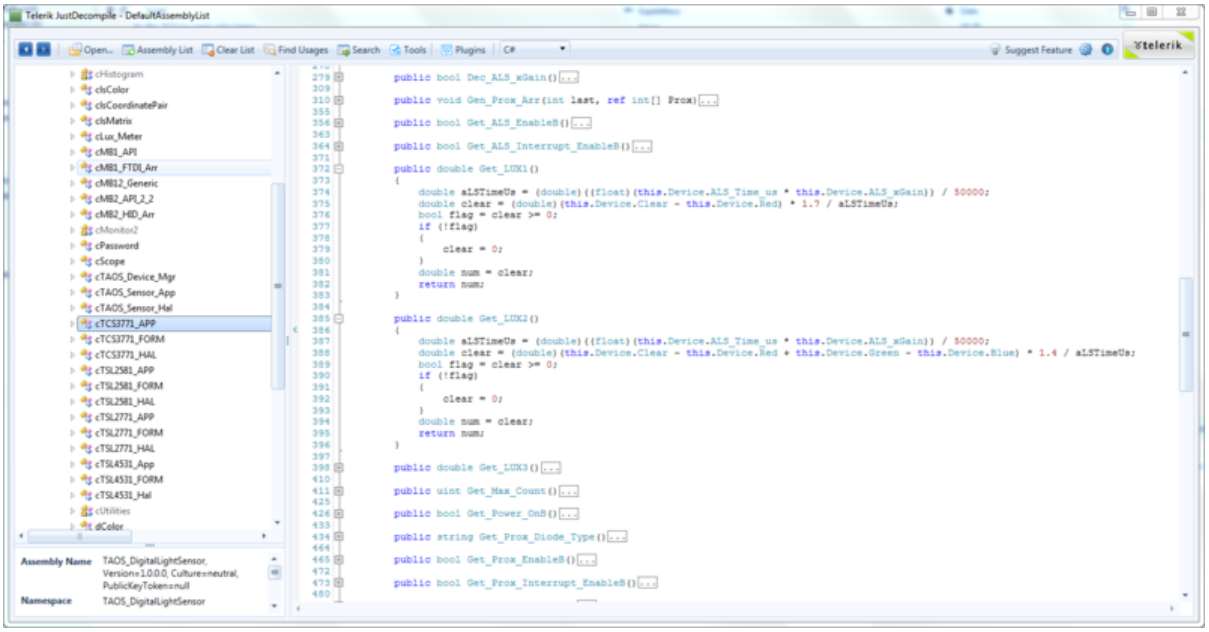
Decompiling .NET Apps

Created by Kevin Townsend

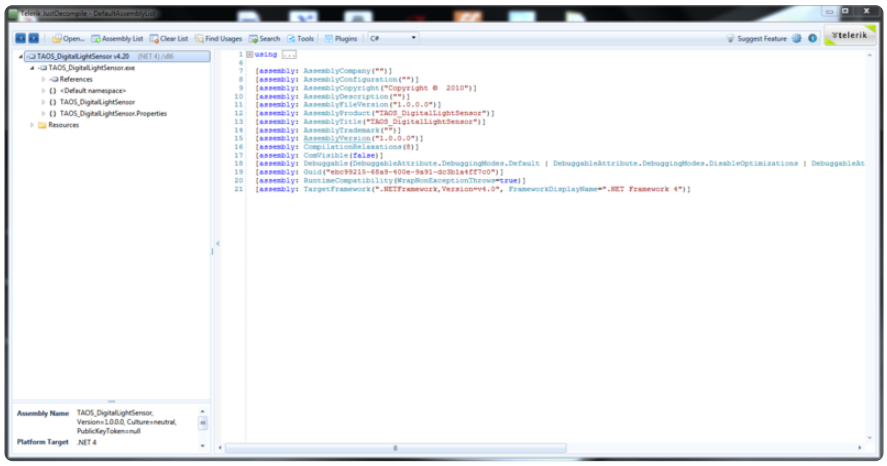


https://learn.adafruit.com/decompiling-net-apps

Last updated on 2024-06-03 01:17:45 PM EDT

Table of Contents

Introduction



I recently had a problem was a new color sensor I was working on: the datasheet was pretty spartan, and there was no information on converting the raw sensor values to something more useful like 'lux' or 'color temperature'.

What I did have was a Windows app for the official evaluation kit that gave me some of the values I was looking for.

I didn't have the eval kit, but it meant that the magic numbers had at least been worked out by someone in the company ... they just didn't make it outside the corporate firewall yet (officially).

Enter the wonderful world of .Net.

If you have a recent Windows based app from a a silicon vendor, there's about 9 chances in 10 it was written in either VB.Net or C#.Net. Thankfully, it's trivial to decompile .Net applications, and you might be able to find the magic numbers or formulas you need in the source code for the application.

Read on to see how easy it is to get from binary to C#, and fill in those blanks in your datasheet ...

Decompiling .Net Binaries

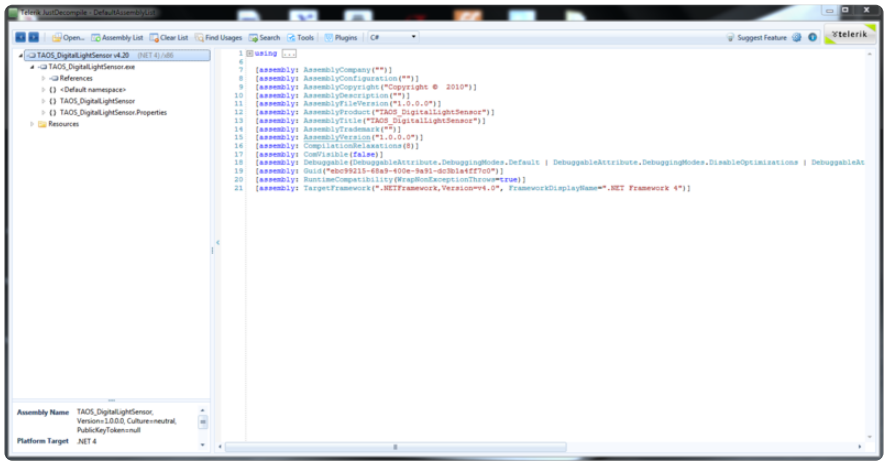
I haven't done any serious C# or .Net development in 3-4 years, but when I did one of the tools I used the most was a free app called .Net Reflector. You could use it to decompile unobfuscated .Net apps, but more importantly you could use it to decompile the core .Net libraries themselves for a direct view of what was going on beneath the surface.

My first thought looking at this demo app from the silicon vendor was to download Reflector ... but alas, Reflector seems to have gone commercial since I last used it. I don't have a problem paying for tools (everyone needs to eat) ... but in this case it's not something I'd use more than once, and so I did a quick search for free alternatives.

The two main free replacements seem to be ILSpy (https://adafru.it/c4l) and Telerik's JustDecompile (https://adafru.it/c4m). They'll both do the job, but I went with JustDecompile simply because I've used Telerik products before.

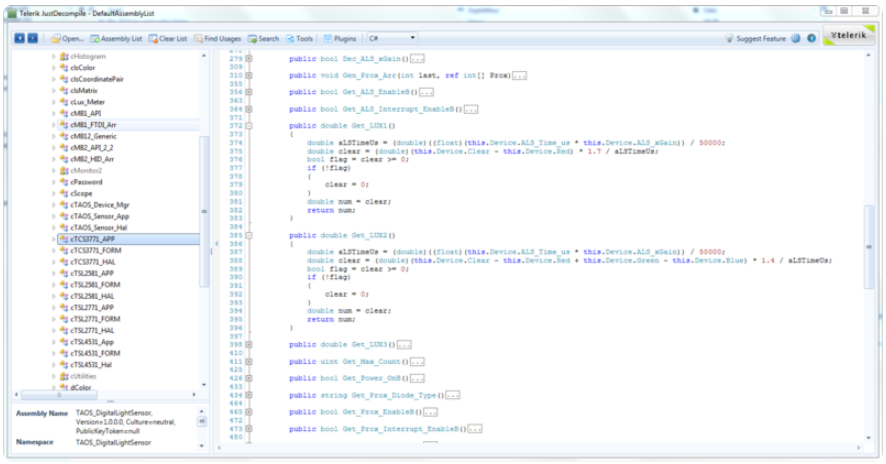
Decompiling Your .Net App with JustDecompile

After downloading and installing JustDecompile, you just need to point to your .exe (or perhaps your .dll), and if it was written using .Net (and isn't obfuscated), you should be able to see some details from the binary and browse the IL code:



From here, you simply need to start exploring the binary, and the tools will decompile the IL code in the binaries back to your .Net language of choice (most tools support at least C# and Visual Basic.Net). The Telerik tool also has a 'search' tool where you can search for types, or do free text searches (for values like 'Lux' for example).

A few seconds of digging, and I came up with this for example:



From here, you should be able to convert the code back to C or whatever other language or platform you're using, and you'll have a working program in no time ... or at least you should get a nudge in the right direction. In any case, it sure beats waiting for your 'bottle tossed into the the ocean' support request to come through!