**繪圖型液晶顯示器**

* 1. **繪圖型液晶顯示器介紹**
* 數字型液晶顯示器祇能顯示 0 ~ 9 的數字，文字型 LCD 則能顯示存在CGROM 內 160 個 5x7 點矩陣字型。除此之外，若要顯示其它的文字、符號可以靠 CG RAM 的造字功能。但若想在 LCD 螢幕上顯示一個圖樣或者動畫，例如史奴比、UFO，或者一點、一線、一圓，‥等，無法用 5x7 點矩陣表現之圖形，只好選擇更高級的繪圖型 LCD 了。
* 本篇針對目前最常用、最便宜的一型作討論。產品編號為雄鐸DG128064和華菱光電所出品的WG12864同屬相容，重要規格如下:
* LCD螢幕橫向128點，縱向64點。採用兩顆HD61202點矩陣液晶驅動器作為64個SEG驅動輸出，以及一顆64個COM驅動晶片HD61203作波形輸出控制。
* 可將128 x 64大小螢幕視為兩塊64 x 64點矩陣LCD，分別由CS1與CS2兩支腳加以致能，亦即控制兩顆HD61202是否動作。
* Y座標共計128點，addr 00~ addr 63由CS1控制; addr 64 ~ addr 127由CS2控制。
* X座標共計64點，劃分為Page 0 ~ Page7八頁，每一頁佔有八點，剛好是一個Byte資料。
* 腳位說明：

|  |  |  |
| --- | --- | --- |
| 接腳名稱 | 方向 | 功 能 |
| Pin1: Vss | I | 接地端 |
| Pin2: Vcc | I | 電源端，接+5V |
| Pin3: Vo | I | LCD亮度調整端 |
| Pin4: | I | 暫存選擇信號 =0選到指令暫存器  =1選到資料暫存器 |
| Pin5: | I | 讀/寫選擇信號 =0寫入LCD資料  =1讀取LCD資料 |
| Pin6: E | I | 動作致能端 |
| Pin7 ~ Pin14: DB0 ~ DB7 | I/O | 資料匯流排 |
| Pin15: CS1 | I | CS1='1'時, LCD左半螢幕64 x 64點致能 |
| Pin16: CS2 | I | CS2='1'時, LCD右半螢幕64 x 64點致能 |
| Pin17: RST | I | RESET重置端, 低態脈波致能 |
| Pin18: VEE | 0 | 負電壓- 10V輸出端,由LCD驅動晶片產生 |
| Pin19,20: NC |  | 未使用 |

* LCD 模組與ECU 作介面：



* 值得注意的是Pin 3 為LCD亮度調整之輸入接腳 Vo，可用來調整螢幕之亮度及清晰度。LCD之產品大都具有旋鈕，DG-128064廠商提供的亮度調整電路如圖7-3所示。其中的10K可變電阻便是實際產品中所使用到的旋鈕。
* 其次，Pin15(CS1)以及Pin16(CS2)分別用來控制LCD左右螢幕的致能接腳，Pin18(VEE)送出-10V左右的負電壓。購買華菱光電請務必注意，有關WG12864繪圖模式LCD有兩種Type，分別為

① WG12864A-NGG-V#A : Pin18腳位無法送出-10V之負電壓。

② WG12864A-NGG-V#N : Pin18腳位可以送出-10V之負電壓。

購買前者雖然較為便宜，但必須額外準備一個 -10V 的電壓源。

* 繪圖模式LCD所執行之動作，完全由以及兩條信號線

來決定，歸納於下表中。

|  |  |  |
| --- | --- | --- |
|  |  | GLCD 操作狀態 |
| 0 | 0 | 將命令碼寫入指令暫存器中 |
| 0 | 1 | 讀取LCD目前之狀態資料 |
| 1 | 0 | 將顯示資料寫入輸入暫存器中 |
| 1 | 1 | 由輸出暫存器讀取顯示資料 |

* DG-128064 指令集：



顯示器 ON/OFF:



* E=1 : 顯示器開啟
* E=0 : 顯示器關閉

(2) 顯示起始列



將 "RRRRRR" 資料寫入起始列暫存器中，繪圖型LCD將根據暫存器之內容來設定顯示列在距離螢幕頂端多少列處開始顯示。

(3) 設定頁數(X 位址)



顯示資料RAM中的X位址是以頁數為單位，整個螢幕共計64個像素，共計分為八頁，由 "XXX" 三個位元作設定。當位元設定完成，此後任何MPU讀寫資料都在這個頁數上進行，直到下一次頁數被設定為止。

(4) 設定位址(Y位址)



顯示資料RAM中的Y位址是以Y位址計數器所設定，共計128個像素由CS1與CS2兩支腳位區分為左右兩塊，每一塊區間由 "YYYYYY" 加以定址。當Y 位址設定完畢，每次MPU讀寫資料完畢後，Y計數器內容自動加1。

(5) 狀態讀取



三個位元旗標，所代表的意義如下:

* BUSY: 當 BUSY 為 "1" 時，表示LCDM正在執行內部動作，無法接受任何指令。因此，我們在執行任何一個指令前，必須讀取狀態資料，確定BUSY旗標是否拉為 "0"。
* ON/OFF: 該旗標指示液晶的顯示狀態 — ON ("1") 或 OFF ("0")。
* RESET: 當 RESET 旗標為 "1"，表示目前系統被初始化，除了狀態讀取指令外，無法接受任何命令。當 RESET為 "0"，表示LCDM初始化動作結束，系統進入正常操作。

(6) 顯示資料的寫入



下達該指令可將八位元資料 "DDDDDDDD" 寫入顯示資料RAM中。此時，Y位址計數器將自動加 1。

(7) 顯示資料的讀取



* 下達該指令可由顯示資料RAM中讀出目前位址指標所指之資料，且Y位址計數器將自動加1。
  1. **繪圖型液晶顯示器程式撰寫**

螢幕顯示是以一個Byte為單位，顯示步驟如下:

* Step1: 設定X頁地址 (0~7)
* Step2: 設定Y行地址 (0~63)
* Step3: 將顯示資料(為一個Byte)寫入RAM緩衝器中。

圖5-2為CS1=1，CS2=0，X=2(Page)、Y=9(addr)，且將11010001資料寫入顯示 RAM 之情形。



**WG12864\_COMMAND程式流程:**

(1) 設定GLCD的 腳及 腳位信號 "Lo"。

(2) 將GLCD的致能信號接腳E拉為 "Hi"。

(3) 將命令碼(R2)寫至GLCD的巴士匯流排DB0 ~ DB7。

(4) 將GLCD的致能信號接腳E回復為 "Lo"。

(5) 呼叫RG12864\_BUSY\_FLAG外部模組，以測試忙碌旗標。

* 參閱指令集，設定 =0及 =0是將命令碼寫至GLCD內部暫存器中，執行的功能包括: 設定X頁數(Page) ，設定Y地址(Address)，設定顯示器ON/OFF，以及設定顯示起始列等。
* 呼叫本模組前，應將命令碼先存至R2暫存器中。

設定顯示位置X=03 (頁) → MOV R2,#10111011B

設定顯示位置Y=03 (位址) → MOV R2,#01000011B

**WG12864\_COMMAND：**

|  |
| --- |
| **;---------------------------------------------------------------**  **;WG12864\_COMMAND**  **;(1)FUNCTION: WRITE DG-128064 COMMAND to (IR)**  **;(2)INPUT : R2**  **;(3)OUTPUT : NONE**  **;(4)USE REG.: R2**  **;---------------------------------------------------------------**  **EXTRN CODE(RG12864\_BUSY\_FLAG)**  **PUBLIC WG12864\_COMMAND**    **ROM SEGMENT CODE**  **RSEG ROM**    **GLCD\_E EQU P3.4**  **GLCD\_RW EQU P3.3**  **GLCD\_DI EQU P3.2**  **GLCD\_BUS EQU P2**    **WG12864\_COMMAND:**  **CLR GLCD\_RW**  **CLR GLCD\_DI**  **SETB GLCD\_E**  **MOV GLCD\_BUS,R2**  **CLR GLCD\_E**  **SETB GLCD\_DI**  **SETB GLCD\_RW**  **CALL RG12864\_BUSY\_FLAG**  **RET**  **END** |

**WDG1\_DATA程式流程:**

(1) 設定GLCD的 信號為 "Lo"， 信號為 "Hi"。

(2) 將GLCD 的致能信號接腳拉為 "Hi"。

(3) 將顯示資料R2寫至GLCD的Bus匯流排上。

(4) 將GLCD 的致能信號接腳回復為 "Lo"。

(5) 呼叫RGl2864\_BUSY\_FLAG外部模組，以測試忙碌旗標。

* 祇要將點對應(Dot Mapping)資料存至R2中，再呼叫本模組，即可

達到顯示之目的。例如，希望顯示如下之畫面:



程式設計如下:

MOV R2,#10111011B

MOV WG12864\_COMMAND ;設定X頁數為3

MOV R2,#01000011B

MOV WG12864\_COMMAND ;設定Y位址為3

MOV R2,#10101010B

MOV WG12864\_DATA ;將顯示資料寫入顯示RAM

MOV R2,#01010101B

MOV WG12864\_DATA ;將顯示資料寫入顯示RAM

**WDG1\_DATA：**

|  |
| --- |
| **;---------------------------------------------------------------**  **;WG12864\_COMMAND**  **;(1)FUNCTION: WRITE DG-128064 COMMAND to (IR)**  **;(2)INPUT : R2**  **;(3)OUTPUT : NONE**  **;(4)USE REG.: R2**  **;---------------------------------------------------------------**  **;--------------------------------------------------------------**  **;WG12864\_DATA**  **;(1) FUNCTION: WRITE DISPLAY DATA TO DR**  **;(2) INPUT : R2 ==> DISPLAY DATA**  **;(3) OUTPUT : NONE**  **;(4) USE REG.: R2**  **;---------------------------------------------------------------**  **EXTRN CODE(RG12864\_BUSY\_FLAG)**  **PUBLIC WG12864\_DATA**  **ROM SEGMENT CODE**  **RSEG ROM**  **GLCD\_E EQU P3.4**  **GLCD\_RW EQU P3.3**  **GLCD\_DI EQU P3.2**  **GLCD\_BUS EQU P2**  **WG12864\_DATA:**  **CLR GLCD\_RW**  **SETB GLCD\_DI**  **SETB GLCD\_E**  **MOV GLCD\_BUS,R2**  **CLR GLCD\_E**  **SETB GLCD\_DI**  **SETB GLCD\_RW**  **CALL RG12864\_BUSY\_FLAG**  **RET**  **END** |

**CG12864\_SCREEN程式流程:**

(1) 重置GLCD ，即在 接腳輸入一個符合重置信號規格的低態脈波。

(2) 令CS1= "1"，致能GLCD的左半區間

(3) 令CS2= "1"，致能GLCD的右半區間

(4) 設定顯示器為ON

(5) 設定顯示起始列為 00

(6) 以R1作為外迴路指標，設定X為第0頁。

(7) 設定Y位址為0。

(8) 設定外迴路指標X=0~7，內迴路指標Y=0~63，計數次數R0=64。將 "00" 寫至顯示資料 RAM 中，藉以清除整個GLCD之螢幕。

* + 本模組之功能有三

 重置GLCD，由規格書中得知 重置信號為一低態脈衝。重置後GLCD處於以下之狀態:

(1) 顯示器OFF。

(2) 設定顯示起始列暫存器為00。

 設定CS1及CS2，致能GLCD的左半區間及右半區間。

 將顯示資料 "00" 填入整個顯示資料RAM中，亦即清除整個GLCD螢幕。

* + 程式一開始請先呼叫本模組，可將螢幕清除乾淨，致能左右區間，

並處於初始狀態，並可確定這片繪圖型LCD動作正常。

**CG12864\_SCREEN：**

|  |
| --- |
| **;------------------------------------------------**  **;CG12864\_SCREEN**  **;(1) FUNCTION: CLEAR DG-128064 SCREEN**  **;(2) INPUT : NONE**  **;(3) OUTPUT : NONE**  **;(4) USE REG.: R0, R1, R2**  **;------------------------------------------------**  **EXTRN CODE(WG12864\_COMMAND)**  **EXTRN CODE(WG12864\_DATA)**  **PUBLIC CG12864\_SCREEN**    **ROM SEGMENT CODE**  **RSEG ROM**    **GLCD\_CS1 EQU P3.5**  **GLCD\_CS2 EQU P3.6**  **GLCD\_RST EQU P3.7**  **CG12864\_SCREEN:**  **; (1) Reset GLCD**  **CLR GLCD\_RST**  **NOP**  **SETB GLCD\_RST**  **; (2) CS1=1**  **SETB GLCD\_CS1**  **; (3) CS2=1**  **SETB GLCD\_CS2**  **; (4) DISPLAY ON**  **MOV R2,#3FH**  **CALL WG12864\_COMMAND**  **; (5) SET DISPLAY START LINE = 0**  **MOV R2,#0C0H**  **CALL WG12864\_COMMAND**  **; (6) SET PAGE X**  **MOV R1,#0B8H**  **CLR\_X\_PAGE:**  **MOV A,R1**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **;(7) SET Y ADDRESS**  **MOV R2,#40H**  **CALL WG12864\_COMMAND**  **;(8) WRITE 00 DATA TO DISPLAY RAM**  **MOV R0,#64**  **MOV R2,#00H**  **CLR\_Y\_ADDR:**  **CALL WG12864\_DATA**  **DJNZ R0,CLR\_Y\_ADDR**  **INC R1**  **CJNE R1,#0C0H,CLR\_X\_PAGE**  **RET**  **END** |

**CG12864\_SCREEN程式流程:**

(1) 20~23: 由輸入暫存器R0來設定X頁數(0~ 7)。

(2) 24~28: 由輸入暫存器R1來設定 Y位址(0~63)。

(3) 29~37: 以R5圖案造型位元組個數作為計數器，R6作為指標，讀取

DPTR指定表格的資料，依次寫入由X頁數以及Y位址所指定的顯示資料RAM中。

* + 本模組可以讓使用者所設計的圖案造型很方便地顯示在 GLCD的任何一個位置上。呼叫前必需先設定R0(X頁數)、Rl (Y位址)來指定顯示位置、R5 (圖案造型Byte數目)以及DPTR(指定造型表格之名稱)才可將造型碼寫入顯示資料RAM中。
  + 本模組的應用有其限制存在，即造型寬度X最好不要超過一頁 (8 Bit)，長度Y則無限制。若圖案造型超過一頁，可要分兩次作寫入。

**CG12864\_SCREEN：**

|  |
| --- |
| **;--------------------------------------------------------------------------------**  **;EG12864\_GRAPH\_PATTERN**  **;(1)FUNCTION: CREATE GRAPHIC PATTERN FROM TABLE**  **;(2)INPUT: R5→THE NUMBER OF GRAPHIC BYTE (1-64)**  **; DPTR → LCD GARPHIC TEBLE NAME**  **; R0 → X PAGE (0 - 7)**  **; R1 → Y ADDRESS (0 -63)**  **;(3)OUTPUT : NONE**  **;(4)USE REG.: R0, R1, R2, R5, R6, R7**  **;----------------------------------------------------------------------------------**  **EXTRN CODE(WG12864\_COMMAND)**  **EXTRN CODE(WG12864\_DATA)**  **PUBLIC EG12864\_GRAPH\_PATTERN**  **ROM SEGMENT CODE**  **RSEG ROM**    **EG12864\_GRAPH\_PATTERN:**  **; (1) SET PAGE X USING R0**  **MOV A,#0B8H**  **ADD A,R0**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **; (2) SET ADDRESS Y USING R1**  **MOV A,#40H**  **ADD A,R1**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **; (3) GET GRAPHIC PATTERN FROM TABLE**  **MOV R6,#00**  **GRAPH\_LOOP:**  **MOV A,R6**  **MOVC A,@A+DPTR**  **MOV R2,A**  **CALL WG12864\_DATA**  **INC R6**  **DJNZ R5,GRAPH\_LOOP**  **RET**  **END** |

**EGl2864\_FIGURE\_PATTERN程式流程:**

(1)以R3作為一個指標，來檢測造形圖案中每一行所顯示的位置是位於LCD螢幕的左半邊或者右半邊。模組中以輸入暫存器R1作為放置圖案造型所在的位址，利用(R1+R3) – 64 < 0 ?

YES: JMP FIG\_LOOP1，檢測到的那一行位於LCD的左半邊。

NO: JMP FIG\_LOOP2，檢測到的那一行位於LCD的右半邊。

(2)檢測到的那一行位於LCD的右半邊，令CS1=0、CS2=1。由輸入暫存器R1及指標暫存器R3來設定 Y位址(0~63)。

(3)檢測到的那一行位於LCD的左半邊，令CS1=1、CS2=0。由輸入暫存器R1及指標暫存器R3來設定 Y位址(0~63)。

(4)由輸入暫存器R0來設定X頁數(0~ 7)。

(5)以R5作為圖案造型寫入計數器，R3作為圖案造型各行指標，R4作為表格資料指標，讀取DPTR指定表格的資料，依次寫入由X頁數以及Y位址所指定的顯示資料RAM中。

* + 本模組是為了改良EG12864\_GRAPH\_PATTERN模組所設計，最大差異在於模組本身會自動判斷圖案造型中的每一行到底是位於LCD的右半邊或者左半邊，並且自動切換CS1以及CS2的邏輯準位。輸入暫存器R1(Y位址)的範圍變成0~127像素，輸入暫存器R0(X頁數)仍舊保持0~7頁。呼叫本模組確實可以讓使用者所設計的圖案造型很方便地顯示在 GLCD的任何一個位置上。
  + 本模組在存放表格資料上作了一些修改，使得所有的圖案造型可以同時放在一個表格內，再藉由輸入暫存器R4作為表格指標指到欲作顯示的造型即可。舉個例子來說明這樣作的好處何在? 倘若在繪圖模式液晶顯示器上秀出0、1、2……9阿拉伯數字，每個數字佔有5 x 7 字型。此時，若將每個阿拉伯數字分別作成一個表格，則程式中會有10個表格出現，難免過於混亂。最好還是將所有阿拉伯數字通通納入同一個表格內，要秀哪一個數字再從相對應的位址讀取該數字的字型碼。

**EGl2864\_FIGURE\_PATTERN：**

|  |
| --- |
| **;---------------------------------------------------------------------------**  **;EG12864\_FIGURE\_PATTERN**  **;(1)FUNCTION: CREATE FIGURE PATTERN FROM TABLE**  **;(2)INPUT : R5 => THE NUMBER OF FIGURE BYTE (1-64)**  **; : DPTR => LCD FIGURE TEBLE NAME**  **; : R0 => X PAGE (0 - 7)**  **; : R1 => Y ADDRESS (0 - 127)**  **; : R4 => INDEX IN TABLE = NUM \* R5**  **;(3)OUTPUT : NONE**  **;(4)USE REG.: R0,R1,R2,R3,R4,R5**  **;---------------------------------------------------------------------------**  **EXTRN CODE(WG12864\_COMMAND)**  **EXTRN CODE(WG12864\_DATA)**  **PUBLIC EG12864\_FIGURE\_PATTERN**    **ROM SEGMENT CODE**  **RSEG ROM**    **GLCD\_CS1 EQU P3.5**  **GLCD\_CS2 EQU P3.6**    **EG12864\_FIGURE\_PATTERN:**  **MOV R3,#0**  **FIG\_LOOP4:**  **MOV A,R1**  **ADD A,R3**  **CLR C**  **SUBB A,#64**  **JC FIG\_LOOP1**  **FIG\_LOOP2:**  **SETB GLCD\_CS2**  **CLR GLCD\_CS1**  **MOV R2,#40H**  **ADD A,R2**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **JMP FIG\_LOOP3**  **FIG\_LOOP1:**  **SETB GLCD\_CS1**  **CLR GLCD\_CS2**  **MOV A,R1**  **ADD A,R3**  **MOV R2,#40H**  **ADD A,R2**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **FIG\_LOOP3:**  **MOV A,#0B8H**  **ADD A,R0**  **MOV R2,A**  **CALL WG12864\_COMMAND**    **MOV A,R4**  **MOVC A,@A+DPTR**  **MOV R2,A**  **CALL WG12864\_DATA**  **INC R4**  **INC R3**  **DJNZ R5,FIG\_LOOP4**  **RET**  **END** |

**CG12864\_GRAPH\_ PATTERN程式流程:**

(1) 由輸入暫存器R0來設定X頁數 (0~7)。

(2) 由輸入暫存器R1來設定Y地址 (0~63)。

(3) 以R5清除圖案造型的Byte數目為計數值，將 "00" 寫至X頁數(R0)與Y地址(R1)所指定的顯示資料 RAM 中。

* + EGl2864\_GRAPH\_PATTERN模組是建立圖案造型並寫入顯示RAM 中，而CG12864\_GRAPH\_PATTERN正好相反，乃作為清除圖案之用。
  + 要清除GLCD螢幕上的一塊圖案，祇要將 "00" 寫入這塊區間所對映的顯示RAM中即可。故呼叫本模組前，要先指定造型所在位置及圖案大小(Byte數)。

**CG12864\_GRAPH\_PATTERN：**

|  |
| --- |
| **;---------------------------------------------------------------------------**  **;CG12864\_GRAPH\_PATTERN**  **;(1)FUNCTION: CLEAR GRAPHIC PATTERN**  **;(2)INPUT: R5→THE NUMBER OF GARPHIC BYTE (1-64)**  **; R0→X PAGE (0 - 7)**  **; R1→Y ADDRESS (0 -63)**  **;(3)OUTPUT : NONE**  **;(4)USE REG: R0, R1, R2, R5**  **;------------------------------------------------------------**  **EXTRN CODE(WG12864\_COMMAND)**  **EXTRN CODE(WG12864\_DATA)**  **PUBLIC CG12864\_GRAPH\_PATTERN**    **ROM SEGMENT CODE**  **RSEG ROM**    **CG12864\_GRAPH\_PATTERN:**  **; (1) SET PAGE X USING R0**  **MOV A,#0B8H**  **ADD A,R0**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **; (2) SET ADDRESS Y USING R1**  **MOV A,#40H**  **ADD A,R1**  **MOV R2,A**  **CALL WG12864\_COMMAND**  **; (3) WRITE ‘00’ TO DISPLAY DATA RAM**  **MOV R2,#00H**  **CLEAR\_LOOP:**  **CALL WG12864\_DATA**  **DJNZ R5,CLEAR\_LOOP**  **RET**  **END** |

**RG12864\_DATA程式流程:**

(1) 假讀動作，由資料匯流排讀入顯示RAM的資料。

(2) 呼叫RG12864\_BUSY\_FLAG外部模組，以測試忙碌旗標。

(3) 真正讀取，由資料匯流排讀入輸出暫存器資料。

(4) 呼叫RG12864\_BUSY\_FLAG外部模組，以測試忙碌旗標。EGl2864\_GRAPH\_PATTERN模組是建立圖案造型並寫入顯示RAM 中，而CG12864\_GRAPH\_PATTERN正好相反，乃作為清除圖案之用。

* + 模組的設計完全根據圖7-6 DG-128064讀取時序而設計，重點在於多了一步假讀之動作。
  + 使用本模組前，請先完成X頁數及Y位址之設定。執行本模組後，Y 位址自動加1 ，可直接讀取之。

**RG12864\_DATA：**

|  |
| --- |
| **;----------------------------------------------------------**  **;RG12864\_DATA**  **;(1)FUNCTION: READ DG128064 RAM DATA**  **;(2)INPUT : NONE**  **;(3)OUTPUT : R2**  **;(4)USE REG.: R2**  **;-------------------------------------------------------------**    **EXTRN CODE(RG12864\_BUSY\_FLAG)**  **PUBLIC RG12864\_DATA**    **ROM SEGMENT CODE**  **RSEG ROM**    **GLCD\_E EQU P3.4**  **GLCD\_RW EQU P3.3**  **GLCD\_DI EQU P3.2**  **GLCD\_BUS EQU P2**  **RG12864\_DATA:**  **;(1) DUMMY READ DATA**  **SETB GLCD\_DI**  **SETB GLCD\_RW**  **SETB GLCD\_E**  **MOV GLCD\_BUS,#0FFH**  **MOV R2,GLCD\_BUS**  **CLR GLCD\_E**  **CLR GLCD\_DI**  **CALL RG12864\_BUSY\_FLAG**    **SETB GLCD\_DI**  **SETB GLCD\_E**  **MOV R2,GLCD\_BUS**  **CLR GLCD\_E**  **CALL RG12864\_BUSY\_FLAG**  **RET**  **END** |