**GP7 SDK**

Rev 1.0

GHADIRCO.NET

**فهرست :**

1. مقدمه
   1. توضیحات و نحوه استفاده از DLL
2. توابع

|  |  |  |
| --- | --- | --- |
| ورژن | تاریخ | توضیحات |
| V1.0 | 01/10/1391 | در این ورژن یک سری از توابع اولیه برای برقراری ارتباط ، گرفتن و تنظیم تاریخ و ساعت ، تبدیل تاریخ میلادی به شمسی و جمع آوری ترددهای پرسنل موجود می باشد .  لیست توابع عبارتند از :  Open\_Serial  Open\_Ehternet  Close\_Serial  Close\_Ehternet  CLOSE  ConvertToShamsi  Test\_Connect  Set\_Time  Get\_Time  Set\_Date  Get\_Date  Collect\_Record |
| V1.1 | 25/10/1391 | در این ورژن دو تابع برای انتقال پرسنل از دستگاه ها به کامپیوتر و بالعکس اضافه شده اند .  لیست توابع عبارتند از :  Get\_UserInfo  Enroll\_User |

1. مقدمه

با توجه به نیاز ارتباط دستگاه با سایر نرم افزارها وهمچنین یکپارچگی برنامه این DLL نوشته شده است . این DLL به زبان C++ است که علاوه بر حجم بسیار پایین و سادگی استفاده ، انعطاف زیادی برای استفاده در سایر زبانهای برنلمه نویسی ( شامل win32 یا .net) را فراهم می کند.

در این SDK یک مثال ساده و کاربردی به زبان C# و نحوه Import واستفاده از DLL موجود می باشد که با توجه به آن نیاز برنامه نویس به نحوه استفاده از آن کاملا بر طرف می گردد.

* 1. توضیحات و نحوه استفاده از DLL

نحوه استفاده از این DLL کاملا استاندارد و مطابق بر DLL C++ می باشد که بعد از کپی کردن DLL در شاخه پروژه و Import کردن آن و تعرف تابع مورد نظر آنرا در برنامه استفاده می کنیم .

مثال در C#

[System.Runtime.InteropServices.DllImport("GP7-dll.dll")]

private static extern int Open\_Serial(IntPtr Port, int Buadrate

, ref int Handle);

1. توابع

در این قسمت به بررسی تمام توابع DLL و پارامترهای ورودی و خروجی و مقدار بازگشتی آنها می پردازیم .

---------------------------------------------------------------------------------------------------

int Open\_Serial(char\* Port, int Buadrate, int& Handle)

یک ارتباط با دستگاه از طریق پورت سریال ایجاد می کند.

پارامترها :

Port : نام پورت مورد نظر برای ارتباط با دستگاه ( مثلا COM1 )

Buadrate : سرعت (باود) پورت سریال که برابر با 57600 می باشد.

Handle : یک اشاره گر به کانال ارتباطی با دستگاه بر می گرداند که مشخص می کند از چه طریقی به دستگاه وصل هستیم (سریال یا شبکه)

مقدار بازگشتی :

0 : OK

---------------------------------------------------------------------------------------------------

int Open\_Ehternet(char\* IP, int Port, int& Handle)

یک ارتباط با دستگاه از طریق شبکه ایجاد می کند.

پارامترها :

IP : IP دستگاه مورد نظر

Port : port دستگاه مورد نظر

Handle : یک اشاره گر به کانال ارتباطی با دستگاه بر می گرداند که مشخص می کند از چه طریقی به دستگاه وصل هستیم (سریال یا شبکه)

مقدار بازگشتی :

0 : OK

---------------------------------------------------------------------------------------------------

void Close\_Serial(int Handle)

پورت سریال را می بندد.

---------------------------------------------------------------------------------------------------

void Close\_Ehternet(int Handle)

پورت شبکه را می بندد.

---------------------------------------------------------------------------------------------------

void CLOSE(int Handle)

با توجه به Handle پورت سریال یا شبکه را می بندد.

---------------------------------------------------------------------------------------------------

void ConvertToShamsi (char\* Date)

تبدیل تاریخ میلادی به شمسی

پارامترها :

Date : تاریخ میلادی با فرمت "روز / ماه / سال" را گرفته و به شمسی با همان فرمت تبدیل می کند.

---------------------------------------------------------------------------------------------------

bool Test\_Connect(int Handle, int DeviceID)

چک می کند آیا ارتباط با دستگاه برقرار است یا خیر ؟

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

---------------------------------------------------------------------------------------------------

bool Set\_Time(int Handle, char\* Time, int DeviceID)

زمان را تنظیم می کند.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

Time : زمان را با فرمت " ثانیه : دقیقه : ساعت" گرفته و در دستگاه ست می کند.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

مقدار بازگشتی :

true : OK

---------------------------------------------------------------------------------------------------

bool Get\_Time(int Handle, char\* Time, int DeviceID)

زمان را از دستگاه دریافت می کند.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

Time : زمان با فرمت " ثانیه : دقیقه : ساعت" را از دستگاه می گیرد.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

مقدار بازگشتی :

true : OK

---------------------------------------------------------------------------------------------------

bool Set\_Date(int Handle, char\* Date, int DeviceID)

تاریخ را تنظیم می کند.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

Date : زمان را با فرمت "روز / ماه / سال " گرفته و در دستگاه ست می کند.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

مقدار بازگشتی :

true : OK

---------------------------------------------------------------------------------------------------

bool Get\_Date(int Handle, char\* Date, int DeviceID)

تاریخ را از دستگاه دریافت می کند.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

Date : زمان را با فرمت "روز / ماه / سال را از دستگاه می گیرد.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

مقدار بازگشتی :

true : OK

---------------------------------------------------------------------------------------------------

int Collect\_Record(int Handle, char\* fromDate, char\* toDate, char\* fromTime, char\* toTime, int DeviceID, TrafficRecord\* TF)

رکوردها را در یک بازه زمانی از دستگاه جمع آوری می کند.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

fromDate : از تاریخ مورد نظر

toDate : تا تاریخ مورد نظر

fromTime : از زمان مورد نظر

toTime : تا زمان مورد نظر

DeviceID : سریال دستگاه مورد نظر را می گیرد.

TrafficRecord : آرایه ای از structure زیر را که شامل ترددهای پرسنل می باشد را برمیگرداند.

struct TrafficRecord

{

DWORD UserID;

BYTE PIS;

BYTE PC;

BYTE DateTime[7];

};

* PIS( Personal Identify Scenario ) : نوع تردد پرسنل را مشخص می کند ( کارت یا اثر انگشت یا ...)

enum PersonalIdentifyScenario

{

None = 0,

CardOrFinger,

FingerOnly,

PersonalNumberOrCardAndFinger,

CardAndFinger,

PersonalNumberAndFinger,

DeviceMode

};

* PC( Pre Card ) : پیش کارت رکورد را مشخص می کند ( عادی ، ورود ، خروج ، مرخصی ، ... )
* DateTime :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ثانیه | دقیقه | ساعت | روز | ماه | سال(byte) | 8<< سال |

مقدار بازگشتی :

1- : خطا

Else : تعداد رکورهای خوانده شده را یر گرداند.

---------------------------------------------------------------------------------------------------

int Get\_UserInfo(int Handle, int DeviceID, PersonInformation &PI,

FingerTemplate\* FingerPrintTemplate);

کلیه اطلاعات مربوط به یک پرسنل و اثر انگشتهایش را بر میگرداند.

تذکر : در هنگام صدا کردن این تابع باید شماره پرسنلی کاربر مورد نظر را درون PI.PersonalNumber قرار دهید.

پارامترها :

Handle : با توجه به این اشاره گر با پورتهای شبکه یا سریال ارتباط برقرار می کند.

DeviceID : سریال دستگاه مورد نظر را می گیرد.

PersonInformation: آرایه ای از structure زیر را که شامل اطلاعات یک پرسنل می باشد را برمیگرداند.

struct PersonInformation

{

DWORD PersonNumber;

DWORD Password;

DWORD FingerID;

DWORD CardID;

BYTE PIS;

BYTE PAR;

BYTE FingerTemplateNumber;

};

* PIS( Personal Identify Scenario ) : توضیحات در تابع Collect\_Record
* PAR( Personal Access Right ) : نوع دسترسی پرسنل را مشخص می کند(کاربر عادی یا خاص یا مدیر)

enum PesonalAccessRight

{

None = 0,

Administrator,

SpecialUser,

User

};

* FingerTemplateNumber : تعداد اثرانگشتهای شخص را مشخص می کند.

FingerTemplate : آرایه ای از structure زیر را که اثر انگشتهای یک پرسنل می باشد را برمیگرداند.

struct FingerTemplate

{

BYTE Template[512];

};

مقدار بازگشتی :

1. : خطا

Else : تعداد اثر انگشت

---------------------------------------------------------------------------------------------------

CommunicationFlag Enroll\_User(int Handle, int DeviceID, PersonInformation PI,

FingerTemplate\* FingerPrintTemplate, int NumOfFPtemplate);

پرسنل با مشخصات تعیین شده را در دستگاه ثبت می کند.

پارامترها :

NumOfFPtemplate : تعداد اثر انگشتهای کاربر را می گیرد.

بقیه پارامترها مانند تابع Get\_UserInfo می باشند.

مقدار بازگشتی :

یک مقدار از خطاهایی که ممکن است در هنگام ثبت کاربر روی دهد را با توجه به لیست شمارشی زیر برمی گرداند.

enum CommunicationFlag

{

None = 0,

PacketError = 2,

ProccessSuccess = 3,

CommunicatonFail = 4,

CommunicationError = 7,

CardIsRepetitive = 9,

FingerIsRepetitive = 10,

PersonIsRepetitive = 11

};