1

## مثال : اجراى دستور جمع توسط C-Action

پس از تعریف سه عدد تگ داخلی بصورت Unsigned 32 bit با نامهای f1 ، f1 و fsum تنظیمات زیر

را انجام داده و کد C-Action مربوطه را به تحریک موس اتصال می دهیم .

Main.pdl X		
F1: 0,000 F2: 0,000	=F1+F2	
SUM	•	
	Edit Action	
<	■■■■■■	ĉe   👑 🤪 A 🕘 🕩
Object Properties Properties Events	Project functions Standard functions Internal functions	Hinclude "apdefap.h" void OnLButtonDown(char* lpszPictureName, char* lpsz( { int f1,f2,f3;
Rectangle     Execute on     Action       Mouse     Mouse Click     Mouse Click       Focus     Focus     Focus       Miscellaneous     Press Left     Mouse Click       Property Topics     Release Left     Release Right       Image: Secondary Constraint     Mouse Click     Release Click		f1=GetTagWord("f1"); f2=GetTagWord("f2"); f3=f1+f2; SetTagWord("fsum",f3); // WINCC:TAGNAME_SECTION_START // syntax: #define TagNameInAction "DMTagName" // next TagID : 1 // WINCC:TAGNAME_SECTION_END
F1: 65,000 F2: 25,000	Fsum=F1+F2 90.0	000



مثال : برنامه متوسط گیری از سه عدد توسط C-Action

پس از تعریف جهار عدد تگ داخلی بصورت Float 32 bit با نامهای NUM1 الی NUM3 و AVR ، در C-Action مربوط به تحریک کلید چپ یک شی مستطیل که به منظور فرمان انجام متوسط گیری در نظر گرفته شده است ، دستورات زیر تایپ شده و سپس با قراردادن چهار عدد ابزار I/O Filed برای نمایش چهار تگ فوق و مقداردهی در محیط RunTime برنامه را چک می نماییم.

#include "apdefap.h"

void OnLButtonDown(char\* lpszPictureName, char\* lpszObjectName, char\* lpszPropertyName, UINT nFlags, int x, int y)

float val1,val2,val3,avr\_tag;

val1=GetTagFloat("NUM1"); //Return-Type: float val2=GetTagFloat("NUM2"); //Return-Type: float val3=GetTagFloat("NUM3"); //Return-Type: float

avr\_tag=((val1+val2+val3)/3); SetTagFloat("AVR",avr\_tag); //Return-Type: BOOL

// WINCC:TAGNAME\_SECTION\_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME\_SECTION\_END

// WINCC:PICNAME\_SECTION\_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME\_SECTION\_END
}



(كانال تخصصي آموزش اتوماسيون صنعتي ، ابزاردقيق و الكترونيك )control0ta100@

مثال : ساخت Push-Botton بصورت Momumentry توسط C-Action مربوط به تحریک Push-Botton بر روی ابتدا یک عدد تگ بصورت Bool تعریف نموده و سپس در C-Action مربوط به تحریک Mouse بر روی یک ابزار مستطیل کد زیر را وارد می کنیم. در صورتی که کلید تحریک شده و مقدار تگ برابر یک باشد ، مقدار صفر جایگزین شده و در حالت عکس و در صورتی که مقدار تگ برابر یک باشد ، مقدار صفر جایگزین می شود . در این صورت کلید مورد نظر ساخته شده است .

if(GetTagBit("valve\_tag")==1)
SetTagBit("valve\_tag",0); //Return-Type: BOOL
else
SetTagBit("valve\_tag",1); //Return-Type: BOOL



مثال : تغییر رنگ پس زمینه توسط **برنامه نویسی با کد C** 

در این مثال می خواهیم در محیط RunTime با تحریک یک کلید ، ر نگ پس زمینه صفحه اصلی تغییر یابد ۱- ایجاد یک یالت رنگ جدید و تعریف رنگ های دلخواه :



(كانال تخصصي آموزش اتوماسيون صنعتي ، ابزاردقيق و الكترونيك )controlOta100@

1

Color Selection	Color Selection
Colors	Olors
	(°)
Transparency 2	Transparency 16
Red 0	Red 148
Green 0	Green 30
Blue 0	Blue 130
HTML code: 000000	HTML code: 9 1E82 To Palette
	Proiect Palette
	New color range
20 21 22 23 24 25 26 27 28 29	
30 31 32 33 34 35 36 37 38 39	40         41         42         43         44         45         46         47         48         49
	50 51 52 53 54 55 56 57 58 59
ک را در خانه ای از پالت تنظیم نمایید.	بر اساس الگوی فوق تعدادی کد رنگ ایجاد نموده و هر یا

در صفحه اصلی WinCC برنامه C-Editor را باز نمایید و پس از ایجاد یک Project Function جدید ، کد زیر را با رعایت حروف کوچک و بزرگ در آن بنویسید .

🔏 WinCC Explorer - C:\Users\Public\Documents\Siemens\W	nCCFrojects\112\112.MCF [ Active	
File Edit View Tools Help		
[]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]	<b>?</b> Name	Global Script C - \\TAVANA-PC\WinCC Project 112\112.mcp : Color MT-C.f.t
Computer Tag Management Menus and too bars Menus and too bars Alarn Logging Tag Logging C-Editor Global Script Global Script Standard functions WSS-Editor Text Library Crest-Computer Cross-Reference Constructions Redundancy User Administrator Constructions Redundancy User Administrator Redundancy User Arbive	Computer Tag Management A Graphics Designer Menus and toolbars Alarm Logging Tag Logging Report Designer Global Script Tase Ibary Tose Distributor Orss-Refarence A Load Online Changes Redundarcy User Archive C Time synchronization D Hoin P Citure Trae Manager Lifebeat Monitoring C Ospingert Editor	Global Script C - (VIAVANA-PC/WINCC_Project_I12/I12.mcp: Color MI-C.frt File Edit View Window Options Help Global Script C • • • × Global Script C • • • × Global Script C • • • × GetPaetteColor GetPaetteColor (int ColorIndex) GetPaetteColor (int ColorIndex) File ColorIndex = 199; return pow (2, 31) + ColorIndex; J

(كانال تخصصي آموزش اتوماسيون صنعتي ، ابزاردقيق و الكترونيك )control0ta100@

توجه :شماره هر مربع که رنگی در آن ریخته می شود در ادامه به عنوان شماره رنگ مورد نظر در دستورات استفاده می شود . البته بصورت پیش فرض خود نرم افزار تعدادی از شماره های اول این خانه را با رنگ های پیش فرض پر نموده است که اگر بخواهیم از آنها استفاده کنیم نیازی به طی مراحل فوق نمی باشد .

```
long GetPaletteColor(int ColorIndex)
{
    if (ColorIndex > 199)
    ColorIndex = 199;
    return pow(2,31) + ColorIndex;
}
and any of the colorIndex and any of the colorIndex;
}
Zout ColorIndex and any of the colorIndex any of the colorIndex and any of the colorIndex a
```

۲- تنظیمات محیط گرافیک

در محیط گرافیک ابتدا در مشخصه صفحه پروژه تنظیمات Global Color Scheme را غیر فعال نمایید. سپس یک شی مستطیل در صفحه قرار داده و در تحریک موس آن یک کد Action تعریف نموده و دستور زیر را در آن بنویسید . دقت نمایید Maín نام صفحه اصلی است که می خواهیم رنگ پس زمینه آن تعویض شود و عدد یک شماره رنگ مورد نظر از پالت رنگ ها است .

#include "apdefap.h"

void OnClick(char\* lpszPictureName, char\* lpszObjectName, char\*
lpszPropertyName)

SetBackColor(lpszPictureName, "Main", GetPaletteColor(1));

توجه در صورتی که در این دستور نام مربوط به هر شی دیگر را جایگزین نموده و برای آن شی نیز Global Color Tocheme را غیرفعال کنیم همین شرایط انجام شده و رنگ پس زمینه تغییر می کند. دستور فوق دستور SetBackColor یعنی تغییر رنگ پس زمینه می باشد . در این دستور از یک فانکشن که قبلا در محیط برنامه نویسی C ساخته بودیم (GetPaletteColor(1)) برای فراخوانی رنگ از پالت استفاده کرده ایم .

در ادامه توضیحاتی در خصوص نحوه ایجاد فانکشن و سپس فراخوانی آن در Action ذکر می گردد.



ساختار کد نویسی C در WinCC به این نحو می باشد که می توان بصورت مستقیم یک C-Action در محیط گرافیک ایجاد نمود و آنرا به یک تریگر ( به عنوان مثال تحریک موس بر روی یک شیء) متصل نمود . البته تریگر ها می تواند در انواع مختلف دیگری از جمله تغییرات مقدار یک تگ ، اتمام یک زمان سنجی و... نیز تعریف گردد .همچنین می توان با ایجاد یک فانکشن ، آنرا در داخل C-Action فراخوانی نمود .



فانکشن ها در سه دسته قابل تعریف می باشند :

Project Function \_ Internal Function - Standard Function

که دو دسته اول توسط شرکت سازنده تعریف و ایجاد شده و کاربر می تواند بدون امکان ویرایش آنها را در برناهه خود و بدون نیاز به تریگر استفاده نماید . نوع سوم یا Project Function ها توسط کاربر ایجاد و ویرایش می شوند .

برای ایجاد یک فانکشن همانگونه که د رمثال فوق توضیح داده شد می توان در محیط Global C-editor یک فانکشن جدید ایجاد نموده و پس از کدنویسی آنرا ذخیره نمود . در این حال این فانکشن همواره وجود داشته و در صورت لزوم می توان آنرا مورد استفاده قرار داد.

جهت ایجاد Action هم می توان همانند شکل زیر بصورت Global یا LOCAL ( فقط در یک Clinet ) نسبت به تعریف Action اقدام نمود .البته در محیط گرافیک نیز می توان همزمان تریگر و Action را مطابق با توضیحات مثالهای قبل ایجاد کرد.



1

File Edit View Wind	ow Options Help
: 🗋 🗿 🗐 🗎 🔂 🖌	( 🗅 📇 🖉 ୯ 🖨 😽 🍃
Global Script C	<b>▲</b> 廿 Ⅹ
Internal functions	
Actions	A-PC
Actions	A-PC New
Actions Actions : TAVAN	A-PC New Copy
Actions	A-PC New Copy Paste
Actions	A-PC New Copy Paste Delete
Actions	A-PC New Copy Paste Delete Find in files

جهت اتصال یک تریگر به یک Action می توان در محیط برنامه نویسی یک Action کلیک راست نموده



و گزینه info/Trigger را انتخاب نمود و سپس با توجه به نیاز یک تریگر تعریف نمود.



برنامه نویسی توسط C-Action در WinCC

E	Add
⊡∰ Cyclic	Delete
Tag	Change

در تنظیمات فوق در حالت سیکلی در یک بازه زمانی بر اساس الگوی تعریف شده Action اجرا می شود . در حالت Acyclic در یک زمان مشخص شده فقط یکبار اجرا می شود و در حالت Tag می توان برای یک Action یک یا چند تگ تعریف نموده که در صورت تغییر در آنها Action اجرا گردد .

مثال : ذخیره سازی اطلاعات ورود / خروج کاربران با بهره گیری از کد C

با توجه به اینکه با هر ورود و خروج کاربران ( Log out / Log out ) ، تگ داخلی سیستمی به نام CurrentUserName تغییر مقدار می دهد ، لذا می توان از این تغییرات این تگ به عنوان تریگر نمودن یک Action با کد دستورات زیر بهره گرفت . این Action با توجه به وضعیت Log In یا Log Out یک تگ بیتی متناظر را SET می کند . لذا دو عدد تگ بیتی داخلی نیز به نامهای LoginBit و LogoutBit تعریف می نماییم .

با باز نمودن C-Editor با کلیک راست بر روی قسمت Global Action یک فایل جدید ایجاد نموده و کد زیر را در داخل تابع می نویسیم . در این تابع دو متغیر جهت ذخیره سازی نام کاربری قبل و نام کاربری جدید در نظر گرفته شده است که اطلاعات خود را از تگ های سیستمی مربوطه دریافت می نمایند . در مقایسه انجام



شده ، در صورتی که طول کاراکتر در نام کاربری جدید مخالف صفر باشد ، تگ Logout برابر صفر شده و در تگ Log In یک لبه بالارونده ایجاد می شود .در غیر این صورت در تگ Logout یک لبه بالارونده ایجاد می شود .

```
#include "apdefap.h"
int gscAction( void )
#pragma code ("kernel32.dll")
void Sleep (int a);
#pragma code ()
char *NewUser, *OldUser;
OldUser=GetTagChar("@OldUser");
if (strlen(NewUser)!=0)
ł
SetTagBit("LogoutBit",0);
SetTagBit("LoginBit",0);
Sleep(1000);
SetTagBit("LoginBit",1);
SetTagChar("@OldUser", NewUser);
}
else
ł
SetTagBit("LoginBit",0);
Sleep(1000);
SetTagBit("LogoutBit",1);
}
return 0;
```

پس از نوشتن دستورات فوق ، در محلی خالی از صفحه کد نویسی کلیک راست نموده و جهت اعمال یک تریگر ، گزینه Info/Trigger را انتخاب می کنیم . در صفحه ای مطابق با شکل زیر تریگر از نوع تگ را به تگ سیستمی نام کاربری (CurrentUserName) ) متصل می کنیم تا در صورت تغییر وضعیت این تگ ، Action نوشته شده فراخوانی شود . در مرحله بعد Action را ذخیره نمود و سپس جهت نمایش اطلاعات مربوط به ورود/ خروج کاربران ، از Alarm Logging استفاده می کنیم. پس از باز نمودن محیط آلارم ، جهت نمایش نام کاربر ، ابتدا از قسمت Message Block یه کادر Process value به سیستم نمایش آلارم اضافه می نماییم. سپس در قسمت آلارم ها مطابق با شکل زیر دو عدد آلارم را به تگ های مربوطه متصل نموده و در کادر Process value نیز نام کاربر جاری را ذکر می کنیم.

<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp															
Alarm logging	«		Message	s [Alarm ]										Find	
🖃 🖂 Messages	*		Number	Message tag	Mes Statu	Stat	Ackr A	ckr I	Message	Mes Mess	Priority	Message text	Poin Process val	ue: 1	
Error 			1	1	LoginBit	0	0	0	E	Error	Alar	0	User Login Is:	@CurrentU:	serName
		2	2	LogoutBit	0	0	0	E	Error	Alar	0	User Logout Is:	@CurrentU:	serName	
		3	談	X											
		4													
🕀 🖼 System, requires acknov	1	5													
🕀 🖼 System, without acknow		6													
E Message blocks		7													
		8													

در محيط Runtime پس از هر ورود/ خروج آلارمي با ذكر تاريخ و زمان و نام كاربر مربوطه ايجاد مي گردد .

<mark>1</mark> W	WinCC AlarmControl								
0	) 🛃   隆	0 🛛 🖓 🕞 🍊	<b>.</b>	📓 😤 📓 🧧	ŝ   🔒				
	Datum	Uhrzeit	Nummer	Prozesswertblock	Anwen	derte	Anwer		
6	28.02.16	09:53:00 AM	2						
7	28.02.16	09:53:01 AM	1	tavana					
8	28.02.16	09:53:12 AM	1	tavana					
9	28.02.16	09:53:13 AM	1	tavana					
10	28.02.16	09:53:13 AM	1	assbco					
11	28.02.16	09:53:18 AM	1	assbco					
12	28.02.16	09:53:19 AM	2						
13	28.02.16	09:53:19 AM	2						
14					1				
4			III				F.		
stehe	end: 2 Zu d	quittieren: 2 Aus	geblendet: 0 Lis	te: 13		9:53:	49		



مثال : برنامه ای بنویسید که ابتدا دو عدد را از اپراتور دریافت نموده و سپس با استفاده از C-Action مربوط به تحریک یک کلید حاصل جمع آنها را محاسبه نماید . سپس با تحریک کلید دوم ، ابتدا لامپی را روشن نموده و با شمارش معکوس زمان به میزان عدد حاصل جمع مرحله قبل لامپ را خاموش نماید .



Tag Management	~		Tags [ Internal tags ]			
🗐 🛄 Tag Management	*		Name	Data type	Length	Format ada
		13	@TLGRT_AVERAGE_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8	
	=	14	@TLGRT_SIZEOF_NLL_INPUT_QUEUE	Floating-point number 64-bit IEEE	8	
		15	@TLGRT_SIZEOF_NOTIFY_QUEUE	Floating-point number 64-bit IEEE	8	
		16	@TLGRT_TAGS_PER_SECOND	Floating-point number 64-bit IEEE	8	
		17	cat	Text reference	4	
		18	dt	Unsigned 16-bit value	2	
П П ТСР/Р	55	19	language_eng	Text reference	4	
	<b>T</b>	20	language per	Text reference	4	
Tag Management		21	sum_var	Unsigned 16-bit value	2	
11 ····		22	x1_var	Unsigned 16-bit value	2	
🛃 Alarm logging		23	x2_var	Unsigned 16-bit value	2	
		24	k_var	Unsigned 16-bit value	2	
Tag Logging		25	bit_lamp_color	Binary Tag	1	
at Tart Library		26	514			
		27	2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2			
Illoor Administrator		28				





L Edit Action				? <b>X</b>
■ fl   X 車 車 ☆   ×   ら	Ĉe   🛗 📦 🛝 🕑 🕞	📺   Italian (Italy)		
Project functions Standard functions Internal functions	<pre>#include "apdefap.h" void OnClick(char* IpszPictureNat {     int l.t.g:     SetBackColor(IpszPictureName,     l=GetTagWord("sum_var");     while(l&gt;1)         {            I;             SetTagWord("k_var".l);             g=100;             while(g&gt;1)             {                g;                 t=10000;                 while(t&gt;1)                 {</pre>	ame, char* lpszObjectName, char* lpszProper "Circle1",RGB(255,0,0)); "Circle1",RGB(60,0,0));	tyName)	