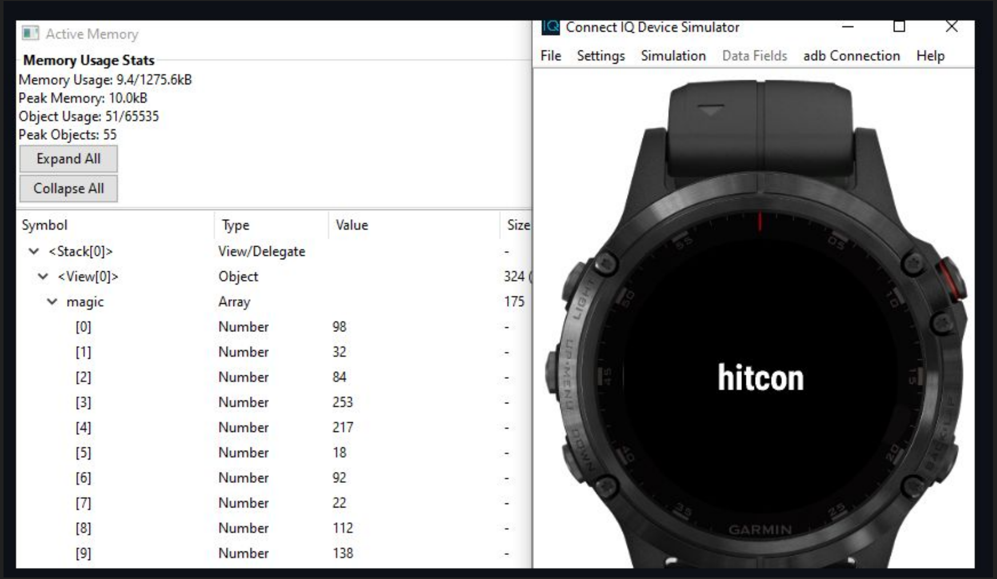
Unlimited Power

CTF Challenge by TheLuka

Ozadje

hitcon

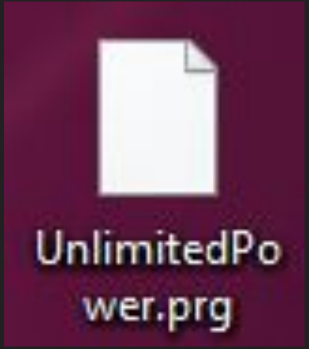


Memory Usage Stats
Memory Usage: 9.4/1275.6kB
Peak Memory: 10.0kB
Object Usage: 51/65535
Peak Objects: 55

Symbol	Type	Value	Size
<Stack[0]>	View/Delegate		-
<View[0]>	Object		324
magic	Array		175
[0]	Number	98	-
[1]	Number	32	-
[2]	Number	84	-
[3]	Number	253	-
[4]	Number	217	-
[5]	Number	18	-
[6]	Number	92	-
[7]	Number	22	-
[8]	Number	112	-
[9]	Number	138	-

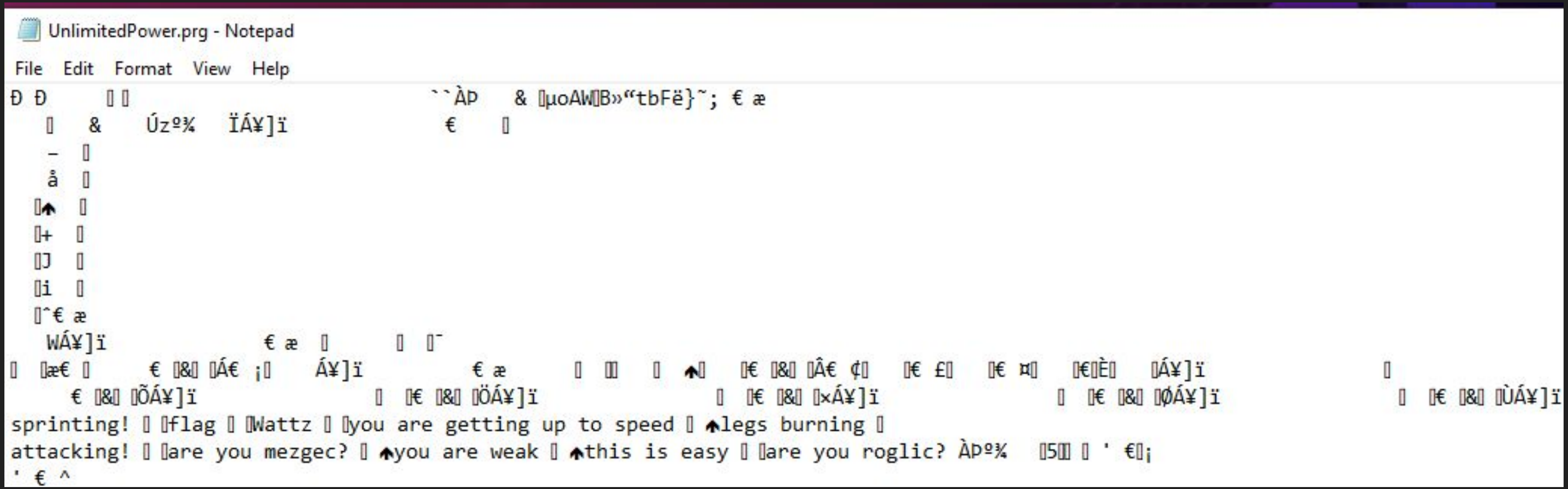
1.del

hint: Hack my watts



2.del

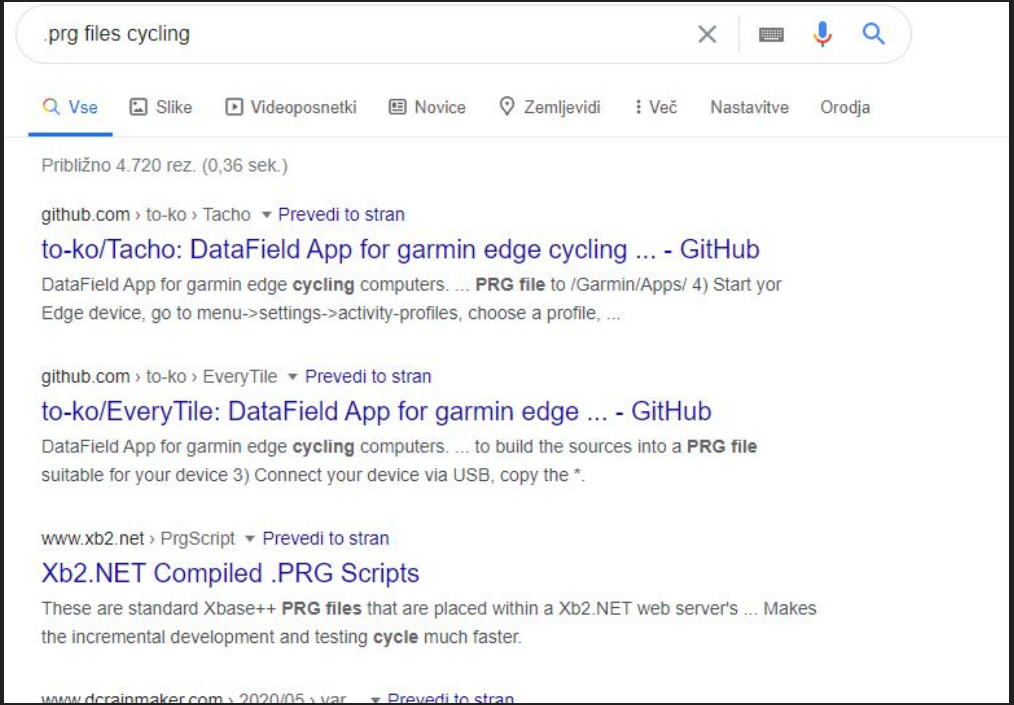
kaj je .prg?



UnlimitedPower.prg - Notepad

```
Ð Ð     ▯ ▯                            ``ÀÞ   & ▯µoAW▯B»“tbFë}˜; € æ
   ▯   &    Úzº¾   ÏÁ¥]ï              €    ▯
   –  ▯
   å  ▯
  ▯🡅   ▯
  ▯+   ▯
  ▯J   ▯
  ▯i   ▯
  ▯ˆ€ æ
   WÁ¥]ï              € æ  ▯      ▯  ▯¯
▯  ▯æ€ ▯      € ▯&▯ ▯Á€ ¡▯    Á¥]ï          € æ      ▯  ▯▯   ▯  🡅▯    ▯€ ▯&▯ ▯Â€ ¢▯   ▯€ £▯   ▯€ ¤▯   ▯€▯È▯   ▯Á¥]ï               ▯
     € ▯&▯ ▯ÕÁ¥]ï                ▯  ▯€ ▯&▯ ▯ÖÁ¥]ï                 ▯  ▯€ ▯&▯ ▯×Á¥]ï                 ▯  ▯€ ▯&▯ ▯ØÁ¥]ï                 ▯  ▯€ ▯&▯ ▯ÙÁ¥]ï
sprinting! ▯ ▯flag ▯ ▯Wattz ▯ ▯you are getting up to speed ▯ 🡅legs burning ▯
attacking! ▯ ▯are you mezgec? ▯ 🡅you are weak ▯ 🡅this is easy ▯ ▯are you roglic? ÀÞº¾   ▯5▯▯ ▯ ' €▯¡
' € ^
```

3.del Google



.prg files cycling

Vse · Slike · Videoposnetki · Novice · Zemljevidi · Več · Nastavitve · Orodja

Približno 4.720 rez. (0,36 sek.)

github.com › to-ko › Tacho ▾ Prevedi to stran

to-ko/Tacho: DataField App for garmin edge cycling ... - GitHub

DataField App for garmin edge **cycling** computers. ... **PRG file** to /Garmin/Apps/ 4) Start yor Edge device, go to menu->settings->activity-profiles, choose a profile, ...

github.com › to-ko › EveryTile ▾ Prevedi to stran

to-ko/EveryTile: DataField App for garmin edge ... - GitHub

DataField App for garmin edge **cycling** computers. ... to build the sources into a **PRG file** suitable for your device 3) Connect your device via USB, copy the *.

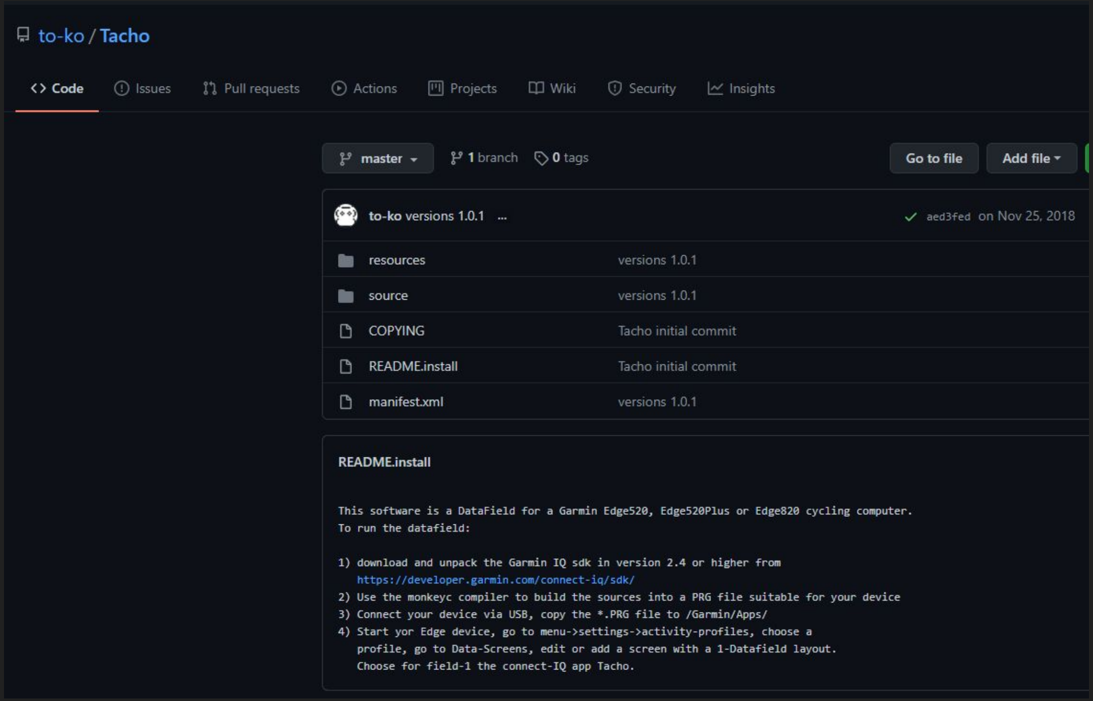
www.xb2.net › PrgScript ▾ Prevedi to stran

Xb2.NET Compiled .PRG Scripts

These are standard Xbase++ **PRG files** that are placed within a Xb2.NET web server's ... Makes the incremental development and testing **cycle** much faster.

www.dcrainmaker.com › 2020/05 › var... ▾ Prevedi to stran

4.del Garmin SDK



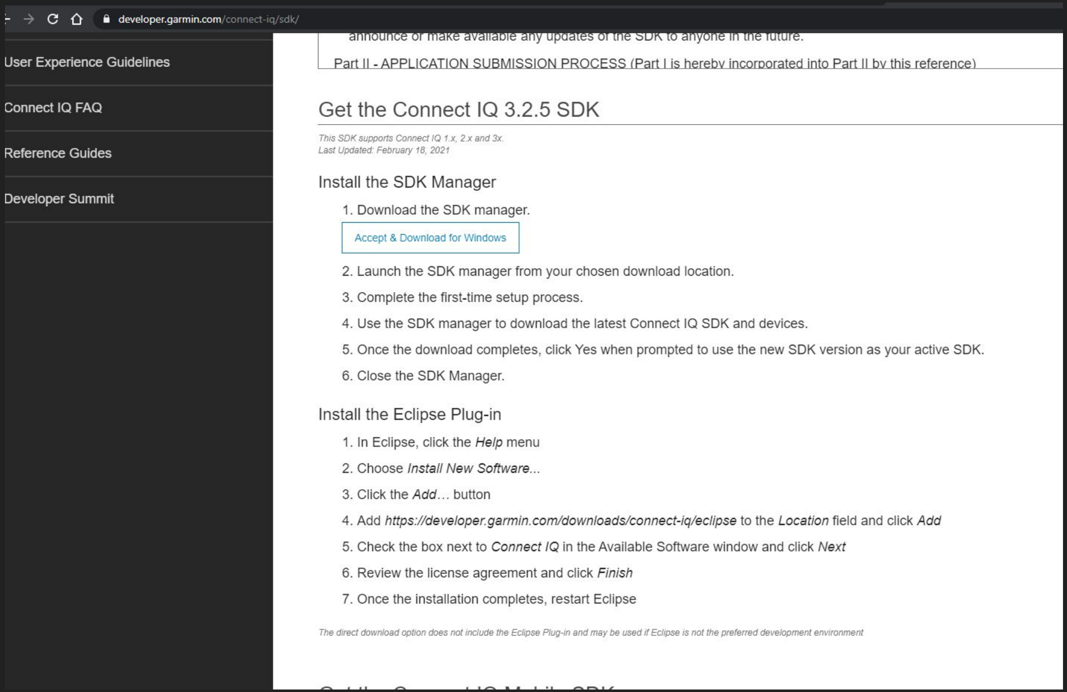
to-ko versions 1.0.1 ... aed3fed on Nov 25, 2018

resources	versions 1.0.1
source	versions 1.0.1
COPYING	Tacho initial commit
README.install	Tacho initial commit
manifest.xml	versions 1.0.1

README.install

```
This software is a DataField for a Garmin Edge520, Edge520Plus or Edge820 cycling computer.
To run the datafield:

1) download and unpack the Garmin IQ sdk in version 2.4 or higher from
   https://developer.garmin.com/connect-iq/sdk/
2) Use the monkeyc compiler to build the sources into a PRG file suitable for your device
3) Connect your device via USB, copy the *.PRG file to /Garmin/Apps/
4) Start yor Edge device, go to menu->settings->activity-profiles, choose a
   profile, go to Data-Screens, edit or add a screen with a 1-Datafield layout.
   Choose for field-1 the connect-IQ app Tacho.
```

5.del setup garmin SDK


developer.garmin.com/connect-iq/sdk/
User Experience Guidelines
Connect IQ FAQ
Reference Guides
Developer Summit

announce or make available any updates of the SDK to anyone in the future.

Part II - APPLICATION SUBMISSION PROCESS (Part I is hereby incorporated into Part II by this reference)

Get the Connect IQ 3.2.5 SDK

This SDK supports Connect IQ 1.x, 2.x and 3x.
Last Updated: February 18, 2021

Install the SDK Manager

1. Download the SDK manager.

Accept & Download for Windows

2. Launch the SDK manager from your chosen download location.
3. Complete the first-time setup process.
4. Use the SDK manager to download the latest Connect IQ SDK and devices.
5. Once the download completes, click Yes when prompted to use the new SDK version as your active SDK.
6. Close the SDK Manager.

Install the Eclipse Plug-in

1. In Eclipse, click the *Help* menu
2. Choose *Install New Software...*
3. Click the *Add…* button
4. Add *https://developer.garmin.com/downloads/connect-iq/eclipse* to the *Location* field and click *Add*
5. Check the box next to *Connect IQ* in the Available Software window and click *Next*
6. Review the license agreement and click *Finish*
7. Once the installation completes, restart Eclipse

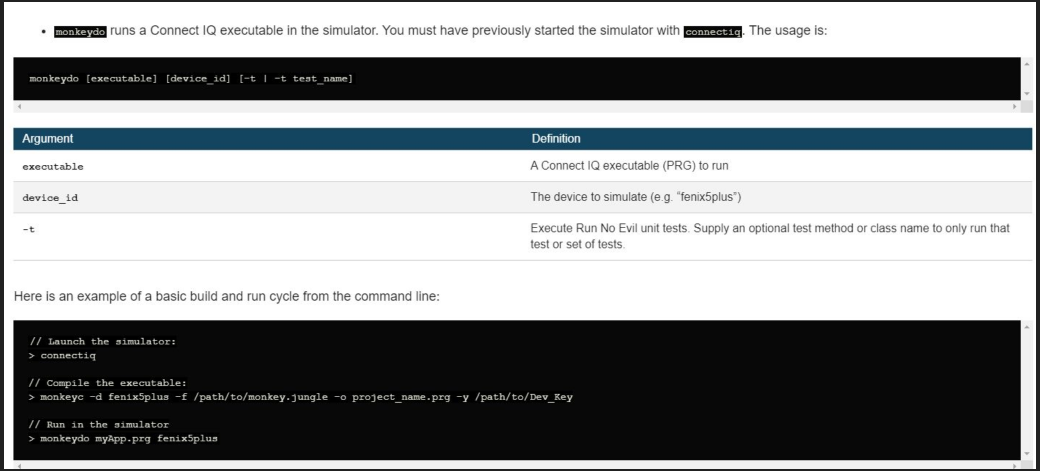
The direct download option does not include the Eclipse Plug-in and may be used if Eclipse is not the preferred development environment

6.del running the simulator

connectiq

monkeydo UnlimitedPower.prg edge_520

C:\Users\lukad\AppData\Roaming\Garmin\ConnectIQ\Sdks\connectiq-sdk-win-3.2.5-2021-02-12-6d31f4357\bin



- `monkeydo` runs a Connect IQ executable in the simulator. You must have previously started the simulator with `connectiq`. The usage is:

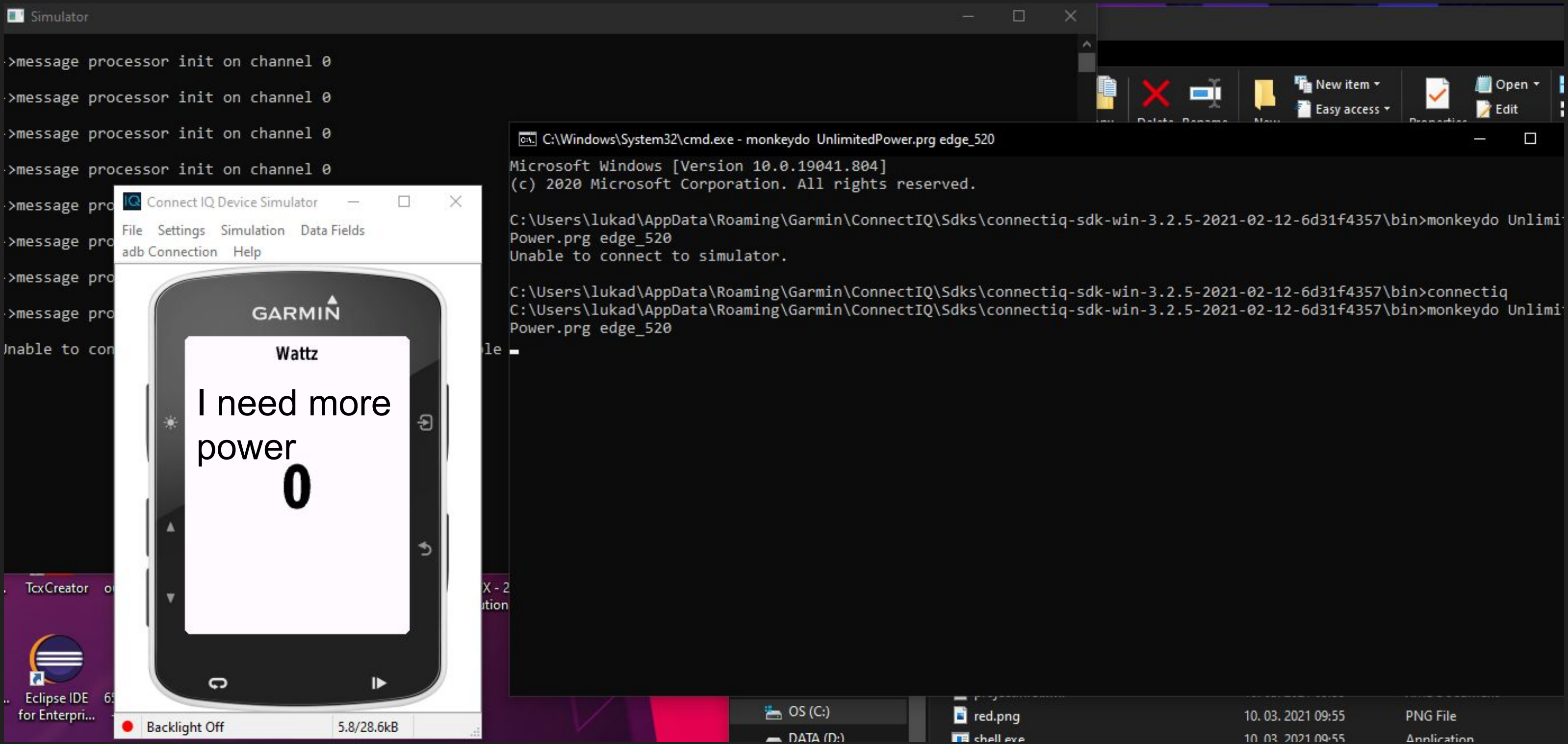
```
monkeydo [executable] [device_id] [-t | -t test_name]
```

Argument	Definition
`executable`	A Connect IQ executable (PRG) to run
`device_id`	The device to simulate (e.g. "fenix5plus")
`-t`	Execute Run No Evil unit tests. Supply an optional test method or class name to only run that test or set of tests.

Here is an example of a basic build and run cycle from the command line:

```
// Launch the simulator:
> connectiq

// Compile the executable:
> monkeyc -d fenix5plus -f /path/to/monkey.jungle -o project_name.prg -y /path/to/Dev_Key

// Run in the simulator
> monkeydo myApp.prg fenix5plus
```

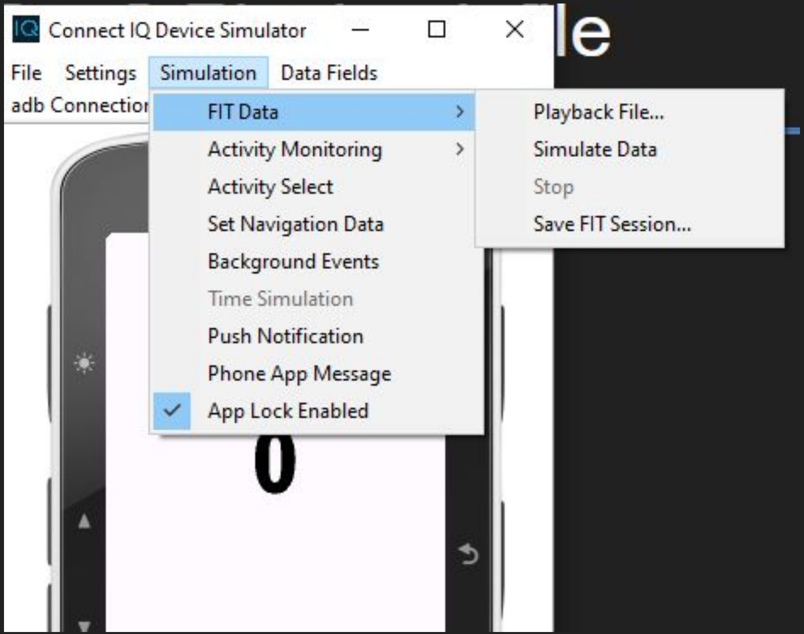
7.Run the simulation



```
Simulator
>message processor init on channel 0
>message processor init on channel 0
>message processor init on channel 0
>message processor init on channel 0
>message pro
>message pro
>message pro
>message pro
Unable to con
```

```
C:\Windows\System32\cmd.exe - monkeydo  UnlimitedPower.prg edge_520
Microsoft Windows [Version 10.0.19041.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\lukad\AppData\Roaming\Garmin\ConnectIQ\Sdks\connectiq-sdk-win-3.2.5-2021-02-12-6d31f4357\bin>monkeydo Unlimi
Power.prg edge_520
Unable to connect to simulator.

C:\Users\lukad\AppData\Roaming\Garmin\ConnectIQ\Sdks\connectiq-sdk-win-3.2.5-2021-02-12-6d31f4357\bin>connectiq
C:\Users\lukad\AppData\Roaming\Garmin\ConnectIQ\Sdks\connectiq-sdk-win-3.2.5-2021-02-12-6d31f4357\bin>monkeydo Unlimi
Power.prg edge_520
```

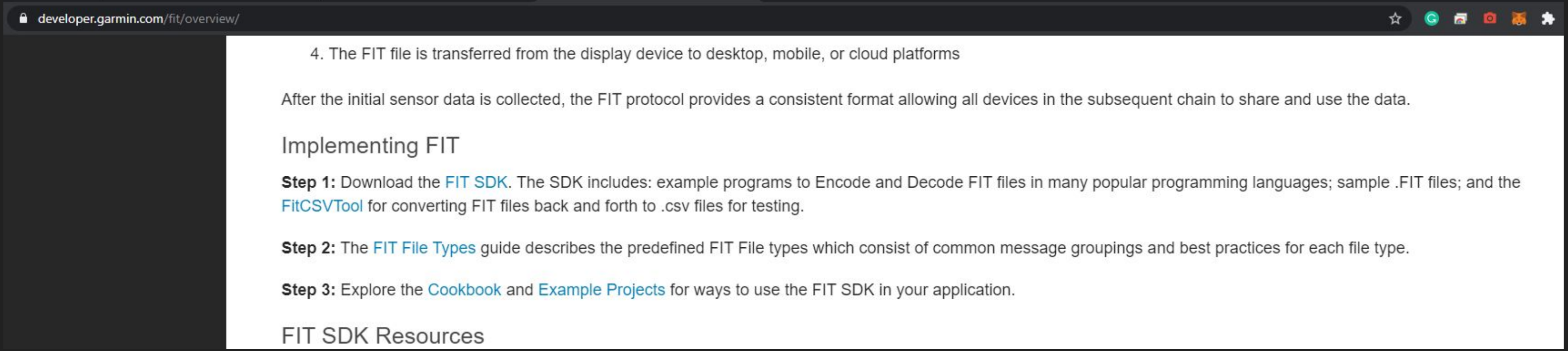
8.Playback file



9. Pridobivanje fit file

Lahko uporabiš tudi svojega za testiranje

Cilj je dobiti nadzor nad Power Data



developer.garmin.com/fit/overview/

4. The FIT file is transferred from the display device to desktop, mobile, or cloud platforms

After the initial sensor data is collected, the FIT protocol provides a consistent format allowing all devices in the subsequent chain to share and use the data.

Implementing FIT

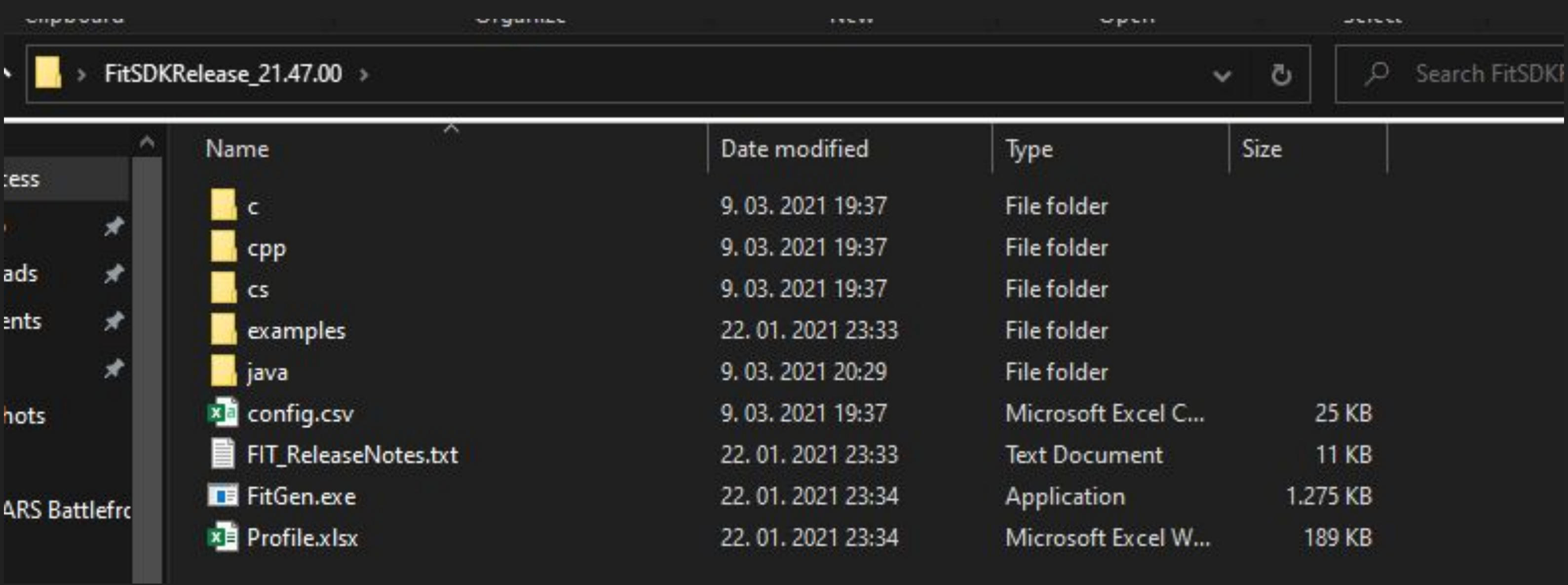
Step 1: Download the FIT SDK. The SDK includes: example programs to Encode and Decode FIT files in many popular programming languages; sample .FIT files; and the FitCSVTool for converting FIT files back and forth to .csv files for testing.

Step 2: The FIT File Types guide describes the predefined FIT File types which consist of common message groupings and best practices for each file type.

Step 3: Explore the Cookbook and Example Projects for ways to use the FIT SDK in your application.

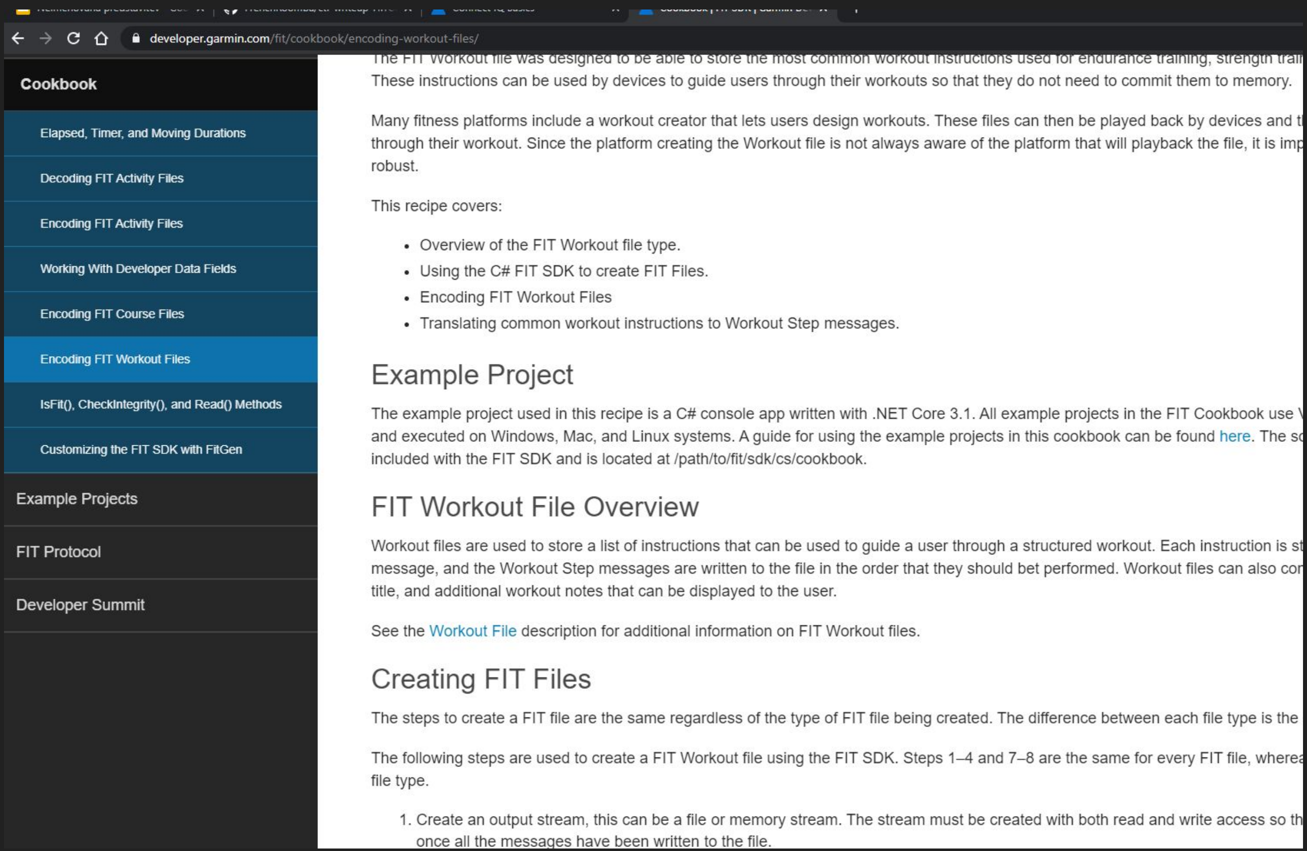
FIT SDK Resources

10. Examples



Name	Date modified	Type	Size
c	9. 03. 2021 19:37	File folder	
cpp	9. 03. 2021 19:37	File folder	
cs	9. 03. 2021 19:37	File folder	
examples	22. 01. 2021 23:33	File folder	
java	9. 03. 2021 20:29	File folder	
config.csv	9. 03. 2021 19:37	Microsoft Excel C...	25 KB
FIT_ReleaseNotes.txt	22. 01. 2021 23:33	Text Document	11 KB
FitGen.exe	22. 01. 2021 23:34	Application	1.275 KB
Profile.xlsx	22. 01. 2021 23:34	Microsoft Excel W...	189 KB

11.Got em


developer.garmin.com/fit/cookbook/encoding-workout-files/

Cookbook

- Elapsed, Timer, and Moving Durations
- Decoding FIT Activity Files
- Encoding FIT Activity Files
- Working With Developer Data Fields
- Encoding FIT Course Files
- Encoding FIT Workout Files
- IsFit(), CheckIntegrity(), and Read() Methods
- Customizing the FIT SDK with FitGen

Example Projects

FIT Protocol

Developer Summit

The FIT Workout file was designed to be able to store the most common workout instructions used for endurance training, strength trai
These instructions can be used by devices to guide users through their workouts so that they do not need to commit them to memory.

Many fitness platforms include a workout creator that lets users design workouts. These files can then be played back by devices and t
through their workout. Since the platform creating the Workout file is not always aware of the platform that will playback the file, it is imp
robust.

This recipe covers:

- Overview of the FIT Workout file type.
- Using the C# FIT SDK to create FIT Files.
- Encoding FIT Workout Files
- Translating common workout instructions to Workout Step messages.

Example Project

The example project used in this recipe is a C# console app written with .NET Core 3.1. All example projects in the FIT Cookbook use \
and executed on Windows, Mac, and Linux systems. A guide for using the example projects in this cookbook can be found here. The sc
included with the FIT SDK and is located at /path/to/fit/sdk/cs/cookbook.

FIT Workout File Overview

Workout files are used to store a list of instructions that can be used to guide a user through a structured workout. Each instruction is st
message, and the Workout Step messages are written to the file in the order that they should bet performed. Workout files can also cor
title, and additional workout notes that can be displayed to the user.

See the Workout File description for additional information on FIT Workout files.

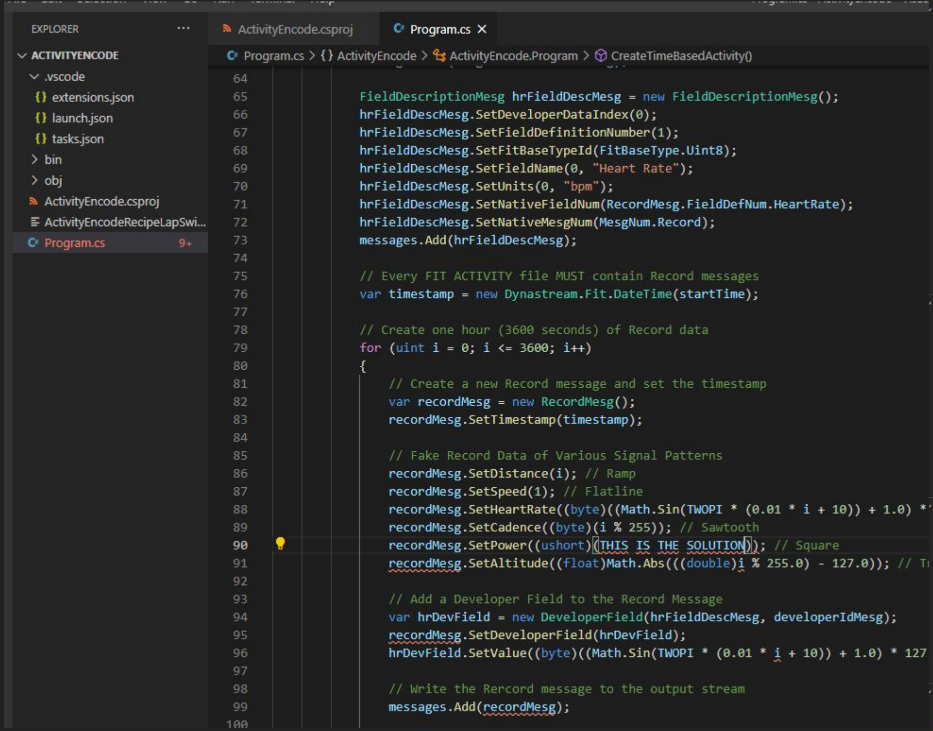
Creating FIT Files

The steps to create a FIT file are the same regardless of the type of FIT file being created. The difference between each file type is the

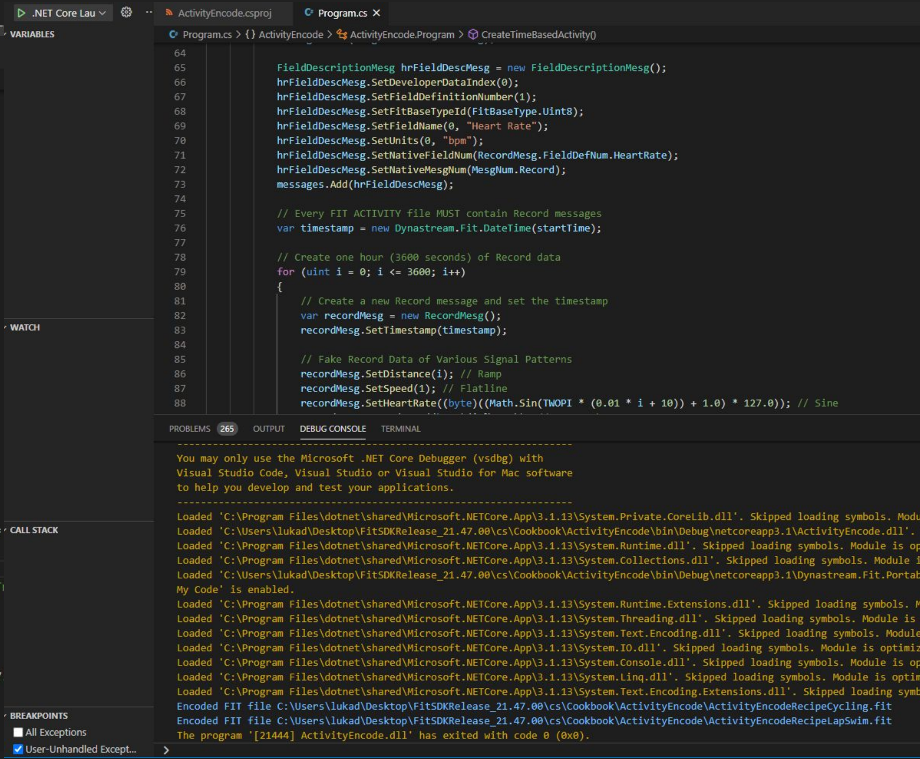
The following steps are used to create a FIT Workout file using the FIT SDK. Steps 1–4 and 7–8 are the same for every FIT file, wherea
file type.

1. Create an output stream, this can be a file or memory stream. The stream must be created with both read and write access so th
   once all the messages have been written to the file.

12. Using the code



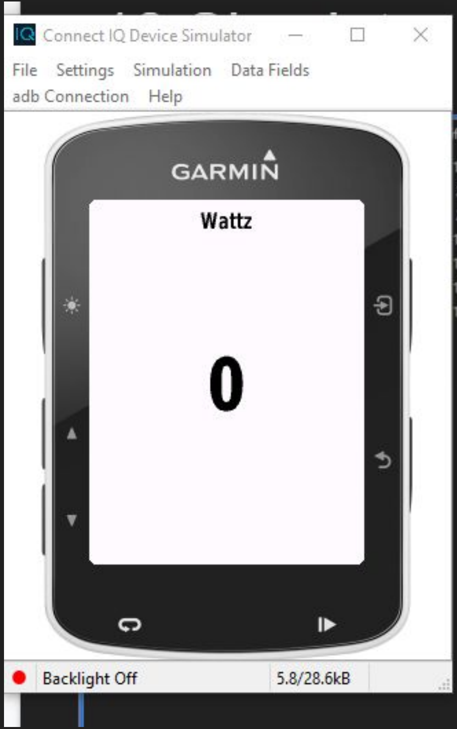
```
            FieldDescriptionMesg hrFieldDescMesg = new FieldDescriptionMesg();
            hrFieldDescMesg.SetDeveloperDataIndex(0);
            hrFieldDescMesg.SetFieldDefinitionNumber(1);
            hrFieldDescMesg.SetFitBaseTypeId(FitBaseType.Uint8);
            hrFieldDescMesg.SetFieldName(0, "Heart Rate");
            hrFieldDescMesg.SetUnits(0, "bpm");
            hrFieldDescMesg.SetNativeFieldNum(RecordMesg.FieldDefNum.HeartRate);
            hrFieldDescMesg.SetNativeMesgNum(MesgNum.Record);
            messages.Add(hrFieldDescMesg);

            // Every FIT ACTIVITY file MUST contain Record messages
            var timestamp = new Dynastream.Fit.DateTime(startTime);

            // Create one hour (3600 seconds) of Record data
            for (uint i = 0; i <= 3600; i++)
            {
                // Create a new Record message and set the timestamp
                var recordMesg = new RecordMesg();
                recordMesg.SetTimestamp(timestamp);

                // Fake Record Data of Various Signal Patterns
                recordMesg.SetDistance(i); // Ramp
                recordMesg.SetSpeed(1); // Flatline
                recordMesg.SetHeartRate((byte)((Math.Sin(TWOPI * (0.01 * i + 10)) + 1.0) *
                recordMesg.SetCadence((byte)(i % 255)); // Sawtooth
                recordMesg.SetPower((ushort)(THIS IS THE SOLUTION)); // Square
                recordMesg.SetAltitude((float)Math.Abs(((double)i % 255.0) - 127.0)); // T

                // Add a Developer Field to the Record Message
                var hrDevField = new DeveloperField(hrFieldDescMesg, developerIdMesg);
                recordMesg.SetDeveloperField(hrDevField);
                hrDevField.SetValue((byte)((Math.Sin(TWOPI * (0.01 * i + 10)) + 1.0) * 127

                // Write the Rercord message to the output stream
                messages.Add(recordMesg);
```

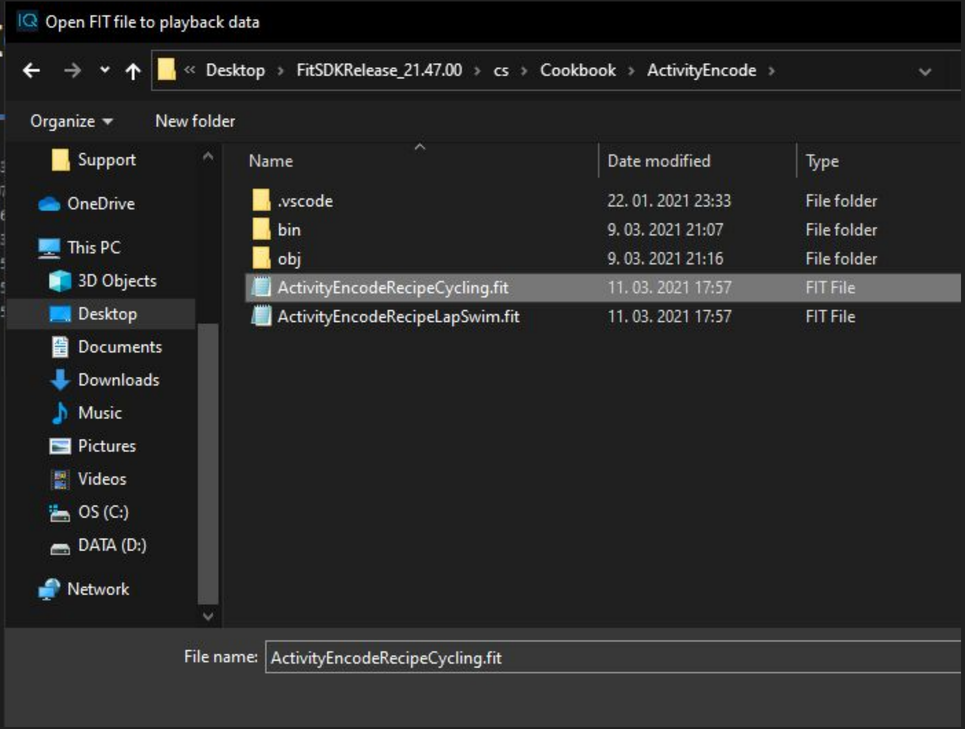


```
            FieldDescriptionMesg hrFieldDescMesg = new FieldDescriptionMesg();
            hrFieldDescMesg.SetDeveloperDataIndex(0);
            hrFieldDescMesg.SetFieldDefinitionNumber(1);
            hrFieldDescMesg.SetFitBaseTypeId(FitBaseType.Uint8);
            hrFieldDescMesg.SetFieldName(0, "Heart Rate");
            hrFieldDescMesg.SetUnits(0, "bpm");
            hrFieldDescMesg.SetNativeFieldNum(RecordMesg.FieldDefNum.HeartRate);
            hrFieldDescMesg.SetNativeMesgNum(MesgNum.Record);
            messages.Add(hrFieldDescMesg);

            // Every FIT ACTIVITY file MUST contain Record messages
            var timestamp = new Dynastream.Fit.DateTime(startTime);

            // Create one hour (3600 seconds) of Record data
            for (uint i = 0; i <= 3600; i++)
            {
                // Create a new Record message and set the timestamp
                var recordMesg = new RecordMesg();
                recordMesg.SetTimestamp(timestamp);

                // Fake Record Data of Various Signal Patterns
                recordMesg.SetDistance(i); // Ramp
                recordMesg.SetSpeed(1); // Flatline
                recordMesg.SetHeartRate((byte)((Math.Sin(TWOPI * (0.01 * i + 10)) + 1.0) * 127.0)); // Sine
```

```
-------------------------------------------------------------------
You may only use the Microsoft .NET Core Debugger (vsdbg) with
Visual Studio Code, Visual Studio or Visual Studio for Mac software
to help you develop and test your applications.
-------------------------------------------------------------------
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Private.CoreLib.dll'. Skipped loading symbols. Modu
Loaded 'C:\Users\lukad\Desktop\FitSDKRelease_21.47.00\cs\Cookbook\ActivityEncode\bin\Debug\netcoreapp3.1\ActivityEncode.dll'.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Runtime.dll'. Skipped loading symbols. Module is op
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Collections.dll'. Skipped loading symbols. Module i
Loaded 'C:\Users\lukad\Desktop\FitSDKRelease_21.47.00\cs\Cookbook\ActivityEncode\bin\Debug\netcoreapp3.1\Dynastream.Fit.Portab
My Code' is enabled.
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Runtime.Extensions.dll'. Skipped loading symbols. M
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Threading.dll'. Skipped loading symbols. Module is
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Text.Encoding.dll'. Skipped loading symbols. Module
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.IO.dll'. Skipped loading symbols. Module is optimiz
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Console.dll'. Skipped loading symbols. Module is op
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Linq.dll'. Skipped loading symbols. Module is optim
Loaded 'C:\Program Files\dotnet\shared\Microsoft.NETCore.App\3.1.13\System.Text.Encoding.Extensions.dll'. Skipped loading symb
Encoded FIT file C:\Users\lukad\Desktop\FitSDKRelease_21.47.00\cs\Cookbook\ActivityEncode\ActivityEncodeRecipeCycling.fit
Encoded FIT file C:\Users\lukad\Desktop\FitSDKRelease_21.47.00\cs\Cookbook\ActivityEncode\ActivityEncodeRecipeLapSwim.fit
The program '[21444] ActivityEncode.dll' has exited with code 0 (0x0).
```

13.Simulator testing



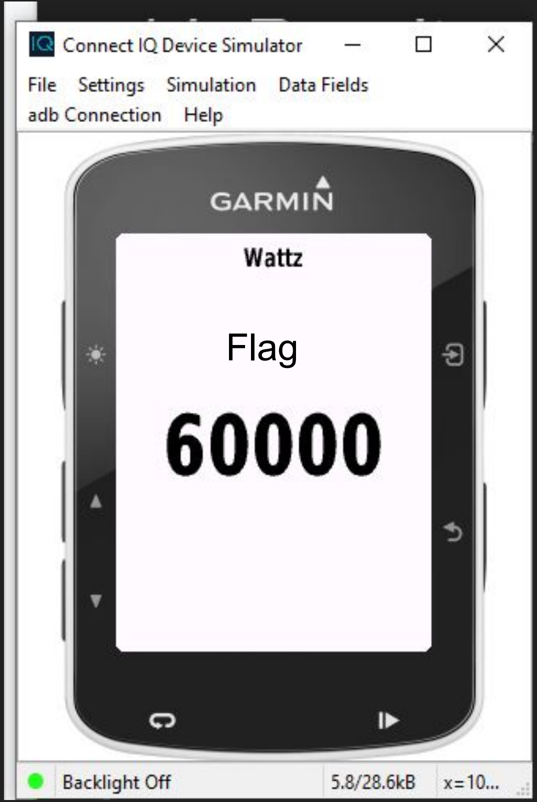


Name	Date modified	Type
.vscode	22. 01. 2021 23:33	File folder
bin	9. 03. 2021 21:07	File folder
obj	9. 03. 2021 21:16	File folder
ActivityEncodeRecipeCycling.fit	11. 03. 2021 17:57	FIT File
ActivityEncodeRecipeLapSwim.fit	11. 03. 2021 17:57	FIT File

File name: ActivityEncodeRecipeCycling.fit

14. Result

Altering Power Data with custom fit generation

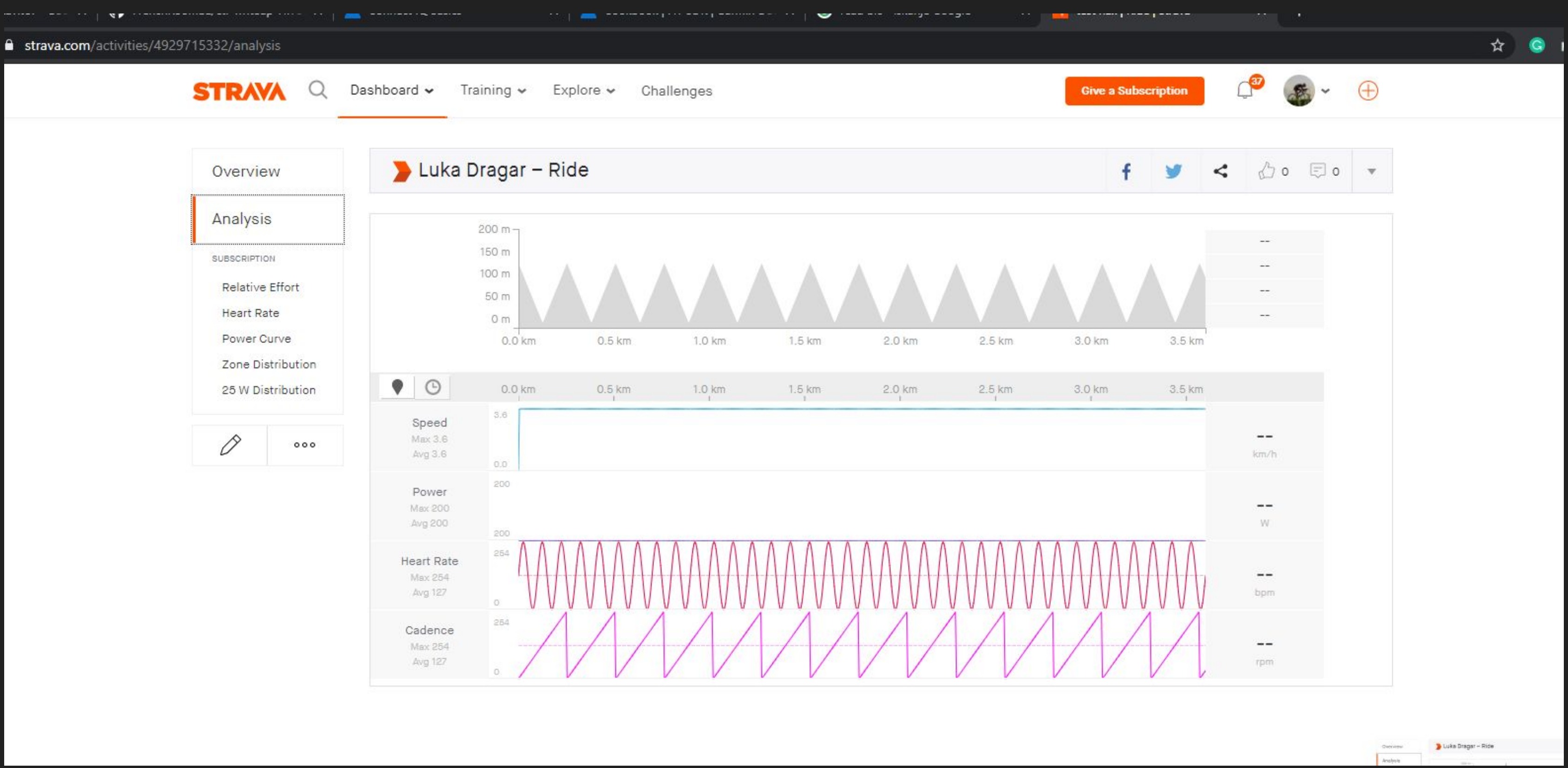


Wattz

Flag

60000

Real world usage


strava.com/activities/4929715332/analysis