5.3) How to get GPS coordinates ?

How to extract the latitudes and latitudes (expressed in meters) of all GPS positions for a given sensor and for a given sequence and display it ?

gpsDataExtraction.cpp

```
#include <iostream>
#include <sstream>
#include <vector>
#include <string>

#include "containers.h"
#include "config.h"
#include "sensor_data.h"

int main(int argc, char* argv[])
{

   setlocale (LC_ALL,"C");

   //  Usage: ./GpsDataExtract sequencePath sensorAlias
   if(argc==3)
   {
     std::string sequencePath(argv[1]);
     std::string sensorAlias(argv[2]);

     ConfigTable ctable;
     std::stringstream msg;
     msg << CONFIGFOLDERPATH << "/config_table";
     ctable.readTable(msg.str());

     std::vector<Unit> data;

     std::string
timestamps_file_name=ctable.table[sensorAlias].folderName+ctable.table[sensorAlias].suffix;
     std::string
sensor_data_folder=sequencePath+"/"+ctable.table[sensorAlias].folderName+"/";

     getReadings(ctable.table[sensorAlias].sensorName, sensor_data_folder,
timestamps_file_name, ctable.table[sensorAlias].tfm, data, TimestampInterval());

     std::cout << "================" << std::endl;
     std::cout << "The GPS positions, expressed in meters, for the sensor " << sensorAlias << "
in the sequence stored in " << sequencePath << " are: "<< std::endl;
     for(unsigned int i=0;i<data.size();i++)
     {
          gps_data mygpsdata(data[i].reading);
          std::cout << "  ( " <<  mygpsdata.lat_l2e << " ; " << mygpsdata.lng_l2e << " )" <<
std::endl;
     }
     std::cout << "================" << std::endl;
   }

  return 0;
}
```

Explanation :

- The content is relatively similar to the previous example. We just have to cast the generic container in a sensor-specific object. In that aim, we have to do :

```
#include "sensor_data.h"
```

- Each Unit can be casted to a gps_data and then all elements can be access :

```
gps_data mygpsdata(data[i].reading);
std::cout << "  ( " <<  mygpsdata.lat_l2e << " ; " << mygpsdata.lng_l2e << " )" << std::endl;
```

This solution can be used for all referenced structures i.e. : gps_data, pose_data, odom_motor_data, steering_data, odo_wheels_data, accelerometer_data and gyroscope_data

9. (refer to sensor_data.h)

CMakeLists.txt

```
project(GpsDataExtraction)
cmake_minimum_required(VERSION 2.6)

set( CMAKE_BUILD_TYPE Release )
add_definitions( -Wall )

set(libIPDS_include_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/include/")
set(libIPDS_lib_dir "/home/courbon/IPDS/IPDSCodes/IPDSLib/build/lib/")


# IPDS library inclusion
include_directories(${libIPDS_include_dir})
link_directories(${libIPDS_lib_dir})

add_executable (${PROJECT_NAME} gpsDataExtraction.cpp)

target_link_libraries(
    ${PROJECT_NAME}
    IPDSlib
)
```

Note that the paths to the IPDS include and library (libIPDS_include_dir and libIPDS_lib_dir) have to be modified to suit your set-up.

(the source code is in the archive GPSDataExtraction.zip)