Annexe 2

Code du module «Capteurs»
codeEmetteurNanoVDistortion.ino

```
/*****************************************************************
***********  code pour l'emeteur des angles Yaw,Roll,Pitcc  **********
***********      made by the MovingTones team              **********
***********              website :)                        **********
* Licensed under a Creative Commons Attribution 3.0 Unported License *
*****************************************************************/



/* Les angles ici envoyes seront les angles Yaw,Roll,Pich
plus d'info : http://en.wikipedia.org/wiki/Aircraft_principal_axes
C'est necesaire d'avoir les libreries suivantes:
            * FreeSixIMU
            * FIMU_ADXL345
            * FIMU_ITG3200

Vous pouvez les trouver sur:
http://bildr.org/2012/03/stable-orientation-digital-imu-6dof-arduino/*/


//Importation des bilbiotheques

        #include <FreeSixIMU.h>
        #include <FIMU_ADXL345.h>
        #include <FIMU_ITG3200.h>
        #include <Wire.h>


float angles[3];                            //Ici on va stocker les angles.
//float accel[3];                          //Si jamais on a besoin des accelerations
FreeSixIMU sixDOF = FreeSixIMU();          // Set the FreeSixIMU object



void setup() {

        Wire.begin();
        Serial.begin(57600);            //C'est necesaire pour envoyer les messages via XBee
        delay(5);
        sixDOF.init();                   //begin the IMU
        delay(5);
}

void loop() {


      //sixDOF.getEuler(angles);              //Si jamais on a besoin des angles d'euler
      //sixDOF.getValues(accel);              //Si jamais on a besoin des accelerations
      sixDOF.getYawPitchRoll(angles);        //On a choisi d'utiliser ces angles car sont plus intuitives dans le
mouvement

      angles[0]=abs(angles[0]);                // Comme l'angle va de -180 a +180 on le change pour etre de 0 ---
150--180--150--0 etc...

      //Mais en fait l'angle Yaw va de -180 jusqu'a +180, donc on va se restrindre a la plage 0 180.
      //TODO: Take advantaje of this large angle.
     // if (angles[0]>180){angles[0]=180;}
     // if (angles[0]<0){angles[0]=0;}

      angles[1]=angles[1]+90;              // Comme l'angle va de -90 a +90 on le change pour etre de 0 a 180
      angles[2]=angles[2]+90;              // Comme l'angle va de -90 a +90 on le change pour etre de 0 a 180


      //Si on a besoin de voir les angles qu'on envoie, on peut les afficher sur l'ecran de l'ordinateur
      //Mais il y aura des problemes dans les messages recoit par la pedale: L'ordinateur et le Xbee utilisent le
meme port serial de l'arduino nano (unique)

            /* Serial.print(angles[0]);
              Serial.print(" | ");
              Serial.print(angles[1]);
              Serial.print(" | ");
              Serial.print(angles[2]);
              Serial.println();*/


      envoyeAngles(angles[1],angles[0],angles[2]);      //On envoie finnalement les angles a la pedale a travers
du port Serial.
```

```
}

void envoyeAngles(float psi, float theta, float phi){

          int16_t psiInt = (int16_t)(psi);            //On va envoyer des integers, car c'est plus facile et on
n'a pas besoin d'assez de precition
          int16_t thetaInt = (int16_t)(theta);        //On va envoyer des integers, car c'est plus facile et on
n'a pas besoin d'assez de precition
          int16_t phiInt = (int16_t)(phi);            //On va envoyer des integers, car c'est plus facile et on
n'a pas besoin d'assez de precition

          //Ici, pour chaque angle en variable type integer, on va lui diviser en deux octets en on va lui
envoyer un par un
          //Toud d'abord on envoye le "header" qui est 0xAAAA= -21846
          Serial.write(0xAAAA/ 256);    //on envoye la premiere partie
          Serial.write(0xAAAA% 256);    //on envoye la deuxieme partie

          Serial.write(psiInt / 256);   //on envoye la premiere partie
          Serial.write(psiInt% 256);    //on envoye la deuxieme partie

          Serial.write(thetaInt / 256);  //on envoye la premiere partie
          Serial.write(thetaInt% 256);


          Serial.write(phiInt/ 256);    //on envoye la premiere partie
          Serial.write(phiInt% 256);    //on envoye la deuxieme partie

}
```