1. Basic Device Protocol

CID Easy is a standart Usb/Hid (Human interface device) device . The programmers can use any standart HID libraries or OS HID services to communicate with device easily.

The device has a simple custom protocol. Application softwares must handle this protocol for communicating with the device. This protocol is related to manage of calls which stored in device memory. (Device stores raw (unparsed) call informations in own memory/queue before send its to application program.)

This communication flows like this :

CID EASY DEVICE		APP. PROGRAM
“ENQ”	→	
	←	“ACK”
<DATA> (if any data exist)	→	
	←	“DCK” or “NAK”
<DATA>(if any data exist)	→	
	←	“DCK” or “NAK”
<DATA>(if any data exist)	→	
	←	“DCK” or “NAK”
<DATA>(if any data exist)	→	
	←	“DCK” or “NAK”
“ETB”	→	
	←	“ACK”

If there are no call data in device memory this handshake flows like this :

CID EASY DEVICE		APP. PROGRAM
“ENQ”	→	
	←	“ACK”
“ETB”	→	
	←	“ACK”

The device send ENQuery to application program periodicaly (about every 6-8 second). And the app program must be response this enquery with ACK (positive acknownledge)

If any package waiting in memory/queue, the device send first packege after first ACK from application. If application program receive the data-package must be answer this package with DCK (data acknownledge) or NAK. (negative acknownledge)

DCK cause clearing package from device memory. If there are any problem in package you can answer this query with NAK. In this case device send same package again while receiving DCK.

This steps repeated for all packages stored in device memory

When the data sequence complated (all data sended waiting in queue) device send ETB (End of Transmission Block) to appliaction. Application program must answer this with ACK also.

If there are no call info, device immediatly send ETB after receiving ACK for ENQ

Note 1.: All strings must be terminated vith CR+LF / #13#10
Note 2.: ENQ decleration from device includes firmware version info also. Example : ENQ-v121045
Note 3.: When the new call occoured, device send a new ENQ to application program immediately without waiting for the timing.
Note 4 : You can analyze this protocol using any standart hid protocol analyzer.

The followed VB6 sample can describes this method. This sample based on MCHID.DLL standart hid dll library.

https://www.cideasy.com.tr/download/MCHIDSamples.zip

This is basic and low-level way to manage the devices. But of course this way has some handicaps :

1. This way is bounded to the firmware version. Protocols can be change depending on new firmwares in future. So you must support all of avaible protocols in your application programs. (Device protocol number sended with ENQ string)
2. General purpose DLL's (MCHID.DLL etc.) generaly supports one device at time. We product 1,2 and 4 line models of devices. If you want more then 4 lines, you must use more than 1 device. But general purpose HID dll's can't support this.
3. In this way you receice raw-cli data from phone lines. Your program must be parse the data to generate pure call info. And there are more than one package type in phone cli standart. Generaly the telecom companies use more then one standart in their network circuits.

We build a .NET dll for eleminate this problems. You can use this DLL directly in .NET environment, Or -advanced than this- can use our HOST application based on same .NET dll to connect to devices, named as EasyHost.

By this way your application programs is fully-isolated from device and version managament requirements.

2. Using Native .NET (CEHIDLibrary.DLL) library (for Windows programmers)

CEHidLibrary is C# (.NET) DLL library for Windows application programmers.

This library manage connected device(s) (up to 8 devices). Also It can manage diffrerent versions of devices simultaneously. And also parse all type in CLI datas in common package types and gives pure call informations to application programs.

You can download C# sample for this library from :

https://www.cideasy.com.tr/download/CEHIDLibSample.zip

3. Using EASY HOST (for Windows programmers)

Easy Host is a low weight .Net application based on CEHIDLibrary.DLL.

EasyHost is starting with Windows automaticly and starts manage attached devices to PC. Easy Host receives calls (stored or new arrived) from devices, parse the calls and store it standart data file. When the application program connect to EasyHost, program inform your application about stored call informations and new calls.

Easy HOST provide a Windows application socket for the application programs that can manage and access to call list.

Application programs can connect to EasyHost over Windows socket (1468 normaly, and can be changable by user) . Easy host has a small command set for manage call list and sharing it with client program.

Followed samples is about the connect and communucate to EasyHost over Socket in VB6 and Delphi

https://www.cideasy.com.tr/download/EasyHostClientDemos.zip

You can download EasyHost latest version with followed link

https://www.cideasy.com.tr/download/EasyHost2Setup-2.1.zip

Easy Host comamnd set is :

COMAMND	ANSWER	DESC.
HALO	HALO=<Halo Info>	Return Server Infos : Local IP, Remote IP, Server Version.
TIME	TIME TIME=<Server Time Info>	Get Server Time. Date and time info

		of PC server running on
STATUS	STATUS= [DEVICE PLUGGED] | [DEVICE UNPLUGGED]	Device status : Is any device plugged or not. Status answer automaticly generated when device remove and arrival
COUNT	COUNT=[<Calls Count>]	Calls count in list. Normaly application program must enquery this info when connect to Easy Host and after must receive all call-datas waiting
DATA <n>	DATA[<n>]=..........	Get n'th data from list. N is index of call in list
KILL <n>	KILL[<n>]=..........	Kill n'th data from list. N is index of call in list Application program must kill data from Easy Host after receive complate with DATA command
Special State	NEWCALL[<n>]=...	This is special state. EasyHost immediately generate NEWCALL response after any new call arrived from any device if any client currently connectted N is index of call in list

4. Using Java Class Library (For Android Programmers)

CidEasyBridge is android based open source java class for manage conneccted caller id devices on OTG port.

You can download sample application included this class by following link.

https://www.cideasy.com.tr/download/Android-CidEasyBridge.zip

Note: When device connected over OTG port you must move ADB connection over to TCPIP ports. The commands following allows you to do this

```
ADB TCPIP 5555
ADB CONNECT192.168.XXX.YYY:5555
```

Where the 5555 is Ip port number and 192.168.XXX.YYY is emulator device IP address.