

معرفی و آموزش استفاده از DLL دستگاه کالر آیدی مانیتور 2013

CallerID Monitor DLL

ویژگیهای DLL کالر آیدی مانیتور:

- سرعت بسیار بالا در پردازش اطلاعات دریافتی از کالر آیدی
- عدم نیاز به تعیین شماره پورت دستگاه جهت برقراری ارتباط (Auto Detect)
- امکان ارتباط و مدیریت چند دستگاه کالر آیدی به صورت هم زمان بر روی یک سیستم
- ادامه اجرای برنامه به صورت صحیح، در صورت قطع و وصل شدن کابل USB کالر آیدی (بدون نیاز به باز کردن مجدد برنامه یا پورت)
- کنترل خروجی به دو حالت تمام موارد (CallerID, RingCount, Tone Decoder, Onhook, Offhook) یا فقط شماره ی کالر آیدی
- مجزا کردن خودکار رشته های به هم چسبیده ورودی از بافر پورت سریال

نحوه ی تعریف یک متغیر از نوع کلاس Dll در زبان VB 6

تعریف متغیر جدید در دو مرحله صورت می گیرد. مرحله اول در قسمت عمومی و سراسری برنامه به شکل زیر:

```
Dim CID As CallerID_ActiveX_class
```

و در مرحله دوم تعریف فضای حافظه جدید برای این کلاس در قسمت لود شدن فرم برنامه:

```
Set CID = New CallerID_ActiveX_class
```

نحوه ی استفاده از فرامین کلاس Dll

۱- **Total_Devices:**

این تابع تعداد کالر آیدی های متصل به سیستم را بر می گرداند.

مثال:

```
Dim Var1 as Byte
```

```
Var1 = CID.Total_Devices
```

تعداد کالر آیدی های متصل به سیستم → Var1

۲- Select_Device:

با استفاده از این دستور می توانید اندیس کالرآیدی مورد نظر برای دسترسی را مشخص کنید. زمانی که تنها یک کالرآیدی به سیستم متصل است این مقدار همیشه عدد یک خواهد بود ، به عنوان مثال زمانی که ۳ دستگاه کالرآیدی به سیستم متصل است این اندیس می تواند حداکثر عدد سه باشد.

در مثال زیر کالرآیدی شماره ی (۱) به عنوان کالرآیدی هدف انتخاب شده است، حال تمام فرامین متغیر کلاس DLL با این کالرآیدی مرتبط خواهد بود.

CID.Select_Device (1)

۳- Enable_CallerID:

این دستور کالرآیدی انتخاب شده را فعال کرده و پورت مربوط به آن را باز خواهد کرد و پس از اجرای این دستور خروجی دستگاه پس از هر عکس العمل در متغیر بافر ذخیره خواهد شد.

مثال:

CID.Enable_CallerID

۴- CID.Read_Buffer:

با استفاده از این تابع رشته خروجی کالرآیدی را دریافت خواهید کرد (خواندن متغیر بافر). یکی از ویژگیهای مهم این تابع این است که اگر دو رویداد با هم و هم زمان به وقوع بپیوندند در هر مرتبه خواندن یک رویداد برای شما ارسال خواهد شد. برای درک بهتر به مثال زیر توجه کنید:

تماس جدید در خط ۱ و گوشی برداشتن در خط ۲ هم زمان رخ داده است که نتیجه رشته ورودی در پورت سریال سیستم به شکل زیر است:

L1:05118447276@L2:HookOn@

همان طور که می بینید شما قبلا باید دو رشته را از یکدیگر جدا می کردید و در برخی اوقات به علت پر شدن بافر سیستم، نیمی از رشته ناقص ظاهر می شد و نیمه دیگر آن با فاصله زمانی کوتاهی ظاهر می گشت و برخی برنامه نویسان در اتصال این دو رشته و تفکیک نهایی دچار مشکل و سر در گمی می شدند.

با ویژگی جدید Dll در هر بار خواندن مقدار تابع Read_Buffer یک بخش کامل رشته به صورت تفکیک شده ارسال خواهد شد و برنامه نویس برای الحاق و تفکیک نهایی مشکلی نخواهد داشت. خروجی تابع Read_Buffer برای مثال بالا به شکل زیر است:

S1 = CID.Read_Buffer → S1= "L1:05118447276"

S2 = CID.Read_Buffer → S2= "L2:HookOn"

حال اگر شما کالرآیدی را با دستور Enable_CallerID فعال کنید و در فاصله های زمانی کوتاهی به صورت پیوسته این تابع را فراخوانی کنید اطلاعات دریافتی را به صورت درست ، کامل و دقیق در دسترس خواهید داشت.

برای این کار می توانید از روش زیر استفاده کنید:

۱ - فعال کردن کالرآیدی و فعال کردن یک تایمر با دوره ی زمانی ۵۰۰ میلی ثانیه برای فراخوانی مکرر

CID.Enable_CallerID

Timer_ReadBuffer.Interval=500

Timer_ReadBuffer.Enabled = True

۲ - نوشتن دستورات دلخواه برای پردازش و عکس العمل مناسب در روال تایمر مذکور

Private Sub Timer_ReadBuffer_Timer()

s = **CID**.Read_Buffer

If s <> "" Then Print s

End Sub

۵-Show_Line_Number:

با فعال بودن این خصوصیت در رشته خروجی کالرآیدی که از متغیر بافر دریافت خواهید کرد شماره ی خط حذف خواهد شد. مثال:

s = **CID**.Read_Buffer → S="L1:HookOff"

CID.Show_Line_Number (False)

s = **CID**.Read_Buffer → S="HookOff"

۶-ShowOnlyNumber:

با غیر فعال بودن این خصیصه فقط شماره کالرآیدی در متغیر بافر قرار خواهد گرفت و مابقی اطلاعات مانند گوشی برداشته، گذاشته و ... به صورت خودکار حذف خواهد شد.
مثال: اگر خط ۱ زنگ بخورد و شما گوشی تلفن را پس از ۱ زنگ بردارید:

s = **CID**.Read_Buffer → s="L1:05118447276"

s = **CID**.Read_Buffer → s="L1:Ring:1"

s = **CID**.Read_Buffer → s="L1:HookOn"

CID.ShowOnlyNumber (True) → فعال کردن خصیصه فقط شماره و تکرار مثال بالا

s = **CID**.Read_Buffer → s="L1:05118447276"

s = **CID**.Read_Buffer → s=""

s = **CID**.Read_Buffer → s=""

۷- Last_Status:

خروجی این تابع آخرین مقدار رشته خروجی در پورت و یا رویداد کالرایدی است. مثال:

S= **CID**.Last_Status → s="L1:HookOn"

توجه: چند رویداد دیگر نیز غیر از رویدادهای مربوط به خط تلفن وجود دارد که در جدول زیر آمده است:

عنوان	شرح عملکرد
Connect	این رویداد نشان دهنده وقوع ارتباط درست با دستگاه کالرایدی است
DisConnect	این رویداد نشان دهنده قطع شدن کابل یو اس بی از دستگاه است
DeviceNotFound	این رویداد زمانی رخ خواهد داد که شماره ایندکس کالرایدی تعیین شده وجود نداشته باشد / یا دسترسی در حین کار دچار مشکل شده باشد که با قطع و وصل کردن کابل یو اس بی مشکل حل شده و رویداد Connect رخ خواهد داد
DeviceOpenError	این رویداد زمانی رخ خواهد داد که امکان دسترسی به کالرایدی وجود نداشته باشد

۸- port_number:

خروجی این تابع شماره پورتهای دستگاه کالرایدی جاری به آن متصل شده است.

Print **CID**.port_number

۹- Disable_CallerID:

با استفاده از این دستور DII غیر فعال شده و ارتباط با دستگاه کالرایدی قطع خواهد شد و پورت مربوطه بسته می شود. توجه داشته باشید که در هنگام بسته شدن برنامه حتما از این دستور استفاده نمائید تا پورت دستگاه بسته شود، برای این منظور بهتر است این خط دستور را در قسمت Unload برنامه خود قرار دهید:

CID.Disable_CallerID

*** زمانی که DLL در حال اجرا است، اگر کابل دستگاه قطع شود و چنانچه کابل مجددا وصل شود پس از چند ثانیه DLL دستگاه را به صورت خودکار تشخیص خواهد داد و نیازی به کد نویسی برای این مورد نخواهید داشت.

*** لازم به ذکر است که پس از نصب درایور خودکار، که بر روی سی دی همراه با دستگاه قرار دارد، DLL نیز در مسیر windows\system32 کپی و رجیستر خواهد شد، و دیگر نیاز نیست برنامه نویسی فایل DLL را در داخل Setup بسته نرم افزار قرار دهد و همچنین نیاز به رجیستر کردن DLL در ویندوز نیست.

در سی دی ارائه شده همراه با دستگاه، مثال های استفاده عملی از DLL به زبان های Visual Basic 6 و Visual Basic

.Net. قرار دارد، برای استفاده بهتر به این مثال ها مراجعه کنید.

توجه داشته باشید که در مثال های ارائه شده کلید های فرامین به صورت ستونی از بالا به پایین چیده شده است

که ترتیب اجرای هر یک از کلید ها به همان ترتیب بالا به پایین است.

برای درک بهتر برنامه مثال را کامپایل و اجرا کنید، سپس از بالا به پایین کلید ها را کلیک کنید و دستگاه را تست

بگیرید.