# Mise en œuvre d'une centrale inertielle

Version 0.b

F Camps

LAAS / CNRS

Gestion document

| Versions | Date modification | Objet modification | Responsable modification |
|---|---|---|---|
| 0.a | 20/03/2007 | Création document | F Camps LAAS/CNRS |
| 0.b | 28/03/2007 | Modification de l'API | F Camps LAAS/CNRS |

Document disponible : www.laas.fr/~fcamps/MTI.pdf

Ce document s'applique aux versions de centrale inertielle Xsens suivantes n : 00300639, le numéro de série est inscrit sur la centrale inertielle.

Clé SDK : 6213-607-60822259
Version logiciel : 3144-415-32025555
Class C++ MTComm 1.2.0 du 27/02/2006
Nouvelle version SDK sur www.xsens.com

SOMMAIRE

# 1 Introduction

Dans les projets RIA, il est souvent utile d'avoir des informations de positionnement pour les robots terrestres et aériens. Ce document présente le développement détaillé d'une API pour l'utilisation de la centrale inertielle XSens (5G, 300°S) s/n : 00300639. L'API fournit les fonctionnalités de bas niveau pour lire des données de la centrale inertielle. Cette API est ensuite utilisée pour la communication de données dans un module Genom. L'API peut être utilisée sur l'ensemble des robots terrestres et ou aériens.

L'API doit fournir les informations de roulis, tangage, lacet, repère d'Euler en ligne de commande. Un module Genom doit être également développé pour publier ces informations de positionnement. Par défaut (sortie d'usine) la centrale inertielle est configurée pour donner des informations en mode quaternion 100 fois par seconde [1]. La fréquence de mesure est de 100Hz (valeur par défaut), la fréquence maximale est de 120Hz.

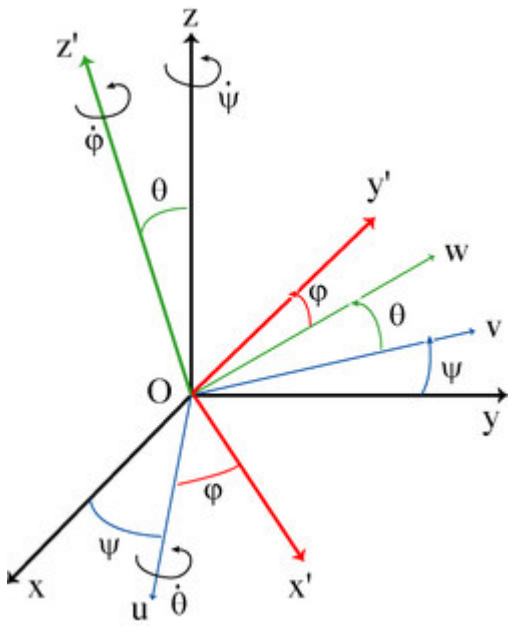
# 2 Fonctionnement d'une centrale inertielle

Une centrale à inertie ou centrale inertielle [3] est un appareil de navigation de précision comportant des gyroscopes, des capteurs d'accélération et de vitesse angulaire et calculant en temps réel à partir de ces mesures l'évolution du vecteur vitesse et la position du véhicule à bord duquel il est installé, ainsi que de son attitude (roulis, tangage, cap). Les centrales à inertie sont installées à bord de navire, d'aéronefs, de missiles et de véhicules spatiaux. La centrale Xsens possède 3 magnétomètres (compas) avec un processeur embarqué qui fournit en temps réel le roulis (roll), tangage (pitch) et le lacet (yaw) donc 9 données.


Les termes correspondant en anglais sont inertial unit, inertial platform, inertial measurement unit (IMU).

# 3 Repère d'Euler

On s'intéresse seulement ici à la description du mouvement du solide en rotation quelconque autour du point O, qui peut être un point fixe du solide dans le référentiel de référence Oxyz ou le centre de masse. Les angles d'Euler sont choisis de façon à permettre une mémorisation simple de la construction du vecteur rotation instantané du solide, nécessaire à l'étude de la cinématique du solide. On passe du référentiel fixe Oxyz au référentiel lié au solide Ox'y'z' par trois rotations successives. La précession ψ, autour de l'axe OzX




, fait passer de Oxyz au référentiel Ouvz. La rotation θ, autour de l'axe Ou, fait passer de Ouvz à Ouwz'. La rotation propre, autour de l'axe Oz', fait passer de Ouwz' au référentiel lié au solide Ox'y'z'.
Dans ces conditions le vecteur rotation instantané du solide est donné par la simple somme où les vecteurs apparaissant dans le membre de droite sont les vecteurs unitaires des axes correspondants.

$$\vec{\Omega} = \dot{\psi}\,\vec{z} + \dot{\theta}\,\vec{u} + \dot{\varphi}\,\vec{z'}$$

On remarquera que l'expression simple précédente utilise une base non orthogonale. Ces angles sont ensuite utilisés par les robots pour décrire leur mouvement.

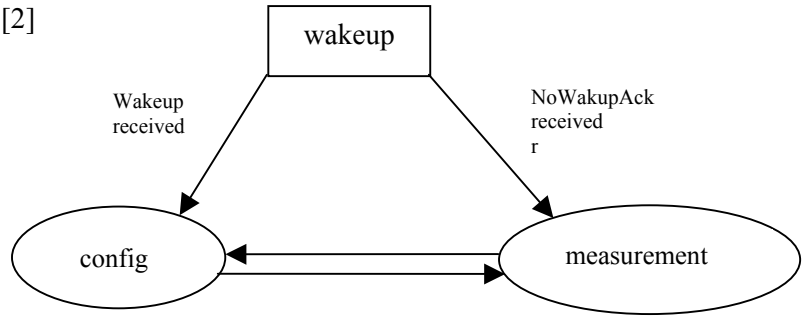
NB. L'axe Ou est porté par l'intersection des plans Oxy et Ox'y'

# 4 Lecture des informations

Le protocole de communication utilisé pat la sonde MTI est compatible avec MotionTracker communication protocol, un protocole développé par XSens pour l'ensemble de ses produits.

La sonde fonctionne selon 2 modes : configuration et mesure. A la mise sous tension la sonde envoie un message *Wakeup*, si la sonde ne reçoit aucune réponse alors elle se place en mode mesure. Mais si elle reçoit le message *Wakeupack* dans les 500ms alors la sonde entre dans le mode configuration. Avant de passer dans le mode mesure un message *Configuration* est envoyé vers le système embarqué, c'est la configuration utilisée pour les mesures. Il est également possible d'utiliser GotoConfig ou GotoMeasurement lorsque la sonde est dans le mode mesure.

[2]




La sonde est connectée au système embarqué par un port RS232 en utilisant la bibliothèque termio :

```
VMIN = 0
VTIME = 1
```

**VMIN** représente un nombre de caractères de 0 à 255 et VTIME est une mesure de temps avec un pas de 0.1 seconde, de 0 à 25.5 secondes.

**VMIN = 0 et VTIME = 0**
C'est une lecture non bloquante du read() – l'appel est satisfait immédiatement par le pilote du port série. Si des données sont disponibles, alors elles sont transmises directement à l'utilisateur dans un buffer de nbytes. Sinon le read() renvoie 0. Cette façon de lire les données s'appelle le polling, cette méthode consomme du temps processeur et n'est pas très efficace. Il est préférable de ne pas l'utiliser, à moins de vraiment y être contraint.

**VMIN = 0 and VTIME > 0**
Cette méthode permet de lire le port série pendant un temps donné, c'est un timer. Si des données sont disponibles dans la file d'attente, alors elles sont transférées à l'appel à l'utilisateur avec un maximum de nbyte. Sinon le read() est bloquant jusqu'à l'arrivée de données ou lorsque VTIME expire depuis le début de l'appel. Si le timer expire sans données, alors read() renvoie 0. Un seul octet permet de satisfaire l'appel du read(), mais si plus de données sont présentes dans le buffer série alors elles sont retournées.

**VMIN > 0 and VTIME > 0**
L'appel du read() est satisfait lorsque VMIN caractères ont été transférés dans le buffer appelant ou si VTIME expire entre des caractères. Dans la mesure où le timer n'est démarré qu'à l'arrivée du premier caractère, alors l'appel peut bloquer indéfiniment si la ligne série est en attente ou débranchée. VTIME est un timout inter-caractère. Cet appel ne retourne jamais zéro octet.

**VMIN > 0 and VTIME = 0**
Dans ce cas read() est un compteur de données qui est satisfait uniquement lorsque au moins VMIN caractères ont été transférés dans le buffer – dans ce cas le temps n'est pas pris en compte. L'appel peut être bloquant indéfiniment, si la ligne série est en attente ou débranchée.

Le fichier MTComm.cpp contient le code qui permet de configurer les valeurs de VMIN et VTIME.

# 5 Modification de l'API MTCOMM :

Par défaut l'API MTCOMM positionne VMIN=0 et VTIME=5**.** Pour réaliser une mesure à 100Hz VTIME a été modifiée à VTIME=1.

Le mode de mesure a été positionné de la façon suivante et le menu utilisateur a été supprimé :

```
void getUserInputs(char *device)
{
  outputMode = 3;
…
  outputSettings = OUTPUTSETTINGS_ORIENTMODE_EULER;
…
}
```

## 5.1 Utilisation de l'API

La fonction suivante permet de lancer l'API avec un passage de paramètre :

```
int startMTI(int argc, char *argv[])
```

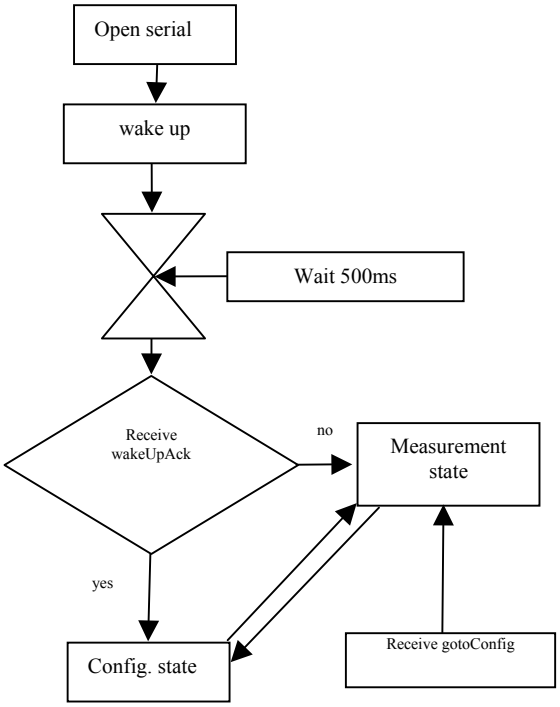
Mode verbeux sur le port ttyS0 :
/MTITest –v /dev/ttyS0 // affichage sur la sortie standard

Mode verbeux avec log dans un fichier
/MTITest –v /dev/ttyS0 -l logfile.txt // affichage et log dans un fichier

## 5.2 Détramage des informations

Le détramage des informations est assurée par la classe MTComm livrée par le constructeur.

## 5.3 Machine à état de l'API [2]:




# 6 Configuration de la centrale

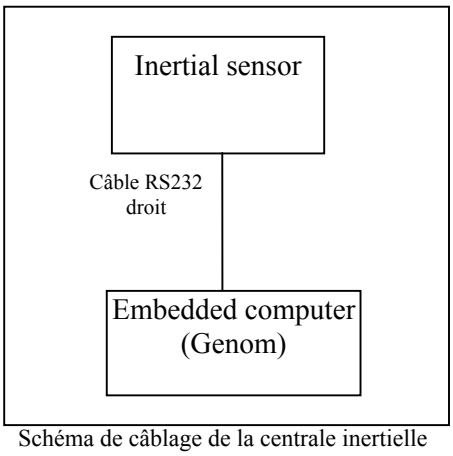



Schéma de câblage de la centrale inertielle

# 7 API

L'API fournit un programme de test pour la centrale en mode de commande.

Pour lancer l'API avec un affichage sur la sortie standard (repère Euler, ,

```
./MTITest -v /dev/ttyS0 -d 3 -o 2
```

Option : o
d : output mode :
// 1 - Quaternions
// 2 - Euler angles
// 3 - Matrix
Option : d
o : data output format :
// 1 - Calibrated data
// 2 - Orientation data
// 3 - Both Calibrated and Orientation data

Pour lancer l'API avec un enregistrement dans un fichier et un affichage sur la sortie standard:

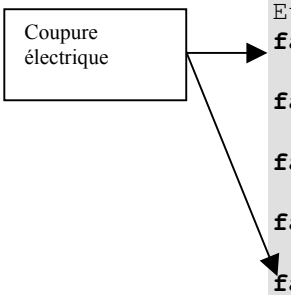
```
MTITest -v /dev/ttyS0 -o 3 -d 1
ACC 1175591741.934895000 0.405339 -0.013323 9.82722
GYR 1175591741.934895000 0.00926154 -0.0126498 -0.000776927
MAGN 1175591741.934895000 -0.346904 -0.162479 -0.507031
Quaternion  0.192  0.023  -0.005   0.981

MTITest -v /dev/ttyS0 -o 3 -d 2
GYR 1175591641.95478000 0.00684433 -0.00434485 0.000391047
MAGN 1175591641.95478000 -0.359825 -0.150648 -0.441962
Euler 1175591641.95520000 -0.0280683 -2.57175 157.115

MTITest -v /dev/ttyS0 -o 3 -d 3
ACC 1175591533.162465000 0.393923 1.14481e-05 9.76012
GYR 1175591533.162465000 -0.00720804 -0.00418098 -0.000433221
MAGN 1175591533.162465000 -0.344148 -0.163636 -0.510993
Matrix -0.926    0.376   0.045
Matrix -0.376   -0.926  -0.003
Matrix  0.040   -0.019   0.999
```

# 8 Horodatage

L'horodatage est assuré lors de l'utilisation de l'API, dans le module Genom l'horodatage n'est pas assuré le module « appelant » peut fournir l'horodatage.

# 9 Coupure électrique en mode commande

Le test consiste à couper l'alimentation électrique puis de re-alimenter la centrale. A la reconnexion la centrale reprend le décodage des informations :




```
Euler 1172051312.587676000 3.66294 5.92737 148.278
ACC 1172051312.597592000 -1.03854 0.603196 9.72968
GYR 1172051312.597592000 0.000532169 -0.00972301 0.00515331
MAGN 1172051312.597592000 -0.111044 -0.254017 -0.915454
Euler 1172051312.597646000 3.66178 5.92606 148.165
failed to read message, code (2)

failed to read message, code (2)

failed to read message, code (2)

failed to read message, code (2)

failed to read message, code (2)

ACC 1172051315.543369000 -1.03854 0.603196 9.72968
GYR 1172051315.543369000 0.000532169 -0.00972301 0.00515331
MAGN 1172051315.543369000 -0.111044 -0.254017 -0.915454
Euler 1172051315.543423000 3.66178 5.92606 148.165
```

## 10 Fonctionnement normal dans hardlib

L'API s'appel MTI et se trouve dans cvs.laas.fr:/cvs/openrobots :

```
/home/fcamps/DOCS/RIA/hardlib/MTI/test/.libs
shuttle[.libs] ./MTITest -v /dev/ttyS0
*****> mode2
MTI Calibrated sensor data - LAAS/CNRS 2006

ACCX ACCY ACCZ : unity m/s2
GYRX GYRY GYRZ : unity rad/s
MAGNX MAGNY MAGNZ : arbitrary units normalized to earth field
strength
EulerX EulerY EulerZ : unity degree

ACC 1173274639.200957000 0.570438 0.858297 9.75441
GYR 1173274639.200957000 -0.00173001 -0.0133367 0.00629628
MAGN 1173274639.200957000 0.0150826 -0.170309 -0.682367
Euler 1173274639.201066000 5.02746 -3.34213 63.2148
ACC 1173274639.201098000 0.573888 0.858575 9.72572
GYR 1173274639.201098000 -0.000102778 -0.0133447 0.00578157
MAGN 1173274639.201098000 -0.135765 -0.492307 -1.05934
Euler 1173274639.201129000 5.02746 -3.3504 63.5616
ACC 1173274639.204280000 0.559526 0.87304 9.75357
GYR 1173274639.204280000 0.00541642 -0.0124868 -0.00497945
MAGN 1173274639.204280000 0.0426676 -0.123192 -0.630444
Euler 1173274639.204329000 5.03184 -3.35684 63.3297
```

## 11 Module Genom

Le module Genom s'appel MTI et se trouve dans cvs.laas.fr:/cvs/openrobots :

```
h2 init
Initializing pocolibs devices: OK
shuttle[MTI] pocolibs execution environment version 2.1
Copyright (c) 1999-2005 CNRS-LAAS

shuttle[MTI] MTI -b
pocolibs execution environment version 2.2
Copyright (c) 1999-2007 CNRS-LAAS
MTI: spawning control task.
MTICntrlInitTask: created mailbox
MTICntrlInitTask: initialized mailbox as a server
MTICntrlInitTask: installed requests
MTICntrlInitTask: installed abort request
MTICntrlInitTask: created control poster
MTI: spawning task MTIMTITask.
MTIMTITaskInitTaskFunc: timer allocated
MTIMTITaskInitTaskFunc: timer started
MTIMTITaskInitTaskFunc: posters created
MTIMTITaskInitTaskFunc: client posters initialized
MTIMTITaskInitTaskFunc: ok
MTI: all tasks are spawned
shuttle[MTI] MTITest 0
pocolibs execution environment version 2.2
Copyright (c) 1999-2007 CNRS-LAAS
client init ...ok.   Poster init ...ok.
-------------------------------------------------------------------
-----
  0:  MTIGetData  (nE)            1:  GPSSetData  (nE)            2:
GetMTIConfiguration

55: posters   66: abort   77: replies(0)   88: state   99: QUIT   -
99: END
-------------------------------------------------------------------
-----
MTI0 (88)> 0

Get current inertialInit (y/n) ? (n) y
                ------------------------------------
                0:  request GetMTIConfiguration
                1:  poster MTI_INFOMTISensor
                other: don't get
                ------------------------------------
Select function > (-1)  0
Final reply: OK

-- Enter struct INERTIAL_INIT inertialInit:
  device[24]  (/dev/ttyS0)
  mode  (3)
  ACC[4]:
    | [0] (0.000000)
    | [1] (0.000000)
    | [2] (0.000000)
```

```
    | [3] (0.000000)
  GYR[4]:
    | [0] (0.000000)
    | [1] (0.000000)
    | [2] (0.000000)
    | [3] (0.000000)
  MAG[4]:
    | [0] (0.000000)
    | [1] (0.000000)
    | [2] (0.000000)
    | [3] (0.000000)
  EULER[4]:
    | [0] (0.000000)
    | [1] (0.000000)
    | [2] (0.000000)
    | [3] (0.000000)
Wait final reply (y/n/a) ? : y

Activity 0 started
MTI Calibrated sensor data - LAAS/CNRS 2006

ACCX ACCY ACCZ : unity m/s2
GYRX GYRY GYRZ : unity rad/s
MAGNX MAGNY MAGNZ :       arbitrary units normalized to earth field
strength
EulerX EulerY EulerZ : unity degree

ACC 1173880453.87844000 -0.744109 0.939623 9.72442
GYR 1173880453.87844000 0.0084367 -0.011531 0.00033025
MAGN 1173880453.87844000 -0.077955 -0.547501 -0.562148
mode = 3, device = /dev/ttyS0
pointer = 0xb7cf7938
***********start poster*************
ACC : X=-0.744109  Y=0.939623  Z=9.724424
GYR : X=-0.077955  Y=-0.547501  Z=-0.562148
MAG : X=-0.077955  Y=-0.547501  Z=-0.562148
Euler : X=5.523922  Y=4.348711  Z=104.200249
***********end poster*************
MTI Calibrated sensor data - LAAS/CNRS 2006

ACCX ACCY ACCZ : unity m/s2
GYRX GYRY GYRZ : unity rad/s
MAGNX MAGNY MAGNZ :       arbitrary units normalized to earth field
strength
EulerX EulerY EulerZ : unity degree

ACC 1173880453.333926000 -0.766007 0.971062 9.71188
GYR 1173880453.333926000 0.00181259 -0.0129211 -0.0073448
MAGN 1173880453.333926000 -0.0914663 -0.829825 -0.848278
mode = 3, device = /dev/ttyS0
pointer = 0xb7cf7938
```

# 12 Conclusion

L'API a été utilisé avec succès sur le ballon dirigeable. Le module Genom MTI peut maintenant fournir les informations de positionnement aux différents modules de calcul :

```
ACC 0.570438 0.858297 9.75441
GYR -0.00173001 -0.0133367 0.00629628
MAGN 0.0150826 -0.170309 -0.682367
Euler 5.02746 -3.34213 63.2148
```

# 13 Documentations de référence

[1] MTi and MTx Low level communication documentation – Doc MT0101P, Rev E march 2, 2005

[2] MTi and MTx User Manual and Technical Documentation – Doc MT0100P – Re G, March 2, 2006

[3]http://fr.wikipedia.org