HE853 Control Documentation

Thomas Ascher

May 22, 2012

Contents

1 Introduction

The HE853 Control project is dedicated to create a Windows SDK for the HE853 USB dongle of the Home Easy home automation products since the vendor does not provide one.

The SDK consists of GPL licensed tools and a LGPL licensed library that can be used from other applications.

Out target audience are power users and software developers who whish to extend the usage of the HE853 USB dongle beyond the limits of the vendor software.

Most information this project is based on was gathered through reverse engineering. So not all details of the used communiation protocol are known and documented.

For more information about the Home Easy home automation products please refer to the vendor page: http://www.elro.eu/en/products/category/home_automation/home_easy/zenders2/pc_afstandsbediening_usb_dongle

For more information and the latest packages please refer to the project page: http://he853control.sourceforge.net/

1.1 Prequesites

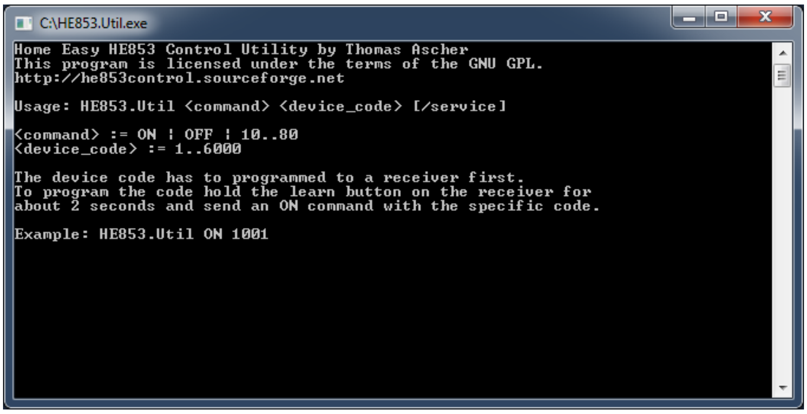
For installation, usage and development Microsoft .NET Framework 2.0 has to be installed on your system: http://go.microsoft.com/fwlink/?LinkId=131000

2 Tools

All tools are located in the installation directory and support the following commands:

- On: receiver will perform switch on
- Off: receiver will perform switch off
- Dim 10–80: reciver will perform dim to specific level if supported

2.1 Command Line Utility – HE853.Util.exe



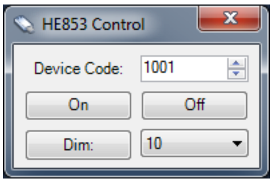
```
Home Easy HE853 Control Utility by Thomas Ascher
This program is licensed under the terms of the GNU GPL.
http://he853control.sourceforge.net

Usage: HE853.Util <command> <device_code> [/service]

<command> := ON | OFF | 10..80
<device_code> := 1..6000

The device code has to programmed to a receiver first.
To program the code hold the learn button on the receiver for
about 2 seconds and send an ON command with the specific code.

Example: HE853.Util ON 1001
```

Usage: HE853.Util <command><device_code>[/service]

<command>:= ON — OFF — 10..80 <device_code>:= 1..6000

Example: HE853.Util ON 1001

2.2 GUI Application – HE853.App.exe



Device Code: 1001

Dim: 10

Enter a specifi device code and click the buttons to send a specific command.

2.3 Coupling Dongle with Receivers

Before a receiver react to commands of the USB dongle a specific device code has to be programmed to the receiver first. A device code is a number between 1 and 6000 and can be used by multiple receivers.

To program a device code use the command line utility or the GUI application. Press the learn button on a receiver for about 1 second. Then send an ON command with the desired device code from the command line utility or the GUI application. The receiver will react to commands with this device code.

To clear the programmed device code press the learn button on a receiver util the LED on the receiver starts to flash.

2.4 Service – HE853.Service.exe

The service HE853.Service is for advanced users. It is meant for scenarios where multiple applications have to access the HE853 dongle concurrently. Both the utility and the GUI application support the */service* switch that makes them use the service instead of using the device directly.

2.4.1 Configuration

The service is configured to start manually per default. This behavior has to be changed in the service control panel. Configure the HE853 RPC Service to start automatically and or start it.

2.4.2 Manual Installation

If the service was not installed with the MSI installer is has to be registered with the Installer Tool (Installutil.exe).

3 Development

For development at least Microsoft .NET Framework 2.0 and or Visual Studio 2005 is required.

3.1 Using the Library – HE853.dll

The library can be used by any .NET language by referencing the HE853.dll assembly from the installation directory or via COM. Please note that the assembly is compiled as Any CPU and works with x86 and x64 platform configurations.

Visual Studio 2010 sample projects for the languages C++, C++/CLI, C# and Visual Basic are located in the installation folder in \Samples.

Please note that the library is licensed under the term of the LGPL. This means it is possible to use the library in commercial applications but code changes to the library have to be made available on distribution.

3.1.1 C# Sample Code

```
HE853.IDevice device = new HE853.Device();
if (device.Open())
{
    device.On(1001);
    device.Off(1001);
    device.Close();
}
```

3.2 Using the RPC Library – HE853.RPC.dll

The RPC library ist not designed for general usage yet and is not available via COM. It uses .NET Remoting IPC for communication. The library does only work in combination with the service application.

Use the folloging code in combination with the RPC library:

```
HE853.RPC.RegisterClient();
HE853.IDevice device = new HE853.Device();
if (device.Open())
{
    device.On(1001);
    device.Off(1001);
    device.Close();
}
```

3.3 Depolyment

The easiest method of depolyment is by installing the HE853 Control MSI setup on the target system. The installation performs all required steps:

- Installation of HE853.dll to the global assembly cache
- Registration of HE853.dll for COM usage
- Registration of HE853.Service.exe

If you deploy .NET applications you can simply install or copy the HE853.dll with your application as private assembly.

4 Support HE853 Control

You can support this project by testing, writing code or documentation and suggesting features and reporting bugs. For writing documentation a good knowledge of LaTeX is required. For writing code one should have knowledge of:

- Visual Studio 2010 (required for development)
- C#
- PInvoke and Windows API
- HID Devices
- Unit Tests

4.1 TODOs

- Correct spelling errors in documentation
- Improve user documentation
- Improve source documentation (at least for public interfaces)
- Better input validation in program
- Rewrite code to use exceptions instead of return values
- Refactor code so it is more suiteable for test driven development
- Write more unit tests
- Add installation path to PATH variable during installation
- Make Command classes internal again, exposed now for unit tests
- Test on more 32 and 64 bit platforms: XP, Vista, Windows 7, Windows 8
- Implement GUI as custom control
- Improve GUI application (help integration, layout, etc.)
- Maybe translate the package into other languages