UNIVERZITA OBRANY

Dokumentace ke knihovně InsDevice

Výsledek řešení projektu PRO K-209

Petr Františ

4.1.2012

Programátorská dokumentace pro použití knihovny InsDevice určené k začlenění podpory inerciálních navigačních jednotek do vlastních programů a projektů.

Obsah

Popis 2
Podporované inerciální navigační jednotky 2
Popis implementace 3
Schéma 4
Poskytované funkce 4
Získání hodnot inerciální navigační jednotky 5
Příklad užití knihovny 6

Popis

Knihovna *InsDevice* je soubor tříd v jazyce C++, který poskytuje podporu pro zpracování informací z inerciálních navigačních jednotek. V této verzi jsou podporovány tři typy inerciálních navigačních jednotek. Knihovna *InsDevice* je navržena tak, aby mohla být jednoduše rozšířena o podporu dalších inerciálních navigačních jednotek. Významnou vlastností této knihovny je, že v sobě zapouzdřuje implementované jednotky a navenek tak poskytuje jednotné rozhraní. Proto pro software využívající tuto knihovnu se všechny jednotky jeví jako stejné. Software využívající služeb této knihovny může pracovat s kteroukoliv jednotkou, aniž by musel být vnitřně modifikován.

Podporované inerciální navigační jednotky

Inerciální navigační jednotka je zařízení, které umožňuje získat poziční a polohovou informaci nezávisle na jiných přístrojích. Ke své funkci využívá přesný GPS přijímač a soustavu akcelerometrů a magnetometrů. Údaje z těchto vnitřních komponent jsou zpracovány zabudovaným signálovým procesorem a poskytovány přes komunikační rozhraní (RS 232 nebo USB).

Obvykle inerciální navigační jednotky poskytují tyto údaje:

- poloha (zeměpisné souřadnice),
- nadmořská výška,
- orientace (otočení kolem třech os),
- vektor rychlosti (jeho složky v jednotlivých osách).

Tyto základní údaje poskytují všechny jednotky, rozdíly jsou především v přesnosti a někdy v použitých jednotkách. V knihovně *InsDevice* je zabudována podpory pro tři jednotky. Je to jednotka NAV440CA od firmy Crossbow, Mti-G firmy XSENS a eGYRO-GC-PLUS od firmy PCFlightSystems.

Všechny tyto jednotky poskytují všechny uvedené základní údaje, navíc každá jednotka poskytuje další doplňkové údaje. Knihovna pro obsluhu těchto jednotek musí údaje z těchto jednotek transformovat na standardní formát a jednotky, tak aby hlavní program využívající tuto knihovnu mohl ke všem jednotkám přistupovat stejně.

V následující tabulce je srovnání dvou testovaných jednotek:

	NAV440CA	MTi-G
Poloha	zeměpisné souřadnice	zeměpisné souřadnice
nadmořká výška	dle GPS v metrech	barometrický senzor + GPS
Orientace	eulerovy úhly, radiány	eulerovy úhly, quaternion, ve stupních
vektor rychlosti	severovýchodní rychlost a rychlost stoupání	severovýchodní rychlost a rychlost stoupání
Napájení	9-42V 350mA	z USB
Hmotnost	0.58 kg	68 g
Obrázek		

Tabulka 1 – Srovnání testovaných jednotek

Protože jednotky obsahují i magnetometry (pro úvodní kalibraci orientace jednotky), poskytují údaj o magnetickém kurzu. To je důležité zejména pro přistání, protože kurzy přistávacích drah jsou určovány dle magnetického severu.

Je důležité si uvědomit, že nadmořská výška je poskytována jak výška nad elipsoidem systému WGS-84. To znamená, že oproti leteckým mapám, které používají MSL výšku je zde rozdíl kolem 50 metrů nad našim územím. Proto je nutné před použitím systém kalibrovat. Obvykle se tak děje před startem letadla, protože nadmořská výška vzletové a přistávací dráhy je známa.

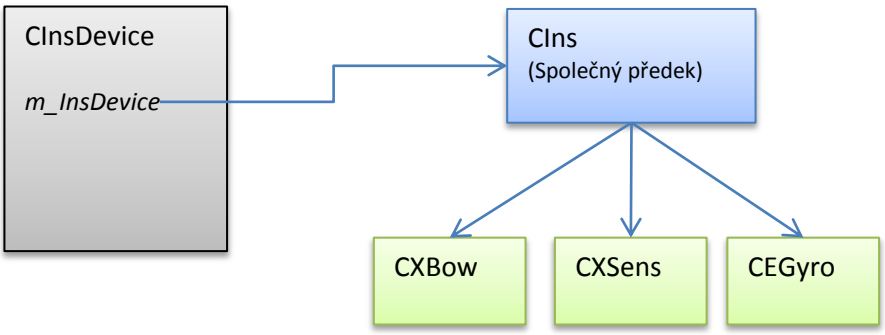
Ani jedna z těchto dvou jednotek neposkytuje informaci o pravé vzdušné rychlosti (nebo alespoň informaci o dynamickém tlaku vzduchu). Jednotka eGYRO-GC-PLUS poskytuje informaci o pravé vzdušné rychlosti a o barometrické výšce.

Popis implementace

Knihovna *InsDevice* se skládá z hlavní třídy *CInsDevice*. Tato třída implementuje rozhraní pro inicializaci a deinicializaci inerciálních navigačních jednotek, podpůrné funkce pro logování a přehrávání z logového souboru a objekt *m_InsDevice* třídy *CIns*.

Třída *CIns* poskytuje funkce pro čtení základních údajů z inerciálních navigačních jednotek. Tyto jednotky jsou implementovány samostatnými třídami (pro každý typ jednotky jedna třída). Uživatel při inicializaci zvolí, která inerciální navigační jednotka má být použita.

Schéma



Obrázek 1 - Schéma dědičnosti

Poskytované funkce

Základní poskytované funkce jsou veřejné funkce třídy CInsDevice. Tyto funkce slouží pro inicializaci inerciální navigační jednotky, pro práci s ní a pro práci s logovým souborem.

```
int StartInsDevice(int deviceid,int portnum,int mode,char *logname);
      //*** mode 0 = standard, mode LOG_SAVE_ON - enable saving data,
creates file with name as current date and time.dat, mode LOG_PLAY_ON opens
a logfile (logname) and reads ins data from this file
      void PauseInsDevice(BOOL pause); //*** true pause, false restore
state (run)
      void RollForward(int num); //*** skip num INS records while playing
from log file
```

Druh použité inerciální navigační jednotky se volí pomocí parametru *deviceid*, tento parametr může nabývat těchto hodnot:

```
#define INS_XBOW 1
#define INS_XSENS 2
#define INS_EGYRO 3
```

Parametr *portnum* určuje číslo sériového portu, ke kterému je připojena inerciální navigační jednotka. Při připojení jednotky přes rozhraní USB je to číslo virtuálního sériového portu (lze zjistit v nastavení zařízení systému).

Parametr *mode* určuje způsob práce s logovým souborem. Pokud je nulový, neprobíhá ukládání dat do logového souboru, pokud je nastaven na hodnotu LOG_SAVE_ON, potom jsou hodnoty z inerciální navigační jednotky ukládány do logového souboru do aktuálního adresáře. Logový soubor se je pojmenován na základě hodnoty parametru *logname* následované hodnotou aktuálního data a času s příponou *.dat*.

Pokud je parametr *mode* nastaven na hodnotu LOG_PLAY, není inerciální navigační jednotka inicializována, ale data jsou brána z logového souboru určeného parametrem *logname*.

Získání hodnot inerciální navigační jednotky

Hodnoty parametrů inerciální navigační jednotky jsou publikovány přes veřejné rozhraní objektu *m_Device*, které je implementací třídy *CIns*. Funkce tohoto rozhraní jsou:

```
//*** Get functions:
      //*** in radians
      float GetRollAngle;
      float GetPitchAngle;
      float GetYawAngleTrue;
      float GetYawAngleMagnetic;

      //*** in degreees
      float GetRollAngleDeg;
      float GetPitchAngleDeg;
      float GetYawAngleTrueDeg;
      float GetYawAngleMagneticDeg;


      float GetXAngRate();
      float GetYAngRate();
      float GetZAngRate();

      float GetXAccel();
      float GetYAccel();
      float GetZAccel();

      float GetNVel();
      float GetEVel();
      float GetDVel();

      float GetLonGPS();
      float GetLatGPS();
      float GetAltGPS();

      float GetTemperature();

      float GetGPSSpeed();
      float GetAirSpeed();

      DWORD GetSyncTime();

      BOOL GetHWFail();
      BOOL GetStatusFail();
```

Příklad užití knihovny

```
#include "insdevice.h"

CInsDevice g_InsDevice;

if (g_InsDevice.StartInsDevice(INS_XSENS,5,LOG_SAVE_ON,"XS")<0)
AfxMessageBox("Can not initialize INS device");

Horizont.SetAngles(g_InsDevice.m_Device->GetPitchAngle(),
g_InsDevice.m_Device->GetRollAngle());

Compass.SetAngles(g_InsDevice.m_Device->GetYawAngleTrue());

Variometer.SetAltitude(g_InsDevice.m_Device->GetAltGPS());

Variometer.SetVSpeed(g_InsDevice.m_Device->GetDVel());

Altimeter.SetAltitude(g_InsDevice.m_Device->GetAltGPS() +
Corrections.altitude);
```