

AmfiTrack Development Kit - Getting started

1. DOWNLOADING SOFTWARE, USB PROTOCOL, DATA LOGGING SCRIPT

Please follow this link in order to download the latest version of the AmfiTrack demo software "AmfiTrack Viewer", a USB protocol description, the data logging python script and the drivers:

<http://www.amfitrack.com/SW/AmfiTrackSoftware.zip>

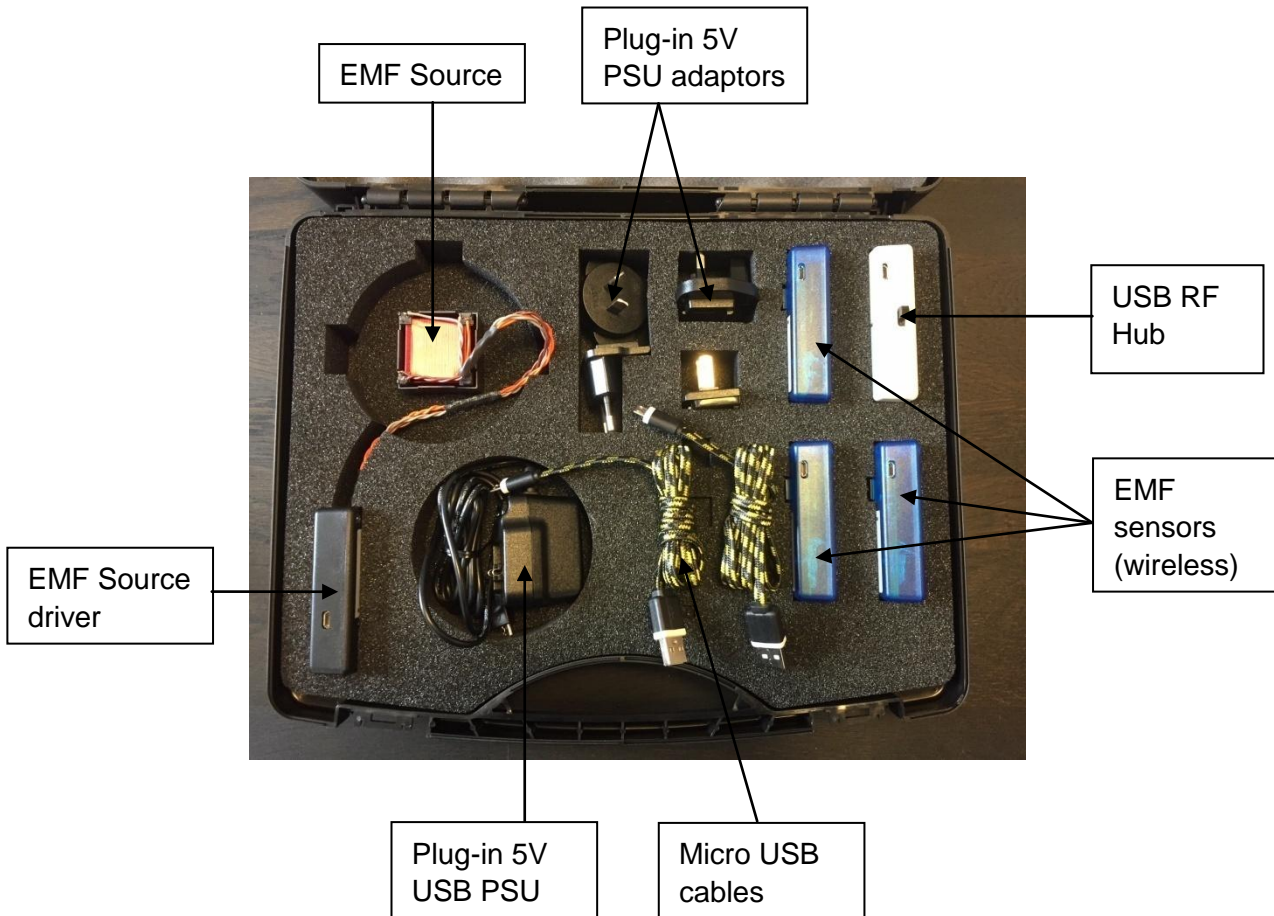
The AmfiTrack Viewer software is used when a visual representation of the tracking system output is necessary.

1.1. Drivers

A C++ driver exists for AmfiTrack alongside a .NET wrapper. These are distributed as a set of x86-64 DLLs ready to be imported into your project. The DLLs you will need are:

- hidapi.dll
- Amfitrack_CPP_Driver.dll
- *(optional, used for .NET)* AmfitrackCPPDriver_CSharpWrapper.dll

2. COMPONENTS OVERVIEW



EMF Source: Emits the EM field.

EMF Source driver: Is powered from 5V through USB port. Communicating with EMF sensors and USB RF hub using 2.4GHz proprietary RF link.

EMF sensors: Picks up the EM field and calculates its position (6DOF) from the EMF source. Communicating with EMF Source driver and USB RF hub using 2.4GHz proprietary RF link. Sensors can be supplied with or without Kalman filter.

USB RF Hub: Picks up the 2.4GHz RF link communication between the EMF Source driver and EMF sensors, and relays the system positioning data through its USB port.

Please note: The USB RF Hub is only needed to present positioning data from the wireless sensors on a PC. All positioning calculations are done embedded by the EMF sensors microcontroller.

Plug-in 5V USB PSU: Can be used for powering the EMF Source driver. Can also be used for charging the EMF sensors.

Plug-in 5V USB adaptors: Interchangeable. Allows the Plug-in PSU to be used with country specific wall outlets.

3. POWERING AND CHARGING

3.1. Powering the source

The source can be powered by 5V USB from either the supplied external plug-in Micro USB supply or a PC USB port.

3.2. Turning the sensors on/off

Press the button until the LED turns on/off.

3.3. Charging the sensor

The sensor contains a rechargeable Lithium battery. The charging starts when the sensor is connected to a 5V USB power source (external 5V USB power supply or PC USB port).

Charging indication:

LED flashing white: Charging in progress

LED constantly white: Charging complete

A complete charge cycle of a sensor from 0-100% will take approx. 1 hour.

4. START-UP CALIBRATION

Please note that you need to calibrate the sensor(s) after power up, in order for the sensors to hemisphere calibrated and recognized by the system/software.

Start-up calibration instructions (and more) can be found in this video:

<https://www.youtube.com/watch?v=X9ALvqmkmAM>

Below is also a picture showing the start-up calibration position, using the EMF source included in the standard Development Kit. The sensor start-up calibration position does not need to be very precise at all, since it is a hemisphere calibration and not a absolute position calibration.

You find the calibration position of the sensor approx 30-40 cm away from the Z/Y side of the source, and 30-40 cm up from the source plane (see picture below).



The sensor will stop flashing and light up constantly yellow, when the calibration position is correct. Then push the button of the sensor, and the sensor should be visible in the AmfiTrack Viewer software.

5. UNITY DEMO

You can download the demo by going to:

<http://www.amfitrack.com/SW/AmfiTrackIslandDemo.zip>

To run it, connect and calibrate Amfitrack as previously described. Unzip *AmfiTrackIslandDemo.zip*, and run *Amfitrack_Unity.exe*

6. UNITY DEMO PROJECT

You can download the demo project by going to:

<http://www.amfitrack.com/SW/UnityDemoProject.zip>

To get started with the unity project demo, first you need to download and install unity from <https://unity3d.com>.

After installing, you should be able to build the provided demo project (made using Unity v2018.2.18f1). To run it, crop the driver DLLs from *<build root>/Amfitrack_Unity_Data/Plugins* to *<build root>*.

To have the demo running in the Unity editor as well, you need to make the drivers available to it. To achieve this, copy *hidapi.dll* and *Amfitrack_CPP_Driver.dll* to *<unity path>/Editor*.