確認定位板的板ID選擇 (撥動開關)。從定位板測試工具連接到定位板的情況下，應確認“確認硬體中斷”的勾選處於取消狀態。 [MC300]
SSC_FUNC_ERR_UNSUPPORTED_DEVICE_DRIVER	非軟元件驅動器的對應版本。 應組合實用軟體中收錄的軟元件驅動器及API庫使用。

要點

應在調用了對要使用板ID的sscOpen函數後，調用各API函數。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscClose (↩ 32頁 sscClose)

sscClose

關閉記憶體訪問埠。

```
int sscClose (  
    int board_id  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號 (0~3)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNOPEN	尚未調用sscOpen函數。
SSC_FUNC_ERR_DEVICE_DRIVER	軟元件驅動器的調用時發生了出錯。 應確認已經安裝了軟元件驅動器。
SSC_FUNC_ERR_DELETE_SEMAPHORE	在CloseHandle函數 (Windows API) 中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

要點

應在結束用戶程式之前，調用已打開記憶體訪問埠的板ID的sscClose函數。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscOpen(🔗 31頁 sscOpen)

4.3 資訊函數群

sscGetControlCycle

獲取控制週期狀態。

```
int sscGetControlCycle (  
    int board_id,  
    int channel,  
    short *ctrl_cycle  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
ctrl_cycle [out]	指向儲存控制週期狀態的2位元組變數(2位元組×1)的指針 ■值 • SSC_CTRL_CYCLE_ERROR: 系統啟動前 • SSC_CTRL_CYCLE_888: 0.888ms • SSC_CTRL_CYCLE_444: 0.444ms • SSC_CTRL_CYCLE_222: 0.222ms

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetBoardVersion

獲取定位板的系統版本資訊。

```
int sscGetBoardVersion (  
    int board_id,  
    int channel,  
    char *version  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
version [out]	指向儲存系統版本資訊的16位元組陣列(1位元組×16)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

版本資訊的末尾無NULL代碼。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetDriverVersion([↗](#) 35頁 sscGetDriverVersion)

sscGetDriverVersion

獲取驅動器版本資訊。

```
int sscGetDriverVersion (  
    int board_id,  
    int channel,  
    int axnum,  
    char *version  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
version [out]	指向儲存驅動器版本資訊的16位元組陣列(1位元組×16)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

版本資訊的末尾無NULL代碼。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetBoardVersion([↩](#) 34頁 sscGetBoardVersion)

sscGetOperationCycleMonitor

獲取運算週期監視資料。

```
int sscGetOperationCycleMonitor (  
    int board_id,  
    int channel,  
    short *now,  
    short *max,  
    short *over,  
    char *status  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
now [out]	指向儲存運算週期當前時間[μs]的2位元組變數的指針
max [out]	指向儲存運算週期最大時間[μs]的2位元組變數的指針
over [out]	指向儲存運算週期超過次數的2位元組變數的指針
status [out]	指向儲存運算週期狀態的1位元組變數(1位元組×1)的指針 使用各值的邏輯或設置已獲取的資料。 ■值 • SSC_BIT_OCME: 運算週期報警中 • SSC_BIT_OCMW: 運算週期警告中

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscClearOperationCycleMonitor([↗](#) 37頁 sscClearOperationCycleMonitor)

sscClearOperationCycleMonitor

清除(初始化)運算週期監視資料。

```
int sscClearOperationCycleMonitor (  
    int board_id,  
    int channel  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_RUNNING_CHANNEL	系統啟動前。 應使用sscSystemStart函數啟動系統。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。

要點

無。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscGetOperationCycleMonitor([↩](#) 36頁 sscGetOperationCycleMonitor)

sscGetBoardSerialNumber [MC300]

獲取定位板的製造編號。

```
int sscGetBoardSerialNumber (  
    int board_id,  
    char *serialnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
serialnum [out]	指向儲存製造編號的16位元組陣列 (1位元組×16) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 製造編號的末尾無NULL代碼。
- Visual C#的工程中的本API函數的char*類型的參數應替換為sbyte[]類型。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetBoardVersion ([↩](#) 34頁 sscGetBoardVersion)

4.4 參數函數群

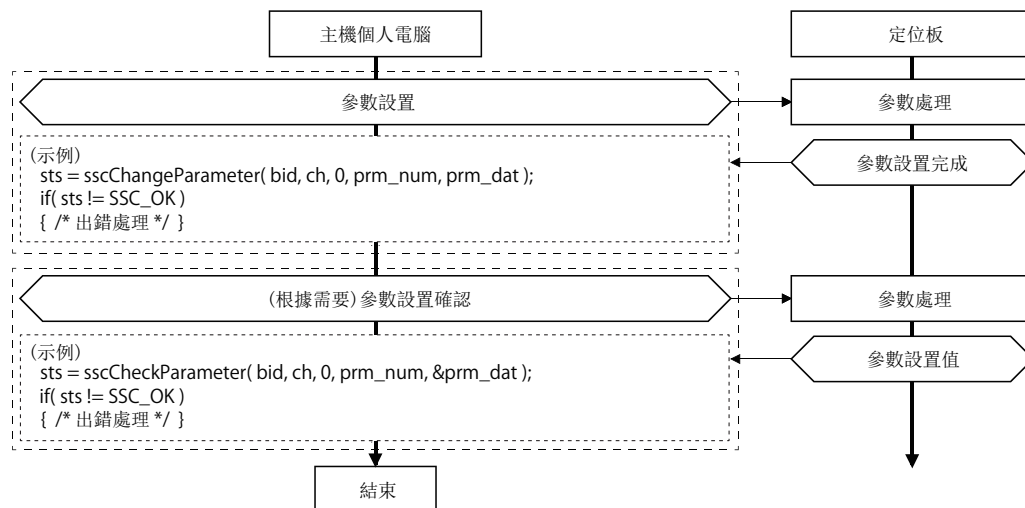
處理步驟

參數處理步驟的示例如下所示。

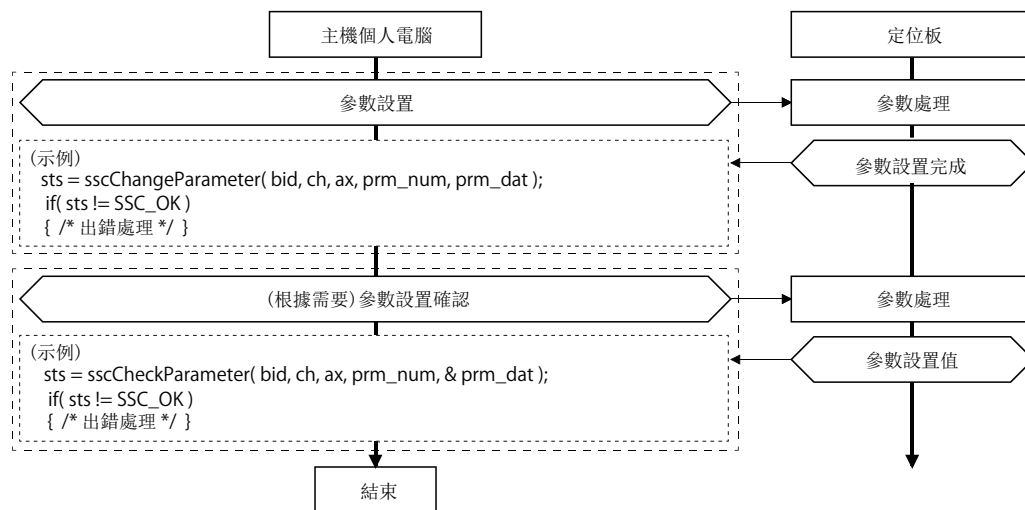
要點

應使用sscResetAllParameter函數將參數全部初始化後，僅更改與初始值不同的參數。

■設置系統參數的情況下



■設置控制/伺服參數的情況下



sscResetAllParameter

將系統啟動前(系統準備完成)的所有參數設置成初始值。

```
int sscResetAllParameter (  
    int board_id,  
    int channel,  
    int timeout  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
timeout [in]	超時時間[ms](0~65535)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNREADY_CHANNEL	系統準備完成以外。 應使用sscReboot函數重啟。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了指定的超時時間。

要點

在超時時間中指定了2秒(2000ms)及以下的情況下, 超時時間將為2秒(2000ms)。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

無。

sscChangeParameter

寫入各參數。

本API函數支援系統參數、伺服參數(MR-J4(W)-_B/MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入。由於不支持MR-J5(W)-_B的伺服參數，因此寫入MR-J5(W)-_B的伺服參數的情況下，應使用sscChangeParameterEx函數。

```
int sscChangeParameter (
    int board_id,
    int channel,
    int axnum,
    short prmnum,
    short prmdata
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統參數 • 1~64: 各軸參數 • -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4 . . . -15: 站15, -16: 站16)
prmnum [in]	參數寫入編號
prmdata [in]	參數寫入資料

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PWEN1	發生了參數寫入編號出錯。 <ul style="list-style-type: none"> • 在參數寫入編號1中設置了超出範圍的值。 • 軸編號與參數寫入編號不匹配。(例: 在軸編號中設置了“系統參數”，在參數寫入編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_STS_BIT_PWED1	在參數寫入資料1中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM1	參數寫入編號1的陳述式與狀態不一致。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA1	參數寫入資料1的陳述式與狀態不一致。
SSC_FUNC_ERR_PARAM_WRITE_EX_NUM1	無法寫入MR-J5(W)-_B的伺服參數。 應使用sscChangeParameterEx函數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以寫入參數。
- 為了進行定位板的回應確認，大約需要2~10個控制週期。
- 對於多個軸連續寫入多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短寫入時間。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChange2Parameter (↗ 43頁 sscChange2Parameter)
- sscCheckParameter (↗ 45頁 sscCheckParameter)
- sscCheck2Parameter (↗ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (↗ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (↗ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (↗ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (↗ 58頁 sscCheck2ParameterEx)

sscChange2Parameter

寫入2點的各參數。

本API函數支援系統參數、伺服參數(MR-J4(W)-_B/MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入。由於不支持MR-J5(W)-_B的伺服參數，因此寫入MR-J5(W)-_B的伺服參數的情況下，應使用sscChange2ParameterEx函數。

```
int sscChange2Parameter (  
    int board_id,  
    int channel,  
    int axnum,  
    short *prmnum,  
    short *prmdata,  
    char *status  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統參數• 1~64: 各軸參數• -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4... -15: 站15, -16: 站16)
prmnum [in]	指向儲存參數寫入編號(2點)的4位元組陣列(2位元組×2)的指針
prmdata [in]	指向儲存參數寫入資料(2點)的4位元組陣列(2位元組×2)的指針
status [out]	指向儲存參數寫入狀態(2點)的2位元組陣列(1位元組×2)的指針 使用各值的邏輯或設置已獲取的資料。 ■值 <ul style="list-style-type: none">• SSC_BIT_PWFIN: 參數寫入完成• SSC_BIT_PWEN: 參數編號不正確• SSC_BIT_PWED: 參數資料超出範圍

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PWEN□ □=1~2: 參數寫入編號(2點)的陣列編號	發生了參數寫入編號出錯。 • 在參數寫入編號中設置了超出範圍的值。 • 軸編號與參數寫入編號不匹配。(例: 在軸編號中設置了“系統參數”，在參數寫入編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_STS_BIT_PWED□ □=1~2: 參數寫入資料(2點)的陣列編號	在參數寫入資料中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM□ □=1~2: 參數寫入編號(2點)的陣列編號	參數寫入編號的陳述式與狀態不一致。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □=1~2: 參數寫入資料(2點)的陣列編號	參數寫入資料的陳述式與狀態不一致。
SSC_FUNC_ERR_PARAM_WRITE_EX_NUM□ □=1~2: 參數寫入資料(2點)的陣列編號	無法寫入MR-J5(W)-B的伺服參數。 應使用sscChange2ParameterEx函數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以寫入參數。
- 只更改1點參數的情況下，應將不使用的參數編號設置為“0”。
- 為了進行定位板的回應確認，大約需要2~10個控制週期。
- 對於多個軸連續寫入多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短寫入時間。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (☞ 41頁 sscChangeParameter)
- sscCheckParameter (☞ 45頁 sscCheckParameter)
- sscCheck2Parameter (☞ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (☞ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (☞ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (☞ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (☞ 58頁 sscCheck2ParameterEx)

sscCheckParameter

讀取指定的參數設置值。

本API函數支援系統參數、伺服參數(MR-J4(W)-_B/MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取。由於不支持MR-J5(W)-_B的伺服參數，因此讀取MR-J5(W)-_B的伺服參數的情況下，應使用sscCheckParameterEx函數。

```
int sscCheckParameter (
    int board_id,
    int channel,
    int axnum,
    short prmnum,
    short *prmdata
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統參數 • 1~64: 各軸參數 • -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
prmnum [in]	參數讀取編號
prmdata [out]	指向儲存參數讀取資料的2位元組變數(2位元組×1)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_STS_BIT_PREN1	發生了參數讀取出錯。 <ul style="list-style-type: none"> • 在參數讀取編號1中設置了超出範圍的值。 • 軸編號與參數讀取編號不匹配。(例: 在軸編號中設置了“系統參數”，在參數讀取編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM1	參數讀取編號1的陳述式與狀態不一致。
SSC_FUNC_ERR_PARAM_READ_EX_NUM1	無法讀取MR-J5(W)-_B的伺服參數。 應使用sscCheckParameterEx函數。

要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以讀取參數。
- 為了進行定位板的回應確認，大約需要2~10個控制週期。
- 從多個軸連續讀取多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短讀取時間。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (↗ 41頁 sscChangeParameter)
- sscChange2Parameter (↗ 43頁 sscChange2Parameter)
- sscCheck2Parameter (↗ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (↗ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (↗ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (↗ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (↗ 58頁 sscCheck2ParameterEx)

sscCheck2Parameter

讀取指定的2點參數設置值。

本API函數支援系統參數、伺服參數(MR-J4(W)-_B/MR-J3(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取。由於不支持MR-J5(W)-_B的伺服參數，因此讀取MR-J5(W)-_B的伺服參數的情況下，應使用sscCheck2ParameterEx函數。

```
int sscCheck2Parameter (
    int board_id,
    int channel,
    int axnum,
    short *prmnum,
    short *prmdata,
    char *status
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統參數 • 1~64: 各軸參數 • -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
prmnum [in]	指向儲存參數讀取編號(2點)的4位元組陣列(2位元組×2)的指針
prmdata [out]	指向儲存參數讀取資料(2點)的4位元組陣列(2位元組×2)的指針
status [out]	指向儲存參數獲取狀態(2點)的2位元組陣列(1位元組×2)的指針 使用各值的邏輯或設置已獲取的資料。 值 <ul style="list-style-type: none"> • SSC_BIT_PRFIN: 參數讀取完成 • SSC_BIT_PREN: 參數編號不正確

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_STS_BIT_PREN□ □=1~2: 參數讀取編號(2點)的陣列編號	發生了參數讀取出錯。 <ul style="list-style-type: none"> • 在參數讀取編號中設置了超出範圍的值。 • 軸編號與參數讀取編號不匹配。 (例: 在軸編號中設置了“系統參數”，在參數讀取編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM□ □=1~2: 參數讀取編號(2點)的陣列編號	參數讀取編號的陳述式與狀態不一致。
SSC_FUNC_ERR_PARAM_READ_EX_NUM□ □=1~2: 參數讀取編號(2點)的陣列編號	無法讀取MR-J5(W)-_B的伺服參數。 應使用sscCheck2ParameterEx函數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以讀取參數。
- 為了唯讀取1點參數設置，應將不使用的參數編號設置為“0”。
- 為了進行定位板的回應確認，大約需要2~10個控制週期。
- 從多個軸連續讀取多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短讀取時間。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (↖ 41頁 sscChangeParameter)
- sscChange2Parameter (↖ 43頁 sscChange2Parameter)
- sscCheckParameter (↖ 45頁 sscCheckParameter)
- sscChangeParameterEx (↖ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (↖ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (↖ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (↖ 58頁 sscCheck2ParameterEx)

sscLoadAllParameterFromFlashROM

從快閃記憶體讀取系統啟動前(系統準備完成)的所有參數。

```
int sscLoadAllParameterFromFlashROM (
    int board_id,
    int channel,
    int timeout
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
timeout [in]	超時時間[ms](0~65535)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNREADY_CHANNEL	系統準備完成以外。 應使用sscReboot函數重啟。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了指定的超時時間。
SSC_FUNC_ERR_FLASHROM_PARAM_LOAD	發生了快閃記憶體參數讀取不正確。


■要點

在超時時間中指定了2秒(2000ms) [MC200]/5秒(5000ms) [MC300]及以下的情況下, 超時時間將為2秒(2000ms) [MC200]/5秒(5000ms) [MC300]。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSaveAllParameterToFlashROM( 50頁 sscSaveAllParameterToFlashROM)

sscSaveAllParameterToFlashROM

將所有參數儲存至快閃記憶體。

```
int sscSaveAllParameterToFlashROM (  
    int board_id,  
    int channel,  
    int timeout  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
timeout [in]	超時時間[ms](0~65535 [MC200]/0~600000 [MC300])

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~3: 超時位置	向定位板發出陳述式後的回應確認中, 經過了指定的超時時間。
SSC_FUNC_ERR_STS_BIT_FRNG	快閃記憶體傳送準備不正確信號(FRNG)變為了ON。
SSC_FUNC_ERR_STS_BIT_FSNG	快閃記憶體傳送不正確信號(FSNG)變為了ON。

要點

- 在超時時間中指定了10秒(1000ms) [MC200]/300秒(300000ms) [MC300]及以下的情況下, 超時時間將為10秒(1000ms) [MC200]/300秒(300000ms) [MC300]。
- 將快閃記憶體中備份的參數內容更改為初始值的情況下, 應通過sscResetAllParameter函數初始化所有參數後再調用本API函數。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscLoadAllParameterFromFlashROM([↩](#) 49頁 sscLoadAllParameterFromFlashROM)

sscCheckSvPrmChangeNumEx

獲取伺服參數更改編號。

```
int sscCheckSvPrmChangeNumEx (  
    int board_id,  
    int channel,  
    int axnum,  
    short *prmnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
prmnum [out]	指向儲存伺服參數更改編號的8位元組陣列(2位元組×4)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- SSCNETⅢ通信方式的情況下, 不會在伺服參數更改編號的高位元6位元組中儲存值。[MC200]
- 對於伺服參數更改編號的詳細資訊, 應使用監視。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2. 40	B7	mc2xxstd.h
MR-MC3_ _	1. 30	A6	mc3xxstd.h

■參照

無。

sscChangeParameterEx

寫入各參數。

本API函數支援系統參數、伺服參數(MR-J5(W)-_B/MR-J4(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入。

```
int sscChangeParameterEx (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned short prmnum,  
    unsigned long prmdata  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統參數• 1~64: 各軸參數• -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4 . . . -15: 站15, -16: 站16)
prmnum [in]	參數寫入編號
prmdata [in]	參數寫入資料

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PWEN1	發生了參數寫入編號出錯。 <ul style="list-style-type: none">• 在參數寫入編號1中設置了超出範圍的值。• 軸編號與參數寫入編號不匹配。 (例: 在軸編號中設置了“系統參數”, 在參數寫入編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_STS_BIT_PWED1	在參數寫入資料1中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM1	參數寫入編號1的陳述式與狀態不一致。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA1	參數寫入資料1的陳述式與狀態不一致。
SSC_FUNC_ERR_UNSUPPORTED_PARAM_EX	定位板不支持MR-J5(W)-_B的伺服參數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以寫入參數。
- 為了進行定位板的回應確認，大約需要2~120個控制週期。
- 對於多個軸連續寫入多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短寫入時間。
- 根據參數寫入編號，參數寫入資料的大小有所不同。

軸編號	參數寫入編號	參數寫入資料大小
0	0000h~FFFFh	2位元組(應設置為低位元2位元組。)
1~64	2000h~2FFFh	4位元組
	上述以外	2位元組(應設置為低位元2位元組。)
-16~-1	0000h~FFFFh	2位元組(應設置為低位元2位元組。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (↖ 41頁 sscChangeParameter)
- sscChange2Parameter (↖ 43頁 sscChange2Parameter)
- sscCheckParameter (↖ 45頁 sscCheckParameter)
- sscCheck2Parameter (↖ 47頁 sscCheck2Parameter)
- sscChange2ParameterEx (↖ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (↖ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (↖ 58頁 sscCheck2ParameterEx)

sscChange2ParameterEx

寫入2點的各參數。

本API函數支援系統參數、伺服參數(MR-J5(W)-_B/MR-J4(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的寫入。

```
int sscChange2ParameterEx (
    int board_id,
    int channel,
    int axnum,
    unsigned short *prmnum,
    unsigned long *prmdata,
    unsigned char *status
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統參數 • 1~64: 各軸參數 • -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4... -15: 站15, -16: 站16)
prmnum [in]	指向儲存參數寫入編號(2點)的4位元組陣列(2位元組×2)的指針
prmdata [in]	指向儲存參數寫入資料(2點)的4位元組陣列(2位元組×2)的指針
status [out]	指向儲存參數寫入狀態(2點)的2位元組陣列(1位元組×2)的指針 使用各值的邏輯或設置已獲取的資料。 ■值 <ul style="list-style-type: none"> • SSC_BIT_PWFIN: 參數寫入完成 • SSC_BIT_PWEN: 參數編號不正確 • SSC_BIT_PWED: 參數資料超出範圍

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PWEN□ □=1~2: 參數寫入編號(2點)的陣列編號	發生了參數寫入編號出錯。 <ul style="list-style-type: none"> • 在參數寫入編號中設置了超出範圍的值。 • 軸編號與參數寫入編號不匹配。 (例: 在軸編號中設置了“系統參數”, 在參數寫入編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_STS_BIT_PWED□ □=1~2: 參數寫入資料(2點)的陣列編號	在參數寫入資料中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM□ □=1~2: 參數寫入編號(2點)的陣列編號	參數寫入編號的陳述式與狀態不一致。
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □=1~2: 參數寫入資料(2點)的陣列編號	參數寫入資料的陳述式與狀態不一致。
SSC_FUNC_ERR_UNSUPPORTED_PARAM_EX	定位板不支持MR-J5(W)-_B的伺服參數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以寫入參數。
- 只更改1點參數的情況下，應將不使用的參數編號設置為“0”。
- 為了進行定位板的回應確認，大約需要2~120個控制週期。
- 對於多個軸連續寫入多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短寫入時間。
- 根據參數寫入編號，參數寫入資料的大小有所不同。

軸編號	參數寫入編號	參數寫入資料大小
0	0000h~FFFFh	2位元組(應設置為低位元2位元組。)
1~64	2000h~2FFFh	4位元組
	上述以外	2位元組(應設置為低位元2位元組。)
-16~-1	0000h~FFFFh	2位元組(應設置為低位元2位元組。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (☞ 41頁 sscChangeParameter)
- sscChange2Parameter (☞ 43頁 sscChange2Parameter)
- sscCheckParameter (☞ 45頁 sscCheckParameter)
- sscCheck2Parameter (☞ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (☞ 52頁 sscChangeParameterEx)
- sscCheckParameterEx (☞ 56頁 sscCheckParameterEx)
- sscCheck2ParameterEx (☞ 58頁 sscCheck2ParameterEx)

sscCheckParameterEx

讀取指定的參數設置值。

本API函數支援系統參數、伺服參數(MR-J5(W)-_B/MR-J4(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取。

```
int sscCheckParameterEx (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned short prmnum,  
    unsigned long *prmdata  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統參數• 1~64: 各軸參數• -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4 ••• -15: 站15, -16: 站16)
prmnum [in]	參數讀取編號
prmdata [out]	指向儲存參數讀取資料的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_STS_BIT_PREN1	發生了參數讀取出錯。 <ul style="list-style-type: none">• 在參數讀取編號中設置了超出範圍的值。• 軸編號與參數讀取編號不匹配。 (例: 在軸編號中設置了“系統參數”, 在參數讀取編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM1	參數讀取編號1的陳述式與狀態不一致。
SSC_FUNC_ERR_UNSUPPORTED_PARAM_EX	定位板不支持MR-J5(W)-_B的伺服參數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以讀取參數。
- 為了進行定位板的回應確認，大約需要2~120個控制週期。
- 從多個軸連續讀取多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短讀取時間。
- 根據參數讀取編號，參數讀取資料的大小有所不同。

軸編號	參數讀取編號	參數讀取資料大小
0	0000h~FFFFh	2位元組(被儲存到低位元2位元組。)
1~64	2000h~2FFFh	4位元組
	上述以外	2位元組(被儲存到低位元2位元組。)
-16~-1	0000h~FFFFh	2位元組(被儲存到低位元2位元組。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscChangeParameter (↖ 41頁 sscChangeParameter)
- sscChange2Parameter (↖ 43頁 sscChange2Parameter)
- sscCheckParameter (↖ 45頁 sscCheckParameter)
- sscCheck2Parameter (↖ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (↖ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (↖ 54頁 sscChange2ParameterEx)
- sscCheck2ParameterEx (↖ 58頁 sscCheck2ParameterEx)

sscCheck2ParameterEx

讀取指定的2點參數設置值。

本API函數支援系統參數、伺服參數(MR-J5(W)-_B/MR-J4(W)-_B、傳感模組)、控制參數、RIO模組參數、RIO控制參數的讀取。

```
int sscCheck2ParameterEx (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned short *prmnum,  
    unsigned long *prmdata,  
    unsigned char *status  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統參數• 1~64: 各軸參數• -16~-1: 各站參數(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
prmnum [in]	指向儲存參數讀取編號(2點)的4位元組陣列(2位元組×2)的指針
prmdata [out]	指向儲存參數讀取資料(2點)的4位元組陣列(2位元組×2)的指針
status [out]	指向儲存參數獲取狀態(2點)的2位元組陣列(1位元組×2)的指針 使用各值的邏輯或設置已獲取的資料。 ■值 <ul style="list-style-type: none">• SSC_BIT_PRFIN: 參數讀取完成• SSC_BIT_PREN: 參數編號不正確

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_STS_BIT_PREN□ □=1~2: 參數讀取編號(2點)的陣列編號	發生了參數讀取出錯。 <ul style="list-style-type: none">• 在參數讀取編號中設置了超出範圍的值。• 軸編號與參數讀取編號不匹配。 (例: 在軸編號中設置了“系統參數”, 在參數讀取編號中設置了“軸參數”等情況下)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM□ □=1~2: 參數讀取編號(2點)的陣列編號	參數讀取編號的陳述式與狀態不一致。
SSC_FUNC_ERR_UNSUPPORTED_PARAM_EX	定位板不支持MR-J5(W)-_B的伺服參數。

■要點

- 對於本API函數，不管在系統啟動前/系統啟動中，均可以讀取參數。
- 為了唯讀取1點參數設置，應將不使用的參數編號設置為“0”。
- 為了進行定位板的回應確認，大約需要2~120個控制週期。
- 從多個軸連續讀取多個參數的情況下，在“系統”及“軸”的各執行緒中調用本API函數時，可以縮短讀取時間。
- 根據參數讀取編號，參數讀取資料的大小有所不同。

軸編號	參數讀取編號	參數讀取資料大小
0	0000h~FFFFh	2位元組(被儲存到低位元2位元組。)
1~64	2000h~2FFFh	4位元組
	上述以外	2位元組(被儲存到低位元2位元組。)
-16~-1	0000h~FFFFh	2位元組(被儲存到低位元2位元組。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

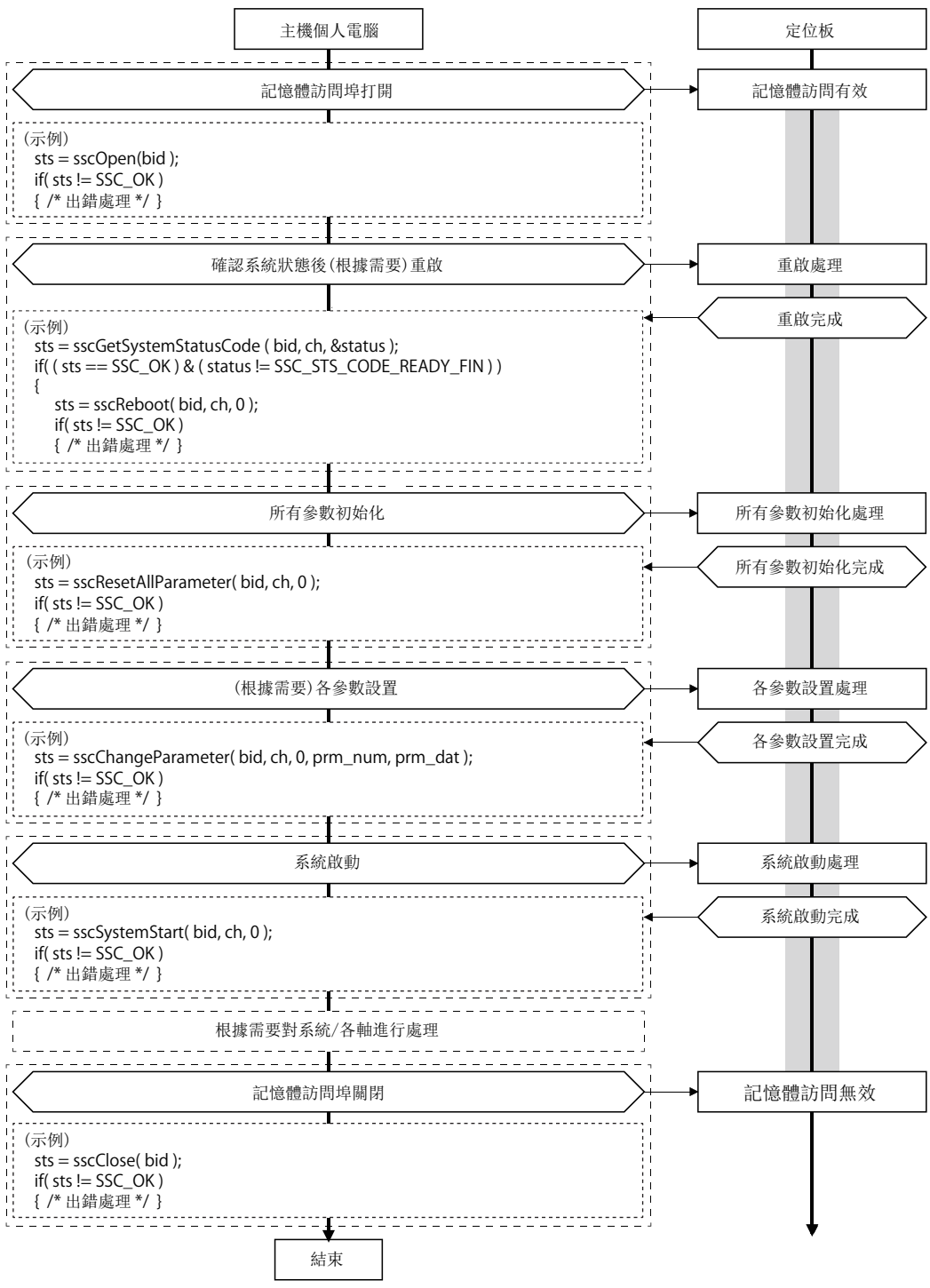
■參照

- sscChangeParameter (☞ 41頁 sscChangeParameter)
- sscChange2Parameter (☞ 43頁 sscChange2Parameter)
- sscCheckParameter (☞ 45頁 sscCheckParameter)
- sscCheck2Parameter (☞ 47頁 sscCheck2Parameter)
- sscChangeParameterEx (☞ 52頁 sscChangeParameterEx)
- sscChange2ParameterEx (☞ 54頁 sscChange2ParameterEx)
- sscCheckParameterEx (☞ 56頁 sscCheckParameterEx)

4.5 系統函數群

處理步驟

啟動(結束)系統之前的系統處理步驟的示例如下所示。



sscReboot

進行重啟(系統啟動中→系統準備完成)。

在函數內部等待直到系統準備完成。

```
int sscReboot (  
    int board_id,  
    int channel,  
    int timeout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
timeout [in]	超時時間[ms](0~65535)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應中, 經過了指定的超時時間。


■要點

- 在超時時間中指定了10秒(10000ms) [MC200]/20秒(20000ms) [MC300]及以下的情況下, 超時時間將為10秒(10000ms) [MC200]/20秒(20000ms) [MC300]。
- 在系統準備完成狀態下執行的情況下也進行重啟。
- 請勿在重啟中從定位板測試工具進行軟元件打開。否則有可能不正常動作。[MC300]

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetSystemStatusCode( 63頁 sscGetSystemStatusCode)

sscSystemStart

伺服驅動器通信初始化後，進行系統啟動(系統準備完成→系統啟動中)。

```
int sscSystemStart (  
    int board_id,  
    int channel,  
    int timeout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
timeout [in]	超時時間[ms](0~65535)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNREADY_CHANNEL	系統準備完成以外。 應使用sscReboot函數重啟。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了指定的超時時間。 應確認定位板側的SSCNET III電纜是否連接。或是SSCNET通信方式不同。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

■要點

- 在超時時間中指定了10秒(10000ms)及以下的情況下，超時時間將為10秒(10000ms)。
- 再次進行系統啟動的情況下，應在重啟後執行。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetSystemStatusCode

獲取系統狀態碼。

```
int sscGetSystemStatusCode (  
    int board_id,  
    int channel,  
    short *statuscode  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
statuscode [out]	指向儲存系統狀態碼的2位元組變數(2位元組×1)的指針 ■值 • SSC_STS_CODE_READY_FIN: 系統準備完成 • SSC_STS_CODE_RUNNING: 系統啟動中

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

關於“系統準備完成”、“系統啟動中”以外的系統狀態碼, 請參閱下述手冊的“系統陳述式/狀態”的“系統狀態”。

📖MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscReconnectSSCNET

開始與設置成控制軸的未通信軸進行SSCNET通信。

```
int sscReconnectSSCNET (  
    int board_id,  
    int channel,  
    unsigned long long *ctrl_axbit,  
    unsigned short *err_code  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
ctrl_axbit [out]	指向儲存控制中軸資訊位及控制中站資訊位(0~0F000000FFFFFFFFh)的8位元組變數(8位元組×1)的指針 位0→軸1、位1→軸2・・・・位31→軸32、位56→站1、位57→站2、位58→站3、位59→站4
err_code [out]	指向儲存斷開後重新連接出錯代碼的2位元組變數(2位元組×1)的指針

■返回值


值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(10秒)。
SSC_FUNC_ERR_STS_BIT_RCE	重新連接不正確信號(RCE)變為了ON。

■要點


關於“斷開後重新連接出錯代碼”，請參閱下述手冊的“斷開/重新連接功能”的“介面”。

 MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscDisconnectSSCNET( 66頁 sscDisconnectSSCNET)

sscReconnectSSCNETEx [MC300]

開始與設置成控制軸的未通信軸進行SSCNET通信。

```
int sscReconnectSSCNETEx (  
    int board_id,  
    int channel,  
    SLAVE_INFO *pSlaveInfo,  
    unsigned short *err_code  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pSlaveInfo [out]	指向儲存軟元件資訊(控制中軸資訊及控制中站資訊)的80位元組結構(80位元組×1)的指針 關於軟元件資訊結構，請參閱下述章節。 ☞ 237頁 SLAVE_INFO結構[MC300]
err_code [out]	指向儲存斷開後重新連接出錯代碼的2位元組變數(2位元組×1)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(10秒)。
SSC_FUNC_ERR_STS_BIT_RCE	重新連接不正確信號(RCE)變為了ON。

要點

關於“斷開後重新連接出錯代碼”，請參閱下述手冊的“斷開/重新連接功能”的“介面”。

☞ MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscDisconnectSSCNETEx(☞ 67頁 sscDisconnectSSCNETEx [MC300])

sscDisconnectSSCNET

關閉通過斷開軸編號設置的SSCNET通信中軟元件設備(伺服驅動器等)及以後連接的SSCNET通信。

```
int sscDisconnectSSCNET (  
    int board_id,  
    int channel,  
    int com_num,  
    unsigned long long *ctrl_axbit,  
    unsigned short *err_code  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
com_num [in]	斷開軸編號(-4~32) • 1~32: 各軸編號 • -4~-1: 各站編號(-1: 站1, -2: 站2, -3: 站3, -4: 站4)
ctrl_axbit [out]	指向儲存控制中軸資訊位及控制中站資訊位(0~0F000000FFFFFFFFh)的8位元組變數(8位元組×1)的指針 位0→軸1、位1→軸2...位31→軸32、位56→站1、位57→站2、位58→站3、位59→站4
err_code [out]	指向儲存斷開後重新連接出錯代碼的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(10秒)。
SSC_FUNC_ERR_STS_BIT_CCE	斷開不正確信號(CCE)變為了ON。

■要點

關於“斷開後重新連接出錯代碼”, 請參閱下述手冊的“斷開/重新連接功能”的“介面”。

📖MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscReconnectSSCNET(📖 64頁 sscReconnectSSCNET)

sscDisconnectSSCNETEx [MC300]

斷開通過斷開軸編號設置的SSCNET通信中軟元件設備(伺服驅動器等)及以後連接的SSCNET通信。

```
int sscDisconnectSSCNETEx (
    int board_id,
    int channel,
    int com_num,
    SLAVE_INFO *pSlaveInfo,
    unsigned short *err_code
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
com_num [in]	斷開軸編號(-16~64) <ul style="list-style-type: none"> • 1~64: 各軸編號 • -16~-1: 各站編號(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
pSlaveInfo [out]	指向儲存軟元件資訊(控制中軸資訊及控制中站資訊)的80位元組結構(80位元組×1)的指針 關於軟元件資訊結構, 請參閱下述章節。 ☞ 237頁 SLAVE_INFO結構[MC300]
err_code [out]	指向儲存斷開後重新連接出錯代碼的2位元組變數(2位元組×1)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(10秒)。
SSC_FUNC_ERR_STS_BIT_CCE	斷開不正確信號(CCE)變為了ON。

要點

關於“斷開後重新連接出錯代碼”, 請參閱下述手冊的“斷開/重新連接功能”的“介面”。

☞ MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscReconnectSSCNETEx (☞ 65頁 sscReconnectSSCNETEx [MC300])

sscGetControllingAxis [MC300]

獲取控制中軸資訊及控制中站資訊。

```
int sscGetControllingAxis(  
    int board_id,  
    int channel,  
    SLAVE_INFO *pSlaveInfo  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pSlaveInfo [out]	指向儲存軟元件資訊(控制中軸資訊及控制中站資訊)的80位元組結構(80位元組×1)的指針 關於軟元件資訊結構,請參閱下述章節。 ☞ 237頁 SLAVE_INFO結構[MC300]

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數,確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

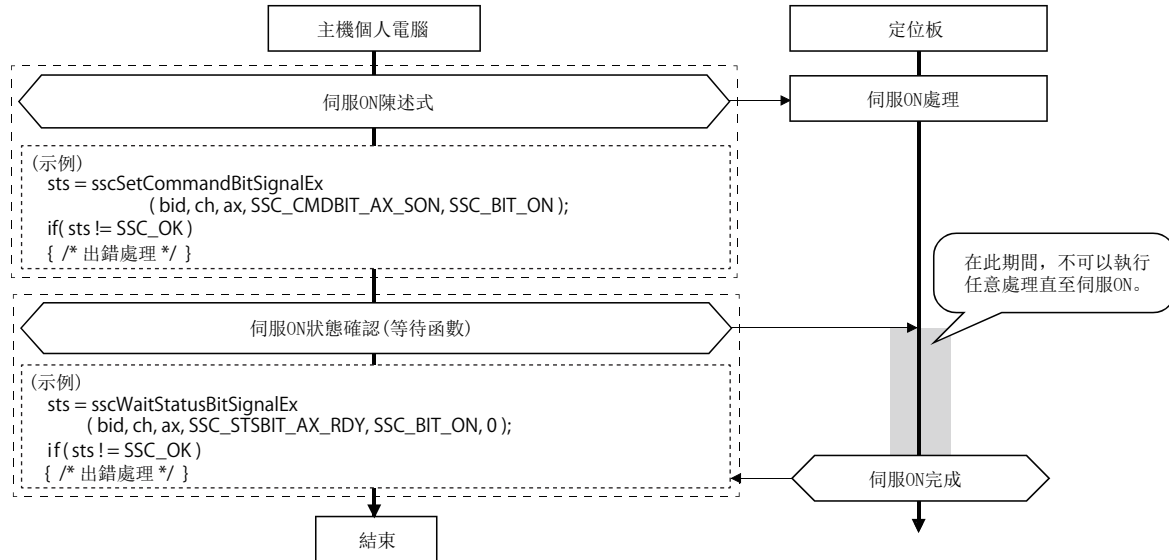
無。

4.6 陳述式/狀態函數群

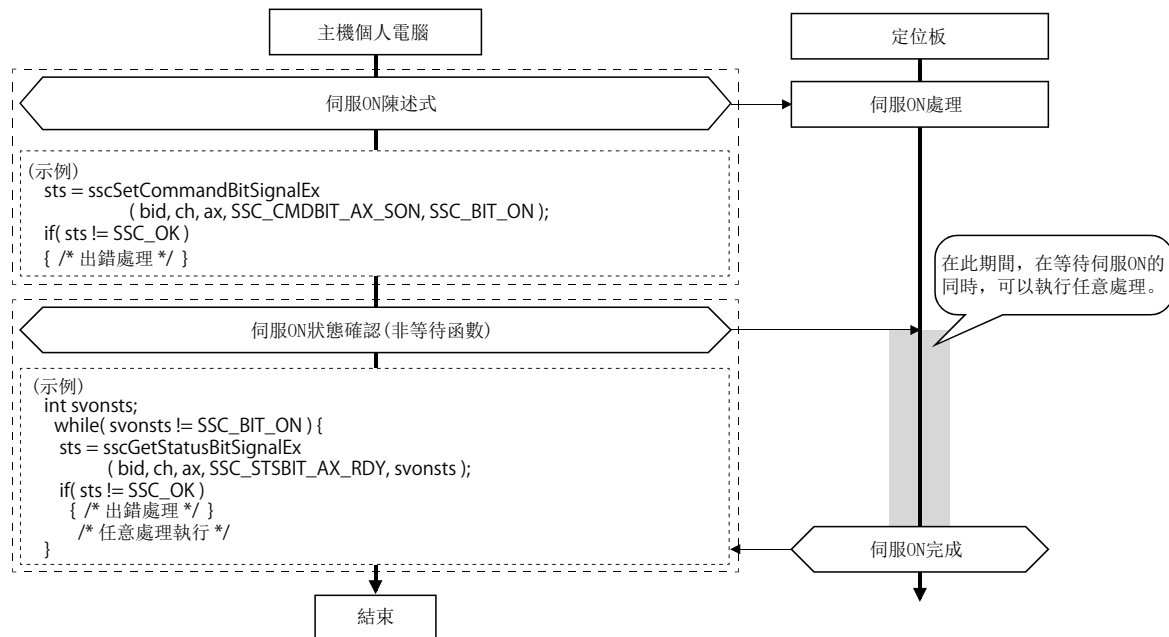
處理步驟

陳述式/狀態處理步驟的示例如下所示。

■通過等待函數進行伺服ON的情況下



■通過非等待函數進行伺服ON的情況下



sscSetCommandBitSignalEx

對指定的陳述式位進行ON或OFF。

```
int sscSetCommandBitSignalEx (  
    int board_id,  
    int channel,  
    int axnum,  
    int bitnum,  
    int bitdata  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) • 0: 系統陳述式位 • 1~64: 各軸陳述式位 • -16~-1: 各站陳述式位(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
bitnum [in]	陳述式位編號 關於陳述式位編號, 請參閱下述章節。 ☞ 238頁 位定義一覽
bitdata [in]	陳述式位資料 ■值 • SSC_BIT_OFF: 位OFF • SSC_BIT_ON: 位ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_ARGUMENT_MISMATCH	軸編號與陳述式位編號不匹配。 (例: 在軸編號中設置了“0”, 在陳述式位編號中設置了“SSC_CMDBIT_AX_SON”等情況下)

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

- sscGetStatusBitSignalEx ([☞ 71頁 sscGetStatusBitSignalEx](#))
- sscWaitStatusBitSignalEx ([☞ 72頁 sscWaitStatusBitSignalEx](#))

sscGetStatusBitSignalEx

獲取指定的狀態位元。

```
int sscGetStatusBitSignalEx (
    int board_id,
    int channel,
    int axnum,
    int bitnum,
    int *bitstatus
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統狀態位元 • 1~64: 各軸狀態位元 • -16~-1: 各站陳述式位(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
bitnum [in]	狀態位元編號 關於狀態位元編號，請參閱下述章節。 ☞ 238頁 位定義一覽
bitstatus [out]	指向儲存狀態位元資料的4位元組變數(4位元組×1)的指針 ■值 <ul style="list-style-type: none"> • SSC_BIT_OFF: 位OFF • SSC_BIT_ON: 位ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ARGUMENT_MISMATCH	軸編號與狀態位元編號不匹配。 (例: 在軸編號中設置了“0”，在狀態位元編號中設置了“SSC_STSBIT_AX_RDY”等情況下)

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetCommandBitSignalEx (☞ 70頁 sscSetCommandBitSignalEx)
- sscWaitStatusBitSignalEx (☞ 72頁 sscWaitStatusBitSignalEx)

sscWaitStatusBitSignalEx

等待至指定的狀態位元變成指定狀態。

```
int sscWaitStatusBitSignalEx (  
    int board_id,  
    int channel,  
    int axnum,  
    int bitnum,  
    int waitstatus,  
    int timeout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) • 0: 系統狀態位元 • 1~64: 各軸狀態位元 • -16~-1: 各站陳述式位(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
bitnum [in]	狀態位元編號 關於狀態位元編號, 請參閱下述章節。 ☞ 238頁 位定義一覽
waitstatus [in]	待機的位元狀態 ■值 • SSC_BIT_OFF: 位OFF • SSC_BIT_ON: 位ON
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中, 經過了指定的超時時間。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_ARGUMENT_MISMATCH	軸編號與狀態位元編號不匹配。 (例: 在軸編號中設置了“0”, 在狀態位元編號中設置了“SSC_STSBIT_AX_RDY”等情況下)

■要點

在超時時間中指定了“SSC_INFINITE”的情況下, 不進行超時檢查, 將無限等待直到狀態位元變為指定的“待機的位元狀態”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetCommandBitSignalEx (☞ 70頁 sscSetCommandBitSignalEx)
- sscGetStatusBitSignalEx (☞ 71頁 sscGetStatusBitSignalEx)

4.7 點位元表函數群

sscSetPointDataEx

設置點資料。

```
int sscSetPointDataEx (  
    int board_id,  
    int channel,  
    int axnum,  
    int pntnum,  
    PNT_DATA_EX *pPntDataEx  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號 (0~319 [MC200]/0~2047 [MC300])
pPntDataEx [in]	指向儲存點資料的32位元組結構 (32位元組×1) [MC200]/48位元組結構 (48位元組×1) [MC300]的指針 關於點資料結構，請參閱下述章節。 ☞ 215頁 PNT_DATA_EX結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。

■要點

不檢查要設置的點資料的內容。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetPointOffset (☞ 75頁 sscSetPointOffset)

sscCheckPointDataEx

獲取點資料。

```
int sscCheckPointDataEx (  
    int board_id,  
    int channel,  
    int axnum,  
    int pntnum,  
    PNT_DATA_EX *pPntDataEx  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號 (0~319 [MC200]/0~2047 [MC300])
pPntDataEx [out]	指向儲存點資料的32位元組結構 (32位元組×1) [MC200]/48位元組結構 (48位元組×1) [MC300]的指針 關於點資料結構，請參閱下述章節。 ☞ 215頁 PNT_DATA_EX結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscCheckPointOffset (☞ 76頁 sscCheckPointOffset)

sscSetPointOffset

設置點編號偏置。

```
int sscSetPointOffset (  
    int board_id,  
    int channel,  
    int axnum,  
    short offset  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
offset [in]	點編號偏置值(0~319 [MC200]/0~2047 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetPointDataEx([↩ 73頁](#) sscSetPointDataEx)

sscCheckPointOffset

獲取點編號偏置。

```
int sscCheckPointOffset (  
    int board_id,  
    int channel,  
    int axnum,  
    short *offset  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
offset [out]	指向儲存點位表偏置的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscCheckPointDataEx([↗](#) 74頁 sscCheckPointDataEx)

sscGetDrivingPointNumber

獲取運行中的點編號。

```
int sscGetDrivingPointNumber (  
    int board_id,  
    int channel,  
    int axnum,  
    short *driving_pnt  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
driving_pnt [out]	指向儲存運行中的點編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“自動運行”或“直線插補[MC200]/插補運行[MC300]”。

■要點

已獲取的運行中的點編號將儲存“運行中點編號+1”的值。停止中的情況下，將儲存“0”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscSetLatestPointNumber

設置最新陳述式點編號。在使用了點位表的循環方式的情況下使用。

```
int sscSetLatestPointNumber (  
    int board_id,  
    int channel,  
    int axnum,  
    short latest_point  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
latest_point [in]	最新陳述式點編號(1~320 [MC200]/1~2048 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

對於最新陳述式點編號的設置, 應設置“點編號+1”的值。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A6	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscAutoStart(↖ 87頁 sscAutoStart)
- sscSetPointDataEx(↖ 73頁 sscSetPointDataEx)

4.8 擋塊控制資料函數群

sscSetPressData

設置擋塊控制資料。

```
int sscSetPressData (  
    int board_id,  
    int channel,  
    int axnum,  
    PRESS_DATA *pPressData  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
pPressData [in]	指向儲存擋塊控制資料的32位元組結構 (32位元組×1) 的指針 關於擋塊控制資料結構，請參閱下述章節。 ☞ 223頁 PRESS_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

不檢查要設置的擋塊控制資料內容。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetPressData (☞ 80頁 sscGetPressData)

sscGetPressData

獲取擋塊控制資料。

```
int sscGetPressData (  
    int board_id,  
    int channel,  
    int axnum,  
    PRESS_DATA *pPressData  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
pPressData [out]	指向儲存擋塊控制資料的32位元組結構(32位元組×1)的指針 關於擋塊控制資料結構，請參閱下述章節。 ☞ 223頁 PRESS_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetPressData([☞ 79頁 sscSetPressData](#))

4.9 運行函數群

sscJogStart

啟動JOG運行。

實施運行中必要的設置並更改為JOG運行模式後，將運動啟動信號(ST)置為ON。

```
int sscJogStart (  
    int board_id,  
    int channel,  
    int axnum,  
    long speed,  
    short actime,  
    short dctime,  
    char dir  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
speed [in]	手動進給速度[速度單位](0~2147483647)
actime [in]	手動進給加速時間常數[ms](0~20000)
dctime [in]	手動進給減速時間常數[ms](0~20000)
dir [in]	移動方向 ■值 • SSC_DIR_PLUS: +方向 • SSC_DIR_MINUS: -方向

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

■要點

將運行啟動信號(ST)置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscJogStop(☞ 83頁 sscJogStop)
- sscJogStopNoWait(☞ 84頁 sscJogStopNoWait)
- sscDriveStop(☞ 94頁 sscDriveStop)
- sscDriveStopNoWait(☞ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(☞ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(☞ 97頁 sscDriveRapidStopNoWait)

sscJogStop

停止JOG運行。

將運行啟動信號(ST)置為OFF之後，在函數內部等待直至運行中信號(OP)變為OFF。(最大20秒)

運行中信號(OP)已經為OFF的情況下，立即完成。

```
int sscJogStop (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“JOG運行”。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式請求後的回應確認待機中，經過了超時時間(20秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

要點

避免在函數內部等待的情況下，應使用sscJogStopNoWait函數。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscJogStart (↗ 81頁 sscJogStart)
- sscJogStopNoWait (↗ 84頁 sscJogStopNoWait)
- sscDriveStop (↗ 94頁 sscDriveStop)
- sscDriveStopNoWait (↗ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop (↗ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait (↗ 97頁 sscDriveRapidStopNoWait)

sscJogStopNoWait

停止JOG運行。

將運行啟動信號(ST)置為OFF，返回運行中信號(OP)的狀態。

```
int sscJogStopNoWait (  
    int board_id,  
    int channel,  
    int axnum,  
    short *stpsts  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
stpsts [out]	指向儲存停止完成狀態的2位元組變數(2位元組×1)的指針 ■值 • SSC_DRIVING: 運行中 • SSC_DRIVE_FIN: 停止完成

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“JOG運行”。

■要點

使用在函數內部的等待的情況下，應使用sscJogStop函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscJogStart (↖ 81頁 sscJogStart)
- sscJogStop (↖ 83頁 sscJogStop)
- sscDriveStop (↖ 94頁 sscDriveStop)
- sscDriveStopNoWait (↖ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop (↖ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait (↖ 97頁 sscDriveRapidStopNoWait)

sscIncStart

啟動增量進給。

進行運行中必要的設置並更改為增量進給模式後，將高速運行啟動信號 (FST) 置為ON。

```
int sscIncStart (  
    int board_id,  
    int channel,  
    int axnum,  
    long distance,  
    long speed,  
    short actime,  
    short dctime  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
distance [in]	增量進給移動量[陳述式單位] (-2147483647~2147483647)
speed [in]	手動進給速度[速度單位] (0~2147483647)
actime [in]	手動進給加速時間常數[ms] (0~20000)
dctime [in]	手動進給減速時間常數[ms] (0~20000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中 (請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯 (E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1秒)。

■要點

- 在移動距離中指定了正值的情況下將向+方向移動，指定了負值的情況下將向-方向移動。
- 將高速運行啟動信號 (FST) 置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■参照

- `sscDriveStop` (🔗 94頁 `sscDriveStop`)
- `sscDriveStopNoWait` (🔗 95頁 `sscDriveStopNoWait`)
- `sscDriveRapidStop` (🔗 96頁 `sscDriveRapidStop`)
- `sscDriveRapidStopNoWait` (🔗 97頁 `sscDriveRapidStopNoWait`)

sscAutoStart

啟動自動運行。

實施運行中必要的設置並更改為自動運行模式後，將高速運行啟動信號(FST)置為ON。

```
int sscAutoStart (  
    int board_id,  
    int channel,  
    int axnum,  
    int point_s,  
    int point_e  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
point_s [in]	起始點編號(0~319 [MC200]/0~2047 [MC300])
point_e [in]	結束點編號(0~319 [MC200]/0~2047 [MC300])

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

要點

將高速運行啟動信號(FST)置為了ON後不進行回應確認。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscDriveStop(☞ 94頁 sscDriveStop)
- sscDriveStopNoWait(☞ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(☞ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(☞ 97頁 sscDriveRapidStopNoWait)

sscHomeReturnStart

啟動原點復位。

實施運行中必要的設置並更改為原點復位模式後，將高速運行啟動信號(FST)置為ON。

```
int sscHomeReturnStart (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

要點

將高速運行啟動信號(FST)置為了ON後不進行回應確認。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscDriveStop(☞ 94頁 sscDriveStop)
- sscDriveStopNoWait(☞ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(☞ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(☞ 97頁 sscDriveRapidStopNoWait)

sscLinearStart

啟動直線插補。

實施運行中必要的設置並更改為直線插補模式後，將高速運行啟動信號 (FST) 置為ON。

```
int sscLinearStart (  
    int board_id,  
    int channel,  
    int axnum,  
    int grpnum,  
    int point_s,  
    int point_e  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	基準軸編號 (1~32 [MC200]/1~64 [MC300])
grpnum [in]	插補組編號 (0~8 [MC200]/0~16 [MC300]) <ul style="list-style-type: none">• 0: 插補組可變有效時• 1~8: 插補組可變無效時 [MC200]• 1~16: 插補組可變無效時 [MC300]
point_s [in]	起始點編號 (0~319 [MC200]/0~2047 [MC300])
point_e [in]	結束點編號 (0~319 [MC200]/0~2047 [MC300])

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_STS_BIT_IPCH_ON	插補組可變設置有效。 應在插補組編號中設置“0”。
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	插補組可變設置無效。 應在插補組編號中設置“1~8 [MC200]/1~16 [MC300]”。
SSC_FUNC_ERR_SUB_AXIS_NUM	插補組可變有效時，點位表中已設置的插補軸編號為不正確的值。 <ul style="list-style-type: none">• 插補軸編號超出設置範圍• 插補軸編號與基準軸編號重覆• 插補軸編號相互重覆
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中 (請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯 (E□□□H)。 應使用 sscGetSystemStatusCode 函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用 sscGetAlarm 函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用 sscGetAlarm 函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1秒)。

■要點

將高速運行啟動信號(FST)置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.90	A9	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(🔗 94頁 sscDriveStop)
- sscDriveStopNoWait(🔗 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(🔗 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(🔗 97頁 sscDriveRapidStopNoWait)

sscInterpolationStart [MC300]

啟動插補運行。

實施運行中必要的設置並更改為插補運行模式後，將高速運行啟動信號(FST)置為ON。

```
int sscInterpolationStart (  
    int board_id,  
    int channel,  
    int axnum,  
    int grpnum,  
    int point_s,  
    int point_e  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	基準軸編號(1~64)
grpnum [in]	插補組編號(0~16) <ul style="list-style-type: none">• 0: 插補組可變有效時• 1~16: 插補組可變無效時
point_s [in]	起始點編號(0~2047)
point_e [in]	結束點編號(0~2047)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_STS_BIT_IPCH_ON	插補組可變設置有效。 應在插補組編號中設置“0”。
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	插補組可變設置無效。 應在插補組編號中設置“1~16”。
SSC_FUNC_ERR_SUB_AXIS_NUM	插補組可變有效時，點位表中已設置的插補軸編號為不正確的值。 <ul style="list-style-type: none">• 插補軸編號超出設置範圍• 插補軸編號與基準軸編號重覆• 插補軸編號相互重覆
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

■要點

將高速運行啟動信號(FST)置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(🔗 94頁 sscDriveStop)
- sscDriveStopNoWait(🔗 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(🔗 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(🔗 97頁 sscDriveRapidStopNoWait)

sscDataSetStart

啟動原點重新設置(資料設置)。

實施運行中必要的設置並更改為原點重新設置(資料設置)模式後，將高速運行啟動信號(FST)置為ON。

```
int sscDataSetStart (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

■要點

將高速運行啟動信號(FST)置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(☞ 94頁 sscDriveStop)
- sscDriveStopNoWait(☞ 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(☞ 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(☞ 97頁 sscDriveRapidStopNoWait)

sscDriveStop

停止運行。

將運行啟動信號(ST)置為OFF，並將運行停止信號(STP)置為ON後，在函數內部等待直至運行中信號(OP)變為OFF。

確認運行中信號(OP)的OFF後，將運行停止信號(STP)置為OFF。

```
int sscDriveStop (  
    int board_id,  
    int channel,  
    int axnum,  
    int timeout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
timeout [in]	超時時間[ms](0~65535)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了指定的超時時間。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

■要點

在超時時間中指定了“0”的情況下，超時時間將為20秒(20000ms)。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStopNoWait (🔗 95頁 sscDriveStopNoWait)
- sscDriveRapidStop (🔗 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait (🔗 97頁 sscDriveRapidStopNoWait)

sscDriveStopNoWait

停止運行。

將運行啟動信號(ST)置為OFF，並將運行停止信號(STP)置為了ON後，返回停止完成狀態。

```
int sscDriveStopNoWait (
    int board_id,
    int channel,
    int axnum,
    short *stpsts
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
stpsts [out]	指向儲存停止完成狀態的2位元組變數(2位元組×1)的指針 ■值 <ul style="list-style-type: none"> SSC_DRIVING: 運行中 SSC_DRIVE_FIN: 停止完成

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

■要點

- 使用在函數內部的等待的情況下，應使用sscDriveStop函數。
- 未確認停止完成狀態儲存變數變為“SSC_DRIVE_FIN”的情況下，運行停止信號(STP)將保持為ON不變。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(🔗 94頁 sscDriveStop)
- sscDriveRapidStop(🔗 96頁 sscDriveRapidStop)
- sscDriveRapidStopNoWait(🔗 97頁 sscDriveRapidStopNoWait)

sscDriveRapidStop

緊急停止運行。

將運行啟動信號(ST)置為OFF，並將運行緊急停止信號(RSTP)置為ON後，在函數內部等待直至運行中信號(OP)變為OFF。確認運行中信號(OP)的OFF後，將運行緊急停止信號(RSTP)置為OFF。

```
int sscDriveRapidStop (  
    int board_id,  
    int channel,  
    int axnum,  
    int timeout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
timeout [in]	超時時間[ms](0~65535)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了指定的超時時間。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

■要點

- 在超時時間中指定了“0”的情況下，超時時間將為20秒(20000ms)。
- 避免在函數內部等待的情況下，應使用sscDriveRapidStopNoWait函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(🔗 94頁 sscDriveStop)
- sscDriveStopNoWait(🔗 95頁 sscDriveStopNoWait)
- sscDriveRapidStopNoWait(🔗 97頁 sscDriveRapidStopNoWait)

sscDriveRapidStopNoWait

緊急停止運行。

將運行啟動信號(ST)置為OFF，並將運行緊急停止信號(RSTP)置為了ON後，返回停止完成狀態。

```
int sscDriveRapidStopNoWait (
    int board_id,
    int channel,
    int axnum,
    short *stpsts
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
stpsts [out]	指向儲存停止完成狀態的2位元組變數(2位元組×1)的指針 ■值 <ul style="list-style-type: none"> SSC_DRIVING: 運行中 SSC_DRIVE_FIN: 停止完成

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。

■要點

- 使用在函數內部的等待的情況下，應使用sscDriveRapidStop函數。
- 未確認停止完成狀態儲存變數變為“SSC_DRIVE_FIN”的情況下，運行緊急停止信號(RSTP)將保持為ON不變。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscDriveStop(🔗 94頁 sscDriveStop)
- sscDriveStopNoWait(🔗 95頁 sscDriveStopNoWait)
- sscDriveRapidStop(🔗 96頁 sscDriveRapidStop)

sscSetDriveMode

更改運行模式。

設置通過其他軸啟動功能進行其他軸啟動的軸的運行模式時使用。

```
int sscSetDriveMode (  
    int board_id,  
    int channel,  
    int axnum,  
    int drv_mode  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
drv_mode [in]	運行模式 ■值 • SSC_DRV_MODE_AUTO: 自動運行模式 • SSC_DRV_MODE_HOME: 原點復位模式 • SSC_DRV_MODE_JOG: JOG運行模式 • SSC_DRV_MODE_INC: 增量進給模式 • SSC_DRV_MODE_LINEAR: 直線插補模式[MC200] • SSC_DRV_MODE_INTERP: 插補運行模式[MC300] • SSC_DRV_MODE_DST: 原點重新設置模式

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。

要點

- 其他軸啟動的軸需要事先更改運行模式的情況下, 使用本API函數更改運行模式。
- 由於啟動運行的函數(sscAutoStart等)在函數內進行運行模式的更改, 因此無需調用本API函數。
- 由於定位板在運行啟動信號(ST)的受理完成時導入運行模式, 因此在調用本API函數後無需待機至運行模式被更改。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■参照

- [sscGetDriveMode](#) (100頁 [sscGetDriveMode](#))
- [sscSetOtherAxisStartData](#) (131頁 [sscSetOtherAxisStartData](#))

sscGetDriveMode

獲取運行模式。

```
int sscGetDriveMode (  
    int board_id,  
    int channel,  
    int axnum,  
    int *drv_mode  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
drv_mode [out]	指向儲存運行模式狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_DRV_MODE_NONE: 運行模式未設置 • SSC_DRV_MODE_AUTO: 自動運行模式中 • SSC_DRV_MODE_HOME: 原點復位模式中 • SSC_DRV_MODE_JOG: JOG運行模式中 • SSC_DRV_MODE_INC: 增量進給模式中 • SSC_DRV_MODE_LINEAR: 直線插補模式中[MC200] • SSC_DRV_MODE_INTERP: 插補運行模式中[MC300] • SSC_DRV_MODE_DST: 原點重新設置模式中

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

要點

其他軸啟動的軸需要事先更改運行模式的情況下, 使用本API函數確認運行模式。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscSetDriveMode([↖](#) 98頁 sscSetDriveMode)
- sscSetOtherAxisStartData([↖](#) 131頁 sscSetOtherAxisStartData)

sscGetDriveFinStatus

確認運行完成狀態。

```
int sscGetDriveFinStatus (  
    int board_id,  
    int channel,  
    int axnum,  
    int fin_type,  
    int *fin_status  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
fin_type [in]	運行完成類型 ■值 • SSC_FIN_TYPE_SMZ: 通過平滑停止的運行完成 • SSC_FIN_TYPE_CPO: 通過粗匹配的運行完成 • SSC_FIN_TYPE_INP: 通過進入位置停止的完成
fin_status [out]	指向儲存運行完成狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_FIN_STS_RDY: 運行啟動受理前 • SSC_FIN_STS_STP: 運行完成 • SSC_FIN_STS_MOV: 運行中 • SSC_FIN_STS_ALM_STP: 報警發生中(停止完成) • SSC_FIN_STS_ALM_MOV: 報警發生中(減速停止中)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 在自動運行、直線運行[MC200]/插補運行[MC300]以外, 將運行完成類型設置為“SSC_FIN_TYPE_CPO”的情況下, 運行完成確認類型無條件以“SSC_FIN_TYPE_SMZ”判定。
- 將運行完成類型設置為“SSC_FIN_TYPE_INP”的情況下, 進入位置信號(INP)為OFF中將始終變為“SSC_FIN_STS_MOV”。
- 互鎖停止・干擾檢查待機中的情況下, 運行完成狀態將變為“SSC_FIN_STS_MOV”。
- 在自動運行、直線插補[MC200]/插補運行[MC300]中將減速檢查方式設置為“SSC_SUBCMD_STOP_INP”的情況下, 即使將運行完成類型設置為“SSC_FIN_TYPE_SMZ”也將變為“SSC_FIN_TYPE_INP”。
- 不對系統報警及系統出錯進行確認。
- 由於本API函數的運行完成狀態通過運行完成類型進行判定, 因此可能會與定位板的運行中信號(OP)及運行完成信號(OPF)不同。
- JOG運行模式的的情況下, 運行完成狀態不變為“SSC_FIN_STS_RDY”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscChangeControlMode

切換伺服驅動器的控制模式。

```
int sscChangeControlMode (
    int board_id,
    int channel,
    int axnum,
    unsigned short ctrl_mode
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■值 SSC_CTRL_MODE_POSITION: 位置控制模式 SSC_CTRL_MODE_PRESS: 擋塊控制模式

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_STS_BIT_IFMO	介面模式中。 在介面模式中更改控制模式的情況下, 應使用sscIfmSetControlMode函數。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。
SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_CTLMCE	控制模式切換不正確信號(CTLMCE)變為了ON。


■要點

為了進行定位板的回應確認, 大約需要6個控制週期+4~6ms。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetPressData( 79頁 sscSetPressData)

4.10 更改函數群

sscChangeManualPosition

在增量進給中進行位置更改。

將位置更改信號(PCHG)置為ON，確認位置更改準備完成信號(PCF)或位置更改不正確信號(PCE)變為ON。

```
int sscChangeManualPosition (  
    int board_id,  
    int channel,  
    int axnum,  
    long position  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
position [in]	更改後的增量進給移動量[陳述式單位](-2147483647~2147483647)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“增量進給”。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PCE	位置更改不正確信號(PCE)變為了ON。
SSC_FUNC_ERR_CHG_POS_DIR	位置更改前與位置更改後的移動方向不同。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscChangeManualSpeed (↖ 108頁 sscChangeManualSpeed)
- sscChangeManualAccTime (↖ 110頁 sscChangeManualAccTime)
- sscChangeManualDecTime (↖ 112頁 sscChangeManualDecTime)

sscChangeAutoPosition

在自動運行中進行位置更改。

將位置更改信號(PCHG)置為ON，確認位置更改準備完成信號(PCF)或位置更改不正確信號(PCE)變為ON。

```
int sscChangeAutoPosition (
    int board_id,
    int channel,
    int axnum,
    int pntnum,
    long position
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號(0~319 [MC200]/0~2047 [MC300])
position [in]	更改後的位置資料[陳述式單位]

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“自動運行”。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PCE	位置更改不正確信號(PCE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetDrivingPointNumber(↖ 77頁 sscGetDrivingPointNumber)
- sscChangeAutoSpeed(↖ 109頁 sscChangeAutoSpeed)
- sscChangeAutoAccTime(↖ 111頁 sscChangeAutoAccTime)
- sscChangeAutoDecTime(↖ 113頁 sscChangeAutoDecTime)

sscChangeLinearPosition

在直線插補中進行位置更改。

```
int sscChangeLinearPosition (  
    int board_id,  
    int channel,  
    int axnum,  
    int grpnum,  
    int pntnum,  
    long *pPosition  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	基準軸編號(1~32 [MC200]/1~64 [MC300])
grpnum [in]	插補組編號(0~8 [MC200]/0~16 [MC300]) <ul style="list-style-type: none">• 0: 插補組可變有效時• 1~8: 插補組可變無效時[MC200]• 1~16: 插補組可變無效時[MC300]
pntnum [in]	點編號(0~319 [MC200]/0~2047 [MC300])
pPosition [in]	指向儲存更改後的位置資料[陳述式單位]的16位元組陣列(4位元組×4)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“直線插補[MC200]/插補運行[MC300]”。
SSC_FUNC_ERR_STS_BIT_IPCH_ON	插補組可變設置有效。 應在插補組編號中設置“0”。
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	插補組可變設置無效。 應在插補組編號中設置“1~8 [MC200]/1~16 [MC300]”。
SSC_FUNC_ERR_NOT_LIP_DRIVING	直線插補中不為運行中。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_PCE	位置更改不正確信號(PCE)變為了ON。

■要點

對於更改後的位置資料陣列, 應從直線插補物件軸中較小的軸編號開始按順序進行設置。(軸1、軸3的直線插補的情況下, 應將軸1的更改後的位置資料設置為pPosition[0], 將軸3的更改後的位置資料設置為pPosition[1]。不使用pPosition[2]與pPosition[3]。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.90	A9	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetDrivingPointNumber(🔗 77頁 sscGetDrivingPointNumber)
- sscChangeAutoSpeed(🔗 109頁 sscChangeAutoSpeed)
- sscChangeAutoAccTime(🔗 111頁 sscChangeAutoAccTime)
- sscChangeAutoDecTime(🔗 113頁 sscChangeAutoDecTime)

sscChangeManualSpeed

在JOG運行或增量進給中進行速度更改。

將速度更改信號(SCHG)置為ON，確認速度更改準備完成信號(SCF)或速度更改不正確信號(SCE)變為ON。

```
int sscChangeManualSpeed (  
    int board_id,  
    int channel,  
    int axnum,  
    long speed  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
speed [in]	更改後的手動進給速度[速度單位](0~2147483647)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“JOG運行”或“增量進給”。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_STS_BIT_SCE	速度更改不正確信號(SCE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscChangeManualPosition (↗ 104頁 sscChangeManualPosition)
- sscChangeManualAccTime (↗ 110頁 sscChangeManualAccTime)
- sscChangeManualDecTime (↗ 112頁 sscChangeManualDecTime)

sscChangeAutoSpeed

在自動運行或直線插補[MC200]/插補運行[MC300]中進行速度更改。

將速度更改信號(SCHG)置為ON，確認速度更改準備完成信號(SCF)或速度更改不正確信號(SCE)變為ON。

```
int sscChangeAutoSpeed (
    int board_id,
    int channel,
    int axnum,
    int pntnum,
    long speed
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號(0~319 [MC200]/0~2047 [MC300])
speed [in]	更改後的進給速度[速度單位](0~2147483647)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“自動運行”或“直線插補[MC200]/插補運行[MC300]”。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_SCE	速度更改不正確信號(SCE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetDrivingPointNumber(↖ 77頁 sscGetDrivingPointNumber)
- sscChangeAutoPosition(↖ 105頁 sscChangeAutoPosition)
- sscChangeAutoAccTime(↖ 111頁 sscChangeAutoAccTime)
- sscChangeAutoDecTime(↖ 113頁 sscChangeAutoDecTime)

sscChangeManualAccTime

在JOG運行或增量進給中進行加速時間常數更改。

將加速時間常數更改信號(TACHG)置為ON，確認加速時間常數更改準備完成信號(TACF)或加速時間常數更改不正確信號(TACE)變為ON。

```
int sscChangeManualAccTime (  
    int board_id,  
    int channel,  
    int axnum,  
    short acctime  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
acctime [in]	更改後的手動進給加速時間常數[ms](0~20000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“JOG運行”或“增量進給”。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_TACE	加速時間常數更改不正確信號(TACE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscChangeManualPosition([↖](#) 104頁 sscChangeManualPosition)
- sscChangeManualSpeed([↖](#) 108頁 sscChangeManualSpeed)
- sscChangeManualDecTime([↖](#) 112頁 sscChangeManualDecTime)

sscChangeAutoAccTime

在自動運行或直線插補[MC200]/插補運行[MC300]中進行加速時間常數更改。

將加速時間常數更改信號(TACHG)置為ON，確認加速時間常數更改準備完成信號(TACF)或加速時間常數更改不正確信號(TACE)變為ON。

```
int sscChangeAutoAccTime (
    int board_id,
    int channel,
    int axnum,
    int pntnum,
    short acctime
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號(0~319 [MC200]/0~2047 [MC300])
acctime [in]	更改後的加速時間常數[ms](0~20000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“自動運行”或“直線插補[MC200]/插補運行[MC300]”。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_TACE	加速時間常數更改不正確信號(TACE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetDrivingPointNumber(↖ 77頁 sscGetDrivingPointNumber)
- sscChangeAutoPosition(↖ 105頁 sscChangeAutoPosition)
- sscChangeAutoSpeed(↖ 109頁 sscChangeAutoSpeed)
- sscChangeAutoDecTime(↖ 113頁 sscChangeAutoDecTime)

sscChangeManualDecTime

在JOG運行或增量進給中進行減速時間常數更改。

將減速時間常數更改信號(TDCHG)置為ON，確認減速時間常數更改準備完成信號(TDCF)或減速時間常數更改不正確信號(TDCE)變為ON。

```
int sscChangeManualDecTime (  
    int board_id,  
    int channel,  
    int axnum,  
    short dectime  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
dectime [in]	更改後的手動進給減速時間常數[ms](0~20000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“JOG運行”或“增量進給”。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_TDCE	減速時間常數更改不正確信號(TDCE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscChangeManualPosition([↖](#) 104頁 sscChangeManualPosition)
- sscChangeManualSpeed([↖](#) 108頁 sscChangeManualSpeed)
- sscChangeManualAccTime([↖](#) 110頁 sscChangeManualAccTime)

sscChangeAutoDecTime

在自動運行或直線插補[MC200]/插補運行[MC300]中進行減速時間常數更改。

將減速時間常數更改信號(TDCHG)置為ON，確認減速時間常數更改準備完成信號(TDCF)或減速時間常數更改不正確信號(TDCE)變為ON。

```
int sscChangeAutoDecTime (  
    int board_id,  
    int channel,  
    int axnum,  
    int pntnum,  
    short dectime  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
pntnum [in]	點編號(0~319 [MC200]/0~2047 [MC300])
dectime [in]	更改後的減速時間常數[ms](0~20000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	運行模式不是“自動運行”或“直線插補[MC200]/插補運行[MC300]”。
SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_TDCE	減速時間常數更改不正確信號(TDCE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetDrivingPointNumber(↖ 77頁 sscGetDrivingPointNumber)
- sscChangeAutoPosition(↖ 105頁 sscChangeAutoPosition)
- sscChangeAutoSpeed(↖ 109頁 sscChangeAutoSpeed)
- sscChangeAutoAccTime(↖ 111頁 sscChangeAutoAccTime)

4.11 報警函數群

sscGetAlarm

獲取報警編號及報警詳細編號。

```
int sscGetAlarm (  
    int board_id,  
    int channel,  
    int axnum,  
    int alarm_type,  
    unsigned short *code,  
    unsigned short *detail_code  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) • 0: 系統報警 • 1~64: 軸報警 • -16~-1: 站報警(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
alarm_type [in]	報警類型 ■值 • SSC_ALARM_SYSTEM: 系統報警 • SSC_ALARM_SERVO: 伺服報警 • SSC_ALARM_OPERATION: 運行報警 • SSC_ALARM_UNIT: RIO模組報警 • SSC_ALARM_UNIT_CTRL: RIO控制報警
code [out]	指向儲存報警編號變數的2位元組變數(2位元組×1)的指針
detail_code [out]	指向儲存報警詳細編號變數的2位元組變數(2位元組×1)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ARGUMENT_MISMATCH	軸編號與報警類型不匹配。 (例: 在軸編號中設置了“0”, 在報警類型中設置了“SSC_ALARM_OPERATION”等情況下)

要點

未發生報警的情況下, 分別提取“0”作為報警編號及報警詳細編號。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscResetAlarm([↩](#) 115頁 sscResetAlarm)

sscResetAlarm

復位報警。

將報警復位信號置為ON後，確認了報警中信號變為OFF後，將報警復位信號置為OFF。

```
int sscResetAlarm (
    int board_id,
    int channel,
    int axnum,
    int alarm_type
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統報警 • 1~64: 軸報警 • -16~-1: 站報警(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
alarm_type [in]	報警類型 ■值 <ul style="list-style-type: none"> • SSC_ALARM_SYSTEM: 系統報警 • SSC_ALARM_SERVO: 伺服報警 • SSC_ALARM_OPERATION: 運行報警 • SSC_ALARM_UNIT: RIO模組報警 • SSC_ALARM_UNIT_CTRL: RIO控制報警

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	發生了無法復位的報警。 應確認報警原因後，實施適當的措施。
SSC_FUNC_ERR_ARGUMENT_MISMATCH	軸編號與報警類型不匹配。 (例: 在軸編號中設置了“0”，在報警類型中設置了“SSC_ALARM_OPERATION”等情況下)

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

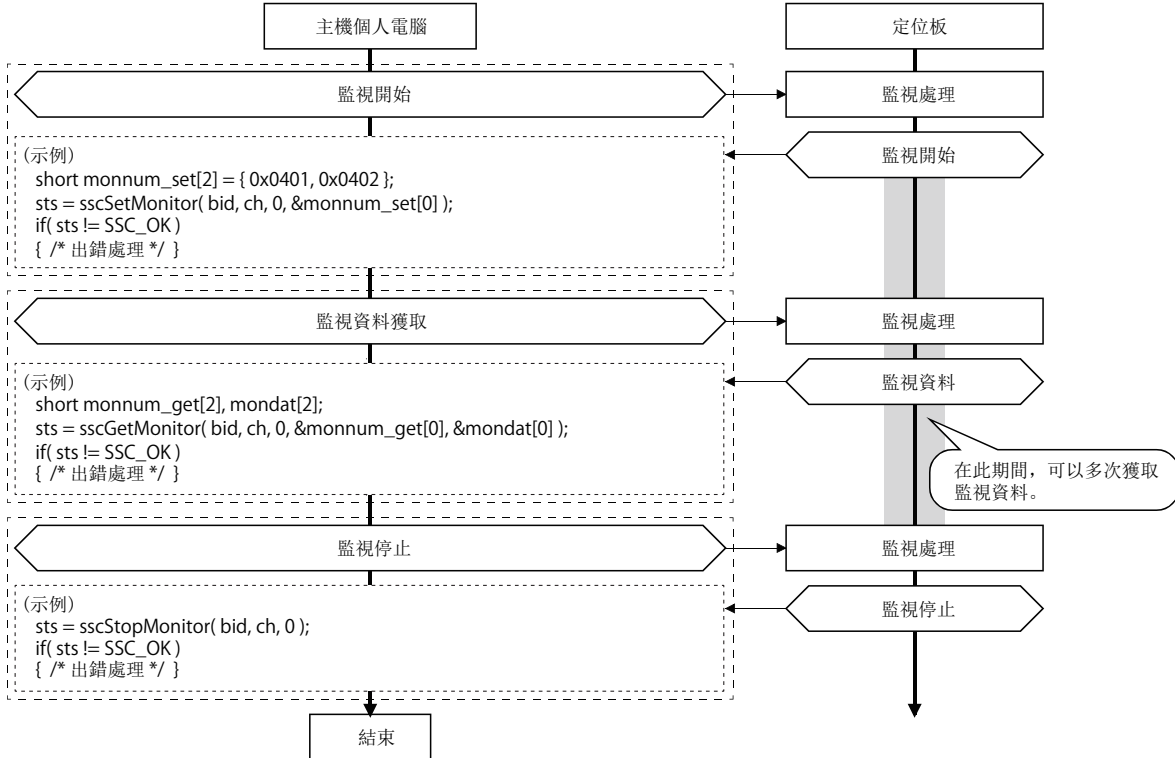
sscGetAlarm([↗](#) 114頁 sscGetAlarm)

4.12 通用監視函數群

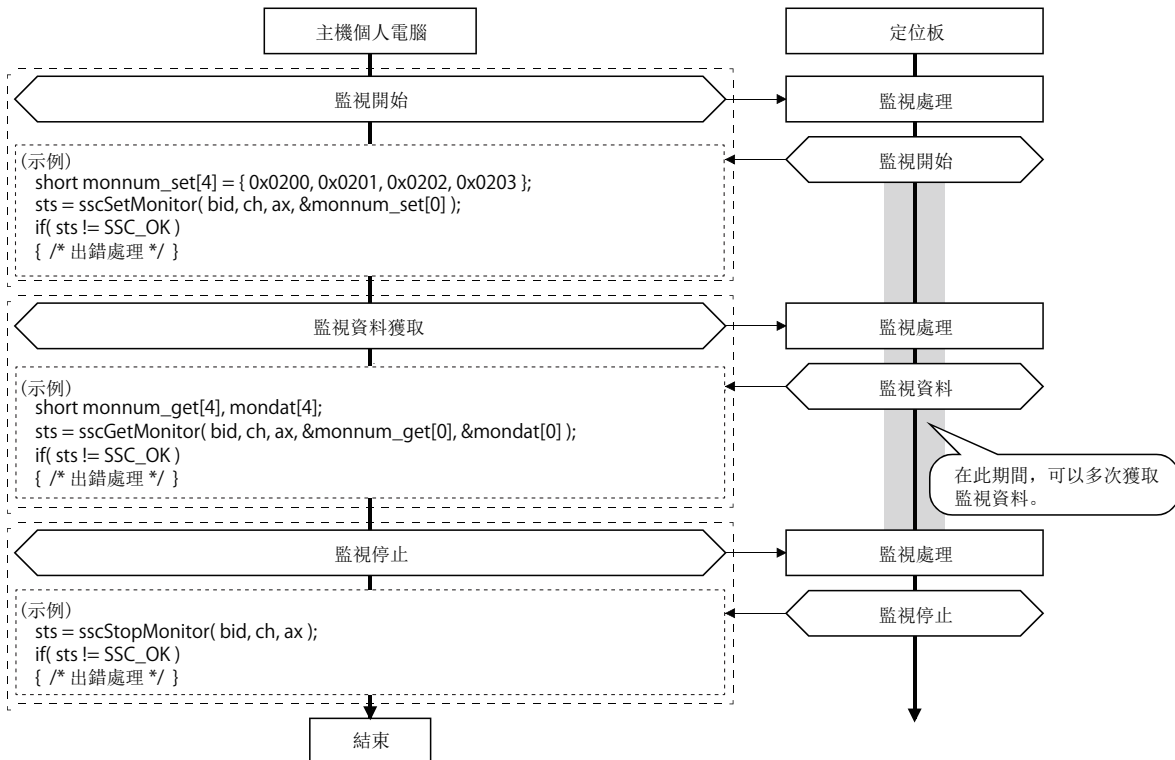
處理步驟

通用監視處理步驟的示例如下所示。

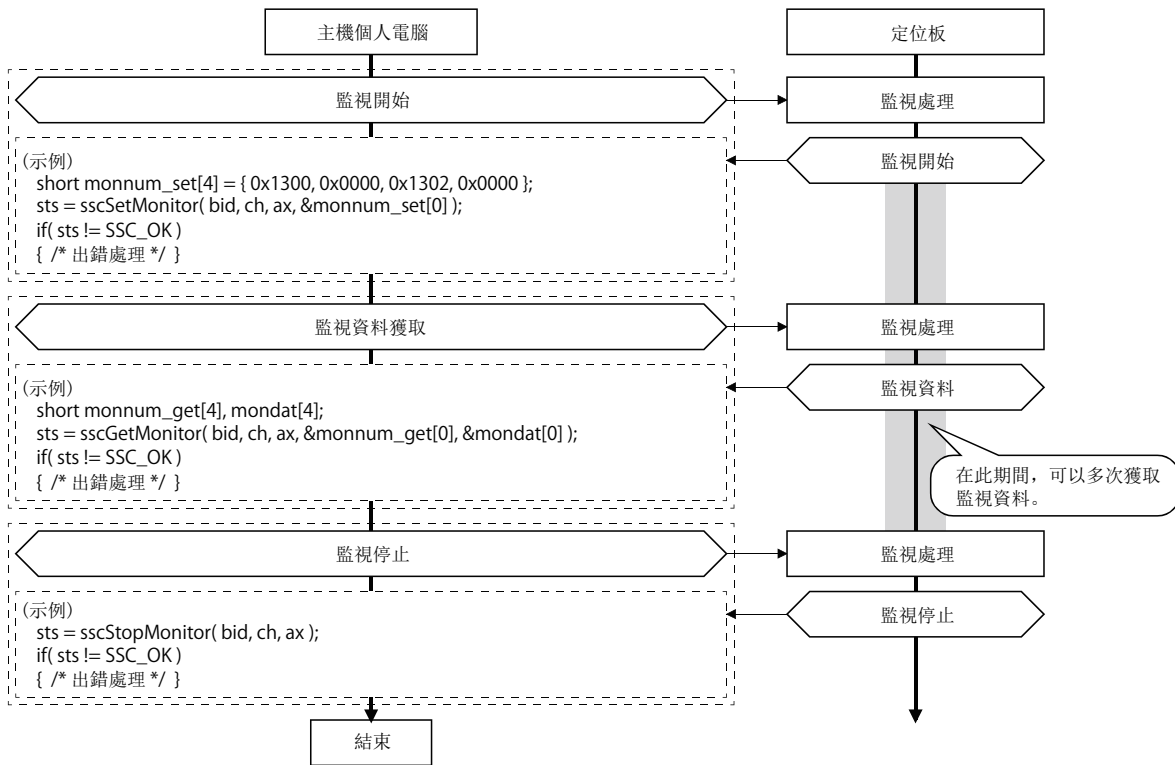
■獲取系統資訊的情況下



■獲取伺服資訊/運行資訊的情況下



■獲取運行資訊(雙字)的情況下



sscSetMonitor

開始監視。

在系統監視時設置2點的監視編號，在軸監視時設置4點的監視編號後，將監視陳述式信號(MON)置為ON，確認監視輸出中信號(MOUT)變為ON。

監視輸出中信號(MOUT)已經處於ON的情況下，將監視陳述式信號(MON)置為了OFF後，再次開始監視。

```
int sscSetMonitor (  
    int board_id,  
    int channel,  
    int axnum,  
    short *monnum  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統監視• 1~64: 軸監視• -16~-1: 站監視(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
monnum [in]	指向儲存監視編號(4點)的8位元組陣列(2位元組×4)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_STS_BIT_MER□ □=1~4: 監視編號(4點)的陣列編號	發生了監視編號出錯。 <ul style="list-style-type: none">• 在監視編號中設置了超出範圍的值。• 軸編號與監視編號不匹配。 (例: 在軸編號中設置了“系統監視”，在監視編號中設置了“軸監視”等情況下)
SSC_FUNC_ERR_STS_BIT_MESV	伺服驅動器未連接時將伺服資訊設置為監視編號。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

要點

- 監視運行資訊(雙字)的情況下，應使用監視編號1與3，並在監視編號2與4中設置“0”。
- 系統監視的情況下，由於不使用監視編號3與4，因此應設置“0”。
- 對於不使用的監視編號，應設置“0”。
- 為了進行定位板的回應確認，根據定位板的控制狀態需要控制週期~數ms。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscStopMonitor(↗ 119頁 sscStopMonitor)
- sscGetMonitor(↗ 120頁 sscGetMonitor)

sscStopMonitor

停止監視。

監視陳述式信號(MON)變為OFF，確認監視輸出中信號(MOUT)變為OFF。

```
int sscStopMonitor (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統監視• 1~64: 軸監視• -16~-1: 站監視(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_MONITOR_STOP	監視已停止中。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

■要點

為了進行定位板的回應確認，根據定位板的控制狀態需要控制週期~數ms。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetMonitor([↩](#) 118頁 sscSetMonitor)

sscGetMonitor

獲取監視資料。

```
int sscGetMonitor (  
    int board_id,  
    int channel,  
    int axnum,  
    short *monnum,  
    short *mondata  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統監視• 1~64: 軸監視• -16~-1: 站監視 (-1: 站1, -2: 站2, -3: 站3, -4: 站4 ••• -15: 站15, -16: 站16)
monnum [out]	指向儲存監視編號的8位元組陣列 (2位元組×4)的指針
mondata [out]	指向儲存監視資料的8位元組陣列 (2位元組×4)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_MONITOR	監視尚未開始。

要點

- 系統監視的情況下, 在監視編號及監視資料的3與4中不儲存值。
- 獲取當前位置、F/B位置、外部信號狀態、移動速度、F/B移動速度、電流回饋的情況下, 通過使用高速監視函數群, 可以更高地獲取資料。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscSetMonitor ([↩](#) 118頁 sscSetMonitor)

4.13 高速監視函數群

sscGetCurrentCmdPositionFast

獲取當前位置。

```
int sscGetCurrentCmdPositionFast (  
    int board_id,  
    int channel,  
    int axnum,  
    long *position  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
position [out]	指向儲存當前位置[陳述式單位]的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

獲取當前位置時, 使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetCurrentFbPositionFast([↗](#) 122頁 sscGetCurrentFbPositionFast)

sscGetCurrentFbPositionFast

獲取F/B位置。

```
int sscGetCurrentFbPositionFast (  
    int board_id,  
    int channel,  
    int axnum,  
    long *position  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
position [out]	指向儲存F/B位置[陳述式單位]的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。


■要點

獲取F/B位置時, 使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetCurrentCmdPositionFast ( 121頁 sscGetCurrentCmdPositionFast)

sscGetIoStatusFast

獲取外部信號 (LSP、LSN、DOG信號) 狀態。

```
int sscGetIoStatusFast (
    int board_id,
    int channel,
    int axnum,
    short *din
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
din [out]	指向儲存外部信號狀態的2位元組變數 (2位元組×1) 的指針 使用各值的邏輯或設置已獲取的資料。 ■值 <ul style="list-style-type: none"> • SSC_BIT_LSP: +側極限開關信號 (LSP) 為ON • SSC_BIT_LSN: -側極限開關信號 (LSN) 為ON • SSC_BIT_DOG: 近點狗輸入信號 (DOG) 為ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

獲取外部信號狀態時, 使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetCmdSpeedFast

獲取移動速度。

```
int sscGetCmdSpeedFast (  
    int board_id,  
    int channel,  
    int axnum,  
    long *speed  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
speed [out]	指向儲存移動速度[速度單位]的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

獲取移動速度時，使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetFbSpeedFast ([↩](#) 125頁 sscGetFbSpeedFast)

sscGetFbSpeedFast

獲取F/B移動速度。

```
int sscGetFbSpeedFast (  
    int board_id,  
    int channel,  
    int axnum,  
    long *speed  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
speed [out]	指向儲存F/B移動速度[速度單位]的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

獲取F/B移動速度時，使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetCmdSpeedFast (↖ 124頁 sscGetCmdSpeedFast)

sscGetCurrentFbFast

獲取電流回饋。

```
int sscGetCurrentFbFast (  
    int board_id,  
    int channel,  
    int axnum,  
    short *currentFb  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
currentFb [out]	指向儲存電流回饋[0.1%]的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

獲取電流回饋時，使用定位板的“高速監視功能”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetPositionDroopFast

獲取位置下降。

```
int sscGetPositionDroopFast (  
    int board_id,  
    int channel,  
    int axnum,  
    long *position_dp  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
position_dp [out]	指向儲存位置下降[pulse]的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 獲取位置下降時, 使用定位板的“高速監視功能”。
- 本功能僅介面模式支援。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 60	A4	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

無。

4.14 用戶看門狗函數群

sscWdEnable

啟用用戶看門狗功能。

```
int sscWdEnable (  
    int board_id,  
    int channel,  
    unsigned short wdcnt  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
wdcnt [in]	看門狗計時器開始計數器

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_ENABLE_WDT	用戶看門狗功能已啟用。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWdDisable([↗](#) 129頁 sscWdDisable)

sscWdDisable

禁用用戶看門狗功能。

```
int sscWdDisable (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_DISABLE_WDT	用戶看門狗功能已禁用。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWdEnable([↩](#) 128頁 sscWdEnable)

sscChangeWdCounter

更新看門狗檢查計數器(+1)。

```
int sscChangeWdCounter (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWdEnable([↗](#) 128頁 sscWdEnable)

4.15 其他軸啟動函數群

sscSetOtherAxisStartData

設置其他軸啟動資料。

```
int sscSetOtherAxisStartData (  
    int board_id,  
    int channel,  
    int oas_num,  
    OAS_DATA *pOasData  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
oas_num [in]	其他軸啟動表編號 (1~32 [MC200]/1~64 [MC300])
pOasData [in]	指向儲存其他軸啟動資料的104位元組結構 (104位元組×1) [MC200]/128位元組結構 (128位元組×1) [MC300]的指針 關於其他軸啟動資料結構，請參閱下述章節。 ☞ 219頁 OAS_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

不檢查設置的其他軸啟動資料內容。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetOtherAxisStartData ([☞ 132頁 sscGetOtherAxisStartData](#))

sscGetOtherAxisStartData

獲取其他軸啟動表資料。

```
int sscGetOtherAxisStartData (  
    int board_id,  
    int channel,  
    int oas_num,  
    OAS_DATA *pOasData  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
oas_num [in]	其他軸啟動表編號 (1~32 [MC200]/1~64 [MC300])
pOasData [out]	指向儲存其他軸啟動資料的104位元組結構 (104位元組×1) [MC200]/128位元組結構 (128位元組×1) [MC300]的指針 關於其他軸啟動資料結構，請參閱下述章節。 ☞ 219頁 OAS_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetOtherAxisStartData ([☞ 131頁 sscSetOtherAxisStartData](#))

sscOtherAxisStartAbortOn

將其他軸啟動中斷信號(OSSTP□)置為ON，中斷其他軸啟動。

```
int sscOtherAxisStartAbortOn (  
    int board_id,  
    int channel,  
    int oas_num  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
oas_num [in]	其他軸啟動表編號(1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

■要點

- 將其他軸啟動中斷信號(OSSTP□)置為了ON後不進行回應確認。
- 對於其他軸啟動狀態的確認，應調用sscGetOtherAxisStartStatus函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscOtherAxisStartAbortOff(☞ 134頁 sscOtherAxisStartAbortOff)
- sscGetOtherAxisStartStatus(☞ 135頁 sscGetOtherAxisStartStatus)

sscOtherAxisStartAbortOff

將其他軸啟動中斷信號 (OSSTP□) 置為OFF，解除其他軸啟動的中斷。

```
int sscOtherAxisStartAbortOff (  
    int board_id,  
    int channel,  
    int oas_num  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
oas_num [in]	其他軸啟動表編號 (1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1秒)。

■要點

對於其他軸啟動狀態的確認，應調用sscGetOtherAxisStartStatus函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscOtherAxisStartAbortOn (🔗 133頁 sscOtherAxisStartAbortOn)
- sscGetOtherAxisStartStatus (🔗 135頁 sscGetOtherAxisStartStatus)

sscGetOtherAxisStartStatus

獲取其他軸啟動狀態(其他軸啟動狀態位元)。

```
int sscGetOtherAxisStartStatus (
    int board_id,
    int channel,
    int oas_num,
    short *status
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
oas_num [in]	其他軸啟動表編號(1~32 [MC200]/1~64 [MC300])
status [out]	指向儲存其他軸啟動狀態位元的2位元組變數(2位元組×1)的指針 使用各值的邏輯或設置已獲取的資料。 ■值 <ul style="list-style-type: none"> • SSC_BIT_OSOP: 其他軸啟動預告 • SSC_BIT_OSFIN: 其他軸啟動完成 • SSC_BIT_OSERR: 其他軸啟動未完成

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

4.16 通過位置中斷函數群

sscSetIntPassPositionData

設置通過位置中斷條件資料。

```
int sscSetIntPassPositionData (  
    int board_id,  
    int channel,  
    int pass_num,  
    unsigned long pass_option,  
    long pass_data  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
pass_num [in]	通過位置條件編號 (1~64 [MC200]/1~128 [MC300])
pass_option [in]	通過位置選單 應使用各值的邏輯或設置要設置的資料。 ■值 [通過方向] • SSC_PASS_DIR_PLUS: +方向通過位置中斷輸出 • SSC_PASS_DIR_MINUS: -方向通過位置中斷輸出 [判定條件] • SSC_PASS_JUDGE_CMD_POS: 當前位置 • SSC_PASS_JUDGE_FB_POS: F/B位置 [中斷條件] [MC300] • SSC_PASS_CANCEL_INP: 進入位置信號 (INP) ON • SSC_PASS_CANCEL_NONE: 無中斷
pass_data [in]	通過位置資料 [陳述式單位] (-2147483648~2147483647)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

對於通過位置選單的判定條件, 僅通過位置條件開始編號的判定條件有效。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscCheckIntPassPositionData (📄 137頁 sscCheckIntPassPositionData)

sscCheckIntPassPositionData

獲取通過位置中斷條件資料。

```
int sscCheckIntPassPositionData (  
    int board_id,  
    int channel,  
    int pass_num,  
    unsigned long *pass_option,  
    long *pass_position  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pass_num [in]	通過位置條件編號(1~64 [MC200]/1~128 [MC300])
pass_option [out]	指向儲存通過位置選單的4位元組變數(4位元組×1)的指針
pass_position [out]	指向儲存通過位置資料[陳述式單位]的4位元組變數(4位元組×1)的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

要點

無。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscSetIntPassPositionData (↩ 136頁 sscSetIntPassPositionData)

sscSetStartingPassNumber

設置通過位置條件開始編號/結束編號。

```
int sscSetStartingPassNumber (  
    int board_id,  
    int channel,  
    int axnum,  
    int pass_start,  
    int pass_end  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
pass_start [in]	通過位置條件開始編號 (1~64 [MC200]/1~128 [MC300])
pass_end [in]	通過位置條件結束編號 (1~64 [MC200]/1~128 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscGetExecutingPassNumber

獲取執行中的通過位置條件編號。

```
int sscGetExecutingPassNumber (
    int board_id,
    int channel,
    int axnum,
    short *executing_pass
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
executing_pass [out]	指向儲存執行中的通過位置條件編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 通過位置條件完成後的執行中通過位置條件編號中儲存通過位置條件的結束編號。
- 通過位置條件未完成後的執行中通過位置條件編號中儲存已中斷的通過位置條件編號。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

4.17 採樣函數群

sscStartSampling

開始採樣。

將採樣開始信號 (SMPS) 置為ON。

```
int sscStartSampling (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_START_SAMPLING	採樣開始信號 (SMPS) 處於ON中。 應使用sscStopSampling函數停止採樣。

■要點

將採樣開始信號 (SMPS) 置為了ON後不進行回應確認。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscStopSampling (↗ 141頁 sscStopSampling)
- sscSetSamplingParameter (↗ 142頁 sscSetSamplingParameter)
- sscGetSamplingStatus (↗ 145頁 sscGetSamplingStatus)

sscStopSampling

停止採樣。

將採樣開始信號 (SMPS) 置為OFF，並等待直至相關的各狀態信號 (採樣觸發等待 (SMPW)、採樣中 (SMPO)、採樣完成 (SMPF)、採樣不正確 (SMPE)) 全部變為OFF。

```
int sscStopSampling (
    int board_id,
    int channel
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1秒)。
SSC_FUNC_ERR_ALREADY_STOP_SAMPLING	採樣已經在停止中。

■要點

採樣完成信號 (SMPF) 或採樣不正確信號 (SMPE) 變為了ON後，應通過本API函數停止採樣。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscStartSampling ([↩](#) 140頁 sscStartSampling)

sscSetSamplingParameter

寫入採樣參數。

```
int sscSetSamplingParameter (  
    int board_id,  
    int channel,  
    short prm_num,  
    long prm_data  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
prm_num [in]	採樣設置寫入編號
prm_data [in]	採樣設置寫入資料

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_SWEN	在採樣設置寫入編號中設置了超出範圍的值。
SSC_FUNC_ERR_STS_BIT_SWED	在採樣設置寫入資料中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_NUM	採樣設置寫入編號的陳述式與狀態不一致。
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_DATA	採樣設置寫入資料的陳述式與狀態不一致。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetSamplingParameter ([↩](#) 143頁 sscGetSamplingParameter)

sscGetSamplingParameter

讀取採樣參數。

```
int sscGetSamplingParameter (  
    int board_id,  
    int channel,  
    short prm_num,  
    long *prm_data  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
prm_num [in]	採樣設置讀取編號
prm_data [out]	指向儲存採樣設置讀取資料的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_STS_BIT_SREN	在採樣設置讀取編號中設置了超出範圍的值。
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	採樣設置讀取編號的陳述式與狀態不一致。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSetSamplingParameter([↗](#) 142頁 sscSetSamplingParameter)

sscGetSamplingError

獲取採樣出錯資訊。

```
int sscGetSamplingError (  
    int board_id,  
    int channel,  
    SMP_ERR *pSmpErr  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pSmpErr [out]	指向儲存採樣出錯資訊的32位元組結構(32位元組×1) [MC200]/48位元組結構(48位元組×1) [MC300]的指針 關於採樣出錯資訊結構，請參閱下述章節。 ☞ 224頁 SMP_ERR結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetSamplingStatus([☞ 145頁 sscGetSamplingStatus](#))

sscGetSamplingStatus

獲取採樣執行狀態 (採樣狀態位元、採樣完成頁碼)。

```
int sscGetSamplingStatus (  
    int board_id,  
    int channel,  
    char *status,  
    short *fin_page  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
status [out]	指向儲存採樣狀態位元的1位元組變數 (1位元組×1) 的指針 ■值 • SSC_BIT_SMPW: 採樣觸發等待 • SSC_BIT_SMP0: 採樣中 • SSC_BIT_SMPF: 採樣完成 • SSC_BIT_SMPPE: 採樣不正確
fin_page [out]	指向儲存採樣完成頁碼的2位元組變數 (2位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetSamplingError (↗ 144頁 sscGetSamplingError)
- sscGetSamplingData (↗ 146頁 sscGetSamplingData)

sscGetSamplingData

獲取採樣讀取有效點數及採樣資料(32點[MC200]/128點[MC300])。
此時，不對採樣讀取有效點數進行確認。

```
int sscGetSamplingData (  
    int board_id,  
    int channel,  
    int page_num,  
    short *valid_num,  
    SMP_DATA *pSmpData  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
page_num [in]	採樣讀取頁碼(0~256 [MC200]/0~512 [MC300])
valid_num [out]	指向採樣讀取有效點數變數的指針
pSmpData [out]	指向儲存採樣資料的4224位元組結構(132位元組×32) [MC200]/16896位元組結構(132位元組×128) [MC300]的指針 關於採樣資料資訊結構，請參閱下述章節。 ☞ 226頁 SMP_DATA結構

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。

要點

- 在採樣開始前，應通過本API函數在採樣讀取頁碼中設置“0”。
- MR-MC3_ _的API版本1.00的情況下，無法獲取32769點(頁碼257)及以後的採樣資料。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscStartSampling(☞ 140頁 sscStartSampling)
- sscGetSamplingStatus(☞ 145頁 sscGetSamplingStatus)

4.18 日誌函數群

sscStartLog

開始日誌。

通過日誌陳述式信號 (LOGC) ON開始日誌，確認日誌動作中信號 (LOGO) 變為ON。

```
int sscStartLog (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1秒)。
SSC_FUNC_ERR_ALREADY_START_LOG	日誌陳述式信號 (LOGC) 處於ON中。 應使用sscStopLog函數停止日誌。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscStopLog ([↗](#) 148頁 sscStopLog)

sscStopLog

停止日誌。

通過日誌陳述式信號 (LOGC) OFF 停止日誌，確認日誌動作中信號 (LOGO) 變為 OFF。

```
int sscStopLog (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後的回應確認中，經過了超時時間 (1 秒)。
SSC_FUNC_ERR_ALREADY_STOP_LOG	日誌已經在停止中。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscStartLog([↗](#) 147頁 sscStartLog)

sscCheckLogStatus

獲取日誌動作狀態。

```
int sscCheckLogStatus (  
    int board_id,  
    int channel,  
    int *status  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
status [out]	指向儲存日誌動作狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_LOGO_OFF: 日誌動作中信號(LOGO)為OFF • SSC_LOGO_ON: 日誌動作中信號(LOGO)為ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscStartLog([↗](#) 147頁 sscStartLog)

sscCheckLogEventNum

獲取日誌資料中記錄的日誌資料有效事件數。

```
int sscCheckLogEventNum (  
    int board_id,  
    int channel,  
    int *eventnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
eventnum [out]	指向儲存日誌資料有效事件數的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscStartLog([↗](#) 147頁 sscStartLog)

sscReadLogData

讀取16個事件的日誌資料。

```
int sscReadLogData (  
    int board_id,  
    int channel,  
    int page_num,  
    LOG_DATA *pLogData  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
page_num [in]	日誌資料讀取頁碼(1~256)
pLogData [out]	指向儲存日誌資料的256位元組結構(16位元組×16)的指針 關於日誌資料資訊結構，請參閱下述章節。 ☞ 228頁 LOG_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_LOGGING	日誌動作中信號(LOGO)處於ON中。 應使用sscStopLog函數停止日誌。
SSC_FUNC_ERR_STS_BIT_LOGRE	日誌資料讀取不正確信號(LOGRE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscClearLogData([☞ 152頁 sscClearLogData](#))

sscClearLogData

清除(初始化)日誌資料。

```
int sscClearLogData (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_NOW_LOGGING	日誌動作中信號(LOGO)處於ON中。 應使用sscStopLog函數停止日誌。
SSC_FUNC_ERR_STS_BIT_LOGIE	日誌資料初始化不正確信號(LOGIE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscReadLogData([↩](#) 151頁 sscReadLogData)

sscGetAlarmHistoryData

讀取4個事件的報警履歷資料。

```
int sscGetAlarmHistoryData (
    int board_id,
    int channel,
    int page_num,
    ALH_DATA *pAlhData
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
page_num [in]	報警履歷資料讀取頁碼(1~512)
pAlhData [out]	指向儲存報警履歷資料的256位元組結構(64位元組×4)的指針 關於報警履歷資料資訊結構，請參閱下述章節。 ☞ 229頁 ALH_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(1秒)。
SSC_FUNC_ERR_UNSUPPORTED_ALH	報警履歷功能未支援。
SSC_FUNC_ERR_STS_BIT_ALHRE	報警履歷讀取不正確信號(ALHRE)變為了ON。

■要點

對於已獲取的校驗和的確認，應通過用戶程式實施。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscClearAlarmHistoryData([☞ 155頁 sscClearAlarmHistoryData](#))

sscCheckAlarmHistoryEventNum

獲取報警履歷資料中記錄的報警履歷資料有效事件數。

```
int sscCheckAlarmHistoryEventNum (  
    int board_id,  
    int channel,  
    int *eventnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
eventnum [out]	指向儲存報警履歷資料有效事件數的4位元組變數(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNUPPORT_ALH	報警履歷功能未支援。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 40	A3	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

無。

sscClearAlarmHistoryData

清除(初始化)報警履歷資料。

```
int sscClearAlarmHistoryData (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(2秒[MC200]/10秒[MC300])。
SSC_FUNC_ERR_UNUPPORT_ALH	報警履歷功能未支援。
SSC_FUNC_ERR_STS_BIT_ALHIE	報警履歷初始化不正確信號(ALHIE)變為了ON。

■要點

對於報警履歷初始化功能, 由於定位板的快閃記憶體寫入次數有限制, 因此應設置為所需最低次數的調用。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetAlarmHistoryData([↩](#) 153頁 sscGetAlarmHistoryData)

4.19 數位輸入輸出函數群

sscGetDigitalInputDataBit

以1點為單位獲取指定的數位輸入DI資料。

```
int sscGetDigitalInputDataBit (  
    int board_id,  
    int channel,  
    int din_num,  
    int *din  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
din_num [in]	數位輸入編號 (0~1023)
din [out]	指向儲存數位輸入的DI資料狀態的4位元組變數 (4位元組×1) 的指針 ■值 • SSC_BIT_OFF: 輸入信號OFF • SSC_BIT_ON: 輸入信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetDigitalInputDataWord ([↩](#) 157頁 sscGetDigitalInputDataWord)

sscGetDigitalInputDataWord

以16點為單位獲取指定的數位輸入DI資料。

```
int sscGetDigitalInputDataWord (
    int board_id,
    int channel,
    int din_word_num,
    unsigned short *din
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
din_word_num [in]	數位輸入字編號(0~63)
din [out]	指向儲存數位輸入的DI資料狀態的2位元組變數(2位元組×1)的指針 (按各位0: 輸出OFF, 1: 輸出ON) 位0→DI_0000, ...位15→DI_8000

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetDigitalInputDataBit ([🔗](#) 156頁 sscGetDigitalInputDataBit)

sscSetDigitalOutputDataBit

以1點為單位設置指定的數位輸出DO資料。

```
int sscSetDigitalOutputDataBit (  
    int board_id,  
    int channel,  
    int dout_num,  
    int dout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
dout_num [in]	數位輸出編號(0~1023)
dout [in](0~1)	數位輸出的DO資料 ■值 • SSC_BIT_OFF: 輸出信號OFF • SSC_BIT_ON: 輸出信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~5: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。

■要點

本API函數使用定位板的排他控制功能進行數位輸出的設置。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetDigitalOutputDataWord([↗](#) 159頁 sscSetDigitalOutputDataWord)
- sscGetDigitalOutputDataBit([↗](#) 160頁 sscGetDigitalOutputDataBit)

sscSetDigitalOutputDataWord

以16點為單位設置指定的數位輸出DO資料。

```
int sscSetDigitalOutputDataWord (
    int board_id,
    int channel,
    int dout_word_num,
    unsigned short dout
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
dout_word_num [in]	數位輸出字編號(0~63)
dout [in] (0000h~FFFFh)	數位輸出的DO資料 (按各位0: 輸出OFF, 1: 輸出ON) 位0→DO_□□□0, . . . 位→DO_8□□□

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~5: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。

■要點

本API函數使用定位板的排他控制功能進行數位輸出的設置。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetDigitalOutputDataBit([↖](#) 158頁 sscSetDigitalOutputDataBit)
- sscGetDigitalOutputDataWord([↖](#) 161頁 sscGetDigitalOutputDataWord)

sscGetDigitalOutputDataBit

以1點為單位獲取指定的數位輸出DO資料。

```
int sscGetDigitalOutputDataBit (  
    int board_id,  
    int channel,  
    int dout_num,  
    int *dout  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
dout_num [in]	數位輸出編號 (0~1023)
dout [out]	指向儲存數位輸出的DO資料狀態的4位元組變數 (4位元組×1) 的指針 ■值 • SSC_BIT_OFF: 輸出信號OFF • SSC_BIT_ON: 輸出信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetDigitalOutputDataBit ([↗](#) 158頁 sscSetDigitalOutputDataBit)
- sscGetDigitalOutputDataWord ([↗](#) 161頁 sscGetDigitalOutputDataWord)

sscGetDigitalOutputDataWord

以16點為單位獲取指定的數位輸出DO資料。

```
int sscGetDigitalOutputDataWord (
    int board_id,
    int channel,
    int dout_word_num,
    unsigned short *dout
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
dout_word_num [in]	數位輸出字編號 (0~63)
dout [out]	指向儲存數位輸出的DO資料狀態的2位元組變數 (2位元組×1) 的指針 (按各位0: 輸出OFF, 1: 輸出ON) 位0→D0_□□□0, ... 位15→D0_8□□□

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetDigitalOutputDataWord ([↗](#) 159頁 sscSetDigitalOutputDataWord)
- sscGetDigitalOutputDataBit ([↗](#) 160頁 sscGetDigitalOutputDataBit)

4.20 標記檢測函數群

sscGetMarkDetectionData

獲取標記檢測資料。

```
int sscGetMarkDetectionData (  
    int board_id,  
    int channel,  
    int axnum,  
    int mark_num,  
    int *read_fin_num,  
    cher *edge,  
    long *position  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
mark_num [in]	標記檢測設置編號(1~2)
read_fin_num [out]	指向儲存讀取完成緩衝編號(1~255)的4位元組變數(4位元組×1)的指針
edge [out]	指向儲存標記檢測邊緣資料的1位元組結構(1位元組×1)的指針 ■值 • SSC_MARK_EDGE_NONE: 未檢測出 • SSC_MARK_EDGE_OFF: OFF邊緣 • SSC_MARK_EDGE_ON: ON邊緣
position [out]	指向儲存標記檢測位置資料的4位元組結構(4位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_MARK_DETECT_UNUSABLE	標記檢測功能無效。 • 不支援標記檢測功能。 • 標記檢測功能因設置無效。
SSC_FUNC_ERR_MARK_DETECT_UNDETECTED	可獲取的標記檢測資料不存在。 應確認更新了定位板的標記檢測次數後, 調用sscGetMarkDetectionData函數。

■要點

- 獲取標記檢測資料後, 更新讀取完成緩衝編號(+1)。(常時檢測模式的情況下, 如果讀取完成緩衝編號超過“255”則設置“1”。環形緩衝模式的情況下, 如果超過連續鎖存資料儲存數則設置“1”。)
- 由於同一標記檢測資料只能獲取一次, 因此需要通過用戶程式保持已獲取的資料。
- 關於指定了NULL指針的參數(讀取完成緩衝編號、標記檢測邊緣資料、標記檢測位置資料), 不獲取資料。(Visual C#的工程的情況下無法指定NULL指針。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetMarkDetectionCounter ([↗](#) 164頁 sscGetMarkDetectionCounter)

sscGetMarkDetectionCounter

獲取標記檢測次數。

```
int sscGetMarkDetectionCounter (  
    int board_id,  
    int channel,  
    int axnum,  
    int mark_num,  
    int *detected_counter  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
mark_num [in]	標記檢測設置編號 (1~2)
detected_counter [out]	指向儲存標記檢測次數 (1~255) 的4位元組變數 (4位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetMarkDetectionData ([↗](#) 162頁 sscGetMarkDetectionData)

sscClearMarkDetectionData

清除(初始化)標記檢測資料。

```
int sscClearMarkDetectionData (  
    int board_id,  
    int channel,  
    int axnum,  
    int mark_num  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
mark_num [in]	標記檢測設置編號(1~2)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~2: 超時位置	向定位板發出陳述式後的回應確認中, 經過了超時時間(1秒)。

■要點

清除標記檢測資料後, 將讀取完成緩衝編號設置為“0”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetMarkDetectionData([↗](#) 162頁 sscGetMarkDetectionData)

4.21 介面模式函數群

sscIfmGetReadErrorCount

獲取讀取錯誤計數器。

```
int sscIfmGetReadErrorCount (  
    int board_id,  
    int channel,  
    short *err_cnt  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
err_cnt [out]	指向儲存讀取錯誤計數器的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscIfmSetHomePosition

進行原點設置，完成後獲取原點資料（“原點多旋轉資料(參數No. 024D)”及“原點1旋轉內位置(參數No. 024E、No. 024F)”）。

```
int sscIfmSetHomePosition (
    int board_id,
    int channel,
    int axnum,
    int mode,
    short *param
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
mode [in]	進入位置信號(INP)確認模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CHK_INP_WAIT: 待機直至進入位置信號(INP)變為ON SSC_IFM_CHK_INP_NOWAIT: 不待機直至進入位置信號(INP)變為ON
param [out]	指向儲存原點資料的6位元組陣列(2位元組×3)的指針 ■陣列編號 <ul style="list-style-type: none"> 0: 原點多旋轉資料(參數No. 024D) 1: 原點1旋轉內位置(參數No. 024E) 2: 原點1旋轉內位置(參數No. 024F)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~3: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(10秒)。
SSC_FUNC_ERR_IFM_INP_OFF	進入位置信號(INP)處於OFF中。 在進入位置信號(INP)確認模式中設置“不待機直至進入位置信號(INP)變為ON”時，應在進入位置信號(INP)處於ON中調用sscIfmSetHomePosition函數。
SSC_FUNC_ERR_STS_BIT_ZSE	原點設置不正確信號(ZSE)變為了ON。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscIfmGetMaximumBufferNumber

獲取位置控制模式的最大緩衝編號。

```
int sscIfmGetMaximumBufferNumber (  
    int board_id,  
    int channel,  
    int axnum,  
    short *bufnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32)
bufnum [out]	指向儲存最大緩衝編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- API版本1.60及以後的情況下, 應使用sscIfmGetMaximumBufferNumberEx函數。[MC200]
- 應使用sscIfmGetMaximumBufferNumberEx函數。[MC300]

☞ 169頁 sscIfmGetMaximumBufferNumberEx

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

無。

sscIfmGetMaximumBufferNumberEx

獲取指定控制模式的最大緩衝編號。

```
int sscIfmGetMaximumBufferNumberEx (
    int board_id,
    int channel
    int axnum,
    unsigned short ctrl_mode,
    short *bufnum
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式
bufnum [out]	指向儲存最大緩衝編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 60	A4	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

無。

sscIfmRenewLatestBuffer

更新位置控制模式的最新陳述式緩衝編號及資料，獲取更新的最新陳述式緩衝編號。

```
int sscIfmRenewLatestBuffer (  
    int board_id,  
    int channel,  
    int axnum,  
    long bufdata,  
    short *bufnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32)
bufdata [in]	更新資料
bufnum [out]	指向儲存最新陳述式緩衝編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_IFM_CMD_BUF_FULL	位置陳述式緩衝無空餘。 應在確認定位板的傳送中緩衝編號已更新後，調用sscIfmRenewLatestBuffer函數或sscIfmRenewLatestBufferEx函數。

■要點

[MC200]

- API版本1.60及以後的情況下，應使用sscIfmRenewLatestBufferEx函數。
↳ 171頁 sscIfmRenewLatestBufferEx
- API版本1.80及以後的情況下，如果在最新陳述式緩衝編號中指定NULL指針，則不獲取資料。(Visual C#的工程的情況下無法指定NULL指針。)
- API版本1.60及以前的情況下，在最新陳述式緩衝編號中指定NULL指針時發生訪問違反，因此請勿指定NULL指針。

[MC300]

- 應使用sscIfmRenewLatestBufferEx函數。
↳ 171頁 sscIfmRenewLatestBufferEx
- 如果在最新陳述式緩衝編號中指定NULL指針，則不獲取資料。(Visual C#的工程的情況下無法指定NULL指針。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscIfmCheckLatestBuffer (↳ 172頁 sscIfmCheckLatestBuffer)
- sscIfmGetTransmitBuffer (↳ 174頁 sscIfmGetTransmitBuffer)

sscIfmRenewLatestBufferEx

更新指定的控制模式的最新陳述式緩衝編號及資料，獲取更新的最新陳述式緩衝編號。

```
int sscIfmRenewLatestBufferEx (
    int board_id,
    int channel,
    int axnum,
    unsigned short ctrl_mode,
    long bufdata,
    short *bufnum
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式
bufdata [in]	更新資料
bufnum [out]	指向儲存最新陳述式緩衝編號的2位元組變數(2位元組×1)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_IFM_CMD_BUF_FULL	位置陳述式緩衝無空餘。 應在確認定位板的傳送中緩衝編號已更新後，調用sscIfmRenewLatestBuffer函數或sscIfmRenewLatestBufferEx函數。

■要點

[MC200]

- API版本1.80及以後的情況下，如果在最新陳述式緩衝編號中指定NULL指針，則不獲取資料。(Visual C#的工程的情況下無法指定NULL指針。)

- API版本1.60及以前的情況下，在最新陳述式緩衝編號中指定NULL指針時發生訪問違反，因此請勿指定NULL指針。

[MC300]

- 如果在最新陳述式緩衝編號中指定NULL指針，則不獲取資料。(Visual C#的工程的情況下無法指定NULL指針。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscIfmCheckLatestBufferEx(📄 173頁 sscIfmCheckLatestBufferEx)
- sscIfmGetTransmitBufferEx(📄 175頁 sscIfmGetTransmitBufferEx)

sscIfmCheckLatestBuffer

獲取位置控制模式的最新陳述式緩衝編號及最新陳述式緩衝資料。

```
int sscIfmCheckLatestBuffer (  
    int board_id,  
    int channel,  
    int axnum,  
    short *bufnum,  
    long *bufdata  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32)
bufnum [out]	指向儲存最新陳述式緩衝編號的2位元組變數 (2位元組×1) 的指針
bufdata [out]	指向儲存最新陳述式緩衝資料的2位元組變數 (2位元組×1) 的指針

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

要點

- 在最新陳述式緩衝資料中指定了NULL指標的情況下，只獲取最新陳述式緩衝編號。(Visual C#的工程的情況下無法指定NULL指針。)
- API版本1.60及以後的情況下，應使用sscIfmCheckLatestBufferEx函數。[MC200]
- 應使用sscIfmCheckLatestBufferEx函數。[MC300]

☞ 173頁 sscIfmCheckLatestBufferEx

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscIfmGetTransmitBuffer (☞ 174頁 sscIfmGetTransmitBuffer)

sscIfmCheckLatestBufferEx

獲取指定控制模式的最新陳述式緩衝編號及最新陳述式緩衝資料。

```
int sscIfmCheckLatestBufferEx (
    int board_id,
    int channel,
    int axnum,
    unsigned short ctrl_mode,
    short *bufnum,
    long *bufdata
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式
bufnum [out]	指向儲存最新陳述式緩衝編號的2位元組變數 (2位元組×1) 的指針
bufdata [out]	指向儲存最新陳述式緩衝資料的2位元組變數 (2位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

在最新陳述式緩衝資料中指定了 NULL 指標的情況下，只獲取最新陳述式緩衝編號。(Visual C# 的工程的情況下無法指定 NULL 指針。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscIfmGetTransmitBufferEx ([↩](#) 175頁 sscIfmGetTransmitBufferEx)

sscIfmGetTransmitBuffer

獲取位置控制模式的傳送中緩衝編號及傳送中緩衝資料。

```
int sscIfmGetTransmitBuffer (  
    int board_id,  
    int channel,  
    int axnum,  
    short *bufnum,  
    long *bufdata  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32)
bufnum [out]	指向儲存傳送中緩衝編號的2位元組變數 (2位元組×1) 的指針
bufdata [out]	指向儲存傳送中緩衝資料的4位元組變數 (4位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 在傳送中緩衝資料中指定了NULL指標的情況下，只儲存傳送中緩衝編號。(Visual C#的工程的情況下無法指定NULL指針。)
- API版本1.60及以後的情況下，應使用sscIfmGetTransmitBufferEx函數。[MC200]
- 應使用sscIfmGetTransmitBufferEx函數。[MC300]

☞ 175頁 sscIfmGetTransmitBufferEx

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscIfmCheckLatestBuffer (☞ 172頁 sscIfmCheckLatestBuffer)

sscIfmGetTransmitBufferEx

獲取指定控制模式的傳送中緩衝編號及傳送中緩衝資料。

```
int sscIfmGetTransmitBufferEx (
    int board_id,
    int channel,
    int axnum,
    unsigned short ctrl_mode,
    short *bufnum,
    long *bufdata
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式
bufnum [out]	指向儲存傳送中緩衝編號的2位元組變數 (2位元組×1) 的指針
bufdata [out]	指向儲存傳送中緩衝資料的4位元組變數 (4位元組×1) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

在傳送中緩衝資料中指定了NULL指標的情況下，只儲存傳送中緩衝編號。(Visual C#的工程的情況下無法指定NULL指針。)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscIfmCheckLatestBufferEx ([↗](#) 173頁 sscIfmCheckLatestBufferEx)

sscIfmTrqSetSpeedLimit

設置轉矩控制用速度限制值。

```
int sscIfmTrqSetSpeedLimit (  
    int board_id,  
    int channel,  
    int axnum,  
    long speed  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
speed [in]	轉矩控制用速度限制值 [0.01r/min] (0~1000000000)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

應在更改為轉矩控制模式前調用本API函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscIfmSetControlMode ([↩](#) 177頁 sscIfmSetControlMode)

sscIfmSetControlMode

設置控制模式。

```
int sscIfmSetControlMode (
    int board_id,
    int channel,
    int axnum,
    unsigned short ctrl_mode
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
ctrl_mode [in]	控制模式 ■運行模式 <ul style="list-style-type: none"> SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼, 消除發生原因。

■要點

由於伺服驅動器的回應確認, 因此需要大約數ms直至切換控制模式。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 60	A4	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscIfmGetControlMode([↗](#) 178頁 sscIfmGetControlMode)

sscIfmGetControlMode

獲取控制模式。

```
int sscIfmGetControlMode (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned short *ctrl_mode,  
    char *status  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])
ctrl_mode [out]	指向儲存控制模式的2位元組變數 (2位元組×1) 的指針 ■運行模式 • SSC_IFM_CTRL_MODE_POSITION: 位置控制模式 • SSC_IFM_CTRL_MODE_SPEED: 速度控制模式 • SSC_IFM_CTRL_MODE_TORQUE: 轉矩控制模式
status [out]	指向儲存控制模式切換不正確狀態的1位元組變數 (1位元組×1) 的指針 ■運行模式 • SSC_IFM_CTRL_MODE_ERR_OFF: 控制模式切換不正確為OFF • SSC_IFM_CTRL_MODE_ERR_ON: 控制模式切換不正確為ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

控制模式切換不正確的情況下, 應設置切換陳述式前的控制模式, 調用sscIfmSetControlMode函數。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 60	A4	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscIfmSetControlMode ([↩](#) 177頁 sscIfmSetControlMode)

sscIfmGetEventStatusBits

使用事件檢測功能，獲取對指定狀態信號的所有軸的狀態位元資訊。

```
int sscIfmGetEventStatusBits (
    int board_id,
    int channel,
    int bitnum,
    unsigned long *status_bits
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
bitnum [in]	狀態位元編號 ■值 <ul style="list-style-type: none"> • SSC_EVENT_AX_RDY: 伺服就緒ON中 • SSC_EVENT_AX_INP: 進入位置 • SSC_EVENT_AX_ZSP: 零速度中 • SSC_EVENT_AX_TLC: 轉矩限制動作中 • SSC_EVENT_AX_SALM: 伺服報警中 • SSC_EVENT_AX_SWRN: 伺服警告中 • SSC_EVENT_AX_ABSE: 絕對位置丟失 • SSC_EVENT_AX_OALM: 運行報警中 • SSC_EVENT_AX_MAK1: 標記檢測中1 • SSC_EVENT_AX_MAK2: 標記檢測中2 • SSC_EVENT_AX_LSP: +側極限開關 • SSC_EVENT_AX_LSN: -側極限開關 • SSC_EVENT_AX_DOG: 近點狗
status_bits [out]	指向儲存所有軸狀態位元資料的4位元組變數(4位元組×1)的指針 位0→軸1、位1→軸2·····位31→軸32 位ON: 1、位OFF: 0 (未控制軸, 常時位OFF)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

- 通過sscIntDisable函數及sscIntEnd函數結束了中斷處理的情況下，由於事件通知功能不動作，因此無法正常獲取狀態位元的值。
- 由於使用事件檢測功能獲取狀態位元，因此與sscGetStatusBitSignalEx函數相比可以更高速度地讀取。
- 應使用sscIfmGetEventStatusBitsEx函數。[MC300]

☞ 180頁 sscIfmGetEventStatusBitsEx [MC300]

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A6	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetStatusBitSignalEx (☞ 71頁 sscGetStatusBitSignalEx)

sscIfmGetEventStatusBitsEx [MC300]

使用事件檢測功能，獲取對指定狀態信號的所有軸的狀態位元資訊。

```
int sscIfmGetEventStatusBitsEx (  
    int board_id,  
    int channel,  
    int bitnum,  
    SLAVE_INFO *pSlaveInfo  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
bitnum [in]	狀態位元編號 ■值 <ul style="list-style-type: none">• SSC_EVENT_AX_RDY: 伺服就緒ON中• SSC_EVENT_AX_INP: 進入位置• SSC_EVENT_AX_ZSP: 零速度中• SSC_EVENT_AX_TLC: 轉矩限制動作中• SSC_EVENT_AX_SALM: 伺服報警中• SSC_EVENT_AX_SWRN: 伺服警告中• SSC_EVENT_AX_ABSE: 絕對位置丟失• SSC_EVENT_AX_OALM: 運行報警中• SSC_EVENT_AX_MAK1: 標記檢測中1• SSC_EVENT_AX_MAK2: 標記檢測中2• SSC_EVENT_AX_LSP: +側極限開關• SSC_EVENT_AX_LSN: -側極限開關• SSC_EVENT_AX_DOG: 近點狗
pSlaveInfo [out]	指向儲存軟元件資訊(所有軸的狀態位元資料)的80位元組結構(80位元組×1)的指針 關於軟元件資訊結構，請參閱下述章節。(未控制軸及站資訊，常時位OFF) ☞ 237頁 SLAVE_INFO結構[MC300]

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

要點

- 通過sscIntDisable函數及sscIntEnd函數結束了中斷處理的情況下，由於事件通知功能不動作，因此無法正常獲取狀態位元的值。
- 由於使用事件檢測功能獲取狀態位元，因此與sscGetStatusBitSignalEx函數相比可以更高速度地讀取。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

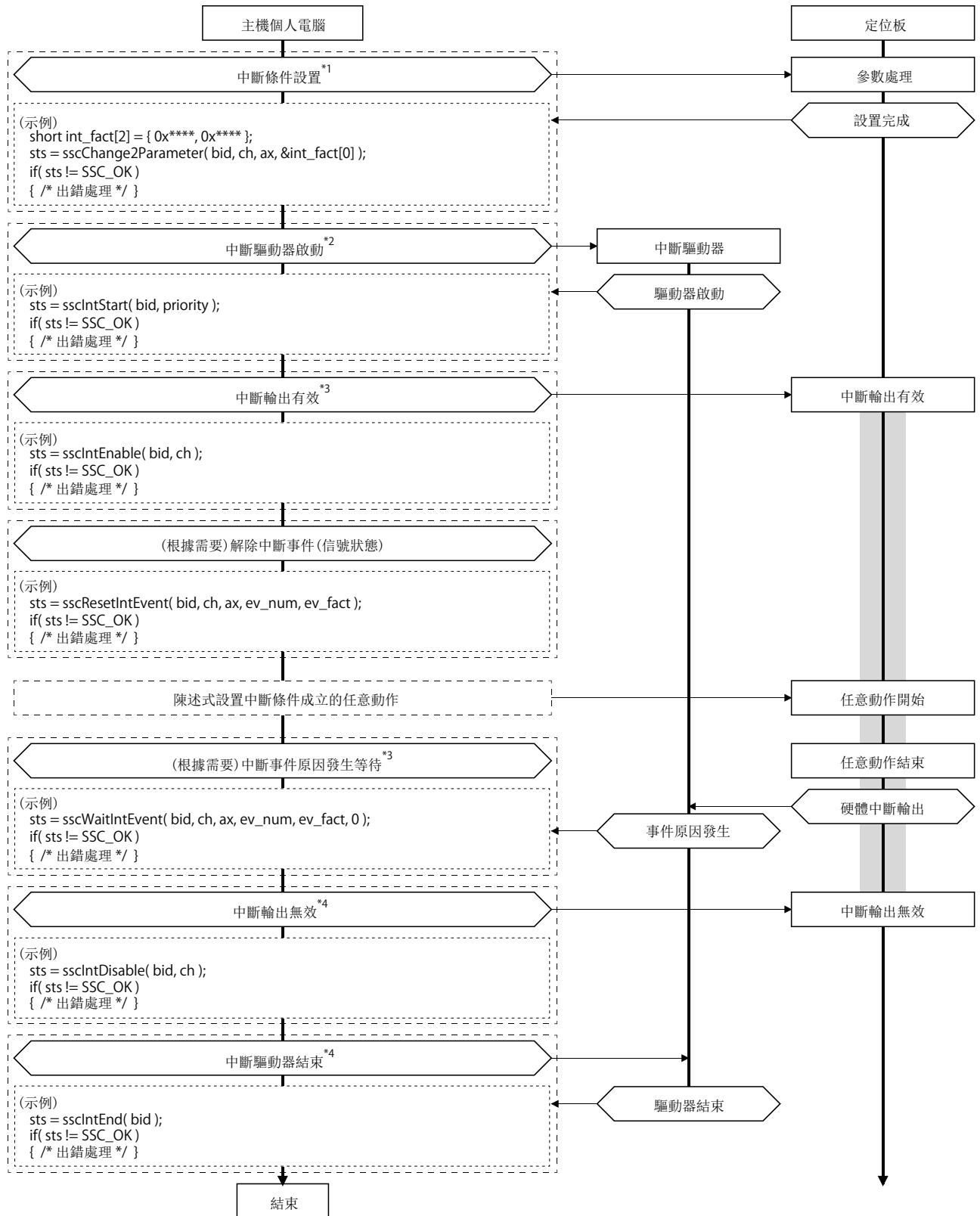
參照

無。

4.22 中斷函數群

處理步驟

使用中斷函數群時的處理步驟如下所示。



- *1 應僅在更改當前設置的中斷條件參數(“中斷條件1(參數No. 0204)”、“中斷條件2(參數No. 0205)”)的情況下調用。
- *2 請務必在中斷驅動器啟動後啟用中斷輸出。(在中斷驅動器未正常執行動作的狀態下輸出了硬體中斷的情況下，無法進行硬體中斷的解除，從而可能會導致主機個人電腦側掛起。)
- *3 中斷開始後，可以通過中斷事件等待函數進行中斷事件等待。此外，還可以通過sscWaitIntDriveFin函數進行運行完成等待。
- *4 請務必在中斷輸出無效後退出中斷驅動器。(與上述*2相同的原因)

sscIntStart

啟動中斷驅動器。
使用中斷事件等待函數進行中斷監視時使用。

```
int sscIntStart (  
    int board_id,  
    int priority  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
priority [in]	優先順序編號

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_START_INT_DRIVER	中斷驅動器已經在啟動中。
SSC_FUNC_ERR_INT_DISABLE_MASK [MC200]	中斷輸出遮罩選擇(撥動開關)有效。
SSC_FUNC_ERR_CREATE_EVENT	在CreateEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_CREATE_THREAD	在CreateThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_THREAD_PRIORITY	在SetThreadPriority函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_RESUME_THREAD	在ResumeThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_CLEAR_INT [MC200]	中斷信號清除暫存器寫入失敗。
SSC_FUNC_ERR_ALREADY_OTHER_PROCESS_INT	中斷驅動器已經在其他進程中啟動。
SSC_FUNC_ERR_DEVICE_DRIVER	軟元件驅動器的調用時發生了出錯。 應確認已經安裝了軟元件驅動器。

要點

- 通過SetThreadPriority函數(Windows API)設置中斷驅動器的優先順序編號。
- 關於優先順序設置值的詳細內容, 請參閱OS的參考手冊等。
- 應在調用了本API函數後通過sscIntEnable函數啟用中斷。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

- sscIntEnd(↖ 184頁 sscIntEnd)
- sscIntEnable(↖ 185頁 sscIntEnable)

sscIntEnd

退出中斷驅動器。

```
int sscIntEnd (  
    int board_id  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_END_INT_DRIVER	中斷驅動器已經在停止中。
SSC_FUNC_ERR_DEVICE_DRIVER	軟元件驅動器的調用時發生了出錯。 應確認已經安裝了軟元件驅動器。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_TIMEOUT_01	在中斷處理程式廢棄待機中, 經過了超時時間(1秒)。
SSC_FUNC_ERR_DELETE_THREAD	在CloseHandle函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_DELETE_EVENT	在CloseHandle函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	在GetExitCodeThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。


要點

無。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

sscIntStart( 183頁 sscIntStart)

sscIntEnable

將中斷輸出有效信號(ITS)置為ON，啟用中斷輸出。

```
int sscIntEnable (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscIntStart(🔗 183頁 sscIntStart)
- sscIntDisable(🔗 186頁 sscIntDisable)

sscIntDisable

將中斷輸出有效信號(ITS)置為OFF，禁用中斷輸出。

```
int sscIntDisable (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscIntEnable([↗](#) 185頁 sscIntEnable)

sscRegisterIntCallback

登記中斷回呼函數。

在中斷發生時，從通過sscIntStart函數啟動的中斷驅動器回檔已登記的函數。

```
int sscRegisterIntCallback (
    int board_id,
    int channel,
    void *cbfunc
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
cbfunc [in]	回呼函數指針 關於回檔結構，請參閱下述章節。 ☞ 233頁 INT_CB_DATA結構

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_REREGISTER_CALLBACK	中斷回呼函數已登記完成。 更改中斷回呼函數的情況下，應調用sscUnregisterIntCallback函數。

■要點

- 使用C++語言的情況下，應在回呼函數中記述_stdcall聲明。
- 在回呼函數內不需要通過中斷處理結束信號(ITE)及中斷原因輸出中信號(ITO)的中斷原因的更新處理。
- 由於回呼函數從中斷驅動器中回檔，因此應安裝所需最小限度的代碼而不實施無限等待處理。
- 回呼函數在sscWaitIntEvent函數等的各種中斷原因發生等待函數之前被回檔。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscIntStart(☞ 183頁 sscIntStart)
- sscIntEnable(☞ 185頁 sscIntEnable)
- sscUnregisterIntCallback(☞ 188頁 sscUnregisterIntCallback)

sscUnregisterIntCallback

解除中斷回呼函數。

```
int sscUnregisterIntCallback (  
    int board_id,  
    int channel  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_ALREADY_UNREGISTER_CALLBACK	中斷回呼函數已解除完成。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscRegisterIntCallback ([↗](#) 187頁 sscRegisterIntCallback)

sscResetIntEvent

將中斷事件設成非信號狀態。

將調用sscWaitIntEvent函數之前發生的中斷事件禁用時使用。

```
int sscResetIntEvent (
    int board_id,
    int channel,
    int axnum,
    int eventnum,
    int eventfactor
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統中斷事件 • 1~64: 各軸中斷事件 • -16~-1: 各站中斷事件(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
eventnum [in]	事件等待編號(0~1)
eventfactor [in]	事件原因 關於事件原因, 請參閱下述章節。 ☞ 286頁 中斷事件原因一覽

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscWaitIntEvent([☞ 191頁 sscWaitIntEvent](#))

sscSetIntEvent

將中斷事件設成信號狀態。

通過sscWaitIntEvent函數待機中，不是定位板的中斷事件，而是在用戶程式的時機從待機狀態恢復時使用。

```
int sscSetIntEvent (  
    int board_id,  
    int channel,  
    int axnum,  
    int eventnum,  
    int eventfactor  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統中斷事件• 1~64: 各軸中斷事件• -16~-1: 各站中斷事件 (-1: 站1, -2: 站2, -3: 站3, -4: 站4 • • • -15: 站15, -16: 站16)
eventnum [in]	事件等待編號 (0~1)
eventfactor [in]	事件原因 關於事件原因，請參閱下述章節。 ☞ 286頁 中斷事件原因一覽

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

調用本API函數並從中斷待機中恢復的情況下，sscWaitIntEvent函數將出錯。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscWaitIntEvent([☞ 191頁 sscWaitIntEvent](#))

sscWaitIntEvent

待機至中斷事件變為信號狀態。

在對針對指定的事件原因的來自定位板的中斷進行待機時使用。

通過改變事件等待編號，可以對同一原因進行最多2次中斷事件的發生等待。

```
int sscWaitIntEvent (
    int board_id,
    int channel,
    int axnum,
    int eventnum,
    int eventfactor,
    int timeout
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統中斷事件 • 1~64: 各軸中斷事件 • -16~-1: 各站中斷事件(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
eventnum [in]	事件等待編號(0~1)
eventfactor [in]	事件原因 關於事件原因，請參閱下述章節。 ☞ 286頁 中斷事件原因一覽
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_WAIT_EVENT	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中，調用了sscIntEnd函數。
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中，在中斷事件通知執行緒中發生了出錯。
SSC_FUNC_ERR_TIMEOUT_01	在針對指定的事件原因的中斷待機中，經過了指定的超時時間。
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中，從用戶程式調用了恢復待機狀態的函數。

■要點

在超時時間中指定了“SSC_INFINITE”的情況下，不進行超時檢查，將無限等待中斷事件發生。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■参照

- [sscResetIntEvent](#) (189頁 [sscResetIntEvent](#))
- [sscSetIntEvent](#) (190頁 [sscSetIntEvent](#))
- [sscWaitIntEventMulti](#) (195頁 [sscWaitIntEventMulti](#))

sscResetIntEventMulti

將多個中斷事件設成非信號狀態。

將調用sscWaitIntEventMulti函數之前發生的多個中斷事件禁用時使用。

```
int sscResetIntEventMulti (
    int board_id,
    int channel,
    int axnum
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統中斷事件 • 1~64: 各軸中斷事件 • -16~-1: 各站中斷事件(-1: 站1, -2: 站2, -3: 站3, -4: 站4 • • • -15: 站15, -16: 站16)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscWaitIntEventMulti (📖 195頁 sscWaitIntEventMulti)

sscSetIntEventMulti

將多個中斷事件設成信號狀態。

通過sscWaitIntEventMulti函數待機中，不是定位板的中斷事件，而是在用戶程式的時機從待機狀態恢復時使用。

```
int sscSetIntEventMulti (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none">• 0: 系統中斷事件• 1~64: 各軸中斷事件• -16~-1: 各站中斷事件(-1: 站1, -2: 站2, -3: 站3, -4: 站4... -15: 站15, -16: 站16)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

調用本API函數並從中斷待機中恢復的情況下，sscWaitIntEventMulti函數將出錯。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntEventMulti (📖 195頁 sscWaitIntEventMulti)

sscWaitIntEventMulti

待機至多個中斷事件變為信號狀態。

在對針對任一多個事件原因的來自定位板的中斷進行待機時使用。

```
int sscWaitIntEventMulti (
    int board_id,
    int channel,
    int axnum,
    int timeout,
    unsigned long *eventcode,
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~32 [MC200]/-16~64 [MC300]) <ul style="list-style-type: none"> • 0: 系統中斷事件 • 1~64: 各軸中斷事件 • -16~-1: 各站中斷事件(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)
eventcode [out]	指向儲存多個事件原因的4位元組變數(4位元組×1)的指針 關於事件原因，請參閱下述章節。 ☞ 286頁 中斷事件原因一覽

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_WAIT_EVENT_MULTI	在WaitForMultipleObjects函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中，調用了sscIntEnd函數。
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中，在中斷事件通知執行緒中發生了出錯。
SSC_FUNC_ERR_TIMEOUT_01	在針對指定的事件原因的中斷待機中，經過了指定的超時時間。
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中，從用戶程式調用了恢復待機狀態的函數。

■要點

- 在超時時間中指定了“SSC_INFINITE”的情況下，不進行超時檢查，將無限等待多個中斷事件發生。
- 同時發生了多個中斷事件的情況下，將儲存最小的事件原因。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■ 参照

- [sscWaitIntEvent](#) (191頁 [sscWaitIntEvent](#))
- [sscResetIntEventMulti](#) (193頁 [sscResetIntEventMulti](#))
- [sscSetIntEventMulti](#) (194頁 [sscSetIntEventMulti](#))

sscResetIntOasEvent

將其他軸啟動的中斷事件設成非信號狀態。

將調用sscWaitIntOasEvent函數之前發生的其他軸啟動中斷事件禁用時使用。

```
int sscResetIntOasEvent (
    int board_id,
    int channel,
    int axnum,
    int oas_num
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	其他軸啟動表的本軸編號(1~32 [MC200]/1~64 [MC300])
oas_num [in]	其他軸啟動表編號(1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntOasEvent (🔗 199頁 sscWaitIntOasEvent)

sscSetIntOasEvent

將其他軸啟動的中斷事件設成信號狀態。

通過sscWaitIntOasEvent函數待機中，不是定位板的中斷事件，而是在用戶程式的時機從待機狀態恢復時使用。

```
int sscSetIntOasEvent (  
    int board_id,  
    int channel,  
    int axnum,  
    int oas_num  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	其他軸啟動表的本軸編號(1~32 [MC200]/1~64 [MC300])
oas_num [in]	其他軸啟動表編號(1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

調用本API函數並從中斷待機中恢復的情況下，sscWaitIntOasEvent函數將出錯。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntOasEvent ([↩](#) 199頁 sscWaitIntOasEvent)

sscWaitIntOasEvent

待機至其他軸啟動的中斷事件變為信號狀態。
在對針對指定的其他軸啟動待機類型的來自定位板的中斷進行待機時使用。
已檢查的狀態被儲存在通過指針指定的變數中。

```
int sscWaitIntOasEvent (  
    int board_id,  
    int channel,  
    int axnum,  
    int oas_num,  
    int oas_type,  
    int *oas_status,  
    int timeout  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	其他軸啟動表的本軸編號(1~32 [MC200]/1~64 [MC300])
oas_num [in]	其他軸啟動表編號(1~32 [MC200]/1~64 [MC300])
oas_type [in]	其他軸啟動待機類型 ■值 • SSC_OAS_WAIT_TYPE_OP: 其他軸啟動預告待機 • SSC_OAS_WAIT_TYPE_FIN: 其他軸啟動完成待機
oas_status [out]	指向儲存其他軸啟動狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_OAS_STS_OP: 其他軸啟動預告 • SSC_OAS_STS_FIN: 其他軸啟動完成 • SSC_OAS_STS_ERR: 其他軸啟動未完成 • SSC_OAS_STS_OP_ERR: 其他軸啟動預告前出錯
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_WAIT_EVENT	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中, 調用了sscIntEnd函數。
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中, 在中斷事件通知執行緒中發生了出錯。
SSC_FUNC_ERR_TIMEOUT_01	在針對指定的事件原因的中斷待機中, 經過了指定的超時時間。
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中, 從用戶程式調用了恢復待機狀態的函數。

■要點

- 在其他軸啟動預告待機中變為了“SSC_OAS_STS_FIN”或“SSC_OAS_STS_ERR”的情況下，將從待機狀態恢復。
- 在其他軸啟動預告前，在其他軸啟動的本軸中發生了報警等情況下，將變為“SSC_OAS_STS_OP_ERR”。
- 在本API函數中，使用下述中斷條件。使用本API函數的情況下，必須通過sscChange2Parameter函數設置相應的中斷條件。
OASF (其他軸啟動中斷原因輸出中)
OPF (運行完成)
- 在超時時間中指定了“SSC_INFINITE”的情況下，不進行超時檢查，將無限等待事件發生。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscResetIntOasEvent (🔗 197頁 sscResetIntOasEvent)
- sscSetIntOasEvent (🔗 198頁 sscSetIntOasEvent)

sscResetIntPassPosition

將通過位置中斷事件設成非信號狀態。

將調用sscWaitIntPassPosition函數之前發生的通過位置中斷事件禁用時使用。

```
int sscResetIntPassPosition (  
    int board_id,  
    int channel,  
    int pass_start,  
    int pass_end  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pass_start [in]	通過位置條件開始編號(1~64 [MC200]/1~128 [MC300])
pass_end [in]	通過位置條件結束編號(1~64 [MC200]/1~128 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntPassPosition([↩](#) 203頁 sscWaitIntPassPosition)

sscSetIntPassPosition

將通過位置中斷事件設成信號狀態。

通過sscWaitIntPassPosition函數待機中，不是定位板的中斷事件，而是在用戶程式的時機從待機狀態恢復時使用。

```
int sscSetIntPassPosition (  
    int board_id,  
    int channel,  
    int pass_start,  
    int pass_end  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pass_start [in]	通過位置條件開始編號(1~64 [MC200]/1~128 [MC300])
pass_end [in]	通過位置條件結束編號(1~64 [MC200]/1~128 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

調用本API函數並從中斷待機中恢復的情況下，sscWaitIntPassPosition函數將出錯。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntPassPosition(🔗 203頁 sscWaitIntPassPosition)

sscWaitIntPassPosition

待機至通過位置中斷事件變為信號狀態。

在對針對指定的通過位置條件編號的來自定位板的中斷進行待機時使用。

已檢查的狀態被儲存在通過指針指定的變數中。

```
int sscWaitIntPassPosition (
    int board_id,
    int channel,
    int pass_num,
    int *pass_status,
    int timeout
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
pass_num [in]	通過位置條件編號(1~64 [MC200]/1~128 [MC300])
pass_status [out]	指向儲存通過位置狀態的4位元組變數(4位元組×1)的指針 ■值 <ul style="list-style-type: none"> SSC_PASS_STS_FIN: 通過位置中斷完成 SSC_PASS_STS_ERR: 通過位置中斷未完成
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_WAIT_EVENT	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中, 調用了sscIntEnd函數。
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中, 在中斷事件通知執行緒中發生了出錯。
SSC_FUNC_ERR_TIMEOUT_01	在針對指定的事件原因的中斷待機中, 經過了指定的超時時間。
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中, 從用戶程式調用了恢復待機狀態的函數。

■要點

在超時時間中指定了“SSC_INFINITE”的情況下, 不進行超時檢查, 將無限等待直至中斷事件變為ON或OFF。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscResetIntPassPosition (🔗 201頁 sscResetIntPassPosition)
- sscSetIntPassPosition (🔗 202頁 sscSetIntPassPosition)

sscResetIntDriveFin

將運行完成的中斷事件設成非信號狀態。

將調用sscWaitIntDriveFin函數之前發生的運行完成中斷事件禁用時使用。

```
int sscResetIntDriveFin (  
    int board_id,  
    int channel,  
    int axnum  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數 (Windows API) 中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntDriveFin ([↩](#) 206頁 sscWaitIntDriveFin)

sscSetIntDriveFin

將運行完成的中斷事件設成信號狀態。

通過sscWaitIntDriveFin函數待機中，不是定位板的中斷事件，而是在用戶程式的時機從待機狀態恢復時使用。

```
int sscSetIntDriveFin (
    int board_id,
    int channel,
    int axnum
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
axnum [in]	軸編號 (1~32 [MC200]/1~64 [MC300])

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。

■要點

調用本API函數並從中斷待機中恢復的情況下，sscWaitIntDriveFin函數將出錯。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscWaitIntDriveFin (🔗 206頁 sscWaitIntDriveFin)

sscWaitIntDriveFin

待機至運行完成的中斷事件變為信號狀態。

在對針對指定的運行完成類型的來自定位板的中斷進行待機時使用。

已檢查的狀態被儲存在通過指針指定的變數中。

```
int sscWaitIntDriveFin (  
    int board_id,  
    int channel,  
    int axnum,  
    int fin_type,  
    int *fin_status,  
    int timeout  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(1~32 [MC200]/1~64 [MC300])
fin_type [in]	運行完成類型 ■值 • SSC_FIN_TYPE_SMZ: 通過平滑停止的運行完成 • SSC_FIN_TYPE_CPO: 通過粗匹配的運行完成 • SSC_FIN_TYPE_INP: 通過進入位置停止的運行完成
fin_status [out]	指向儲存運行完成狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_FIN_STS_STP: 運行完成 • SSC_FIN_STS_MOV: 運行中 • SSC_FIN_STS_ALM_STP: 報警發生中(停止完成) • SSC_FIN_STS_ALM_MOV: 報警發生中(減速停止中)
timeout [in]	超時時間[ms](1~65535、SSC_INFINITE)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
SSC_FUNC_ERR_WAIT_EVENT	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數, 並確認詳細出錯內容。
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中, 調用了sscIntEnd函數。
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中, 在中斷事件通知執行緒中發生了出錯。
SSC_FUNC_ERR_TIMEOUT_01	在針對指定的事件原因的中斷待機中, 經過了指定的超時時間。
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中, 從用戶程式調用了恢復待機狀態的函數。

■要點

- 運行完成檢查的條件取決於通過參數指定的運行完成類型。

運行完成類型	運行完成檢查的條件
“SSC_FIN_TYPE_SMZ” / “SSC_FIN_TYPE_CPO” / “SSC_FIN_TYPE_INP” 的情況下	等待至滿足指定的條件後結束。
“SSC_FIN_TYPE_CPO” 的情況下	在自動運行・直線插補[MC200]/插補運行[MC300]模式下粗匹配信號(CPO)變為了ON時，將變為“SSC_FIN_STS_STP”。
在運行中輸入了運行停止信號(STP)的情況下	停止完成後將變為“SSC_FIN_STS_STP”。

- 粗匹配信號(CPO)僅在自動運行・直線插補[MC200]/插補運行[MC300]模式時被輸出。因此，在自動運行・直線插補[MC200]/插補運行[MC300]模式以外的運行模式時進行運行完成檢查的情況下，在運行完成類型中應使用“SSC_FIN_TYPE_CPO”以外。
- 在本API函數中，使用以下中斷條件。使用本API函數的情況下，必須通過sscChange2Parameter函數設置相應的中斷條件。

中斷條件	備註
INP(進入位置)	未使用運行完成類型的“SSC_FIN_TYPE_INP”的情況下不需要
SALM(伺服報警中)	—
CPO(粗匹配)	未使用運行完成類型的“SSC_FIN_TYPE_CPO”的情況下不需要
OALM(運行報警中)	—
OPF(運行完成)	—
SYSE(系統出錯中)	未使用運行完成類型的“SSC_FIN_TYPE_INP”的情況下不需要

- 應僅設置必要的中斷條件。不需要的中斷條件有可能會降低用戶程式的性能。
- 在超時時間中指定了“SSC_INFINITE”的情況下，不進行超時檢查，將無限等待事件發生。
- 在自動運行、直線插補[MC200]/插補運行[MC300]模式中將減速檢查方式設置為“SSC_SUBCMD_STOP_INP”的情況下，即使將運行完成類型設置為“SSC_FIN_TYPE_SMZ”也將變為“SSC_FIN_TYPE_INP”。
- 超時發生時的運行完成狀態為不定值。
- 由於API函數的運行完成狀態通過運行完成類型進行判定，因此可能會與定位板的運行中信號(OP)及運行完成信號(OPF)不同。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscResetIntDriveFin(🔗 204頁 sscResetIntDriveFin)
- sscSetIntDriveFin(🔗 205頁 sscSetIntDriveFin)
- sscGetDriveFinStatus(🔗 101頁 sscGetDriveFinStatus)

4.23 輸入輸出軟元件函數群

sscGetInputDeviceBit

以1點為單位獲取指定的輸入位軟元件。

```
int sscGetInputDeviceBit (  
    int board_id,  
    int channel,  
    int bit_num,  
    int *dev_in  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
bit_num [in]	輸入位軟元件編號 (0000h~0FFFh [MC200]/0000h~23FFh [MC300])
dev_in [out]	指向儲存輸入位軟元件狀態的4位元組變數 (4位元組×1) 的指針 ■值 • SSC_BIT_OFF: 輸入信號OFF • SSC_BIT_ON: 輸入信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetInputDeviceWord ([↗](#) 209頁 sscGetInputDeviceWord)

sscGetInputDeviceWord

以字單位獲取指定的輸入字軟元件。

```
int sscGetInputDeviceWord (
    int board_id,
    int channel,
    int word_num,
    int word_cnt,
    unsigned short *dev_in
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
word_num [in]	輸入字軟元件編號 (0000h~00FFh [MC200]/0000h~023Fh [MC300])
word_cnt [in]	來自輸入字軟元件編號的字點數 (0001h~0100h [MC200]/0001h~0240h [MC300])
dev_in [out]	指向儲存輸入字軟元件狀態的陣列 (2位元組×word_cnt) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_DVI_TABLE_RANGE_OVER	通過參數指定的 “word_num” + “word_cnt” 超過了輸入軟元件表大小的區域。


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetInputDeviceBit ( 208頁 sscGetInputDeviceBit)

sscSetOutputDeviceBit

使用定位板的雙埠記憶體排他控制功能，以1點為單位設置指定的輸出位軟元件。

```
int sscSetOutputDeviceBit (  
    int board_id,  
    int channel,  
    int bit_num,  
    int dev_out  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
bit_num [in]	輸出位軟元件編號(0000h~0FFFh [MC200]/0000h~23FFh [MC300])
dev_out [in]	設置資料 ■值 • SSC_BIT_OFF: 輸出信號OFF • SSC_BIT_ON: 輸出信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_UNSUPPORTED_BOARD_VERSION	定位板不支援此功能。
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯(E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
SSC_FUNC_ERR_TIMEOUT_0□ □=1~4: 超時位置	向定位板發出陳述式後的回應確認中，經過了超時時間(10秒)。



■要點

本API函數使用定位板的排他控制功能進行輸出位軟元件的設置。但是，由於介面模式時不能使用其他軸啟動功能，因此不使用排他控制功能。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetOutputDeviceWord( 211頁 sscSetOutputDeviceWord)
- sscGetOutputDeviceBit( 212頁 sscGetOutputDeviceBit)

sscSetOutputDeviceWord

不使用定位板的雙埠記憶體排他控制功能，以字單位設置指定的輸出字軟元件。

```
int sscSetOutputDeviceWord (
    int board_id,
    int channel,
    int word_num,
    int word_cnt,
    unsigned short *dev_out
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號 (0~3)
channel [in]	通道編號 (1)
word_num [in]	輸出字軟元件編號 (0000h~00FFh [MC200]/0000h~023Fh [MC300])
word_cnt [in]	來自輸出字軟元件編號的字點數 (0001h~0100h [MC200]/0001h~0240h [MC300])
dev_out [in]	指向儲存設置資料的陣列 (2位元組×word_cnt) 的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗 (使用 sscGetLastError 函數，確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen 函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	通過參數指定的 “word_num” + “word_cnt” 超過了輸出軟元件表大小的區域。

■要點

在通過其他軸啟動功能的輸出信號的控制中，對相應的輸出信號進行輸出字軟元件的設置時，有可能無法保持資料的一致性。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetOutputDeviceBit (↗ 210頁 sscSetOutputDeviceBit)
- sscGetOutputDeviceWord (↗ 213頁 sscGetOutputDeviceWord)

sscGetOutputDeviceBit

以1點為單位獲取指定的輸出位軟元件。

```
int sscGetOutputDeviceBit (  
    int board_id,  
    int channel,  
    int bit_num,  
    int *dev_out  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
bit_num [in]	輸出位軟元件編號(0000h~0FFFh [MC200]/0000h~23FFh [MC300])
dev_out [out]	指向儲存輸出位軟元件狀態的4位元組變數(4位元組×1)的指針 ■值 • SSC_BIT_OFF: 輸出信號OFF • SSC_BIT_ON: 輸出信號ON

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetOutputDeviceWord([↗](#) 213頁 sscGetOutputDeviceWord)
- sscSetOutputDeviceBit([↗](#) 210頁 sscSetOutputDeviceBit)

sscGetOutputDeviceWord

以字單位獲取指定的輸出字軟元件。

```
int sscGetOutputDeviceWord (  
    int board_id,  
    int channel,  
    int word_num,  
    int word_cnt,  
    unsigned short *dev_out  
);
```

詳細說明

■參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
word_num [in]	輸出字軟元件編號(0000h~00FFh [MC200]/0000h~023Fh [MC300])
word_cnt [in]	來自輸出字軟元件編號的字點數(0001h~0100h [MC200]/0001h~0240h [MC300])
dev_out [out]	指向儲存輸出字軟元件狀態的陣列(2位元組× word_cnt)的指針

■返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

■詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	通過參數指定的“word_num” + “word_cnt” 超過了輸出軟元件表大小的區域。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscGetOutputDeviceBit(↗ 212頁 sscGetOutputDeviceBit)
- sscSetOutputDeviceWord(↗ 211頁 sscSetOutputDeviceWord)

4.24 暫態傳送函數群

sscSendReceiveTransientData

對連接至SSCNET的軸或站進行指定的暫態傳送資料的發送接收。

```
int sscSendReceiveTransientData (  
    int board_id,  
    int channel,  
    int axnum,  
    TRANSIENT_CMD *pTransientCmd,  
    TRANSIENT_STS *pTransientSts,  
    int timeout  
);
```

詳細說明

參數

參數	內容
board_id [in]	板ID編號(0~3)
channel [in]	通道編號(1)
axnum [in]	軸編號(-4~-1、1~32 [MC200]/-16~-1、1~64 [MC300]) 1~64: 各軸 -16~-1: 各站(-1: 站1, -2: 站2, -3: 站3, -4: 站4...-15: 站15, -16: 站16)
pTransientCmd [in]	指向儲存暫態傳送陳述式資料的16位元組結構(16位元組×1)的指針 關於暫態傳送陳述式資料結構, 請參閱下述章節。 ☞ 231頁 TRANSIENT_CMD結構
pTransientSts [out]	指向儲存暫態傳送狀態資料的16位元組結構(16位元組×1)的指針 關於暫態傳送狀態資料結構, 請參閱下述章節。 ☞ 232頁 TRANSIENT_STS結構
timeout [in]	超時時間[ms] (0~65535)

返回值

值	內容
SSC_OK	函數成功
SSC_NG	函數失敗(使用sscGetLastError函數, 確認詳細出錯代碼。)
SSC_UNOPEN	sscOpen函式呼叫前

詳細出錯代碼

常數定義	發生原因/處理
SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
SSC_FUNC_ERR_TIMEOUT_01	向定位板發出陳述式後, 在完成暫態處理的確認中, 經過了指定的超時時間。
SSC_FUNC_ERR_TRANSIENT_INVALID_DATA	暫態資料無效。

要點

- 在超時時間中指定了1秒(1000ms)及以下的情況下, 超時時間將為1秒(1000ms)。
- API版本2.20及以前[MC200]/API版本1.10及以前[MC300]的函數名將變為“sscSendRecieveTransientData”。

對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

參照

無。

5 結構一覽

5.1 PNT_DATA_EX結構

該結構是自動運行、直線插補[MC200]/插補運行[MC300]中使用的點資料結構。

PNT_DATA_EX結構[MC200]

```
typedef struct {
/* 0000h */
    long position;
    unsigned long speed;
    unsigned short actime;
    unsigned short dctime;
    unsigned short dwell;
    unsigned short subcmd;
/* 0010h */
    unsigned char oas_num[2];
    unsigned char reserve1[2];
    unsigned char s_curve;
    unsigned char reserve2[3];
    unsigned char sub_axnum[3];
    unsigned char reserve3[5];
/* 0020h */
} PNT_DATA_EX;
```

詳細說明

■成員

成員	內容
position	位置資料
speed	進給速度(0~2147483647)
actime	加速時間常數[ms](0~20000)
dctime	減速時間常數[ms](0~20000)
dwell	停頓[ms]
subcmd	輔助陳述式 應使用各值的邏輯或設置要設置的資料。 ■值 [位置陳述式方式] • SSC_SUBCMD_POS_ABS: 絕對位置陳述式 • SSC_SUBCMD_POS_INC: 相對位置陳述式 [減速檢查方式] • SSC_SUBCMD_STOP_INP: 進入位置停止 • SSC_SUBCMD_STOP_SMZ: 平滑停止 • SSC_SUBCMD_STOP_CONTINUE: 連續運行 [速度切換點指定] • SSC_SUBCMD_PNT_SWITCH_AFTER: 點切換後 • SSC_SUBCMD_PNT_SWITCH_BEFORE: 點切換前 [停頓指定] • SSC_SUBCMD_DWELL: 停頓 • SSC_SUBCMD_PREDWELL: 預停頓 [通過位置中斷指定] • SSC_SUBCMD_PASS_POS_DISABLE: 無效 • SSC_SUBCMD_PASS_POS_ENABLE: 有效 [擋塊控制指定] • SSC_SUBCMD_PRESS_DISABLE: 無效 • SSC_SUBCMD_PRESS_ENABLE: 有效 [循環指定] • SSC_SUBCMD_PNT_LOOP_DISABLE: 無效 • SSC_SUBCMD_PNT_LOOP_START: 循環起始點 • SSC_SUBCMD_PNT_LOOP_END: 循環結束點
oas_num[2]	其他軸啟動指定(0~32)
s_curve	S字比率[%](0~100)

成員	內容
sub_axnum[3]	插補軸編號(0~32) 應設置同一插補組的輔助軸的軸編號。 不使用輔助軸的情況下，應設置“0”。

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.90	A9	mc2xxstd.h

■參照

- sscSetPointDataEx (🔗 73頁 sscSetPointDataEx)
- sscCheckPointDataEx (🔗 74頁 sscCheckPointDataEx)

PNT_DATA_EX結構[MC300]

```
typedef struct {
/* 0000h */
    long position;
    unsigned long speed;
    unsigned short actime;
    unsigned short dctime;
    unsigned short dwell;
    unsigned short subcmd;
/* 0010h */
    unsigned char oas_num[4];
    unsigned char s_curve;
    char reserve1[3];
    unsigned char sub_axnum[4];
    long arc_coord;
/* 0020h */
    unsigned short ac_dc_data[4];
    unsigned short subcmd2;
    unsigned char smoothing_time;
    char reserve2[5];
/* 0030h */
} PNT_DATA_EX;
```

詳細說明

■成員

成員	內容
position	位置資料[陳述式單位]
speed	進給速度[速度單位] (0~2147483647)
actime	加速時間常數[ms] (0~20000)
dctime	減速時間常數[ms] (0~20000)
dwell	停頓[ms]
subcmd	<p>輔助陳述式 應使用各值的邏輯或設置要設置的資料。</p> <p>■值</p> <p>[位置陳述式方式]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_POS_ABS: 絕對位置陳述式 • SSC_SUBCMD_POS_INC: 相對位置陳述式 <p>[減振陳述式濾波器1指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_DAMPING_DISABLE: 減振陳述式濾波器1無效 • SSC_SUBCMD_DAMPING_ENABLE: 減振陳述式濾波器1有效 <p>[減速檢查方式]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_STOP_INP: 進入位置停止 • SSC_SUBCMD_STOP_SMZ: 平滑停止 • SSC_SUBCMD_STOP_CONTINUE: 連續運行 <p>[速度切換點指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PNT_SWITCH_AFTER: 點切換後 • SSC_SUBCMD_PNT_SWITCH_BEFORE: 點切換前 <p>[停頓指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_DWELL: 停頓 • SSC_SUBCMD_PREDWELL: 預停頓 <p>[通過位置中斷指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PASS_POS_DISABLE: 無效 • SSC_SUBCMD_PASS_POS_ENABLE: 有效 <p>[擋塊控制指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PRESS_DISABLE: 無效 • SSC_SUBCMD_PRESS_ENABLE: 有效 <p>[循環指定]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PNT_LOOP_DISABLE: 無效 • SSC_SUBCMD_PNT_LOOP_START: 循環起始點 • SSC_SUBCMD_PNT_LOOP_END: 循環結束點 <p>[插補方式]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_INTERP_LINEAR: 直線插補 • SSC_SUBCMD_INTERP_ARC: 輔助點指定的圓弧插補 • SSC_SUBCMD_INTERP_ARC_CW: 中心點指定的圓弧插補(CW) • SSC_SUBCMD_INTERP_ARC_CCW: 中心點指定的圓弧插補(CCW)
oas_num[4]	其他軸啟動指定(0~64)

成員	內容
s_curve	S字比率[%] (0~100)
sub_axnum[4]	插補軸編號 (0~64) 應設置同一插補組的輔助軸的軸編號。 不使用輔助軸的情況下，應設置“0”。
arc_coord	圓弧座標[陳述式單位] 應設置圓弧的輔助點或中心點的座標。 設置內容根據插補方式而有所不同。
ac_dc_data[4]	加減速資料[0.1%] (0~1000)
subcmd2	輔助陳述式2 應使用各值的邏輯或設置要設置的資料。 ■值 [加減速度方式] • SSC_SUBCMD2_ACCDEC_LINE_S: 直線加減速・S字加減速 • SSC_SUBCMD2_ACCDEC_JERK: JERK比率加減速
smoothing_time	平滑時間常數[ms] (0~100)

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.30	A6	mc3xxstd.h

■參照

- sscSetPointDataEx (73頁 sscSetPointDataEx)
- sscCheckPointDataEx (74頁 sscCheckPointDataEx)

5.2 OAS_DATA結構

該結構是其他軸啟動中使用的其他軸啟動資料結構。

OAS_DATA結構[MC200]

```
typedef struct {
/* 0000h */
    unsigned long opt_own;
    unsigned long opt_observ;
    long data_own;
    long data_observ;
/* 0010h */
    char reserve1[8];
    unsigned long long st_axbit;
/* 0020h */
    unsigned short st_pnt_s;
    unsigned short st_pnt_e;
    char reserve2[12];
/* 0030h */
    char reserve3[40];
    unsigned char dout_ctrl;
    unsigned char dout_num;
    unsigned short dout_ctrlbit;
    unsigned short dout_data;
    char reserve4[10];
/* 0068h */
} OAS_DATA;
```

5

詳細說明

■成員

成員	內容
opt_own	<p>本軸選單 應使用各值的邏輯或設置要設置的資料。</p> <p>■值 [本軸判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OWN_REMAINING_DISTANCE: 剩餘距離指定 • SSC_OAS_OWN_POSITION_PASS: 指定位置通過指定 <p>[本軸判定座標]</p> <ul style="list-style-type: none"> • SSC_OAS_OWN_JUDGE_COORD_FB: F/B位置 • SSC_OAS_OWN_JUDGE_COORD_CMD: 陳述式位置
opt_observ	<p>監視軸選單 應使用各值的邏輯或設置要設置的資料。</p> <p>■值 [監視軸指定]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_DISABLE: 無效 • SSC_OAS_OBSERV_ENABLE: 有效 <p>[監視軸判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_POSITION_PASS: 監視軸指定位置通過指定 <p>[監視軸判定座標]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_JUDGE_COORD_FB: F/B位置 • SSC_OAS_OBSERV_JUDGE_COORD_CMD: 陳述式位置 <p>[監視軸指定位置通過判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_DATA_LESS: 監視軸的位置在監視軸指定位置資料及以下時條件成立 • SSC_OAS_OBSERV_DATA_MORE: 監視軸的位置在監視軸指定位置資料及以上時條件成立
data_own	本軸剩餘距離資料(或本軸通過位置資料)[陳述式單位]
data_observ	監視軸剩餘距離資料(或監視軸通過位置資料)[陳述式單位]
st_axbit	啟動軸指定
st_pnt_s	啟動軸起始點(0~319)
st_pnt_e	啟動軸結束點(0~319)
dout_ctrl	<p>數位輸出信號控制/輸出軟元件信號控制</p> <p>■值 [數位輸出信號控制]</p> <ul style="list-style-type: none"> • SSC_OAS_DO_DISABLE: 無效 • SSC_OAS_DO_ENABLE: 有效

成員	內容
dout_num	數位輸出信號編號 (00h~3Fh) / 輸出軟件信號編號 (00h~FFh)
dout_ctrlbit	數位輸出信號有效選擇 / 輸出軟件信號有效選擇
dout_data	數位輸出信號陳述式 / 輸出軟件信號陳述式

■要點

將監視軸指定設置為“有效 (SSC_OAS_OBSERV_ENABLE)”的情況下，必須設置“監視軸指定位置通過指定 (SSC_OAS_OBSERV_POSITION_PASS)”。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.02	A1	mc2xxstd.h

■參照

- sscSetOtherAxisStartData (📖 131頁 sscSetOtherAxisStartData)
- sscGetOtherAxisStartData (📖 132頁 sscGetOtherAxisStartData)

OAS_DATA結構[MC300]

```
typedef struct {
/* 0000h */
    unsigned long opt_own;
    unsigned long opt_observ;
    long data_own;
    long data_observ;
/* 0010h */
    unsigned char reserve1[24];
    unsigned long long st_axbit;
/* 0030h */
    unsigned short st_pnt_s;
    unsigned short st_pnt_e;
    char reserve2[12];
/* 0040h */
    unsigned char reserve3[40];
    unsigned char dout_ctrl;
    unsigned char dout_num;
    unsigned short dout_ctrlbit;
    unsigned short dout_data;
    char reserve4[2];
/* 0070h */
    char reserve5[16];
/* 0080h */
} OAS_DATA;
```

詳細說明

■成員

成員	內容
opt_own	<p>本軸選單 應使用各值的邏輯或設置要設置的資料。</p> <p>■值</p> <p>[本軸判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OWN_REMAINING_DISTANCE: 剩餘距離指定 • SSC_OAS_OWN_POSITION_PASS: 指定位置通過指定 <p>[本軸判定座標]</p> <ul style="list-style-type: none"> • SSC_OAS_OWN_JUDGE_COORD_FB: F/B位置 • SSC_OAS_OWN_JUDGE_COORD_CMD: 陳述式位置
opt_observ	<p>監視軸選單 應使用各值的邏輯或設置要設置的資料。</p> <p>■值</p> <p>[監視軸指定]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_DISABLE: 無效 • SSC_OAS_OBSERV_ENABLE: 有效 <p>[監視軸判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_POSITION_PASS: 監視軸指定位置通過指定 <p>[監視軸判定座標]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_JUDGE_COORD_FB: F/B位置 • SSC_OAS_OBSERV_JUDGE_COORD_CMD: 陳述式位置 <p>[監視軸指定位置通過判定條件]</p> <ul style="list-style-type: none"> • SSC_OAS_OBSERV_DATA_LESS: 監視軸的位置在監視軸指定位置資料及以下時條件成立 • SSC_OAS_OBSERV_DATA_MORE: 監視軸的位置在監視軸指定位置資料及以上時條件成立
data_own	本軸剩餘距離資料(或本軸通過位置資料)[陳述式單位]
data_observ	監視軸剩餘距離資料(或監視軸通過位置資料)[陳述式單位]
st_axbit	啟動軸指定
st_pnt_s	啟動軸起始點(0~2047)
st_pnt_e	啟動軸結束點(0~2047)
dout_ctrl	<p>數位輸出信號控制/輸出軟元件信號控制</p> <p>■值</p> <p>[數位輸出信號控制]</p> <ul style="list-style-type: none"> • SSC_OAS_DO_DISABLE: 無效 • SSC_OAS_DO_ENABLE: 有效
dout_num	數位輸出信號編號(00h~3Fh)/輸出軟元件信號編號(00h~23h)
dout_ctrlbit	數位輸出信號有效選擇/輸出軟元件信號有效選擇

成員	內容
dout_data	數位輸出信號陳述式/輸出軟元件信號陳述式

■要點

- 將監視軸指定設置為“有效(SSC_OAS_OBSERV_ENABLE)”的情況下，必須設置“監視軸指定位置通過指定(SSC_OAS_OBSERV_POSITION_PASS)”。
- 在“輸入輸出表(參數No. 004A)”中選擇“2: 使用輸入輸出軟元件表(點數擴展方式)”時，應按以下方式設置輸出軟元件信號控制與輸出軟元件信號編號。

<例> 設置DVO_0120~DVO_012F的情況下

```
OAS_DATA oasData;
oasData.dout_ctrl = ( SSC_OAS_DO_ENABLE | ( 0x2 << 4 ) );
oasData.dout_num = 0x01;
```

<例> 設置DVO_23F0~DVO_23FF的情況下

```
OAS_DATA oasData;
oasData.dout_ctrl = ( SSC_OAS_DO_ENABLE | ( 0xF << 4 ) );
oasData.dout_num = 0x23;
```

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetOtherAxisStartData([↖](#) 131頁 sscSetOtherAxisStartData)
- sscGetOtherAxisStartData([↖](#) 132頁 sscGetOtherAxisStartData)

5.3 PRESS_DATA結構

該結構是自動運行(擋塊控制)中使用的擋塊控制資料結構。

```
typedef struct {
/* 0000h */
    long switch_position;
    long position_limit;
    long speed_limit;
    unsigned short target_torque;
    unsigned short continue_time;
/* 0010h */
    unsigned short torque_settle_width;
    unsigned short torque_settle_time;
    unsigned short actime;
    unsigned short dctime;
    unsigned short condition;
    char reserve1[6];
/* 0020h */
} PRESS_DATA;
```

詳細說明

■成員

成員	內容
switch_position	擋塊控制切換位置[陳述式位置]
position_limit	擋塊極限位置[陳述式位置]
speed_limit	擋塊控制用速度限制值[速度單位](1~2147483647)
target_torque	目標轉矩[0.1%](0~32767)
continue_time	擋塊時間[ms]
torque_settle_width	轉矩整定幅度[0.1%]
torque_settle_time	轉矩整定等待時間[ms]
actime	擋塊控制用加速時間常數[ms](0~20000)
dctime	擋塊控制用減速時間常數[ms](0~20000)
condition	擋塊控制動作條件 應使用各值的邏輯或設置要設置的資料。 ■值 [擋塊控制切換開始時條件] • SSC_PRESS_START_AUTO_CMD: 自動切換(陳述式位置) • SSC_PRESS_START_AUTO_FB: 自動切換(F/B位置) • SSC_PRESS_START_MANUAL: 手動切換 [擋塊控制切換結束時條件] • SSC_PRESS_END_AUTO: 自動切換 • SSC_PRESS_END_MANUAL: 手動切換

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscSetPressData([↗](#) 79頁 sscSetPressData)
- sscGetPressData([↗](#) 80頁 sscGetPressData)

5.4 SMP_ERR結構

該結構是採樣中使用的採樣出錯結構。

SMP_ERR結構 [MC200]

```
typedef struct {  
    /* 0000h */  
    unsigned long long err_ax;  
    char reserve1[8];  
    unsigned long err_dat;  
    char reserve2[4];  
    unsigned long err_bit;  
    char reserve3[4];  
    /* 0020h */  
} SMP_ERR;
```

詳細說明

■成員

成員	內容
err_ax	軸出錯資訊
err_dat	資料出錯資訊
err_bit	位出錯資訊


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	A0	mc2xxstd.h

■參照

sscGetSamplingError( 144頁 sscGetSamplingError)

SMP_ERR結構[MC300]

```
typedef struct {
/* 0000h */
    unsigned long long err_ax;
    char reserve1[8];
/* 0010h */
    char reserve2[8];
    unsigned long err_dat;
    char reserve3[4];
/* 0020h */
    unsigned long err_bit;
    char reserve4[4];
    char reserve5[8];
/* 0030h */
} SMP_ERR;
```

詳細說明

■成員

成員	內容
err_ax	軸出錯資訊
err_dat	資料出錯資訊
err_bit	位出錯資訊


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.40	A0	mc3xxstd.h

■參照

sscGetSamplingError( 144頁 sscGetSamplingError)

5.5 SMP_DATA結構

該結構是採樣中使用的採樣資料結構。

SMP_DATA結構[MC200]

```
typedef struct {  
/* 0000h */  
    long smpdata[32];  
/* 0080h */  
    unsigned short smpbit[1];  
    unsigned char reserve1[2];  
/* 0084h */  
} SMP_DATA;
```

詳細說明

■成員

成員	內容
smpdata[32]	資料1~32
smpbit[1]	位資訊

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h

■參照

sscGetSamplingData([↗](#) 146頁 sscGetSamplingData)

SMP_DATA結構[MC300]

```
typedef struct {  
/* 0000h */  
    long smpdata[32];  
/* 0080h */  
    unsigned long smpbit[1];  
/* 0084h */  
} SMP_DATA;
```

詳細說明

■成員

成員	內容
smpdata[32]	資料1~32
smpbit[1]	位資訊

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscGetSamplingData([↗](#) 146頁 sscGetSamplingData)

5.6 LOG_DATA結構

該結構是日誌功能中使用的日誌資料結構。

```
typedef struct {  
/* 0000h */  
    short axnum;  
    short eventcode;  
    long eventtime;  
    short eventdata[4];  
/* 0010h */  
} LOG_DATA;
```

詳細說明

■成員

成員	內容
axnum	軸編號
eventcode	事件代碼
eventtime	時間戳記
eventdata[4]	各事件資訊

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscReadLogData ([↩](#) 151頁 sscReadLogData)

5.7 ALH_DATA結構

該結構是報警履歷功能中使用的報警履歷資料結構。

ALH_DATA結構[MC200]

```
typedef struct {
/* 0000h */
    long long system_time;
    unsigned long free_run_cnt;
    unsigned char ctrl_cycle;
    unsigned char event_code;
    char reserve1[2];
/* 0010h */
    unsigned char sscnet_type;
    unsigned char ctrl_mode;
    char reserve2[2];
    unsigned short axnum;
    unsigned short alarm_code;
    unsigned char drive_mode;
    char reserve3[3];
    long cmd_pos;
/* 0020h */
    long fb_pos;
    unsigned short alarm_code_j5;
    unsigned short alarm_detail_code_j5;
    char reserve4[23];
    unsigned char check_sum;
/* 0040h */
} ALH_DATA;
```

5

詳細說明

■成員

成員	內容
system_time	系統啟動時刻
free_run_cnt	自由運行計數器
ctrl_cycle	控制週期
event_code	事件代碼
sscnet_type	通信模式
ctrl_mode	控制模式
axnum	發生軸編號
alarm_code	報警編號
drive_mode	運行模式
cmd_pos	當前位置[陳述式單位]
fb_pos	F/B位置[陳述式單位]
alarm_code_j5	報警編號(MR-J5(W)-B)
alarm_detail_code_j5	報警詳細編號(MR-J5(W)-B)
check_sum	校驗和

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	2.40	B7	mc2xxstd.h

■參照

sscGetAlarmHistoryData (🔗 153頁 sscGetAlarmHistoryData)

ALH_DATA結構[MC300]

```
typedef struct {
/* 0000h */
    long long system_time;
    unsigned long free_run_cnt;
    unsigned char ctrl_cycle;
    unsigned char event_code;
    char reserve1[2];
/* 0010h */
    char reserve2[4];
    unsigned short axnum;
    unsigned short alarm_code;
    unsigned char drive_mode;
    char reserve3[3];
    long cmd_pos;
/* 0020h */
    long fb_pos;
    unsigned short alarm_code_j5;
    unsigned short alarm_detail_code_j5;
    char reserve4[23];
    unsigned char check_sum;
/* 0040h */
} ALH_DATA;
```

詳細說明

■成員

成員	內容
system_time	系統啟動時刻
free_run_cnt	自由運行計數器
ctrl_cycle	控制週期
event_code	事件代碼
axnum	發生軸編號
alarm_code	報警編號
drive_mode	運行模式
cmd_pos	當前位置[陳述式單位]
fb_pos	F/B位置[陳述式單位]
alarm_code_j5	報警編號(MR-J5(W)-B)
alarm_detail_code_j5	報警詳細編號(MR-J5(W)-B)
check_sum	校驗和


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.40	A6	mc3xxstd.h

■參照

sscGetAlarmHistoryData( 153頁 sscGetAlarmHistoryData)

5.8 TRANSIENT_CMD結構

該結構是暫態傳送資料發送接收中使用的暫態傳送陳述式資料結構。

```
typedef struct {
/* 0000h */
    unsigned short cmd_req;
    unsigned short command;
    unsigned short req_data[4];
    unsigned short reserve[2];
/* 0010h */
} TRANSIENT_CMD;
```

詳細說明

■成員

成員	內容
cmd_req	陳述式發送請求 ■值 • SSC_TRANSIENT_CMD_SINGLE: 暫態請求
command	暫態陳述式
req_data[4]	請求資料


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscSendReceiveTransientData( 214頁 sscSendReceiveTransientData)

5.9 TRANSIENT_STS結構

該結構是暫態傳送資料發送接收中使用的暫態傳送狀態資料結構。

```
typedef struct {  
    /* 0000h */  
    unsigned short status;  
    unsigned short reserve1;  
    unsigned short ans_data[4];  
    unsigned short reserve2[2];  
    /* 0010h */  
} TRANSIENT_STS;
```

詳細說明

■成員

成員	內容
status	暫態狀態 使用各值的邏輯或設置已獲取的資料。 ■值 • SSC_TRANSIENT_STS_WAITING: 暫態陳述式處理完成等待 • SSC_TRANSIENT_STS_START: 暫態請求開始 • SSC_TRANSIENT_STS_RECEIVING: 暫態接收中 • SSC_TRANSIENT_STS_RECEIVE_FIN: 暫態正常接收完成 • SSC_TRANSIENT_STS_VALID_DATA: 暫態資料有效 • SSC_TRANSIENT_STS_INVALID_DATA: 暫態資料無效
ans_data[4]	回應資料


■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1. 80	A8	mc2xxstd.h
MR-MC3_ _	1. 00	A0	mc3xxstd.h

■參照

sscSendReceiveTransientData( 214頁 sscSendReceiveTransientData)

5.10 INT_CB_DATA結構

該結構是中斷回呼函數中使用的中斷資料結構。

INT_CB_DATA結構[MC200]

```
typedef struct {
/* 0000h */
    int board_id;
    int channel;
    unsigned long free_run_cnt;
    unsigned char sys_factor_bit;
    char reserve1;
    unsigned short sys_factor;
/* 0010H */
    unsigned long long axis_factor_bit;
    char reserve2[8];
    unsigned long axis_factor[48];
/* 00E0H */
    unsigned short unit_factor_bit;
    char reserve3[14];
    unsigned short unit_factor[8];
/* 0100H */
    unsigned long oas_factor_bit;
    char reserve4[12];
    unsigned char oas_factor[32];
/* 0130H */
    unsigned long long pass_factor_bit;
    char reserve5[8];
    unsigned char pass_factor[64];
/* 0180H */
    char reserve6[0x180];
/* 0300H */
} INT_CB_DATA;
```

5

詳細說明

■成員

成員	內容
board_id	板ID編號
channel	通道編號
free_run_cnt	自由運行計數器
sys_factor_bit	系統中斷原因輸出中位元
sys_factor	系統中斷原因
axis_factor_bit	軸中斷原因輸出中位
axis_factor[48]	軸中斷原因
unit_factor_bit	站中斷原因輸出中位
unit_factor[8]	站中斷原因
oas_factor_bit	其他軸啟動中斷原因
oas_factor[32]	其他軸啟動中斷原因詳細
pass_factor_bit	通過位置中斷原因
pass_factor[64]	通過位置中斷原因詳細

■要點

API版本1.61及以後的情況下，如果啟用介面模式的事件監測並啟動系統，則在軸中斷原因輸出中位及軸中斷原因中將儲存以下資訊。這些值會在每次事件發生時更新。

- 軸中斷原因輸出中位：在對應的軸上發生事件時ON。
 - 軸中斷原因：儲存事件檢測資訊。
- 關於儲存的資料內容，請參閱下述手冊的“事件原因”。

📖MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC2_ _	1.80	A8	mc2xxstd.h

■參照

sscRegisterIntCallback([↩](#) 187頁 sscRegisterIntCallback)

INT_CB_DATA結構[MC300]

```
typedef struct {
/* 0000h */
    int board_id;
    int channel;
    unsigned long free_run_cnt;
    unsigned char sys_factor_bit;
    char reserve1;
    unsigned short sys_factor;
/* 0010H */
    unsigned long long axis_factor_bit;
    char reserve2[8];
    unsigned long axis_factor[128];
/* 0220H */
    unsigned short unit_factor_bit;
    char reserve3[14];
    unsigned short unit_factor[32];
/* 0270H */
    unsigned long long oas_factor_bit;
    char reserve4[8];
    unsigned char oas_factor[64];
/* 02C0H */
    unsigned long long pass_factor_bit[2];
    char reserve5[16];
    unsigned char pass_factor[128];
/* 0360H */
    unsigned long long event_factor_bit;
    char reserve6[8];
    unsigned long long event_factor[64];
/* 0570H */
    char reserve7[0x290];
/* 0800H */
} INT_CB_DATA;
```

5

詳細說明

■成員


成員	內容
board_id	板ID編號
channel	通道編號
free_run_cnt	自由運行計數器
sys_factor_bit	系統中斷原因輸出中位元
sys_factor	系統中斷原因
axis_factor_bit	軸中斷原因輸出中位
axis_factor[128]	軸中斷原因
unit_factor_bit	站中斷原因輸出中位
unit_factor[32]	站中斷原因
oas_factor_bit	其他軸啟動中斷原因
oas_factor[64]	其他軸啟動中斷原因詳細
pass_factor_bit[2]	通過位置中斷原因
pass_factor[128]	通過位置中斷原因詳細
event_factor_bit;	事件中斷原因
event_factor[64];	事件中斷原因詳細

■要點

如果啟用介面模式的事件監測並啟動系統，則在軸中斷原因輸出中位及軸中斷原因中將儲存以下資訊。這些值會在每次事件發生時更新。

- 軸中斷原因輸出中位：在對應的軸上發生事件時ON。
- 軸中斷原因：儲存事件檢測資訊。


關於儲存的資料內容，請參閱下述手冊的“事件原因”。

 MR-MC200/MR-MC300系列定位板用戶手冊(詳細篇)

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

sscRegisterIntCallback( 187頁 sscRegisterIntCallback)

5.11 SLAVE_INFO結構[MC300]

該結構是SSCNET通信的斷開・重新連接函數及控制中軸資訊獲取函數等中使用的軟元件資訊結構。

```
typedef struct {  
    /* 0000h */  
    unsigned char axis[64];  
    /* 0040h */  
    unsigned char unit[16];  
    /* 0050h */  
} SLAVE_INFO;
```

詳細說明

■成員

成員	內容
axis[64]	軸資訊 ■值 • SSC_SLAVE_ON: 控制軸或狀態位元ON • SSC_SLAVE_OFF: 未控制軸或狀態位元OFF
unit[16]	站資訊 ■值 • SSC_SLAVE_ON: 控制軸或狀態位元ON • SSC_SLAVE_OFF: 未控制軸或狀態位元OFF

■要點

無。

■對應版本

定位板	API版本	Board版本	頭檔案
MR-MC3_ _	1.00	A0	mc3xxstd.h

■參照

- sscReconnectSSCNETEx ([↗](#) 65頁 sscReconnectSSCNETEx [MC300])
- sscDisconnectSSCNETEx ([↗](#) 67頁 sscDisconnectSSCNETEx [MC300])
- sscIfmGetEventStatusBitsEx ([↗](#) 180頁 sscIfmGetEventStatusBitsEx [MC300])

6 位定義一覽

在sscSetCommandBitSignalEx函數、sscGetStatusBitSignalEx函數、sscWaitStatusBitSignalEx函數中指定的位元定義如下所示。

6.1 系統陳述式位

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_ITE	中斷處理結束
SSC_CMDBIT_SYS_ITS	中斷輸出有效
SSC_CMDBIT_SYS_03	廠商設置用
SSC_CMDBIT_SYS_04	
SSC_CMDBIT_SYS_HMA	用戶程式側記憶體訪問中
SSC_CMDBIT_SYS_06	廠商設置用
SSC_CMDBIT_SYS_07	
SSC_CMDBIT_SYS_08	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_SMPS	採樣開始
SSC_CMDBIT_SYS_10	廠商設置用
SSC_CMDBIT_SYS_11	
SSC_CMDBIT_SYS_12	
SSC_CMDBIT_SYS_13	
SSC_CMDBIT_SYS_14	
SSC_CMDBIT_SYS_15	
SSC_CMDBIT_SYS_16	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_SEMI	軟體緊急停止
SSC_CMDBIT_SYS_18	廠商設置用
SSC_CMDBIT_SYS_19	
SSC_CMDBIT_SYS_20	
SSC_CMDBIT_SYS_21	
SSC_CMDBIT_SYS_22	
SSC_CMDBIT_SYS_23	
SSC_CMDBIT_SYS_24	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_25	廠商設置用
SSC_CMDBIT_SYS_26	
SSC_CMDBIT_SYS_27	
SSC_CMDBIT_SYS_28	
SSC_CMDBIT_SYS_29	
SSC_CMDBIT_SYS_30	
SSC_CMDBIT_SYS_31	
SSC_CMDBIT_SYS_32	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_ITFE	中斷處理高速結束
SSC_CMDBIT_SYS_34	廠商設置用
SSC_CMDBIT_SYS_35	
SSC_CMDBIT_SYS_36	
SSC_CMDBIT_SYS_37	
SSC_CMDBIT_SYS_38	
SSC_CMDBIT_SYS_39	
SSC_CMDBIT_SYS_40	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_41	廠商設置用
SSC_CMDBIT_SYS_42	
SSC_CMDBIT_SYS_43	
SSC_CMDBIT_SYS_44	
SSC_CMDBIT_SYS_45	
SSC_CMDBIT_SYS_46	
SSC_CMDBIT_SYS_47	
SSC_CMDBIT_SYS_48	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_ASYN1	非同步陳述式(組1)
SSC_CMDBIT_SYS_ASYN2	非同步陳述式(組2)
SSC_CMDBIT_SYS_ASYN3	非同步陳述式(組3)
SSC_CMDBIT_SYS_ASYN4	非同步陳述式(組4)
SSC_CMDBIT_SYS_ASYN5	非同步陳述式(組5)
SSC_CMDBIT_SYS_ASYN6	非同步陳述式(組6)
SSC_CMDBIT_SYS_ASYN7	非同步陳述式(組7)
SSC_CMDBIT_SYS_ASYN8	非同步陳述式(組8)
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_57	廠商設置用
SSC_CMDBIT_SYS_58	
SSC_CMDBIT_SYS_59	
SSC_CMDBIT_SYS_60	
SSC_CMDBIT_SYS_61	
SSC_CMDBIT_SYS_62	
SSC_CMDBIT_SYS_63	
SSC_CMDBIT_SYS_64	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_RBR	重啟準備
SSC_CMDBIT_SYS_RBS	重啟執行
SSC_CMDBIT_SYS_CRST	系統報警復位
SSC_CMDBIT_SYS_68	廠商設置用
SSC_CMDBIT_SYS_SMON	系統監視陳述式
SSC_CMDBIT_SYS_SMONR	系統監視鎖存陳述式
SSC_CMDBIT_SYS_71	廠商設置用
SSC_CMDBIT_SYS_72	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_73	廠商設置用
SSC_CMDBIT_SYS_74	
SSC_CMDBIT_SYS_75	
SSC_CMDBIT_SYS_76	
SSC_CMDBIT_SYS_77	
SSC_CMDBIT_SYS_78	
SSC_CMDBIT_SYS_79	
SSC_CMDBIT_SYS_80	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_LOGC	日誌陳述式
SSC_CMDBIT_SYS_LOGC	日誌資料讀取陳述式
SSC_CMDBIT_SYS_83	廠商設置用
SSC_CMDBIT_SYS_LOGI	日誌資料初始化陳述式
SSC_CMDBIT_SYS_85	廠商設置用
SSC_CMDBIT_SYS_OCMC	運算週期監視清除
SSC_CMDBIT_SYS_87	廠商設置用
SSC_CMDBIT_SYS_88	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_RCC	重新連接陳述式
SSC_CMDBIT_SYS_90	廠商設置用
SSC_CMDBIT_SYS_91	
SSC_CMDBIT_SYS_CCC	斷開陳述式
SSC_CMDBIT_SYS_93	廠商設置用
SSC_CMDBIT_SYS_94	
SSC_CMDBIT_SYS_95	
SSC_CMDBIT_SYS_96	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_97	廠商設置用
SSC_CMDBIT_SYS_98	
SSC_CMDBIT_SYS_99	
SSC_CMDBIT_SYS_100	
SSC_CMDBIT_SYS_101	
SSC_CMDBIT_SYS_102	
SSC_CMDBIT_SYS_103	
SSC_CMDBIT_SYS_104	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_105	廠商設置用
SSC_CMDBIT_SYS_106	
SSC_CMDBIT_SYS_107	
SSC_CMDBIT_SYS_108	
SSC_CMDBIT_SYS_109	
SSC_CMDBIT_SYS_110	
SSC_CMDBIT_SYS_111	
SSC_CMDBIT_SYS_112	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_113	廠商設置用
SSC_CMDBIT_SYS_114	
SSC_CMDBIT_SYS_115	
SSC_CMDBIT_SYS_116	
SSC_CMDBIT_SYS_117	
SSC_CMDBIT_SYS_118	
SSC_CMDBIT_SYS_119	
SSC_CMDBIT_SYS_120	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_121	廠商設置用
SSC_CMDBIT_SYS_122	
SSC_CMDBIT_SYS_123	
SSC_CMDBIT_SYS_124	
SSC_CMDBIT_SYS_125	
SSC_CMDBIT_SYS_126	
SSC_CMDBIT_SYS_127	
SSC_CMDBIT_SYS_128	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_SPWRT	參數寫入陳述式
SSC_CMDBIT_SYS_130	廠商設置用
SSC_CMDBIT_SYS_131	
SSC_CMDBIT_SYS_132	
SSC_CMDBIT_SYS_133	
SSC_CMDBIT_SYS_134	
SSC_CMDBIT_SYS_135	
SSC_CMDBIT_SYS_136	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_SPRD	參數讀取陳述式
SSC_CMDBIT_SYS_138	廠商設置用
SSC_CMDBIT_SYS_139	
SSC_CMDBIT_SYS_140	
SSC_CMDBIT_SYS_141	
SSC_CMDBIT_SYS_142	
SSC_CMDBIT_SYS_143	
SSC_CMDBIT_SYS_144	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_SMPSW	採樣設置寫入陳述式
SSC_CMDBIT_SYS_146	廠商設置用
SSC_CMDBIT_SYS_147	
SSC_CMDBIT_SYS_148	
SSC_CMDBIT_SYS_SMPSR	採樣設置讀取陳述式
SSC_CMDBIT_SYS_150	廠商設置用
SSC_CMDBIT_SYS_151	
SSC_CMDBIT_SYS_152	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_153	廠商設置用
SSC_CMDBIT_SYS_154	
SSC_CMDBIT_SYS_155	
SSC_CMDBIT_SYS_156	
SSC_CMDBIT_SYS_157	
SSC_CMDBIT_SYS_158	
SSC_CMDBIT_SYS_159	
SSC_CMDBIT_SYS_160	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_161	廠商設置用
SSC_CMDBIT_SYS_162	
SSC_CMDBIT_SYS_163	
SSC_CMDBIT_SYS_164	
SSC_CMDBIT_SYS_165	
SSC_CMDBIT_SYS_166	
SSC_CMDBIT_SYS_167	
SSC_CMDBIT_SYS_168	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_169	廠商設置用
SSC_CMDBIT_SYS_170	
SSC_CMDBIT_SYS_171	
SSC_CMDBIT_SYS_172	
SSC_CMDBIT_SYS_173	
SSC_CMDBIT_SYS_174	
SSC_CMDBIT_SYS_175	
SSC_CMDBIT_SYS_176	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_FTR	快閃記憶體傳送準備
SSC_CMDBIT_SYS_FTS	快閃記憶體傳送執行
SSC_CMDBIT_SYS_179	廠商設置用
SSC_CMDBIT_SYS_180	
SSC_CMDBIT_SYS_FIR	快閃記憶體初始化準備
SSC_CMDBIT_SYS_FIS	快閃記憶體初始化執行
SSC_CMDBIT_SYS_183	廠商設置用
SSC_CMDBIT_SYS_184	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_ALHR	報警履歷讀取陳述式
SSC_CMDBIT_SYS_186	廠商設置用
SSC_CMDBIT_SYS_ALHI	
SSC_CMDBIT_SYS_188	報警履歷初始化陳述式
SSC_CMDBIT_SYS_189	
SSC_CMDBIT_SYS_190	
SSC_CMDBIT_SYS_191	
SSC_CMDBIT_SYS_192	
SSC_CMDBIT_SYS_192	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_193	廠商設置用
SSC_CMDBIT_SYS_194	
SSC_CMDBIT_SYS_195	
SSC_CMDBIT_SYS_196	
SSC_CMDBIT_SYS_197	
SSC_CMDBIT_SYS_198	
SSC_CMDBIT_SYS_199	
SSC_CMDBIT_SYS_200	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_201	廠商設置用
SSC_CMDBIT_SYS_202	
SSC_CMDBIT_SYS_203	
SSC_CMDBIT_SYS_204	
SSC_CMDBIT_SYS_205	
SSC_CMDBIT_SYS_206	
SSC_CMDBIT_SYS_207	
SSC_CMDBIT_SYS_208	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_209	廠商設置用
SSC_CMDBIT_SYS_210	
SSC_CMDBIT_SYS_211	
SSC_CMDBIT_SYS_212	
SSC_CMDBIT_SYS_213	
SSC_CMDBIT_SYS_214	
SSC_CMDBIT_SYS_215	
SSC_CMDBIT_SYS_216	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_217	廠商設置用
SSC_CMDBIT_SYS_218	
SSC_CMDBIT_SYS_219	
SSC_CMDBIT_SYS_220	
SSC_CMDBIT_SYS_221	
SSC_CMDBIT_SYS_222	
SSC_CMDBIT_SYS_223	
SSC_CMDBIT_SYS_224	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_225	廠商設置用
SSC_CMDBIT_SYS_226	
SSC_CMDBIT_SYS_227	
SSC_CMDBIT_SYS_228	
SSC_CMDBIT_SYS_229	
SSC_CMDBIT_SYS_230	
SSC_CMDBIT_SYS_231	
SSC_CMDBIT_SYS_232	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_233	廠商設置用
SSC_CMDBIT_SYS_234	
SSC_CMDBIT_SYS_235	
SSC_CMDBIT_SYS_236	
SSC_CMDBIT_SYS_237	
SSC_CMDBIT_SYS_238	
SSC_CMDBIT_SYS_239	
SSC_CMDBIT_SYS_240	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_241	廠商設置用
SSC_CMDBIT_SYS_242	
SSC_CMDBIT_SYS_243	
SSC_CMDBIT_SYS_244	
SSC_CMDBIT_SYS_245	
SSC_CMDBIT_SYS_246	
SSC_CMDBIT_SYS_247	
SSC_CMDBIT_SYS_248	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_249	廠商設置用
SSC_CMDBIT_SYS_250	
SSC_CMDBIT_SYS_251	
SSC_CMDBIT_SYS_252	
SSC_CMDBIT_SYS_253	
SSC_CMDBIT_SYS_254	
SSC_CMDBIT_SYS_255	
SSC_CMDBIT_SYS_256	

系統陳述式位 [MC300]

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_257	廠商設置用
SSC_CMDBIT_SYS_258	
SSC_CMDBIT_SYS_259	
SSC_CMDBIT_SYS_260	
SSC_CMDBIT_SYS_261	
SSC_CMDBIT_SYS_262	
SSC_CMDBIT_SYS_263	
SSC_CMDBIT_SYS_264	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_265	廠商設置用
SSC_CMDBIT_SYS_266	
SSC_CMDBIT_SYS_267	
SSC_CMDBIT_SYS_268	
SSC_CMDBIT_SYS_269	
SSC_CMDBIT_SYS_270	
SSC_CMDBIT_SYS_271	
SSC_CMDBIT_SYS_272	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_273	廠商設置用
SSC_CMDBIT_SYS_274	
SSC_CMDBIT_SYS_275	
SSC_CMDBIT_SYS_276	
SSC_CMDBIT_SYS_277	
SSC_CMDBIT_SYS_278	
SSC_CMDBIT_SYS_279	
SSC_CMDBIT_SYS_280	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_281	廠商設置用
SSC_CMDBIT_SYS_282	
SSC_CMDBIT_SYS_283	
SSC_CMDBIT_SYS_284	
SSC_CMDBIT_SYS_285	
SSC_CMDBIT_SYS_286	
SSC_CMDBIT_SYS_287	
SSC_CMDBIT_SYS_288	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_289	廠商設置用
SSC_CMDBIT_SYS_290	
SSC_CMDBIT_SYS_291	
SSC_CMDBIT_SYS_292	
SSC_CMDBIT_SYS_293	
SSC_CMDBIT_SYS_294	
SSC_CMDBIT_SYS_295	
SSC_CMDBIT_SYS_296	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_297	廠商設置用
SSC_CMDBIT_SYS_298	
SSC_CMDBIT_SYS_299	
SSC_CMDBIT_SYS_300	
SSC_CMDBIT_SYS_301	
SSC_CMDBIT_SYS_302	
SSC_CMDBIT_SYS_303	
SSC_CMDBIT_SYS_304	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_305	廠商設置用
SSC_CMDBIT_SYS_306	
SSC_CMDBIT_SYS_307	
SSC_CMDBIT_SYS_308	
SSC_CMDBIT_SYS_309	
SSC_CMDBIT_SYS_310	
SSC_CMDBIT_SYS_311	
SSC_CMDBIT_SYS_312	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_313	廠商設置用
SSC_CMDBIT_SYS_314	
SSC_CMDBIT_SYS_315	
SSC_CMDBIT_SYS_316	
SSC_CMDBIT_SYS_317	
SSC_CMDBIT_SYS_318	
SSC_CMDBIT_SYS_319	
SSC_CMDBIT_SYS_320	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_321	廠商設置用
SSC_CMDBIT_SYS_322	
SSC_CMDBIT_SYS_323	
SSC_CMDBIT_SYS_324	
SSC_CMDBIT_SYS_325	
SSC_CMDBIT_SYS_326	
SSC_CMDBIT_SYS_327	
SSC_CMDBIT_SYS_328	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_329	廠商設置用
SSC_CMDBIT_SYS_330	
SSC_CMDBIT_SYS_331	
SSC_CMDBIT_SYS_332	
SSC_CMDBIT_SYS_333	
SSC_CMDBIT_SYS_334	
SSC_CMDBIT_SYS_335	
SSC_CMDBIT_SYS_336	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_337	廠商設置用
SSC_CMDBIT_SYS_338	
SSC_CMDBIT_SYS_339	
SSC_CMDBIT_SYS_340	
SSC_CMDBIT_SYS_341	
SSC_CMDBIT_SYS_342	
SSC_CMDBIT_SYS_343	
SSC_CMDBIT_SYS_344	
位編號(常數)	信號名稱
SSC_CMDBIT_SYS_345	廠商設置用
SSC_CMDBIT_SYS_346	
SSC_CMDBIT_SYS_347	
SSC_CMDBIT_SYS_348	
SSC_CMDBIT_SYS_349	
SSC_CMDBIT_SYS_350	
SSC_CMDBIT_SYS_351	
SSC_CMDBIT_SYS_352	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_353	廠商設置用
SSC_CMDBIT_SYS_354	
SSC_CMDBIT_SYS_355	
SSC_CMDBIT_SYS_356	
SSC_CMDBIT_SYS_357	
SSC_CMDBIT_SYS_358	
SSC_CMDBIT_SYS_359	
SSC_CMDBIT_SYS_360	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_361	廠商設置用
SSC_CMDBIT_SYS_362	
SSC_CMDBIT_SYS_363	
SSC_CMDBIT_SYS_364	
SSC_CMDBIT_SYS_365	
SSC_CMDBIT_SYS_366	
SSC_CMDBIT_SYS_367	
SSC_CMDBIT_SYS_368	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_369	廠商設置用
SSC_CMDBIT_SYS_370	
SSC_CMDBIT_SYS_371	
SSC_CMDBIT_SYS_372	
SSC_CMDBIT_SYS_373	
SSC_CMDBIT_SYS_374	
SSC_CMDBIT_SYS_375	
SSC_CMDBIT_SYS_376	

位編號(常數)	信號名稱
SSC_CMDBIT_SYS_377	廠商設置用
SSC_CMDBIT_SYS_378	
SSC_CMDBIT_SYS_379	
SSC_CMDBIT_SYS_380	
SSC_CMDBIT_SYS_381	
SSC_CMDBIT_SYS_382	
SSC_CMDBIT_SYS_383	
SSC_CMDBIT_SYS_384	

6.2 系統狀態位元

位編號(常數)	信號名稱
SSC_STSBIT_SYS_ITO	中斷原因輸出中
SSC_STSBIT_SYS_IITO	介面模式中斷有效中
SSC_STSBIT_SYS_EVDO	事件檢測有效中
SSC_STSBIT_SYS_HRIF	高回應I/F有效中
SSC_STSBIT_SYS_BMA	系統程式側記憶體訪問中
SSC_STSBIT_SYS_PRINF	擋塊控制對應資訊
SSC_STSBIT_SYS_07	廠商設置用
SSC_STSBIT_SYS_IFMO	介面模式中
位編號(常數)	信號名稱
SSC_STSBIT_SYS_SMPW	採樣觸發等待
SSC_STSBIT_SYS_SMPO	採樣中
SSC_STSBIT_SYS_SMPF	採樣完成
SSC_STSBIT_SYS_SMPE	採樣不正確
SSC_STSBIT_SYS_13	廠商設置用
SSC_STSBIT_SYS_AHINF	報警履歷資訊
SSC_STSBIT_SYS_15	廠商設置用
SSC_STSBIT_SYS_16	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_EMIO	緊急停止中
SSC_STSBIT_SYS_18	廠商設置用
SSC_STSBIT_SYS_TSTO	測試模式中
SSC_STSBIT_SYS_20	廠商設置用
SSC_STSBIT_SYS_21	
SSC_STSBIT_SYS_22	
SSC_STSBIT_SYS_EMID	外部緊急停止無效中
SSC_STSBIT_SYS_24	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_SYS_25	廠商設置用
SSC_STSBIT_SYS_26	
SSC_STSBIT_SYS_27	
SSC_STSBIT_SYS_28	
SSC_STSBIT_SYS_29	
SSC_STSBIT_SYS_30	
SSC_STSBIT_SYS_IPCH	插補組可變
SSC_STSBIT_SYS_32	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_SYS_33	廠商設置用
SSC_STSBIT_SYS_34	
SSC_STSBIT_SYS_35	
SSC_STSBIT_SYS_36	
SSC_STSBIT_SYS_37	
SSC_STSBIT_SYS_38	
SSC_STSBIT_SYS_39	
SSC_STSBIT_SYS_40	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_41	廠商設置用
SSC_STSBIT_SYS_42	
SSC_STSBIT_SYS_43	
SSC_STSBIT_SYS_44	
SSC_STSBIT_SYS_45	
SSC_STSBIT_SYS_46	
SSC_STSBIT_SYS_47	
SSC_STSBIT_SYS_48	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_ASY01	非同步中(組1)
SSC_STSBIT_SYS_ASY02	非同步中(組2)
SSC_STSBIT_SYS_ASY03	非同步中(組3)
SSC_STSBIT_SYS_ASY04	非同步中(組4)
SSC_STSBIT_SYS_ASY05	非同步中(組5)
SSC_STSBIT_SYS_ASY06	非同步中(組6)
SSC_STSBIT_SYS_ASY07	非同步中(組7)
SSC_STSBIT_SYS_ASY08	非同步中(組8)
位編號(常數)	信號名稱
SSC_STSBIT_SYS_SYE01	同步中(組1)
SSC_STSBIT_SYS_SYE02	同步中(組2)
SSC_STSBIT_SYS_SYE03	同步中(組3)
SSC_STSBIT_SYS_SYE04	同步中(組4)
SSC_STSBIT_SYS_SYE05	同步中(組5)
SSC_STSBIT_SYS_SYE06	同步中(組6)
SSC_STSBIT_SYS_SYE07	同步中(組7)
SSC_STSBIT_SYS_SYE08	同步中(組8)
位編號(常數)	信號名稱
SSC_STSBIT_SYS_RBOK	重啟準備OK
SSC_STSBIT_SYS_RBNG	重啟準備不正確
SSC_STSBIT_SYS_CALM	系統報警中
SSC_STSBIT_SYS_68	廠商設置用
SSC_STSBIT_SYS_SMOUT	系統監視輸出中
SSC_STSBIT_SYS_SMRCH	系統監視鎖存中
SSC_STSBIT_SYS_SMER1	系統監視編號不正確1
SSC_STSBIT_SYS_SMER2	系統監視編號不正確2
位編號(常數)	信號名稱
SSC_STSBIT_SYS_73	廠商設置用
SSC_STSBIT_SYS_74	
SSC_STSBIT_SYS_75	
SSC_STSBIT_SYS_76	
SSC_STSBIT_SYS_77	
SSC_STSBIT_SYS_78	
SSC_STSBIT_SYS_79	
SSC_STSBIT_SYS_80	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_LOGO	日誌動作中
SSC_STSBIT_SYS_LOGRF	日誌資料讀取完成
SSC_STSBIT_SYS_LOGRE	日誌資料讀取不正確
SSC_STSBIT_SYS_LOGIF	日誌資料初始化完成
SSC_STSBIT_SYS_LOGIE	日誌資料初始化不正確
SSC_STSBIT_SYS_OCMCO	運算週期監視清除中
SSC_STSBIT_SYS_OCME	運算週期報警中
SSC_STSBIT_SYS_OCMW	運算週期警告中
位編號(常數)	信號名稱
SSC_STSBIT_SYS_RCO	重新連接處理中
SSC_STSBIT_SYS_RCF	重新連接完成
SSC_STSBIT_SYS_RCE	重新連接不正確
SSC_STSBIT_SYS_CCO	斷開處理中
SSC_STSBIT_SYS_CCF	斷開完成
SSC_STSBIT_SYS_CCE	斷開不正確
SSC_STSBIT_SYS_95	廠商設置用
SSC_STSBIT_SYS_96	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_97	廠商設置用
SSC_STSBIT_SYS_98	
SSC_STSBIT_SYS_99	
SSC_STSBIT_SYS_100	
SSC_STSBIT_SYS_101	
SSC_STSBIT_SYS_102	
SSC_STSBIT_SYS_103	
SSC_STSBIT_SYS_104	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_105	廠商設置用
SSC_STSBIT_SYS_106	
SSC_STSBIT_SYS_107	
SSC_STSBIT_SYS_108	
SSC_STSBIT_SYS_109	
SSC_STSBIT_SYS_110	
SSC_STSBIT_SYS_111	
SSC_STSBIT_SYS_112	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_113	廠商設置用
SSC_STSBIT_SYS_114	
SSC_STSBIT_SYS_115	
SSC_STSBIT_SYS_116	
SSC_STSBIT_SYS_117	
SSC_STSBIT_SYS_118	
SSC_STSBIT_SYS_119	
SSC_STSBIT_SYS_120	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_121	廠商設置用
SSC_STSBIT_SYS_122	
SSC_STSBIT_SYS_123	
SSC_STSBIT_SYS_124	
SSC_STSBIT_SYS_125	
SSC_STSBIT_SYS_126	
SSC_STSBIT_SYS_127	
SSC_STSBIT_SYS_128	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_SPWFIN1	參數寫入完成1
SSC_STSBIT_SYS_SPWEN1	參數編號不正確1
SSC_STSBIT_SYS_SPWED1	參數資料超出範圍1
SSC_STSBIT_SYS_132	廠商設置用
SSC_STSBIT_SYS_SPWFIN2	參數寫入完成2
SSC_STSBIT_SYS_SPWEN2	參數編號不正確2
SSC_STSBIT_SYS_SPWED2	參數資料超出範圍2
SSC_STSBIT_SYS_136	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_SYS_SPRFIN1	參數讀取完成1
SSC_STSBIT_SYS_SPREN1	參數編號不正確1
SSC_STSBIT_SYS_SPRFIN2	參數讀取完成2
SSC_STSBIT_SYS_SPREN2	參數編號不正確2
SSC_STSBIT_SYS_141	廠商設置用
SSC_STSBIT_SYS_142	
SSC_STSBIT_SYS_143	
SSC_STSBIT_SYS_144	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_SWFIN	採樣設置寫入完成
SSC_STSBIT_SYS_SWEN	採樣設置編號不正確
SSC_STSBIT_SYS_SWED	採樣設置資料超出範圍
SSC_STSBIT_SYS_148	廠商設置用
SSC_STSBIT_SYS_SRFIN	採樣設置讀取完成
SSC_STSBIT_SYS_SREN	採樣設置編號不正確
SSC_STSBIT_SYS_151	廠商設置用
SSC_STSBIT_SYS_152	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_153	廠商設置用
SSC_STSBIT_SYS_154	
SSC_STSBIT_SYS_155	
SSC_STSBIT_SYS_156	
SSC_STSBIT_SYS_157	
SSC_STSBIT_SYS_158	
SSC_STSBIT_SYS_159	
SSC_STSBIT_SYS_160	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_161	廠商設置用
SSC_STSBIT_SYS_162	
SSC_STSBIT_SYS_163	
SSC_STSBIT_SYS_164	
SSC_STSBIT_SYS_165	
SSC_STSBIT_SYS_166	
SSC_STSBIT_SYS_167	
SSC_STSBIT_SYS_168	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_169	廠商設置用
SSC_STSBIT_SYS_170	
SSC_STSBIT_SYS_171	
SSC_STSBIT_SYS_172	
SSC_STSBIT_SYS_173	
SSC_STSBIT_SYS_174	
SSC_STSBIT_SYS_175	
SSC_STSBIT_SYS_176	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_FROK	快閃記憶體傳送準備OK
SSC_STSBIT_SYS_FRNG	快閃記憶體傳送準備不正確
SSC_STSBIT_SYS_FSOK	快閃記憶體傳送完成
SSC_STSBIT_SYS_FSNG	快閃記憶體傳送不正確
SSC_STSBIT_SYS_FIROK	快閃記憶體初始化準備OK
SSC_STSBIT_SYS_FIRNG	快閃記憶體初始化準備不正確
SSC_STSBIT_SYS_FIOK	快閃記憶體初始化完成
SSC_STSBIT_SYS_FING	快閃記憶體初始化不正確
位編號(常數)	信號名稱
SSC_STSBIT_SYS_ALHRF	報警履歷讀取完成
SSC_STSBIT_SYS_ALHRE	報警履歷讀取不正確
SSC_STSBIT_SYS_ALHIF	報警履歷初始化完成
SSC_STSBIT_SYS_ALHIE	報警履歷初始化不正確
SSC_STSBIT_SYS_189	廠商設置用
SSC_STSBIT_SYS_190	
SSC_STSBIT_SYS_191	
SSC_STSBIT_SYS_192	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_193	廠商設置用
SSC_STSBIT_SYS_194	
SSC_STSBIT_SYS_195	
SSC_STSBIT_SYS_196	
SSC_STSBIT_SYS_197	
SSC_STSBIT_SYS_198	
SSC_STSBIT_SYS_199	
SSC_STSBIT_SYS_200	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_201	廠商設置用
SSC_STSBIT_SYS_202	
SSC_STSBIT_SYS_203	
SSC_STSBIT_SYS_204	
SSC_STSBIT_SYS_205	
SSC_STSBIT_SYS_206	
SSC_STSBIT_SYS_207	
SSC_STSBIT_SYS_208	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_209	廠商設置用
SSC_STSBIT_SYS_210	
SSC_STSBIT_SYS_211	
SSC_STSBIT_SYS_212	
SSC_STSBIT_SYS_213	
SSC_STSBIT_SYS_214	
SSC_STSBIT_SYS_215	
SSC_STSBIT_SYS_216	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_217	廠商設置用
SSC_STSBIT_SYS_218	
SSC_STSBIT_SYS_219	
SSC_STSBIT_SYS_220	
SSC_STSBIT_SYS_221	
SSC_STSBIT_SYS_222	
SSC_STSBIT_SYS_223	
SSC_STSBIT_SYS_224	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_225	廠商設置用
SSC_STSBIT_SYS_226	
SSC_STSBIT_SYS_227	
SSC_STSBIT_SYS_228	
SSC_STSBIT_SYS_229	
SSC_STSBIT_SYS_230	
SSC_STSBIT_SYS_231	
SSC_STSBIT_SYS_232	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_233	廠商設置用
SSC_STSBIT_SYS_234	
SSC_STSBIT_SYS_235	
SSC_STSBIT_SYS_236	
SSC_STSBIT_SYS_237	
SSC_STSBIT_SYS_238	
SSC_STSBIT_SYS_239	
SSC_STSBIT_SYS_240	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_241	廠商設置用
SSC_STSBIT_SYS_242	
SSC_STSBIT_SYS_243	
SSC_STSBIT_SYS_244	
SSC_STSBIT_SYS_245	
SSC_STSBIT_SYS_246	
SSC_STSBIT_SYS_247	
SSC_STSBIT_SYS_248	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_249	廠商設置用
SSC_STSBIT_SYS_250	
SSC_STSBIT_SYS_251	
SSC_STSBIT_SYS_252	
SSC_STSBIT_SYS_253	
SSC_STSBIT_SYS_254	
SSC_STSBIT_SYS_255	
SSC_STSBIT_SYS_256	

系統狀態位元[MC300]

位編號(常數)	信號名稱
SSC_STSBIT_SYS_257	廠商設置用
SSC_STSBIT_SYS_258	
SSC_STSBIT_SYS_259	
SSC_STSBIT_SYS_260	
SSC_STSBIT_SYS_261	
SSC_STSBIT_SYS_262	
SSC_STSBIT_SYS_263	
SSC_STSBIT_SYS_264	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_265	廠商設置用
SSC_STSBIT_SYS_266	
SSC_STSBIT_SYS_267	
SSC_STSBIT_SYS_268	
SSC_STSBIT_SYS_269	
SSC_STSBIT_SYS_270	
SSC_STSBIT_SYS_271	
SSC_STSBIT_SYS_272	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_273	廠商設置用
SSC_STSBIT_SYS_274	
SSC_STSBIT_SYS_275	
SSC_STSBIT_SYS_276	
SSC_STSBIT_SYS_277	
SSC_STSBIT_SYS_278	
SSC_STSBIT_SYS_279	
SSC_STSBIT_SYS_280	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_281	廠商設置用
SSC_STSBIT_SYS_282	
SSC_STSBIT_SYS_283	
SSC_STSBIT_SYS_284	
SSC_STSBIT_SYS_285	
SSC_STSBIT_SYS_286	
SSC_STSBIT_SYS_287	
SSC_STSBIT_SYS_288	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_289	廠商設置用
SSC_STSBIT_SYS_290	
SSC_STSBIT_SYS_291	
SSC_STSBIT_SYS_292	
SSC_STSBIT_SYS_293	
SSC_STSBIT_SYS_294	
SSC_STSBIT_SYS_295	
SSC_STSBIT_SYS_296	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_297	廠商設置用
SSC_STSBIT_SYS_298	
SSC_STSBIT_SYS_299	
SSC_STSBIT_SYS_300	
SSC_STSBIT_SYS_301	
SSC_STSBIT_SYS_302	
SSC_STSBIT_SYS_303	
SSC_STSBIT_SYS_304	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_305	廠商設置用
SSC_STSBIT_SYS_306	
SSC_STSBIT_SYS_307	
SSC_STSBIT_SYS_308	
SSC_STSBIT_SYS_309	
SSC_STSBIT_SYS_310	
SSC_STSBIT_SYS_311	
SSC_STSBIT_SYS_312	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_313	廠商設置用
SSC_STSBIT_SYS_314	
SSC_STSBIT_SYS_315	
SSC_STSBIT_SYS_316	
SSC_STSBIT_SYS_317	
SSC_STSBIT_SYS_318	
SSC_STSBIT_SYS_319	
SSC_STSBIT_SYS_320	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_321	廠商設置用
SSC_STSBIT_SYS_322	
SSC_STSBIT_SYS_323	
SSC_STSBIT_SYS_324	
SSC_STSBIT_SYS_325	
SSC_STSBIT_SYS_326	
SSC_STSBIT_SYS_327	
SSC_STSBIT_SYS_328	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_329	廠商設置用
SSC_STSBIT_SYS_330	
SSC_STSBIT_SYS_331	
SSC_STSBIT_SYS_332	
SSC_STSBIT_SYS_333	
SSC_STSBIT_SYS_334	
SSC_STSBIT_SYS_335	
SSC_STSBIT_SYS_336	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_337	廠商設置用
SSC_STSBIT_SYS_338	
SSC_STSBIT_SYS_339	
SSC_STSBIT_SYS_340	
SSC_STSBIT_SYS_341	
SSC_STSBIT_SYS_342	
SSC_STSBIT_SYS_343	
SSC_STSBIT_SYS_344	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_345	廠商設置用
SSC_STSBIT_SYS_346	
SSC_STSBIT_SYS_347	
SSC_STSBIT_SYS_348	
SSC_STSBIT_SYS_349	
SSC_STSBIT_SYS_350	
SSC_STSBIT_SYS_351	
SSC_STSBIT_SYS_352	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_353	廠商設置用
SSC_STSBIT_SYS_354	
SSC_STSBIT_SYS_355	
SSC_STSBIT_SYS_356	
SSC_STSBIT_SYS_357	
SSC_STSBIT_SYS_358	
SSC_STSBIT_SYS_359	
SSC_STSBIT_SYS_360	

位編號(常數)	信號名稱
SSC_STSBIT_SYS_361	廠商設置用
SSC_STSBIT_SYS_362	
SSC_STSBIT_SYS_363	
SSC_STSBIT_SYS_364	
SSC_STSBIT_SYS_365	
SSC_STSBIT_SYS_366	
SSC_STSBIT_SYS_367	
SSC_STSBIT_SYS_368	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_369	廠商設置用
SSC_STSBIT_SYS_370	
SSC_STSBIT_SYS_371	
SSC_STSBIT_SYS_372	
SSC_STSBIT_SYS_373	
SSC_STSBIT_SYS_374	
SSC_STSBIT_SYS_375	
SSC_STSBIT_SYS_376	
位編號(常數)	信號名稱
SSC_STSBIT_SYS_377	廠商設置用
SSC_STSBIT_SYS_378	
SSC_STSBIT_SYS_379	
SSC_STSBIT_SYS_380	
SSC_STSBIT_SYS_381	
SSC_STSBIT_SYS_382	
SSC_STSBIT_SYS_383	
SSC_STSBIT_SYS_384	

6.3 軸陳述式位

位編號(常數)	信號名稱
SSC_CMDBIT_AX_SON	伺服ON
SSC_CMDBIT_AX_2	廠商設置用
SSC_CMDBIT_AX_3	
SSC_CMDBIT_AX_4	
SSC_CMDBIT_AX_TL	
SSC_CMDBIT_AX_SRST	伺服報警復位
SSC_CMDBIT_AX_7	廠商設置用
SSC_CMDBIT_AX_8	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_ST	運行啟動
SSC_CMDBIT_AX_DIR	移動方向
SSC_CMDBIT_AX_STP	運行停止
SSC_CMDBIT_AX_RSTP	運行緊急停止
SSC_CMDBIT_AX_13	廠商設置用
SSC_CMDBIT_AX_ORST	運行報警復位
SSC_CMDBIT_AX_15	廠商設置用
SSC_CMDBIT_AX_16	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_AUT	自動運行模式
SSC_CMDBIT_AX_ZRN	原點復位模式
SSC_CMDBIT_AX_JOG	JOG運行模式
SSC_CMDBIT_AX_S	增量進給模式
SSC_CMDBIT_AX_21	廠商設置用
SSC_CMDBIT_AX_LIP	直線插補模式[MC200]/插補運行模式[MC300]
SSC_CMDBIT_AX_DST	原點重新設置模式
SSC_CMDBIT_AX_24	廠商設置用
位編號(常數)	信號名稱
SSC_CMDBIT_AX_25	廠商設置用
SSC_CMDBIT_AX_26	
SSC_CMDBIT_AX_27	
SSC_CMDBIT_AX_28	
SSC_CMDBIT_AX_29	
SSC_CMDBIT_AX_30	
SSC_CMDBIT_AX_31	
SSC_CMDBIT_AX_32	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_ITL	互鎖
SSC_CMDBIT_AX_RMONR	高速監視鎖存陳述式
SSC_CMDBIT_AX_35	廠商設置用
SSC_CMDBIT_AX_36	
SSC_CMDBIT_AX_LSPC	+側極限開關輸入
SSC_CMDBIT_AX_LSNC	-側極限開關輸入
SSC_CMDBIT_AX_DOGC	近點狗輸入
SSC_CMDBIT_AX_40	廠商設置用

位編號(常數)	信號名稱
SSC_CMDBIT_AX_SCHG	速度更改
SSC_CMDBIT_AX_TACHG	加速時間常數更改
SSC_CMDBIT_AX_TDCHG	減速時間常數更改
SSC_CMDBIT_AX_PCHG	位置更改
SSC_CMDBIT_AX_45	廠商設置用
SSC_CMDBIT_AX_46	
SSC_CMDBIT_AX_47	
SSC_CMDBIT_AX_48	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_FST	高速運行啟動
SSC_CMDBIT_AX_50	廠商設置用
SSC_CMDBIT_AX_51	
SSC_CMDBIT_AX_52	
SSC_CMDBIT_AX_53	
SSC_CMDBIT_AX_54	
SSC_CMDBIT_AX_55	
SSC_CMDBIT_AX_56	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_PP1STP	中斷通過位置中斷
SSC_CMDBIT_AX_58	廠商設置用
SSC_CMDBIT_AX_59	
SSC_CMDBIT_AX_60	
SSC_CMDBIT_AX_61	
SSC_CMDBIT_AX_62	
SSC_CMDBIT_AX_63	
SSC_CMDBIT_AX_64	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_GAIN	增益切換陳述式
SSC_CMDBIT_AX_FCLS	全封閉控制切換陳述式
SSC_CMDBIT_AX_67	廠商設置用
SSC_CMDBIT_AX_CPC	PID控制陳述式
SSC_CMDBIT_AX_69	廠商設置用
SSC_CMDBIT_AX_70	
SSC_CMDBIT_AX_71	
SSC_CMDBIT_AX_72	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_73	廠商設置用
SSC_CMDBIT_AX_74	
SSC_CMDBIT_AX_75	
SSC_CMDBIT_AX_76	
SSC_CMDBIT_AX_77	
SSC_CMDBIT_AX_78	
SSC_CMDBIT_AX_79	
SSC_CMDBIT_AX_80	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_81	廠商設置用
SSC_CMDBIT_AX_82	
SSC_CMDBIT_AX_83	
SSC_CMDBIT_AX_84	
SSC_CMDBIT_AX_ZSC	原點設置陳述式
SSC_CMDBIT_AX_86	廠商設置用
SSC_CMDBIT_AX_87	
SSC_CMDBIT_AX_88	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_89	廠商設置用
SSC_CMDBIT_AX_MKC1	標記檢測清除陳述式1
SSC_CMDBIT_AX_MKD1	標記檢測無效陳述式1
SSC_CMDBIT_AX_MKSEN1	標記檢測設置有效陳述式1
SSC_CMDBIT_AX_93	廠商設置用
SSC_CMDBIT_AX_MKC2	標記檢測清除陳述式2
SSC_CMDBIT_AX_MKD2	標記檢測無效陳述式2
SSC_CMDBIT_AX_MKSEN2	標記檢測設置有效陳述式2
位編號(常數)	信號名稱
SSC_CMDBIT_AX_97	廠商設置用
SSC_CMDBIT_AX_98	
SSC_CMDBIT_AX_99	
SSC_CMDBIT_AX_100	
SSC_CMDBIT_AX_CTLMC	控制模式切換陳述式
SSC_CMDBIT_AX_102	廠商設置用
SSC_CMDBIT_AX_103	
SSC_CMDBIT_AX_104	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_105	廠商設置用
SSC_CMDBIT_AX_106	
SSC_CMDBIT_AX_107	
SSC_CMDBIT_AX_108	
SSC_CMDBIT_AX_109	
SSC_CMDBIT_AX_110	
SSC_CMDBIT_AX_111	
SSC_CMDBIT_AX_112	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_113	廠商設置用
SSC_CMDBIT_AX_114	
SSC_CMDBIT_AX_115	
SSC_CMDBIT_AX_116	
SSC_CMDBIT_AX_117	
SSC_CMDBIT_AX_118	
SSC_CMDBIT_AX_119	
SSC_CMDBIT_AX_120	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_GAIN2	增益切換2陳述式
SSC_CMDBIT_AX_122	廠商設置用
SSC_CMDBIT_AX_123	
SSC_CMDBIT_AX_124	
SSC_CMDBIT_AX_125	
SSC_CMDBIT_AX_126	
SSC_CMDBIT_AX_127	
SSC_CMDBIT_AX_128	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_MON	監視陳述式
SSC_CMDBIT_AX_MONR	監視鎖存陳述式
SSC_CMDBIT_AX_131	廠商設置用
SSC_CMDBIT_AX_132	
SSC_CMDBIT_AX_133	
SSC_CMDBIT_AX_134	
SSC_CMDBIT_AX_135	
SSC_CMDBIT_AX_136	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_137	廠商設置用
SSC_CMDBIT_AX_138	
SSC_CMDBIT_AX_139	
SSC_CMDBIT_AX_140	
SSC_CMDBIT_AX_141	
SSC_CMDBIT_AX_142	
SSC_CMDBIT_AX_143	
SSC_CMDBIT_AX_144	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_145	廠商設置用
SSC_CMDBIT_AX_146	
SSC_CMDBIT_AX_147	
SSC_CMDBIT_AX_148	
SSC_CMDBIT_AX_149	
SSC_CMDBIT_AX_150	
SSC_CMDBIT_AX_151	
SSC_CMDBIT_AX_152	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_153	廠商設置用
SSC_CMDBIT_AX_154	
SSC_CMDBIT_AX_155	
SSC_CMDBIT_AX_156	
SSC_CMDBIT_AX_157	
SSC_CMDBIT_AX_158	
SSC_CMDBIT_AX_159	
SSC_CMDBIT_AX_160	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_PWRT	參數寫入陳述式
SSC_CMDBIT_AX_162	廠商設置用
SSC_CMDBIT_AX_163	
SSC_CMDBIT_AX_164	
SSC_CMDBIT_AX_165	
SSC_CMDBIT_AX_166	
SSC_CMDBIT_AX_167	
SSC_CMDBIT_AX_PSF	伺服參數讀取完成
位編號(常數)	信號名稱
SSC_CMDBIT_AX_PRD	參數讀取陳述式
SSC_CMDBIT_AX_170	廠商設置用
SSC_CMDBIT_AX_171	
SSC_CMDBIT_AX_172	
SSC_CMDBIT_AX_173	
SSC_CMDBIT_AX_174	
SSC_CMDBIT_AX_175	
SSC_CMDBIT_AX_176	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_177	廠商設置用
SSC_CMDBIT_AX_178	
SSC_CMDBIT_AX_179	
SSC_CMDBIT_AX_180	
SSC_CMDBIT_AX_181	
SSC_CMDBIT_AX_182	
SSC_CMDBIT_AX_183	
SSC_CMDBIT_AX_184	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_185	廠商設置用
SSC_CMDBIT_AX_186	
SSC_CMDBIT_AX_187	
SSC_CMDBIT_AX_188	
SSC_CMDBIT_AX_189	
SSC_CMDBIT_AX_190	
SSC_CMDBIT_AX_191	
SSC_CMDBIT_AX_192	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_193	廠商設置用
SSC_CMDBIT_AX_194	
SSC_CMDBIT_AX_195	
SSC_CMDBIT_AX_196	
SSC_CMDBIT_AX_197	
SSC_CMDBIT_AX_198	
SSC_CMDBIT_AX_199	
SSC_CMDBIT_AX_200	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_201	廠商設置用
SSC_CMDBIT_AX_202	
SSC_CMDBIT_AX_203	
SSC_CMDBIT_AX_204	
SSC_CMDBIT_AX_205	
SSC_CMDBIT_AX_206	
SSC_CMDBIT_AX_207	
SSC_CMDBIT_AX_208	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_209	廠商設置用
SSC_CMDBIT_AX_210	
SSC_CMDBIT_AX_211	
SSC_CMDBIT_AX_212	
SSC_CMDBIT_AX_213	
SSC_CMDBIT_AX_214	
SSC_CMDBIT_AX_215	
SSC_CMDBIT_AX_216	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_217	廠商設置用
SSC_CMDBIT_AX_218	
SSC_CMDBIT_AX_219	
SSC_CMDBIT_AX_220	
SSC_CMDBIT_AX_221	
SSC_CMDBIT_AX_222	
SSC_CMDBIT_AX_223	
SSC_CMDBIT_AX_224	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_225	廠商設置用
SSC_CMDBIT_AX_226	
SSC_CMDBIT_AX_227	
SSC_CMDBIT_AX_228	
SSC_CMDBIT_AX_229	
SSC_CMDBIT_AX_230	
SSC_CMDBIT_AX_231	
SSC_CMDBIT_AX_232	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_233	廠商設置用
SSC_CMDBIT_AX_234	
SSC_CMDBIT_AX_235	
SSC_CMDBIT_AX_236	
SSC_CMDBIT_AX_237	
SSC_CMDBIT_AX_238	
SSC_CMDBIT_AX_239	
SSC_CMDBIT_AX_240	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_241	廠商設置用
SSC_CMDBIT_AX_242	
SSC_CMDBIT_AX_243	
SSC_CMDBIT_AX_244	
SSC_CMDBIT_AX_245	
SSC_CMDBIT_AX_246	
SSC_CMDBIT_AX_247	
SSC_CMDBIT_AX_248	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_249	廠商設置用
SSC_CMDBIT_AX_250	
SSC_CMDBIT_AX_251	
SSC_CMDBIT_AX_252	
SSC_CMDBIT_AX_253	
SSC_CMDBIT_AX_254	
SSC_CMDBIT_AX_255	
SSC_CMDBIT_AX_256	

軸陳述式位 [MC300]

位編號(常數)	信號名稱
SSC_CMDBIT_AX_257	廠商設置用
SSC_CMDBIT_AX_258	
SSC_CMDBIT_AX_259	
SSC_CMDBIT_AX_260	
SSC_CMDBIT_AX_261	
SSC_CMDBIT_AX_262	
SSC_CMDBIT_AX_263	
SSC_CMDBIT_AX_264	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_265	廠商設置用
SSC_CMDBIT_AX_266	
SSC_CMDBIT_AX_267	
SSC_CMDBIT_AX_268	
SSC_CMDBIT_AX_269	
SSC_CMDBIT_AX_270	
SSC_CMDBIT_AX_271	
SSC_CMDBIT_AX_272	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_273	廠商設置用
SSC_CMDBIT_AX_274	
SSC_CMDBIT_AX_275	
SSC_CMDBIT_AX_276	
SSC_CMDBIT_AX_277	
SSC_CMDBIT_AX_278	
SSC_CMDBIT_AX_279	
SSC_CMDBIT_AX_280	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_281	廠商設置用
SSC_CMDBIT_AX_282	
SSC_CMDBIT_AX_283	
SSC_CMDBIT_AX_284	
SSC_CMDBIT_AX_285	
SSC_CMDBIT_AX_286	
SSC_CMDBIT_AX_287	
SSC_CMDBIT_AX_288	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_289	廠商設置用
SSC_CMDBIT_AX_290	
SSC_CMDBIT_AX_291	
SSC_CMDBIT_AX_292	
SSC_CMDBIT_AX_293	
SSC_CMDBIT_AX_294	
SSC_CMDBIT_AX_295	
SSC_CMDBIT_AX_296	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_297	廠商設置用
SSC_CMDBIT_AX_298	
SSC_CMDBIT_AX_299	
SSC_CMDBIT_AX_300	
SSC_CMDBIT_AX_301	
SSC_CMDBIT_AX_302	
SSC_CMDBIT_AX_303	
SSC_CMDBIT_AX_304	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_305	廠商設置用
SSC_CMDBIT_AX_306	
SSC_CMDBIT_AX_307	
SSC_CMDBIT_AX_308	
SSC_CMDBIT_AX_309	
SSC_CMDBIT_AX_310	
SSC_CMDBIT_AX_311	
SSC_CMDBIT_AX_312	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_313	廠商設置用
SSC_CMDBIT_AX_314	
SSC_CMDBIT_AX_315	
SSC_CMDBIT_AX_316	
SSC_CMDBIT_AX_317	
SSC_CMDBIT_AX_318	
SSC_CMDBIT_AX_319	
SSC_CMDBIT_AX_320	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_321	廠商設置用
SSC_CMDBIT_AX_322	
SSC_CMDBIT_AX_323	
SSC_CMDBIT_AX_324	
SSC_CMDBIT_AX_325	
SSC_CMDBIT_AX_326	
SSC_CMDBIT_AX_327	
SSC_CMDBIT_AX_328	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_329	廠商設置用
SSC_CMDBIT_AX_330	
SSC_CMDBIT_AX_331	
SSC_CMDBIT_AX_332	
SSC_CMDBIT_AX_333	
SSC_CMDBIT_AX_334	
SSC_CMDBIT_AX_335	
SSC_CMDBIT_AX_336	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_337	廠商設置用
SSC_CMDBIT_AX_338	
SSC_CMDBIT_AX_339	
SSC_CMDBIT_AX_340	
SSC_CMDBIT_AX_341	
SSC_CMDBIT_AX_342	
SSC_CMDBIT_AX_343	
SSC_CMDBIT_AX_344	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_345	廠商設置用
SSC_CMDBIT_AX_346	
SSC_CMDBIT_AX_347	
SSC_CMDBIT_AX_348	
SSC_CMDBIT_AX_349	
SSC_CMDBIT_AX_350	
SSC_CMDBIT_AX_351	
SSC_CMDBIT_AX_352	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_353	廠商設置用
SSC_CMDBIT_AX_354	
SSC_CMDBIT_AX_355	
SSC_CMDBIT_AX_356	
SSC_CMDBIT_AX_357	
SSC_CMDBIT_AX_358	
SSC_CMDBIT_AX_359	
SSC_CMDBIT_AX_360	

位編號(常數)	信號名稱
SSC_CMDBIT_AX_361	廠商設置用
SSC_CMDBIT_AX_362	
SSC_CMDBIT_AX_363	
SSC_CMDBIT_AX_364	
SSC_CMDBIT_AX_365	
SSC_CMDBIT_AX_366	
SSC_CMDBIT_AX_367	
SSC_CMDBIT_AX_368	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_369	廠商設置用
SSC_CMDBIT_AX_370	
SSC_CMDBIT_AX_371	
SSC_CMDBIT_AX_372	
SSC_CMDBIT_AX_373	
SSC_CMDBIT_AX_374	
SSC_CMDBIT_AX_375	
SSC_CMDBIT_AX_376	
位編號(常數)	信號名稱
SSC_CMDBIT_AX_377	廠商設置用
SSC_CMDBIT_AX_378	
SSC_CMDBIT_AX_379	
SSC_CMDBIT_AX_380	
SSC_CMDBIT_AX_381	
SSC_CMDBIT_AX_382	
SSC_CMDBIT_AX_383	
SSC_CMDBIT_AX_384	

6.4 軸狀態位元

位編號(常數)	信號名稱
SSC_STSBIT_AX_RDY	伺服就緒ON中
SSC_STSBIT_AX_INP	進入位置
SSC_STSBIT_AX_ZSP	零速度中
SSC_STSBIT_AX_ZPAS	已通過Z相
SSC_STSBIT_AX_TLC	轉矩限制動作中
SSC_STSBIT_AX_SALM	伺服報警中
SSC_STSBIT_AX_SWRN	伺服警告中
SSC_STSBIT_AX_ABSE	絕對位置丟失
位編號(常數)	信號名稱
SSC_STSBIT_AX_OP	運行中
SSC_STSBIT_AX_CPO	粗匹配
SSC_STSBIT_AX_PF	定位完成
SSC_STSBIT_AX_ZP	原點復位完成
SSC_STSBIT_AX_SMZ	平滑停止中
SSC_STSBIT_AX_OALM	運行報警中
SSC_STSBIT_AX_OPF	運行完成
SSC_STSBIT_AX_PSW	位置開關
位編號(常數)	信號名稱
SSC_STSBIT_AX_AUTO	自動運行模式中
SSC_STSBIT_AX_ZRNO	原點復位模式中
SSC_STSBIT_AX_JO	JOG運行模式中
SSC_STSBIT_AX_S0	增量進給模式中
SSC_STSBIT_AX_21	廠商設置用
SSC_STSBIT_AX_LIPO	直線插補模式中[MC200]/插補運行模式中[MC300]
SSC_STSBIT_AX_DSTO	原點重新設置模式中
SSC_STSBIT_AX_24	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_AX_25	廠商設置用
SSC_STSBIT_AX_26	
SSC_STSBIT_AX_27	
SSC_STSBIT_AX_28	
SSC_STSBIT_AX_29	
SSC_STSBIT_AX_30	
SSC_STSBIT_AX_31	
SSC_STSBIT_AX_32	
位編號(常數)	信號名稱
SSC_STSBIT_AX_ISTP	互鎖停止中
SSC_STSBIT_AX_RMRCH	高速監視鎖存中
SSC_STSBIT_AX_POV	超過停止位置
SSC_STSBIT_AX_STO	運行啟動受理完成
SSC_STSBIT_AX_37	廠商設置用
SSC_STSBIT_AX_38	
SSC_STSBIT_AX_ZREQ	原點復位請求
SSC_STSBIT_AX_40	廠商設置用

位編號(常數)	信號名稱
SSC_STSBIT_AX_SCF	速度更改準備完成
SSC_STSBIT_AX_TACF	加速時間常數更改準備完成
SSC_STSBIT_AX_TDCF*1	減速時間常數更改準備完成
SSC_STSBIT_AX_PCF	位置更改準備完成
SSC_STSBIT_AX_SCE	速度更改不正確
SSC_STSBIT_AX_TACE*2	加速時間常數更改不正確
SSC_STSBIT_AX_TDCE	減速時間常數更改不正確
SSC_STSBIT_AX_PCE	位置更改不正確

*1 API版本2.00及以前的情況下，將變為“SSC_STSBIT_AX_TACE”。[MC200]

*2 API版本2.00及以前的情況下，將變為“SSC_STSBIT_AX_TDCF”。[MC200]

位編號(常數)	信號名稱
SSC_STSBIT_AX_49	廠商設置用
SSC_STSBIT_AX_50	
SSC_STSBIT_AX_51	
SSC_STSBIT_AX_52	
SSC_STSBIT_AX_53	
SSC_STSBIT_AX_54	
SSC_STSBIT_AX_55	
SSC_STSBIT_AX_56	

位編號(常數)	信號名稱
SSC_STSBIT_AX_PPIOP	通過位置中斷執行中
SSC_STSBIT_AX_PPIFIN	通過位置中斷完成
SSC_STSBIT_AX_PPIERR	通過位置中斷未完成
SSC_STSBIT_AX_60	廠商設置用
SSC_STSBIT_AX_61	
SSC_STSBIT_AX_62	
SSC_STSBIT_AX_63	
SSC_STSBIT_AX_AUTLO	點位表循環中

位編號(常數)	信號名稱
SSC_STSBIT_AX_GAINO	增益切換中
SSC_STSBIT_AX_FCLSO	全封閉控制切換中
SSC_STSBIT_AX_TL50	轉矩限制選擇中
SSC_STSBIT_AX_SPC	PID控制中
SSC_STSBIT_AX_69	廠商設置用
SSC_STSBIT_AX_70	
SSC_STSBIT_AX_71	
SSC_STSBIT_AX_PRSMO	擋塊中

位編號(常數)	信號名稱
SSC_STSBIT_AX_IWT	干擾檢查待機中
SSC_STSBIT_AX_SINP	伺服驅動器進入位置
SSC_STSBIT_AX_75	廠商設置用
SSC_STSBIT_AX_76	
SSC_STSBIT_AX_77	
SSC_STSBIT_AX_78	
SSC_STSBIT_AX_79	
SSC_STSBIT_AX_80	

位編號(常數)	信號名稱
SSC_STSBIT_AX_81	廠商設置用
SSC_STSBIT_AX_82	
SSC_STSBIT_AX_83	
SSC_STSBIT_AX_84	
SSC_STSBIT_AX_ZSF	原點設置完成
SSC_STSBIT_AX_ZSE	原點設置不正確
SSC_STSBIT_AX_87	廠商設置用
SSC_STSBIT_AX_88	
位編號(常數)	信號名稱
SSC_STSBIT_AX_MKIF1	標記檢測對應資訊1
SSC_STSBIT_AX_MKCF1	標記檢測清除完成1
SSC_STSBIT_AX_MKDO1	標記檢測無效中1
SSC_STSBIT_AX_MKSEF1	標記檢測設置完成1
SSC_STSBIT_AX_MKIF2	標記檢測對應資訊2
SSC_STSBIT_AX_MKCF2	標記檢測清除完成2
SSC_STSBIT_AX_MKDO2	標記檢測無效中2
SSC_STSBIT_AX_MKSEF2	標記檢測設置完成2
位編號(常數)	信號名稱
SSC_STSBIT_AX_97	廠商設置用
SSC_STSBIT_AX_98	
SSC_STSBIT_AX_99	
SSC_STSBIT_AX_100	
SSC_STSBIT_AX_CTLMCF	控制模式切換完成
SSC_STSBIT_AX_CTLMCE	控制模式切換不正確
SSC_STSBIT_AX_103	廠商設置用
SSC_STSBIT_AX_104	
位編號(常數)	信號名稱
SSC_STSBIT_AX_105	廠商設置用
SSC_STSBIT_AX_106	
SSC_STSBIT_AX_107	
SSC_STSBIT_AX_108	
SSC_STSBIT_AX_109	
SSC_STSBIT_AX_110	
SSC_STSBIT_AX_111	
SSC_STSBIT_AX_112	
位編號(常數)	信號名稱
SSC_STSBIT_AX_113	廠商設置用
SSC_STSBIT_AX_114	
SSC_STSBIT_AX_115	
SSC_STSBIT_AX_116	
SSC_STSBIT_AX_117	
SSC_STSBIT_AX_118	
SSC_STSBIT_AX_119	
SSC_STSBIT_AX_120	

位編號(常數)	信號名稱
SSC_STSBIT_AX_GAIN20	增益切換2選擇中
SSC_STSBIT_AX_122	廠商設置用
SSC_STSBIT_AX_123	
SSC_STSBIT_AX_124	
SSC_STSBIT_AX_125	
SSC_STSBIT_AX_126	
SSC_STSBIT_AX_127	
SSC_STSBIT_AX_128	
位編號(常數)	信號名稱
SSC_STSBIT_AX_MOUT	監視輸出中
SSC_STSBIT_AX_MRCH	監視鎖存中
SSC_STSBIT_AX_MER1	監視編號不正確1
SSC_STSBIT_AX_MER2	監視編號不正確2
SSC_STSBIT_AX_MER3	監視編號不正確3
SSC_STSBIT_AX_MER4	監視編號不正確4
SSC_STSBIT_AX_MESV	伺服驅動器未連接
SSC_STSBIT_AX_136	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_AX_137	廠商設置用
SSC_STSBIT_AX_138	
SSC_STSBIT_AX_139	
SSC_STSBIT_AX_140	
SSC_STSBIT_AX_141	
SSC_STSBIT_AX_142	
SSC_STSBIT_AX_143	
SSC_STSBIT_AX_144	
位編號(常數)	信號名稱
SSC_STSBIT_AX_145	廠商設置用
SSC_STSBIT_AX_146	
SSC_STSBIT_AX_147	
SSC_STSBIT_AX_148	
SSC_STSBIT_AX_149	
SSC_STSBIT_AX_150	
SSC_STSBIT_AX_151	
SSC_STSBIT_AX_152	
位編號(常數)	信號名稱
SSC_STSBIT_AX_153	廠商設置用
SSC_STSBIT_AX_154	
SSC_STSBIT_AX_155	
SSC_STSBIT_AX_156	
SSC_STSBIT_AX_157	
SSC_STSBIT_AX_158	
SSC_STSBIT_AX_159	
SSC_STSBIT_AX_160	

位編號(常數)	信號名稱
SSC_STSBIT_AX_PWFIN1	參數寫入完成1
SSC_STSBIT_AX_PWEN1	參數編號不正確1
SSC_STSBIT_AX_PWED1	參數資料超出範圍1
SSC_STSBIT_AX_164	廠商設置用
SSC_STSBIT_AX_PWFIN2	參數寫入完成2
SSC_STSBIT_AX_PWEN2	參數編號不正確2
SSC_STSBIT_AX_PWED2	參數資料超出範圍2
SSC_STSBIT_AX_PSCHG	伺服參數有更改
位編號(常數)	信號名稱
SSC_STSBIT_AX_PRFIN1	參數讀取完成1
SSC_STSBIT_AX_PREN1	參數編號不正確1
SSC_STSBIT_AX_PRFIN2	參數讀取完成2
SSC_STSBIT_AX_PREN2	參數編號不正確2
SSC_STSBIT_AX_173	廠商設置用
SSC_STSBIT_AX_174	
SSC_STSBIT_AX_175	
SSC_STSBIT_AX_176	
位編號(常數)	信號名稱
SSC_STSBIT_AX_177	廠商設置用
SSC_STSBIT_AX_178	
SSC_STSBIT_AX_179	
SSC_STSBIT_AX_180	
SSC_STSBIT_AX_181	
SSC_STSBIT_AX_182	
SSC_STSBIT_AX_183	
SSC_STSBIT_AX_184	
位編號(常數)	信號名稱
SSC_STSBIT_AX_185	廠商設置用
SSC_STSBIT_AX_186	
SSC_STSBIT_AX_187	
SSC_STSBIT_AX_188	
SSC_STSBIT_AX_189	
SSC_STSBIT_AX_190	
SSC_STSBIT_AX_191	
SSC_STSBIT_AX_192	
位編號(常數)	信號名稱
SSC_STSBIT_AX_193	廠商設置用
SSC_STSBIT_AX_194	
SSC_STSBIT_AX_195	
SSC_STSBIT_AX_196	
SSC_STSBIT_AX_197	
SSC_STSBIT_AX_198	
SSC_STSBIT_AX_199	
SSC_STSBIT_AX_200	

位編號(常數)	信號名稱
SSC_STSBIT_AX_201	廠商設置用
SSC_STSBIT_AX_202	
SSC_STSBIT_AX_203	
SSC_STSBIT_AX_204	
SSC_STSBIT_AX_205	
SSC_STSBIT_AX_206	
SSC_STSBIT_AX_207	
SSC_STSBIT_AX_208	
位編號(常數)	信號名稱
SSC_STSBIT_AX_209	廠商設置用
SSC_STSBIT_AX_210	
SSC_STSBIT_AX_211	
SSC_STSBIT_AX_212	
SSC_STSBIT_AX_213	
SSC_STSBIT_AX_214	
SSC_STSBIT_AX_215	
SSC_STSBIT_AX_216	
位編號(常數)	信號名稱
SSC_STSBIT_AX_217	廠商設置用
SSC_STSBIT_AX_218	
SSC_STSBIT_AX_219	
SSC_STSBIT_AX_220	
SSC_STSBIT_AX_221	
SSC_STSBIT_AX_222	
SSC_STSBIT_AX_223	
SSC_STSBIT_AX_224	
位編號(常數)	信號名稱
SSC_STSBIT_AX_225	廠商設置用
SSC_STSBIT_AX_226	
SSC_STSBIT_AX_227	
SSC_STSBIT_AX_228	
SSC_STSBIT_AX_229	
SSC_STSBIT_AX_230	
SSC_STSBIT_AX_231	
SSC_STSBIT_AX_232	
位編號(常數)	信號名稱
SSC_STSBIT_AX_233	廠商設置用
SSC_STSBIT_AX_234	
SSC_STSBIT_AX_235	
SSC_STSBIT_AX_236	
SSC_STSBIT_AX_237	
SSC_STSBIT_AX_238	
SSC_STSBIT_AX_239	
SSC_STSBIT_AX_240	

位編號(常數)	信號名稱
SSC_STSBIT_AX_241	廠商設置用
SSC_STSBIT_AX_242	
SSC_STSBIT_AX_243	
SSC_STSBIT_AX_244	
SSC_STSBIT_AX_245	
SSC_STSBIT_AX_246	
SSC_STSBIT_AX_247	
SSC_STSBIT_AX_248	

位編號(常數)	信號名稱
SSC_STSBIT_AX_249	廠商設置用
SSC_STSBIT_AX_250	
SSC_STSBIT_AX_251	
SSC_STSBIT_AX_252	
SSC_STSBIT_AX_253	
SSC_STSBIT_AX_254	
SSC_STSBIT_AX_255	
SSC_STSBIT_AX_256	

軸狀態位元[MC300]

位編號(常數)	信號名稱
SSC_STSBIT_AX_257	廠商設置用
SSC_STSBIT_AX_258	
SSC_STSBIT_AX_259	
SSC_STSBIT_AX_260	
SSC_STSBIT_AX_261	
SSC_STSBIT_AX_262	
SSC_STSBIT_AX_263	
SSC_STSBIT_AX_264	

位編號(常數)	信號名稱
SSC_STSBIT_AX_265	廠商設置用
SSC_STSBIT_AX_266	
SSC_STSBIT_AX_267	
SSC_STSBIT_AX_268	
SSC_STSBIT_AX_269	
SSC_STSBIT_AX_270	
SSC_STSBIT_AX_271	
SSC_STSBIT_AX_272	

位編號(常數)	信號名稱
SSC_STSBIT_AX_273	廠商設置用
SSC_STSBIT_AX_274	
SSC_STSBIT_AX_275	
SSC_STSBIT_AX_276	
SSC_STSBIT_AX_277	
SSC_STSBIT_AX_278	
SSC_STSBIT_AX_279	
SSC_STSBIT_AX_280	

位編號(常數)	信號名稱
SSC_STSBIT_AX_281	廠商設置用
SSC_STSBIT_AX_282	
SSC_STSBIT_AX_283	
SSC_STSBIT_AX_284	
SSC_STSBIT_AX_285	
SSC_STSBIT_AX_286	
SSC_STSBIT_AX_287	
SSC_STSBIT_AX_288	
位編號(常數)	信號名稱
SSC_STSBIT_AX_289	廠商設置用
SSC_STSBIT_AX_290	
SSC_STSBIT_AX_291	
SSC_STSBIT_AX_292	
SSC_STSBIT_AX_293	
SSC_STSBIT_AX_294	
SSC_STSBIT_AX_295	
SSC_STSBIT_AX_296	
位編號(常數)	信號名稱
SSC_STSBIT_AX_297	廠商設置用
SSC_STSBIT_AX_298	
SSC_STSBIT_AX_299	
SSC_STSBIT_AX_300	
SSC_STSBIT_AX_301	
SSC_STSBIT_AX_302	
SSC_STSBIT_AX_303	
SSC_STSBIT_AX_304	
位編號(常數)	信號名稱
SSC_STSBIT_AX_305	廠商設置用
SSC_STSBIT_AX_306	
SSC_STSBIT_AX_307	
SSC_STSBIT_AX_308	
SSC_STSBIT_AX_309	
SSC_STSBIT_AX_310	
SSC_STSBIT_AX_311	
SSC_STSBIT_AX_312	
位編號(常數)	信號名稱
SSC_STSBIT_AX_313	廠商設置用
SSC_STSBIT_AX_314	
SSC_STSBIT_AX_315	
SSC_STSBIT_AX_316	
SSC_STSBIT_AX_317	
SSC_STSBIT_AX_318	
SSC_STSBIT_AX_319	
SSC_STSBIT_AX_320	

位編號(常數)	信號名稱
SSC_STSBIT_AX_321	廠商設置用
SSC_STSBIT_AX_322	
SSC_STSBIT_AX_323	
SSC_STSBIT_AX_324	
SSC_STSBIT_AX_325	
SSC_STSBIT_AX_326	
SSC_STSBIT_AX_327	
SSC_STSBIT_AX_328	
位編號(常數)	信號名稱
SSC_STSBIT_AX_329	廠商設置用
SSC_STSBIT_AX_330	
SSC_STSBIT_AX_331	
SSC_STSBIT_AX_332	
SSC_STSBIT_AX_333	
SSC_STSBIT_AX_334	
SSC_STSBIT_AX_335	
SSC_STSBIT_AX_336	
位編號(常數)	信號名稱
SSC_STSBIT_AX_337	廠商設置用
SSC_STSBIT_AX_338	
SSC_STSBIT_AX_339	
SSC_STSBIT_AX_340	
SSC_STSBIT_AX_341	
SSC_STSBIT_AX_342	
SSC_STSBIT_AX_343	
SSC_STSBIT_AX_344	
位編號(常數)	信號名稱
SSC_STSBIT_AX_345	廠商設置用
SSC_STSBIT_AX_346	
SSC_STSBIT_AX_347	
SSC_STSBIT_AX_348	
SSC_STSBIT_AX_349	
SSC_STSBIT_AX_350	
SSC_STSBIT_AX_351	
SSC_STSBIT_AX_352	
位編號(常數)	信號名稱
SSC_STSBIT_AX_353	廠商設置用
SSC_STSBIT_AX_354	
SSC_STSBIT_AX_355	
SSC_STSBIT_AX_356	
SSC_STSBIT_AX_357	
SSC_STSBIT_AX_358	
SSC_STSBIT_AX_359	
SSC_STSBIT_AX_360	

位編號(常數)	信號名稱
SSC_STSBIT_AX_361	廠商設置用
SSC_STSBIT_AX_362	
SSC_STSBIT_AX_363	
SSC_STSBIT_AX_364	
SSC_STSBIT_AX_365	
SSC_STSBIT_AX_366	
SSC_STSBIT_AX_367	
SSC_STSBIT_AX_368	
位編號(常數)	信號名稱
SSC_STSBIT_AX_369	廠商設置用
SSC_STSBIT_AX_370	
SSC_STSBIT_AX_371	
SSC_STSBIT_AX_372	
SSC_STSBIT_AX_373	
SSC_STSBIT_AX_374	
SSC_STSBIT_AX_375	
SSC_STSBIT_AX_376	
位編號(常數)	信號名稱
SSC_STSBIT_AX_377	廠商設置用
SSC_STSBIT_AX_378	
SSC_STSBIT_AX_379	
SSC_STSBIT_AX_380	
SSC_STSBIT_AX_381	
SSC_STSBIT_AX_382	
SSC_STSBIT_AX_383	
SSC_STSBIT_AX_384	

6.5 站陳述式位

位編號(常數)	信號名稱
SSC_CMDBIT_UT_1	廠商設置用
SSC_CMDBIT_UT_2	
SSC_CMDBIT_UT_3	
SSC_CMDBIT_UT_4	
SSC_CMDBIT_UT_5	
SSC_CMDBIT_UT_RURST	RIO模組報警復位
SSC_CMDBIT_UT_7	廠商設置用
SSC_CMDBIT_UT_8	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_9	廠商設置用
SSC_CMDBIT_UT_10	
SSC_CMDBIT_UT_11	
SSC_CMDBIT_UT_12	
SSC_CMDBIT_UT_13	
SSC_CMDBIT_UT_RCRST	RIO控制報警復位
SSC_CMDBIT_UT_15	廠商設置用
SSC_CMDBIT_UT_16	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_17	廠商設置用
SSC_CMDBIT_UT_18	
SSC_CMDBIT_UT_19	
SSC_CMDBIT_UT_20	
SSC_CMDBIT_UT_21	
SSC_CMDBIT_UT_22	
SSC_CMDBIT_UT_23	
SSC_CMDBIT_UT_24	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_25	廠商設置用
SSC_CMDBIT_UT_26	
SSC_CMDBIT_UT_27	
SSC_CMDBIT_UT_28	
SSC_CMDBIT_UT_29	
SSC_CMDBIT_UT_30	
SSC_CMDBIT_UT_31	
SSC_CMDBIT_UT_32	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_MON	監視陳述式
SSC_CMDBIT_UT_MONR	監視鎖存陳述式
SSC_CMDBIT_UT_35	廠商設置用
SSC_CMDBIT_UT_36	
SSC_CMDBIT_UT_37	
SSC_CMDBIT_UT_38	
SSC_CMDBIT_UT_39	
SSC_CMDBIT_UT_40	

位編號(常數)	信號名稱
SSC_CMDBIT_UT_41	廠商設置用
SSC_CMDBIT_UT_42	
SSC_CMDBIT_UT_43	
SSC_CMDBIT_UT_44	
SSC_CMDBIT_UT_45	
SSC_CMDBIT_UT_46	
SSC_CMDBIT_UT_47	
SSC_CMDBIT_UT_48	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_PWRT	參數寫入陳述式
SSC_CMDBIT_UT_50	廠商設置用
SSC_CMDBIT_UT_51	
SSC_CMDBIT_UT_52	
SSC_CMDBIT_UT_53	
SSC_CMDBIT_UT_54	
SSC_CMDBIT_UT_55	
SSC_CMDBIT_UT_56	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_PRD	參數讀取陳述式
SSC_CMDBIT_UT_58	廠商設置用
SSC_CMDBIT_UT_59	
SSC_CMDBIT_UT_60	
SSC_CMDBIT_UT_61	
SSC_CMDBIT_UT_62	
SSC_CMDBIT_UT_63	
SSC_CMDBIT_UT_64	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_65	廠商設置用
SSC_CMDBIT_UT_66	
SSC_CMDBIT_UT_67	
SSC_CMDBIT_UT_68	
SSC_CMDBIT_UT_69	
SSC_CMDBIT_UT_70	
SSC_CMDBIT_UT_71	
SSC_CMDBIT_UT_72	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_73	廠商設置用
SSC_CMDBIT_UT_74	
SSC_CMDBIT_UT_75	
SSC_CMDBIT_UT_76	
SSC_CMDBIT_UT_77	
SSC_CMDBIT_UT_78	
SSC_CMDBIT_UT_79	
SSC_CMDBIT_UT_80	

位編號(常數)	信號名稱
SSC_CMDBIT_UT_81	廠商設置用
SSC_CMDBIT_UT_82	
SSC_CMDBIT_UT_83	
SSC_CMDBIT_UT_84	
SSC_CMDBIT_UT_85	
SSC_CMDBIT_UT_86	
SSC_CMDBIT_UT_87	
SSC_CMDBIT_UT_88	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_89	廠商設置用
SSC_CMDBIT_UT_90	
SSC_CMDBIT_UT_91	
SSC_CMDBIT_UT_92	
SSC_CMDBIT_UT_93	
SSC_CMDBIT_UT_94	
SSC_CMDBIT_UT_95	
SSC_CMDBIT_UT_96	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_97	廠商設置用
SSC_CMDBIT_UT_98	
SSC_CMDBIT_UT_99	
SSC_CMDBIT_UT_100	
SSC_CMDBIT_UT_101	
SSC_CMDBIT_UT_102	
SSC_CMDBIT_UT_103	
SSC_CMDBIT_UT_104	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_105	廠商設置用
SSC_CMDBIT_UT_106	
SSC_CMDBIT_UT_107	
SSC_CMDBIT_UT_108	
SSC_CMDBIT_UT_109	
SSC_CMDBIT_UT_110	
SSC_CMDBIT_UT_111	
SSC_CMDBIT_UT_112	
位編號(常數)	信號名稱
SSC_CMDBIT_UT_113	廠商設置用
SSC_CMDBIT_UT_114	
SSC_CMDBIT_UT_115	
SSC_CMDBIT_UT_116	
SSC_CMDBIT_UT_117	
SSC_CMDBIT_UT_118	
SSC_CMDBIT_UT_119	
SSC_CMDBIT_UT_120	

位編號(常數)	信號名稱
SSC_CMDBIT_UT_121	廠商設置用
SSC_CMDBIT_UT_122	
SSC_CMDBIT_UT_123	
SSC_CMDBIT_UT_124	
SSC_CMDBIT_UT_125	
SSC_CMDBIT_UT_126	
SSC_CMDBIT_UT_127	
SSC_CMDBIT_UT_128	

6.6 站狀態位元

位編號(常數)	信號名稱
SSC_STSBIT_UT_RURDY	控制器就緒ON受理中
SSC_STSBIT_UT_RUA	DO輸出中
SSC_STSBIT_UT_3	廠商設置用
SSC_STSBIT_UT_4	
SSC_STSBIT_UT_5	
SSC_STSBIT_UT_RUALM	RIO模組報警中
SSC_STSBIT_UT_RUWRN	RIO模組警告中
SSC_STSBIT_UT_8	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_UT_9	廠商設置用
SSC_STSBIT_UT_10	
SSC_STSBIT_UT_11	
SSC_STSBIT_UT_12	
SSC_STSBIT_UT_13	
SSC_STSBIT_UT_RCALM	RIO控制報警中
SSC_STSBIT_UT_15	廠商設置用
SSC_STSBIT_UT_16	
位編號(常數)	信號名稱
SSC_STSBIT_UT_17	廠商設置用
SSC_STSBIT_UT_18	
SSC_STSBIT_UT_19	
SSC_STSBIT_UT_20	
SSC_STSBIT_UT_21	
SSC_STSBIT_UT_22	
SSC_STSBIT_UT_23	
SSC_STSBIT_UT_24	
位編號(常數)	信號名稱
SSC_STSBIT_UT_25	廠商設置用
SSC_STSBIT_UT_26	
SSC_STSBIT_UT_27	
SSC_STSBIT_UT_28	
SSC_STSBIT_UT_29	
SSC_STSBIT_UT_30	
SSC_STSBIT_UT_31	
SSC_STSBIT_UT_32	
位編號(常數)	信號名稱
SSC_STSBIT_UT_MOUT	監視輸出中
SSC_STSBIT_UT_MRCH	監視鎖存中
SSC_STSBIT_UT_MER1	監視編號不正確1
SSC_STSBIT_UT_MER2	監視編號不正確2
SSC_STSBIT_UT_MER3	監視編號不正確3
SSC_STSBIT_UT_MER4	監視編號不正確4
SSC_STSBIT_UT_MER10	RIO模組未連接
SSC_STSBIT_UT_40	廠商設置用

位編號(常數)	信號名稱
SSC_STSBIT_UT_41	廠商設置用
SSC_STSBIT_UT_42	
SSC_STSBIT_UT_43	
SSC_STSBIT_UT_44	
SSC_STSBIT_UT_45	
SSC_STSBIT_UT_46	
SSC_STSBIT_UT_47	
SSC_STSBIT_UT_48	
位編號(常數)	信號名稱
SSC_STSBIT_UT_PWFIN1	參數寫入完成1
SSC_STSBIT_UT_PWEN1	參數編號不正確1
SSC_STSBIT_UT_PWED1	參數資料超出範圍1
SSC_STSBIT_UT_52	廠商設置用
SSC_STSBIT_UT_PWFIN2	參數寫入完成2
SSC_STSBIT_UT_PWEN2	參數編號不正確2
SSC_STSBIT_UT_PWED2	參數資料超出範圍2
SSC_STSBIT_UT_56	廠商設置用
位編號(常數)	信號名稱
SSC_STSBIT_UT_PRFIN1	參數讀取完成1
SSC_STSBIT_UT_PREN1	參數編號不正確1
SSC_STSBIT_UT_PRFIN2	參數讀取完成2
SSC_STSBIT_UT_PREN2	參數編號不正確2
SSC_STSBIT_UT_61	廠商設置用
SSC_STSBIT_UT_62	
SSC_STSBIT_UT_63	
SSC_STSBIT_UT_64	
位編號(常數)	信號名稱
SSC_STSBIT_UT_65	廠商設置用
SSC_STSBIT_UT_66	
SSC_STSBIT_UT_67	
SSC_STSBIT_UT_68	
SSC_STSBIT_UT_69	
SSC_STSBIT_UT_70	
SSC_STSBIT_UT_71	
SSC_STSBIT_UT_72	
位編號(常數)	信號名稱
SSC_STSBIT_UT_73	廠商設置用
SSC_STSBIT_UT_74	
SSC_STSBIT_UT_75	
SSC_STSBIT_UT_76	
SSC_STSBIT_UT_77	
SSC_STSBIT_UT_78	
SSC_STSBIT_UT_79	
SSC_STSBIT_UT_80	

位編號(常數)	信號名稱
SSC_STSBIT_UT_81	廠商設置用
SSC_STSBIT_UT_82	
SSC_STSBIT_UT_83	
SSC_STSBIT_UT_84	
SSC_STSBIT_UT_85	
SSC_STSBIT_UT_86	
SSC_STSBIT_UT_87	
SSC_STSBIT_UT_88	
位編號(常數)	信號名稱
SSC_STSBIT_UT_89	廠商設置用
SSC_STSBIT_UT_90	
SSC_STSBIT_UT_91	
SSC_STSBIT_UT_92	
SSC_STSBIT_UT_93	
SSC_STSBIT_UT_94	
SSC_STSBIT_UT_95	
SSC_STSBIT_UT_96	
位編號(常數)	信號名稱
SSC_STSBIT_UT_97	廠商設置用
SSC_STSBIT_UT_98	
SSC_STSBIT_UT_99	
SSC_STSBIT_UT_100	
SSC_STSBIT_UT_101	
SSC_STSBIT_UT_102	
SSC_STSBIT_UT_103	
SSC_STSBIT_UT_104	
位編號(常數)	信號名稱
SSC_STSBIT_UT_105	廠商設置用
SSC_STSBIT_UT_106	
SSC_STSBIT_UT_107	
SSC_STSBIT_UT_108	
SSC_STSBIT_UT_109	
SSC_STSBIT_UT_110	
SSC_STSBIT_UT_111	
SSC_STSBIT_UT_112	
位編號(常數)	信號名稱
SSC_STSBIT_UT_113	廠商設置用
SSC_STSBIT_UT_114	
SSC_STSBIT_UT_115	
SSC_STSBIT_UT_116	
SSC_STSBIT_UT_117	
SSC_STSBIT_UT_118	
SSC_STSBIT_UT_119	
SSC_STSBIT_UT_120	

位編號(常數)	信號名稱
SSC_STSBIT_UT_121	廠商設置用
SSC_STSBIT_UT_122	
SSC_STSBIT_UT_123	
SSC_STSBIT_UT_124	
SSC_STSBIT_UT_125	
SSC_STSBIT_UT_126	
SSC_STSBIT_UT_127	
SSC_STSBIT_UT_128	

7 中斷事件原因一覽

中斷事件原因一覽如下所示。

系統中斷

事件原因	內容
SSC_INT_SYS_SYSE	系統出錯中
SSC_INT_SYS_CALM	系統報警中
SSC_INT_SYS_EMIO	緊急停止中
SSC_INT_SYS_03	廠商設置用
SSC_INT_SYS_04	
SSC_INT_SYS_05	
SSC_INT_SYS_06	
SSC_INT_SYS_OCME	
SSC_INT_SYS_OASF	運算週期報警中
SSC_INT_SYS_PPI	其他軸啟動中斷原因輸出中
SSC_INT_SYS_10	通過位置中斷原因輸出中
SSC_INT_SYS_11	廠商設置用
SSC_INT_SYS_12	
SSC_INT_SYS_13	
SSC_INT_SYS_14	
SSC_INT_SYS_15	

軸中斷

事件原因	內容
SSC_INT_AX_RDY	伺服就緒ON中
SSC_INT_AX_INP	進入位置
SSC_INT_AX_ZSP	零速度中
SSC_INT_AX_ZPAS	已通過Z相
SSC_INT_AX_TLC	轉矩限制動作中
SSC_INT_AX_SALM	伺服報警中
SSC_INT_AX_SWRN	伺服警告中
SSC_INT_AX_ABSE	絕對位置丟失
SSC_INT_AX_OP	運行中
SSC_INT_AX_CPO	粗匹配
SSC_INT_AX_PF	定位完成
SSC_INT_AX_ZP	原點復位完成
SSC_INT_AX_SMZ	平滑停止中
SSC_INT_AX_OALM	運行報警中
SSC_INT_AX_OPF	運行完成
SSC_INT_AX_PSW	位置開關
SSC_INT_AX_GAINO	增益切換中
SSC_INT_AX_FCLSO	全封閉控制切換中
SSC_INT_AX_TLSO	轉矩限制選擇中
SSC_INT_AX_SPC	PID控制中
SSC_INT_AX_20	廠商設置用
SSC_INT_AX_MAK1	標記檢測中1
SSC_INT_AX_MAK2	標記檢測中2
SSC_INT_AX_PRSMO	擋塊中
SSC_INT_AX_IWT	干擾檢查待機中
SSC_INT_AX_SINP	伺服驅動器進入位置
SSC_INT_AX_26	廠商設置用
SSC_INT_AX_27	
SSC_INT_AX_28	
SSC_INT_AX_29	
SSC_INT_AX_30	
SSC_INT_AX_31	

站中斷

事件原因	內容
SSC_INT_UT_00	廠商設置用
SSC_INT_UT_01	
SSC_INT_UT_02	
SSC_INT_UT_03	
SSC_INT_UT_04	
SSC_INT_UT_RUALM	RIO模組報警中
SSC_INT_UT_RUWRN	RIO模組警告中
SSC_INT_UT_07	廠商設置用
SSC_INT_UT_08	
SSC_INT_UT_09	
SSC_INT_UT_10	
SSC_INT_UT_11	
SSC_INT_UT_12	
SSC_INT_UT_RCALM	RIO控制報警中
SSC_INT_UT_14	廠商設置用
SSC_INT_UT_15	

8 詳細出錯代碼一覽

詳細出錯代碼一覽如下所示。

常見出錯

值	常數定義	發生原因/處理
FFFFFFFFH	SSC_FUNC_ERR_UNKNOWN	使用API函數後，出錯(-1)一次也未發生。
FFFFFFFCH	SSC_FUNC_ERR_UNSUPPORT_BOARD_VERSION	定位板不支援此功能。
0000001H	SSC_FUNC_ERR_ARGUMENT_0□ □=1~9: 參數位置	參數超出設置範圍。
0000002H		
0000003H		
0000004H		
0000005H		
0000006H		
0000007H		
0000008H		
0000009H		
0000100H		
0001000H	SSC_FUNC_ERR_TIMEOUT_0□ □=1~5: 超時位置	發生了超時。 有超時參數的API函數的情況下，應重新修改設置超時值。
0001010H		
0001020H		
0001030H		
0001040H		
0001050H		
0001060H		
0001070H		
0001080H		
0001090H		
00010A0H		
00010B0H		
00010C0H		

軟元件函數群出錯

值	常數定義	發生原因/處理
00020000H	SSC_FUNC_ERR_REOPEN	已調用sscOpen函數。
00020010H	SSC_FUNC_ERR_UNOPEN	尚未調用sscOpen函數。
00021010H	SSC_FUNC_ERR_NOT_FOUND_BOARD	<ul style="list-style-type: none"> 找不到指定板ID的定位板。應確認定位板的板ID選擇(撥動開關)。 從定位板測試工具連接到定位板的情況下，應確認“確認硬體中斷”的勾選處於取消狀態。[MC300] 應確認定位板實用程式2的版本，並使用隨相同的定位板實用程式2附帶的API庫。
00021011H	SSC_FUNC_ERR_GET_CHANNEL_NUM	無法獲取安裝通道資訊。 有時OS不能正常識別定位板。 應通過軟元件管理器確認定位板已正確安裝。
00021012H	SSC_FUNC_ERR_UNSUPPORTED_DEVICE_DRIVER	非軟元件驅動器的對應版本。 應組合實用軟體中收錄的軟元件驅動器及API庫使用。
00022000H	SSC_FUNC_ERR_CREATE_SEMAPHORE	在CreateSemaphore函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022001H	SSC_FUNC_ERR_DELETE_SEMAPHORE	在CloseHandle函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022002H	SSC_FUNC_ERR_WAIT_SEMAPHORE	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022003H	SSC_FUNC_ERR_RELEASE_SEMAPHORE	在ReleaseSemaphore函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022010H	SSC_FUNC_ERR_CREATE_EVENT	在CreateEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022011H	SSC_FUNC_ERR_DELETE_EVENT	在CloseHandle函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022012H	SSC_FUNC_ERR_RESET_EVENT	在ResetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022013H	SSC_FUNC_ERR_SET_EVENT	在SetEvent函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022014H	SSC_FUNC_ERR_WAIT_EVENT	在WaitForSingleObject函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022015H	SSC_FUNC_ERR_WAIT_EVENT_MULTI	在WaitForMultipleObjects函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022020H	SSC_FUNC_ERR_CREATE_THREAD	在CreateThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022021H	SSC_FUNC_ERR_DELETE_THREAD	在CloseHandle函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022022H	SSC_FUNC_ERR_THREAD_PRIORITY	在SetThreadPriority函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022023H	SSC_FUNC_ERR_RESUME_THREAD	在ResumeThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
00022024H	SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	在GetExitCodeThread函數(Windows API)中發生了出錯。 應調用Windows API的GetLastError函數，並確認詳細出錯內容。
000230□□H	SSC_FUNC_ERR_DEVICE_DRIVER	<ul style="list-style-type: none"> 軟元件驅動器的調用時發生了出錯。 應確認已經安裝了軟元件驅動器。 應確認定位板實用程式2的版本，並使用隨相同的定位板實用程式2附帶的API庫。

系統函數群出錯

值	常數定義	發生原因/處理
00030000H	SSC_FUNC_ERR_UNREADY_CHANNEL	系統準備完成以外。 應使用sscReboot函數重啟。
00030020H	SSC_FUNC_ERR_RUNNING_CHANNEL	系統啟動前。 應使用sscSystemStart函數啟動系統。
00030030H	SSC_FUNC_ERR_NOW_ALARM_SYSTEM	發生了系統出錯 (E□□□H)。 應使用sscGetSystemStatusCode函數獲取系統狀態碼，消除發生原因。
00030040H	SSC_FUNC_ERR_FLASHROM_PARAM_LOAD	發生了快閃記憶體參數讀取不正確。
00030041H	SSC_FUNC_ERR_STS_BIT_FRNG	快閃記憶體傳送準備不正確信號 (FRNG) 變為了ON。
00030042H	SSC_FUNC_ERR_STS_BIT_FSNG	快閃記憶體傳送不正確信號 (FSNG) 變為了ON。
00030050H	SSC_FUNC_ERR_STS_BIT_RCE	重新連接不正確信號 (RCE) 變為了ON。
00030051H	SSC_FUNC_ERR_STS_BIT_CCE	斷開不正確信號 (CCE) 變為了ON。
00030060H	SSC_FUNC_ERR_ALREADY_ENABLE_WDT	用戶看門狗功能已啟用。
00030061H	SSC_FUNC_ERR_ALREADY_DISABLE_WDT	用戶看門狗功能已禁用。
00030062H	SSC_FUNC_ERR_STS_BIT_IFMO	介面模式中。 在介面模式中更改控制模式的情況下，應使用sscIfmSetControlMode函數。

參數函數群出錯

值	常數定義	發生原因/處理
00040000H	SSC_FUNC_ERR_STS_BIT_PREN□ □=1~2: 參數讀取編號 (2點) 的陣列編號	發生了參數讀取出錯。 <ul style="list-style-type: none"> 在參數讀取編號中設置了超出範圍的值。 軸編號與參數讀取編號不匹配。 (例: 在軸編號中設置了“系統參數”，在參數讀取編號中設置了“軸參數”等情況下)
00040001H		
00040002H	SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM□ □=1~2: 參數讀取編號 (2點) 的陣列編號	參數讀取編號的陳述式與狀態不一致。
00040003H		
00040004H	SSC_FUNC_ERR_PARAM_READ_EX_NUM□ □=1~2: 參數讀取編號 (2點) 的陣列編號	無法讀取MR-J5(W_)_B的伺服參數。
00040005H		
00040020H	SSC_FUNC_ERR_STS_BIT_PWEN□ □=1~2: 參數寫入編號 (2點) 的陣列編號	發生了參數寫入編號出錯。 <ul style="list-style-type: none"> 在參數寫入編號中設置了超出範圍的值。 軸編號與參數寫入編號不匹配。 (例: 在軸編號中設置了“系統參數”，在參數寫入編號中設置了“軸參數”等情況下)
00040021H		
00040022H	SSC_FUNC_ERR_STS_BIT_PWED□ □=1~2: 參數寫入資料 (2點) 的陣列編號	在參數寫入資料中設置了超出範圍的值。
00040023H		
00040024H	SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM□ □=1~2: 參數寫入編號 (2點) 的陣列編號	參數寫入編號的陳述式與狀態不一致。
00040025H		
00040026H	SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □=1~2: 參數寫入資料 (2點) 的陣列編號	參數寫入資料的陳述式與狀態不一致。
00040027H		
00040028H	SSC_FUNC_ERR_PARAM_WRITE_EX_NUM□ □=1~2: 參數寫入資料 (2點) 的陣列編號	無法寫入MR-J5(W_)_B的伺服參數。
00040029H		
0004002AH	SSC_FUNC_ERR_UNUPPORT_PARAM_EX	定位板不支持MR-J5(W_)_B的伺服參數。

監視函數群出錯

值	常數定義	發生原因/處理
00050000H	SSC_FUNC_ERR_STS_BIT_MER□	發生了監視編號出錯。 <ul style="list-style-type: none"> 在監視編號中設置了超出範圍的值。 軸編號與監視編號不匹配。 (例: 在軸編號中設置了“系統監視”, 在監視編號中設置了“軸監視”等情況下)
00050001H	□=1~4: 監視編號(4點)的陣列編號	
00050002H		
00050003H		
00050004H	SSC_FUNC_ERR_STS_BIT_MESV	伺服驅動器未連接時將伺服資訊設置為監視編號。
00050010H	SSC_FUNC_ERR_ALREADY_MONITOR_STOP	監視已停止中。
00050011H	SSC_FUNC_ERR_NOT_START_MONITOR	監視尚未開始。

軸函數群出錯

值	常數定義	發生原因/處理
00060010H	SSC_FUNC_ERR_NOW_DRIVING	運行中的狀態。
00060011H	SSC_FUNC_ERR_NOW_DRIVING_READY	運行啟動準備中(請求運行啟動後到定位板接收之間)。
00060020H	SSC_FUNC_ERR_NOW_ALARM_SERVO	伺服報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。
00060030H	SSC_FUNC_ERR_NOW_ALARM_DRIVE	運行報警發生中。 應使用sscGetAlarm函數獲取報警編號, 消除發生原因。
00060040H	SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	指定的運行模式以外。 (例: 在自動運行模式中調用了sscChangeManualPosition)
00060041H	SSC_FUNC_ERR_CHG_POS_DIR	位置更改前與位置更改後的移動方向不同。
00060060H	SSC_FUNC_ERR_STS_BIT_PCE	位置更改不正確信號(PCE)變為了ON。
00060070H	SSC_FUNC_ERR_STS_BIT_SCE	速度更改不正確信號(SCE)變為了ON。
00060080H	SSC_FUNC_ERR_STS_BIT_TACE	加速時間常數更改不正確信號(TACE)變為了ON。
00060090H	SSC_FUNC_ERR_STS_BIT_TDCE	減速時間常數更改不正確信號(TDCE)變為了ON。
000600A0H	SSC_FUNC_ERR_POINT_NUMBER_OVER	指定的“點編號+點編號偏置值”超過了點位元表的區域。
000600A1H	SSC_FUNC_ERR_STS_BIT_CTLMCE	控制模式切換不正確信號(CTLMCE)變為了ON。
000600A2H	SSC_FUNC_ERR_STS_BIT_IPCH_ON	插補組可變設置有效。 應在插補組編號中設置“0”。
000600A3H	SSC_FUNC_ERR_STS_BIT_IPCH_OFF	插補組可變設置無效。 應在插補組編號中設置“1~8 [MC200]/1~16 [MC300]”。
000600A4H	SSC_FUNC_ERR_SUB_AXIS_NUM	插補組可變有效時, 點位表中已設置的插補軸編號為不正確的值。 <ul style="list-style-type: none"> 插補軸編號超出設置範圍 插補軸編號與基準軸編號重覆 插補軸編號相互重覆
000600A5H	SSC_FUNC_ERR_NOT_LIP_DRIVING	直線插補中不為運行中。

採樣函數群出錯

值	常數定義	發生原因/處理
00061010H	SSC_FUNC_ERR_ALREADY_START_SAMPLING	採樣開始信號(SMPS)處於ON中。 應使用sscStopSampling函數停止採樣。
00061011H	SSC_FUNC_ERR_ALREADY_STOP_SAMPLING	採樣已經在停止中。
00061020H	SSC_FUNC_ERR_STS_BIT_SREN	在採樣設置讀取編號中設置了超出範圍的值。
00061021H	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	採樣設置讀取編號的陳述式與狀態不一致。
00061022H	SSC_FUNC_ERR_STS_BIT_SWEN	在採樣設置寫入編號中設置了超出範圍的值。
00061023H	SSC_FUNC_ERR_STS_BIT_SWED	在採樣設置寫入資料中設置了超出範圍的值。
00061024H	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_NUM	採樣設置寫入編號的陳述式與狀態不一致。
00061025H	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_DATA	採樣設置寫入資料的陳述式與狀態不一致。

日誌函數群出錯

值	常數定義	發生原因/處理
00062000H	SSC_FUNC_ERR_NOW_LOGGING	日誌動作中信號 (LOGO) 處於ON中。 應使用sscStopLog函數停止日誌。
00062001H	SSC_FUNC_ERR_STS_BIT_LOGRE	日誌資料讀取不正確信號 (LOGRE) 變為了ON。
00062002H	SSC_FUNC_ERR_STS_BIT_LOGIE	日誌資料初始化不正確信號 (LOGIE) 變為了ON。
00062003H	SSC_FUNC_ERR_ALREADY_START_LOG	日誌陳述式信號 (LOGC) 處於ON中。 應使用sscStopLog函數停止日誌。
00062004H	SSC_FUNC_ERR_ALREADY_STOP_LOG	日誌已經在停止中。
00062005H	SSC_FUNC_ERR_UNUPPORT_ALH	報警履歷功能未支援。
00062006H	SSC_FUNC_ERR_STS_BIT_ALHRE	報警履歷讀取不正確信號 (ALHRE) 變為了ON。
00062007H	SSC_FUNC_ERR_STS_BIT_ALHIE	報警履歷初始化不正確信號 (ALHIE) 變為了ON。

中斷函數群出錯

值	常數定義	發生原因/處理
10000100H	SSC_FUNC_ERR_ALREADY_START_INT_DRIVER	中斷驅動器已經在啟動中。
10000101H	SSC_FUNC_ERR_ALREADY_END_INT_DRIVER	中斷驅動器已經在停止中。
10000102H	SSC_FUNC_ERR_ALREADY_OTHER_PROCESS_INT	中斷驅動器已經在其他進程中啟動。
10000103H	SSC_FUNC_ERR_INT_DISABLE_MASK [MC200]	中斷輸出遮罩選擇 (撥動開關) 有效。
10000104H	SSC_FUNC_ERR_CLEAR_INT [MC200]	中斷信號清除暫存器寫入失敗。
10000110H	SSC_FUNC_ERR_NOT_START_INT_DRIVER	中斷驅動器停止中。 應調用sscIntStart函數。
10000111H	SSC_FUNC_ERR_TERMINATE_INT_DRIVER	在針對指定的事件原因的中斷確認待機中，調用了sscIntEnd函數。
10000112H	SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	在針對指定的事件原因的中斷確認待機中，在中斷事件通知執行緒中發生了出錯。
10000113H	SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	在針對指定的事件原因的中斷待機中，從用戶程式調用了恢復待機狀態的函數。
10000200H	SSC_FUNC_ERR_ALREADY_REREGISTER_CALLBACK	中斷回呼函數已登記完成。 更改中斷回呼函數的情況下，應調用sscUnregisterIntCallback函數。
10000201H	SSC_FUNC_ERR_ALREADY_UNREREGISTER_CALLBACK	中斷回呼函數已解除完成。

標記檢測函數群出錯

值	常數定義	發生原因/處理
00067000H	SSC_FUNC_ERR_MARK_DETECT_UNUSABLE	標記檢測功能無效。 • 不支援標記檢測功能。 • 標記檢測功能因設置無效。
00067001H	SSC_FUNC_ERR_MARK_DETECT_UNDETECTED	可獲取的標記檢測資料不存在。 應確認更新了定位板的標記檢測次數後，調用sscGetMarkDetectionData函數。

介面模式函數群出錯

值	常數定義	發生原因/處理
000D0000H	SSC_FUNC_ERR_IFM_INP_OFF	進入位置信號 (INP) 處於OFF中。 在進入位置信號 (INP) 確認模式中設置“不待機直至進入位置信號 (INP) 變為ON”時，應在進入位置信號 (INP) 處於ON中調用sscIfmSetHomePosition函數。
000D0001H	SSC_FUNC_ERR_STS_BIT_ZSE	原點設置不正確信號 (ZSE) 變為了ON。
000D0002H	SSC_FUNC_ERR_IFM_CMD_BUF_FULL	位置陳述式緩衝無空餘。 應在確認定位板的傳送中緩衝編號已更新後，調用sscIfmRenewLatestBuffer函數或sscIfmRenewLatestBufferEx函數。

輸入輸出軟元件函數群出錯

值	常數定義	發生原因/處理
000E0000H	SSC_FUNC_ERR_DVI_TABLE_RANGE_OVER	通過參數指定的“word_num” + “word_cnt” 超過了輸入軟元件表大小的區域。
000E0001H	SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	通過參數指定的“word_num” + “word_cnt” 超過了輸出軟元件表大小的區域。

暫態傳送函數群出錯

值	常數定義	發生原因/處理
000E1001H	SSC_FUNC_ERR_TRANSIENT_INVALID_DATA	暫態資料無效。

附錄

附1 MR-MC2_ _與MR-MC3_ _的不同點

API庫的不同點

MR-MC2_ _用API庫與MR-MC3_ _用API庫的不同點如下所示。

項目	MR-MC2_ _	MR-MC3_ _
最大通道編號	1	1
最大軸編號	32	64
最大站編號	4	16
最大點編號	320	2048
最大點編號偏置值	319	2047
最大其他軸啟動表編號	32	64
最大通過位置條件編號	64	128
最大插補組編號	8	16
輸入輸出位軟元件編號	0000h~0FFFh	0000h~23FFh
輸入輸出字軟元件編號	0000h~0FFFh	0000h~23FFh

MR-MC3_ _用API庫中添加的API函數

MR-MC3_ _用API庫中新添加的API函數如下所示。

分類	函數名稱	內容	參照
資訊函數群	sscGetBoardSerialNumber	獲取定位板的製造編號。	38頁 sscGetBoardSerialNumber [MC300]
系統函數群	sscReconnectSSCNETEx	重新連接SSCNET通信。	65頁 sscReconnectSSCNETEx [MC300]
	sscDisconnectSSCNETEx	斷開SSCNET通信。	67頁 sscDisconnectSSCNETEx [MC300]
	sscGetControllingAxis	獲取控制中軸資訊及控制中站資訊。	68頁 sscGetControllingAxis [MC300]
運行函數群	sscInterpolationStart	啟動插補運行。	91頁 sscInterpolationStart [MC300]
介面模式函數群	sscIfmGetEventStatusBitsEx	使用事件檢測功能，獲取對指定狀態信號的所有軸的狀態位元資訊。	180頁 sscIfmGetEventStatusBitsEx [MC300]

從MR-MC2_ _到MR-MC3_ _的API函數替換

- 下表的MR-MC2_ _用API函數雖然可以在MR-MC3_ _中使用，但使用MR-MC3_ _的情況下，推薦使用MR-MC3_ _用API函數。

分類	函數名稱	
	MR-MC2_ _用API函數	MR-MC3_ _用API函數
系統函數群	sscReconnectSSCNET	sscReconnectSSCNETEx
	sscDisconnectSSCNET	sscDisconnectSSCNETEx
運行函數群	sscLinearStart	sscInterpolationStart

- 下表的MR-MC2_ _用API函數不能在MR-MC3_ _中使用。應替換為MR-MC3_ _用API函數。

分類	函數名稱	
	MR-MC2_ _用API函數	MR-MC3_ _用API函數
介面模式函數群	sscIfmGetMaximumBufferNumber	sscIfmGetMaximumBufferNumberEx
	sscIfmRenewLatestBuffer	sscIfmRenewLatestBufferEx
	sscIfmCheckLatestBuffer	sscIfmCheckLatestBufferEx
	sscIfmGetTransmitBuffer	sscIfmGetTransmitBufferEx
	sscIfmGetEventStatusBits	sscIfmGetEventStatusBitsEx

- 下表的API函數雖然可以照常使用，但在MR-MC3_ _用中由於結構已被擴展，因此請務必在使用前通過“0”進行初始化。

分類	函數名稱	結構
點位元表函數群	sscSetPointDataEx	PNT_DATA_EX結構
	sscCheckPointDataEx	
其他軸啟動函數群	sscSetOtherAxisStartData	OAS_DATA結構
	sscGetOtherAxisStartData	
採樣函數群	sscGetSamplingError	SMP_ERR結構
中斷函數群	sscRegisterIntCallback	INT_CB_DATA結構

索引

A	
ALH_DATA結構	229
I	
INT_CB_DATA結構	233
L	
LOG_DATA結構	228
O	
OAS_DATA結構	219
P	
PNT_DATA_EX結構	215
PRESS_DATA結構	223
S	
SLAVE_INFO結構	237
SMP_DATA結構	226
SMP_ERR結構	224
sscAutoStart	87
sscChange2Parameter	43
sscChange2ParameterEx	54
sscChangeAutoAccTime.	111
sscChangeAutoDecTime.	113
sscChangeAutoPosition	105
sscChangeAutoSpeed	109
sscChangeControlMode.	103
sscChangeLinearPosition	106
sscChangeManualAccTime.	110
sscChangeManualDecTime.	112
sscChangeManualPosition	104
sscChangeManualSpeed.	108
sscChangeParameter	41
sscChangeParameterEx.	52
sscChangeWdCounter	130
sscCheck2Parameter	47
sscCheck2ParameterEx.	58
sscCheckAlarmHistoryEventNum	154
sscCheckIntPassPositionData	137
sscCheckLogEventNum	150
sscCheckLogStatus	149
sscCheckParameter	45
sscCheckParameterEx	56
sscCheckPointDataEx	74
sscCheckPointOffset	76
sscCheckSvPrmChangeNumEx.	51
sscClearAlarmHistoryData	155
sscClearLogData	152
sscClearMarkDetectionData	165
sscClearOperationCycleMonitor	37
sscClose	32
sscDataSetStart.	93
sscDisconnectSSCNET	66
sscDisconnectSSCNETEx	67
sscDriveRapidStop.	96
sscDriveRapidStopNoWait	97
sscDriveStop	94
sscDriveStopNoWait	95
sscGetAlarm	114
sscGetAlarmHistoryData	153
sscGetBoardSerialNumber	38
sscGetBoardVersion	34
sscGetCmdSpeedFast	124
sscGetControlCycle	33
sscGetControllingAxis	68
sscGetCurrentCmdPositionFast	121
sscGetCurrentFbFast	126
sscGetCurrentFbPositionFast	122
sscGetDigitalInputDataBit	156
sscGetDigitalInputDataWord.	157
sscGetDigitalOutputDataBit.	160
sscGetDigitalOutputDataWord	161
sscGetDriveFinStatus	101
sscGetDriveMode.	100
sscGetDriverVersion.	35
sscGetDrivingPointNumber.	77
sscGetExecutingPassNumber	139
sscGetFbSpeedFast.	125
sscGetInputDeviceBit	208
sscGetInputDeviceWord	209
sscGetIoStatusFast	123
sscGetLastError.	28
sscGetMarkDetectionCounter.	164
sscGetMarkDetectionData	162
sscGetMonitor	120
sscGetMountChannel	29
sscGetOperationCycleMonitor	36
sscGetOtherAxisStartData.	132
sscGetOtherAxisStartStatus.	135
sscGetOutputDeviceBit	212
sscGetOutputDeviceWord	213
sscGetPositionDroopFast	127
sscGetPressData.	80
sscGetSamplingData	146
sscGetSamplingError	144
sscGetSamplingParameter	143
sscGetSamplingStatus	145
sscGetStatusBitSignalEx	71
sscGetSystemStatusCode	63
sscHomeReturnStart	88
sscIfmCheckLatestBuffer	172

sscIfmCheckLatestBufferEx	173
sscIfmGetControlMode.	178
sscIfmGetEventStatusBits	179
sscIfmGetEventStatusBitsEx	180
sscIfmGetMaximumBufferNumber	168
sscIfmGetMaximumBufferNumberEx	169
sscIfmGetReadErrorCount	166
sscIfmGetTransmitBuffer	174
sscIfmGetTransmitBufferEx	175
sscIfmRenewLatestBuffer	170
sscIfmRenewLatestBufferEx	171
sscIfmSetControlMode.	177
sscIfmSetHomePosition	167
sscIfmIrqSetSpeedLimit.	176
sscIncStart.	85
sscIntDisable.	186
sscIntEnable	185
sscIntEnd	184
sscInterpolationStart	91
sscIntStart.	183
sscJogStart.	81
sscJogStop	83
sscJogStopNoWait	84
sscLinearStart	89
sscLoadAllParameterFromFlashROM.	49
sscOpen	31
sscOtherAxisStartAbortOff	134
sscOtherAxisStartAbortOn	133
sscReadLogData	151
sscReboot	61
sscReconnectSSCNET	64
sscReconnectSSCNETEx.	65
sscRegisterIntCallback.	187
sscResetAlarm.	115
sscResetAllParameter.	40
sscResetIntDriveFin	204
sscResetIntEvent	189
sscResetIntEventMulti	193
sscResetIntOasEvent	197
sscResetIntPassPosition	201
sscSaveAllParameterToFlashROM.	50
sscSendReceiveTransientData	214
sscSetCommandBitSignalEx.	70
sscSetDigitalOutputDataBit	158
sscSetDigitalOutputDataWord	159
sscSetDriveMode	98
sscSetIntDriveFin	205
sscSetIntEvent	190
sscSetIntEventMulti	194
sscSetIntOasEvent	198
sscSetIntPassPosition	202
sscSetIntPassPositionData	136
sscSetLatestPointNumber	78
sscSetMonitor.	118
sscSetOtherAxisStartData	131
sscSetOutputDeviceBit	210

sscSetOutputDeviceWord	211
sscSetPointDataEx.	73
sscSetPointOffset.	75
sscSetPressData.	79
sscSetSamplingParameter	142
sscSetStartingPassNumber.	138
sscStartLog	147
sscStartSampling	140
sscStopLog	148
sscStopMonitor	119
sscStopSampling.	141
sscSystemStart	62
sscUnregisterIntCallback.	188
sscWaitIntDriveFin	206
sscWaitIntEvent.	191
sscWaitIntEventMulti	195
sscWaitIntOasEvent	199
sscWaitIntPassPosition	203
sscWaitStatusBitSignalEx.	72
sscWdDisable	129
sscWdEnable	128

T

TRANSIENT_CMD結構.	231
TRANSIENT_STS結構.	232

四畫

中斷函數群	181
介面模式函數群	166
日誌函數群	147

五畫

用戶看門狗函數群	128
--------------------	-----

七畫

更改函數群	104
系統函數群	60

八畫

其他軸啟動函數群	131
--------------------	-----

十畫

高速監視函數群	121
-------------------	-----

十一畫

參數函數群	39
採樣函數群	140
軟元件函數群	30
通用監視函數群	116
通過位置中斷函數群	136
陳述式/狀態函數群	69

十二畫

報警函數群 114

十三畫

資訊函數群 33

運行函數群 81

十四畫

輔助函數群 28

十五畫

數位輸入輸出函數群 156

暫態傳送函數群 214

標記檢測函數群 162

十六畫

擋塊控制資料函數群 79

輸入輸出軟元件函數群 208

十七畫

點位元表函數群 73

修訂記錄

*本手冊號在封底的左下角。

修訂日期	*手冊編號	修改內容
2016年10月	IB (NA) -0300305CHT-A	第一版
2019年09月	IB (NA) -0300305CHT-B	第二版 部分修改
2025年03月	IB (NA) -0300305CHT-C	第三版 部分修改

日文原稿手冊：IB-0300224-M

本手冊不授予工業產權或任何其它類型的權利，也不授予任何專利許可。三菱電機對於使用了本手冊中的內容而引起的涉及工業產權的任何問題不承擔責任。

© 2016 MITSUBISHI ELECTRIC CORPORATION

保固

1. 免費保固期限和免費保固範圍

如果產品在免費保固期限發生了因本公司責任而導致的故障或瑕疵(以下統稱“故障”)時,本公司將通過銷售商或本公司的售後服務公司免費對產品進行修理。但如果需要在國內或海外出差維修時,則要收取派遣技術人員的實際費用。此外,因故障單元的更換而發生的現場再調試、試運轉不屬於本公司責任範圍。

[免費保固期限]

關於產品的免費保固期限,請向您的三菱產品銷售商進行諮詢。

[免費保固範圍]

- (1) 首次故障診斷原則上由貴公司負責實施。但應貴公司要求,本公司或者本公司維修網點可有償提供該項業務。此時,如果故障是由於本公司原因而導致的,則該項業務免費。
- (2) 僅限於使用狀態・使用方法及使用環境等均遵照使用說明書、用戶手冊、產品本體注意標籤等規定的條件・注意事項等,並在正常狀態下使用的情況。
- (3) 即使在免費保固期限內,以下情況也要收取維修費用。
 - (i) 因客戶保管或使用不當、疏忽、過失等引起的故障,以及因客戶的硬體或軟體設計內容引起的故障。
 - (ii) 因客戶未經本公司允許對產品進行改造等而引起的故障。
 - (iii) 將本公司產品組合安裝到用戶的設備中使用時,如果用戶的設備上安裝了法規規定的安全裝置或業界標準要求配備的功能和構造後即可避免的故障。
 - (iv) 如果正常維護、更換使用說明書中指定的消耗品即可避免的故障。
 - (v) 消耗品(電池、風扇等)的更換。
 - (vi) 由於火災、異常電壓等不可抗力引起的外部因素以及因地震、雷電、風災水災等自然災害引起的故障。
 - (vii) 根據從本公司出廠時的科技標準還無法預知的原因而導致的故障。
 - (viii) 其他任何非本公司責任或客戶認為非本公司責任的故障。

2. 產品停產後的有償維修期限

- (1) 本公司在本產品停產後的7年內受理該產品的有償維修。關於停產的消息將通過本公司銷售和售後服務人員進行通告。
- (2) 產品停產後,將不再提供產品(包括維修零件)。

3. 海外服務

在海外,由本公司在當地的海外FA中心受理維修業務。但是,請注意各個FA中心的維修條件等可能會有所不同。

4. 機會損失和間接損失等不在保固責任範圍內

無論是否在免費保固期內,本公司對於以下內容都不承擔責任。

- (1) 非本公司責任的原因而導致的損失。
- (2) 因本公司產品故障而引起的用戶機會損失、利潤損失。
- (3) 無論本公司能否預測的特殊事件引起的損失和間接損失、事故賠償、對本公司產品以外的損傷。
- (4) 用戶更換設備、現場機械設備的再調試、啟動試運轉及其他作業的賠償。

5. 產品規格的更改

目錄、手冊或技術資料等所記載的規格如有變更,恕不另行通知。

6. 關於產品的適用範圍

- (1) 在使用本公司定位板時,應該符合以下條件:即使在定位板出現問題或故障時也不會導致重大事故,並且應在設備外部系統地配備能應付任何問題或故障的備用設備及失效安全功能。
- (2) 本公司定位板是以一般工業用途等為對象設計和製造的通用產品。
因此,定位板不適用於面對各電力公司的核電站以及其他發電廠等對公眾有較大影響的用途、及面向各鐵路公司或行政機關等要求構建特殊質量保證體系的用途。此外,定位板也不適用於航空航天、醫療、鐵路、焚燒・燃料裝置、載人運輸裝置、娛樂設備、安全設備等預計對人身財產有較大影響的用途。
但是,對於上述用途,在用戶同意限定用途且無特殊質量要求的條件下,可對其適用性進行研究討論,請與本公司服務窗口聯繫。
- (3) 由於阻斷服務攻擊(DoS攻擊)、非法訪問、電腦病毒及其他網路攻擊而發生的可程式控制器及系統方面的各種問題,三菱電機概不負責。

商標

Microsoft、Visual Basic、Visual C++、Visual C#及Windows是Microsoft集團的企業商標。

PCI Express是PCI-SIG的註冊商標或商標。

CompactPCI是PCI Industrial Computer Manufacturers Group的註冊商標。

本手冊中的公司名、系統名和產品名等是相應公司的註冊商標或商標。

本手冊中，有時未標明商標符號(™、®)。

IB (NA) -0300305CHT-C (2503) STC

MODEL: MRMC2-U-API-CHT

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

Specifications subject to change without notice.