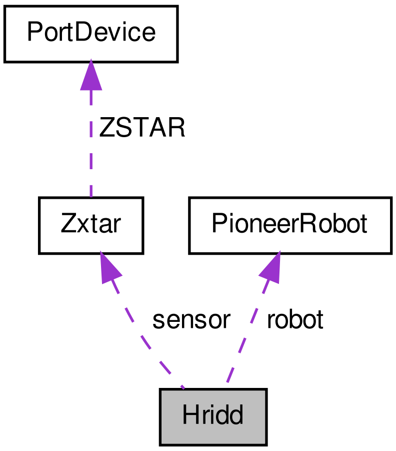
## .0.1 Hridd Class Reference

```
#include <Hridd.h>
```

Collaboration diagram for Hridd:



### Public Member Functions

- int **Kbhit** (void)
- void **GestureControl** ()
- void **SetACCFrameWord** (unsigned int framevalue)

### Public Attributes

- **PioneerRobot robot**
- **Zxtar sensor**

### .0.1.1 Detailed Description

Human-Robot Interaction Dance Demo class is aimed to combine the **Zxtar** (p. 7) accelerometer class, the **PioneerRobot** (p. 3) class so as to control the robot with the user's wearable accelerometer

Definition at line 16 of file Hridd.h.

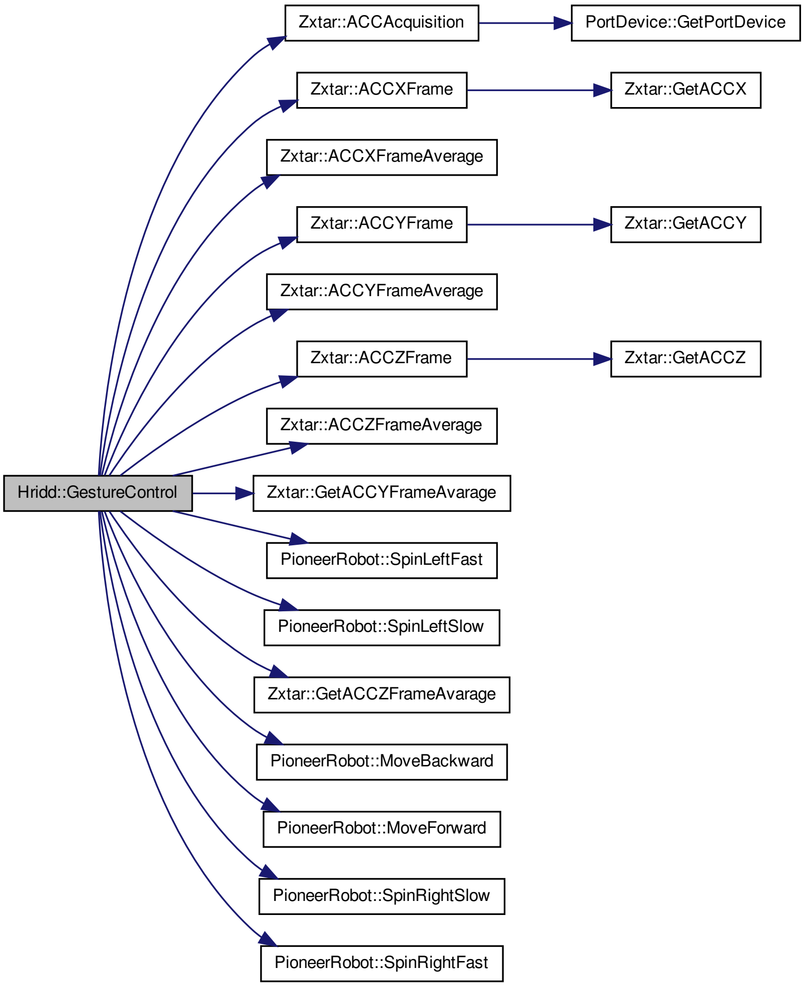
### .0.1.2 Member Function Documentation

#### .0.1.2.1 void **Hridd::GestureControl** ( )

This method contains the main control gesture for the robot movements

Definition at line 40 of file Hridd.cpp.

Here is the call graph for this function:


Hridd::GestureControl
Zxtar::ACCAcquisition
PortDevice::GetPortDevice
Zxtar::ACCXFrame
Zxtar::GetACCX
Zxtar::ACCXFrameAverage
Zxtar::ACCYFrame
Zxtar::GetACCY
Zxtar::ACCYFrameAverage
Zxtar::ACCZFrame
Zxtar::GetACCZ
Zxtar::ACCZFrameAverage
Zxtar::GetACCYFrameAvarage
PioneerRobot::SpinLeftFast
PioneerRobot::SpinLeftSlow
Zxtar::GetACCZFrameAvarage
PioneerRobot::MoveBackward
PioneerRobot::MoveForward
PioneerRobot::SpinRightSlow
PioneerRobot::SpinRightFast

#### .0.1.2.2 int **Hridd::Kbhit** ( void )

Method for the Keyboard hit

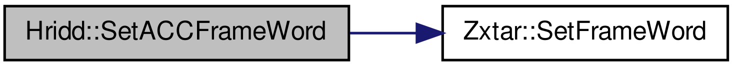
Definition at line 8 of file Hridd.cpp.

#### .0.1.2.3 void **Hridd::SetACCFrameWord** ( unsigned int *framevalue* )

Function to set the Acceleration Frame Word

Definition at line 34 of file Hridd.cpp.

Here is the call graph for this function:



### .0.1.3 Member Data Documentation

#### .0.1.3.1 PioneerRobot Hridd::robot

Data Member for the PionerRobot Class which defines and initializes the robot

Definition at line 23 of file Hridd.h.

#### .0.1.3.2 Zxtar Hridd::sensor

Data Member for the PionerRobot Class which defines and initializes the accelerometer sensor

Definition at line 28 of file Hridd.h.

The documentation for this class was generated from the following files:

- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**Hridd.h**
- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**Hridd.cpp**

## .0.2 PioneerRobot Class Reference

```
#include <PioneerRobot.h>
```

### Public Member Functions

- **PioneerRobot** ()
- ~**PioneerRobot** ()
- virtual bool **init** ()
- virtual void **getCurrentConfiguration** (std::vector< double > &q)

- virtual bool **moveToConfiguration** (std::vector< double > configuration)
- void **MoveForward** ()
- void **MoveBackward** ()
- void **SpinLeftSlow** ()
- void **SpinRightSlow** ()
- void **SpinLeftFast** ()
- void **SpinRightFast** ()

### Public Attributes

- ArRobot **pioneer**

### Protected Member Functions

- double **validateAngle** (double angle)

### .0.2.1 Detailed Description

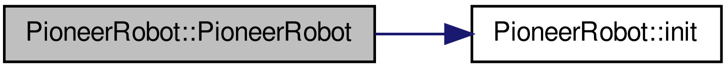
Definition at line 13 of file PioneerRobot.h.

### .0.2.2 Constructor & Destructor Documentation

#### .0.2.2.1 PioneerRobot::PioneerRobot ( )

Definition at line 3 of file PioneerRobot.cpp.

Here is the call graph for this function:



#### .0.2.2.2 PioneerRobot::∼PioneerRobot ( )

Definition at line 9 of file PioneerRobot.cpp.

### .0.2.3 Member Function Documentation

#### .0.2.3.1 void PioneerRobot::getCurrentConfiguration ( std::vector< double > & *q* ) [virtual]

Definition at line 76 of file PioneerRobot.cpp.

#### .0.2.3.2 bool PioneerRobot::init ( ) [virtual]

Definition at line 15 of file PioneerRobot.cpp.

#### .0.2.3.3 void PioneerRobot::MoveBackward ( )

Definition at line 144 of file PioneerRobot.cpp.

#### .0.2.3.4 void PioneerRobot::MoveForward ( )

Definition at line 149 of file PioneerRobot.cpp.

#### .0.2.3.5 bool PioneerRobot::moveToConfiguration ( std::vector< double > *configuration* ) [virtual]

Definition at line 91 of file PioneerRobot.cpp.

#### .0.2.3.6 void PioneerRobot::SpinLeftFast ( )

Definition at line 164 of file PioneerRobot.cpp.

#### .0.2.3.7 void PioneerRobot::SpinLeftSlow ( )

Definition at line 154 of file PioneerRobot.cpp.

#### .0.2.3.8 void PioneerRobot::SpinRightFast ( )

Definition at line 169 of file PioneerRobot.cpp.

#### .0.2.3.9 void PioneerRobot::SpinRightSlow ( )

Definition at line 159 of file PioneerRobot.cpp.

#### .0.2.3.10 double PioneerRobot::validateAngle ( double *angle* ) [protected]

Definition at line 132 of file PioneerRobot.cpp.

### .0.2.4 Member Data Documentation

#### .0.2.4.1 ArRobot PioneerRobot::pioneer

Definition at line 31 of file PioneerRobot.h.

The documentation for this class was generated from the following files:

- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**PioneerRobot.h**
- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**PioneerRobot.-cpp**

## .0.3 PortDevice Class Reference

```
#include <PortDevice.h>
```

### Public Member Functions

- **PortDevice** ()
- int **AllocatePortDevice** ()
- int **ConfigPortDevice** ()
- int **GetPortDevice** ()
- void **CleanPortDevice** ()
- void **ClosePortDevice** ()

### .0.3.1 Detailed Description

Definition at line 27 of file PortDevice.h.

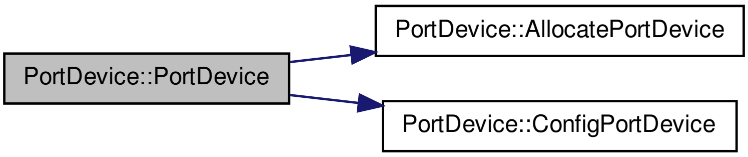
### .0.3.2 Constructor & Destructor Documentation

#### .0.3.2.1 PortDevice::PortDevice ( )

Constructor to reset, to allocate and configurate the ZSTAR USB Stick

Definition at line 4 of file PortDevice.cpp.

Here is the call graph for this function:



### .0.3.3 Member Function Documentation

#### .0.3.3.1 int PortDevice::AllocatePortDevice ( )

SetPortDeviceFD

Definition at line 13 of file PortDevice.cpp.

#### .0.3.3.2 void PortDevice::CleanPortDevice ( )

function to clean the Port Device

Definition at line 89 of file PortDevice.cpp.

#### .0.3.3.3 void PortDevice::ClosePortDevice ( )

ClosePortDevice

Definition at line 83 of file PortDevice.cpp.

#### .0.3.3.4 int PortDevice::ConfigPortDevice ( )

ConfigPortDeviceFD

Definition at line 28 of file PortDevice.cpp.

#### .0.3.3.5 int PortDevice::GetPortDevice ( )

function to get the port's FILE DESCRIPTOR

Definition at line 78 of file PortDevice.cpp.

The documentation for this class was generated from the following files:

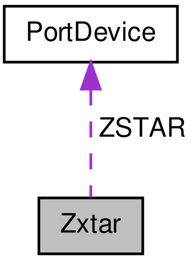
- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**PortDevice.h**

- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**PortDevice.cpp**

## .0.4 Zxtar Class Reference

```
#include <Zxtar.h>
```

Collaboration diagram for Zxtar:



### Public Member Functions

- **Zxtar** ()
- **Zxtar** (int gSensitivity, int setRate)
- void **AvailableACC** ()
- int **Kbhit** (void)
- void **SetSensitivity** (int gSensitivity)
- void **DataRate** (int setRate)
- void **ACCAcquisition** ()
- double **GetACCX** ()
- double **GetACCY** ()
- double **GetACCZ** ()
- void **ACCPrint** ()
- void **startTimer** ()
- double **stopTimer** ()
- void **SetFrameWord** (unsigned int framevalue)
- void **ACCXFrame** ()
- void **ACCXFramePrint** ()
- double **ACCXFrameAverage** ()
- double **GetACCXFrameAvarage** ()
- double **ACCXFrameStdDeviation** ()
- double **GetACCXFrameStdDeviation** ()
- void **ACCYFrame** ()
- void **ACCYFramePrint** ()
- double **ACCYFrameAverage** ()
- double **GetACCYFrameAvarage** ()
- double **ACCYFrameStdDeviation** ()
- double **GetACCYFrameStdDeviation** ()
- void **ACCZFrame** ()

- void **ACCZFramePrint** ()
- double **ACCZFrameAverage** ()
- double **GetACCZFrameAvarage** ()
- double **ACCZFrameStdDeviation** ()
- double **GetACCZFrameStdDeviation** ()

### Static Public Member Functions

- static void **printTime** (double duration)

### Public Attributes

- **PortDevice ZSTAR**

### .0.4.1 Detailed Description

**Zxtar** (p. 7) class permitis to extract a frame, and compute the avarage and Standar Deviation, as well to get these values for the X, Y and Z axis. The following code is an example for its use; uncomment the code for further test.

**Zxtar** (p. 7) sensor; // Declaration of the sensor Zstar class type

sensor.ACCXFrame(); // Getting the ACC of the X-axis frame word

sensor.ACCXFrameAverage(); // X-axis ACC frame mean

sensor.ACCXFrameStdDeviation(); // X-axis ACC frame Standard Deviation

cout << sensor.GetACCXFrameAvarage() << endl; //Printing the ACCX value of the frame avarage

cout << sensor.GetACCXFrameStdDeviation() << endl; //Printing the ACCX value of the frame standard deviation

Definition at line 47 of file Zxtar.h.

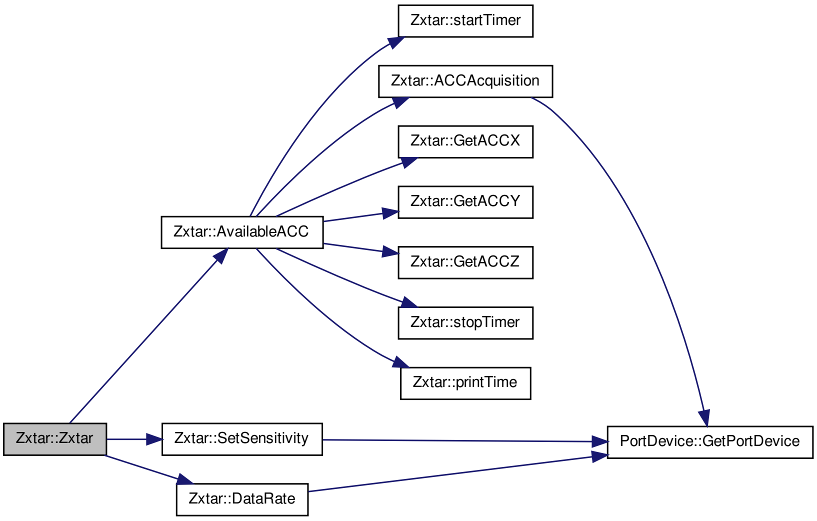
### .0.4.2 Constructor & Destructor Documentation

#### .0.4.2.1 Zxtar::Zxtar ( )

Default Constructor with the default values for sensitibility and data rate: 2g and 60Hz

Definition at line 4 of file Zxtar.cpp.

Here is the call graph for this function:



### .0.4.2.2 Zxtar::Zxtar ( int *gSensitivity,* int *setRate* )

Constructor with parameter options, so as the user is able to configure the sensor by using the following values

**Zxtar** (p. 7) name(g-range, DataRate);

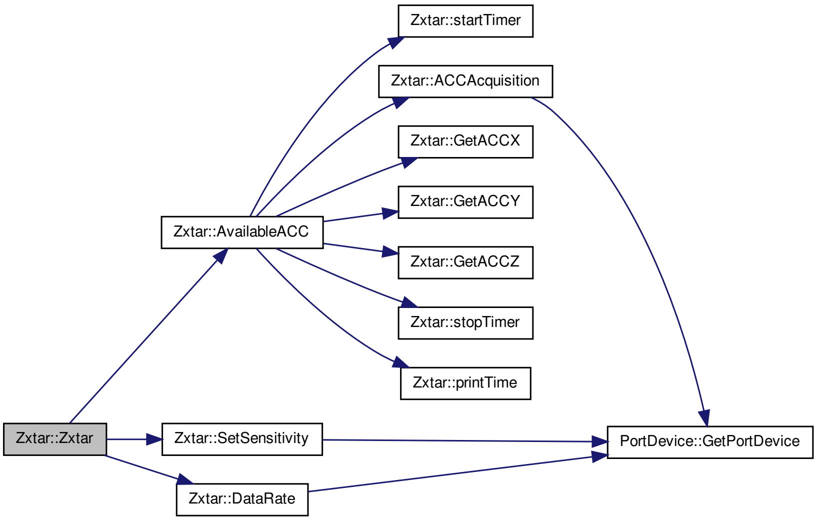
..............|........|

.............|........|--------- Available Values for DataRate are 30, 60, and 120

.............|------------------- Available Values for g-range parameter are 2, 4, and 8.

Definition at line 21 of file Zxtar.cpp.

Here is the call graph for this function:



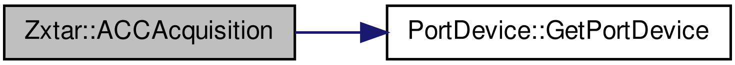
### .0.4.3 Member Function Documentation

#### .0.4.3.1 void Zxtar::ACCAcquisition ( )

Method for the Acceleration Acquisition

Definition at line 149 of file Zxtar.cpp.

Here is the call graph for this function:

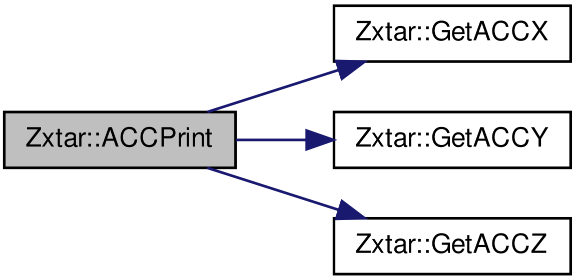


#### .0.4.3.2 void Zxtar::ACCPrint ( )

Method to print the Acceleartion Values

Definition at line 186 of file Zxtar.cpp.

Here is the call graph for this function:

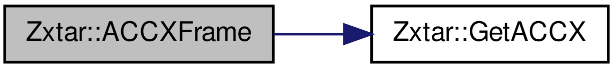


#### .0.4.3.3 void Zxtar::ACCXFrame ( )

Method for the adquisition of X-axis data frame

Definition at line 234 of file Zxtar.cpp.

Here is the call graph for this function:



#### .0.4.3.4 double Zxtar::ACCXFrameAverage ( )

Method to compute the X-axis data frame avarage

Definition at line 261 of file Zxtar.cpp.

#### .0.4.3.5 void Zxtar::ACCXFramePrint ( )

Method for printing the X-axis data frame

Definition at line 248 of file Zxtar.cpp.

#### .0.4.3.6 double Zxtar::ACCXFrameStdDeviation ( )

Method to compute the X-axis data frame standard deviation

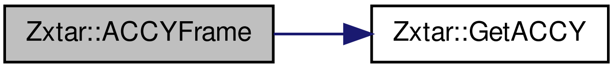
Definition at line 282 of file Zxtar.cpp.

#### .0.4.3.7 void Zxtar::ACCYFrame ( )

Method for the adquisition of Y-axis data frame

Definition at line 304 of file Zxtar.cpp.

Here is the call graph for this function:



#### .0.4.3.8 double Zxtar::ACCYFrameAverage ( )

Method to compute the Y-axis data frame avarage

Definition at line 331 of file Zxtar.cpp.

#### .0.4.3.9 void Zxtar::ACCYFramePrint ( )

Method for printing the Y-axis data frame

Definition at line 318 of file Zxtar.cpp.

#### .0.4.3.10 double Zxtar::ACCYFrameStdDeviation ( )

Method to compute the Y-axis data frame standard deviation

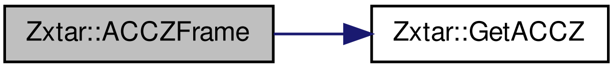
Definition at line 351 of file Zxtar.cpp.

#### .0.4.3.11 void Zxtar::ACCZFrame ( )

Method for the adquisition of Z-axis data frame

Definition at line 376 of file Zxtar.cpp.

Here is the call graph for this function:



#### .0.4.3.12 double **Zxtar::ACCZFrameAverage** ( )

Method to compute the Z-axis data frame avarage

Definition at line 403 of file Zxtar.cpp.

#### .0.4.3.13 void **Zxtar::ACCZFramePrint** ( )

Method for printing the Z-axis data frame

Definition at line 390 of file Zxtar.cpp.

#### .0.4.3.14 double **Zxtar::ACCZFrameStdDeviation** ( )

Method to compute the Z-axis data frame standard deviation

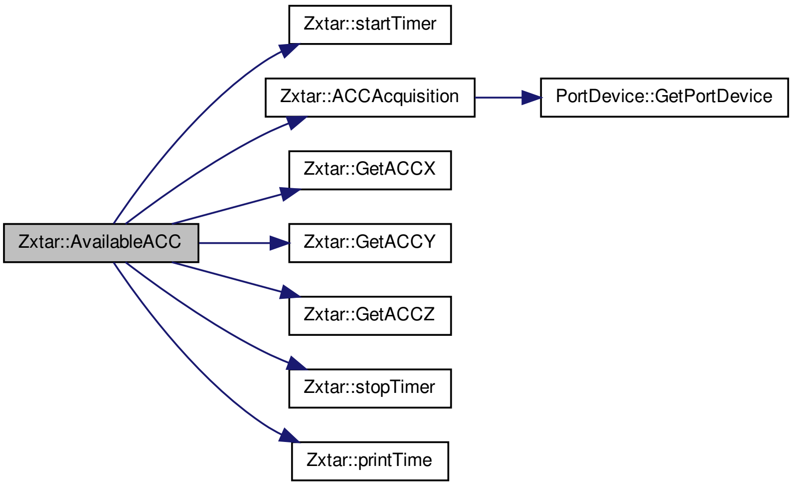
Definition at line 425 of file Zxtar.cpp.

#### .0.4.3.15 void **Zxtar::AvailableACC** ( )

To wait until the data is available by using a while loop

Definition at line 51 of file Zxtar.cpp.

Here is the call graph for this function:

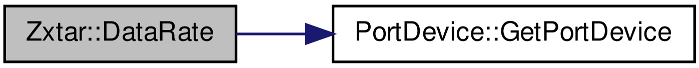


### .0.4.3.16 void **Zxtar::DataRate** ( int *setRate* )

Method to set the Data Rate

Definition at line 124 of file Zxtar.cpp.

Here is the call graph for this function:



### .0.4.3.17 double **Zxtar::GetACCX** ( )

Method to get X-axis Acceleration

Definition at line 171 of file Zxtar.cpp.

#### .0.4.3.18 double **Zxtar::GetACCXFrameAvarage** ( )

Method to get the X-axis data frame avarage

Definition at line 276 of file Zxtar.cpp.

#### .0.4.3.19 double **Zxtar::GetACCXFrameStdDeviation** ( )

Method to get the X-axis data frame standard deviation

Definition at line 298 of file Zxtar.cpp.

#### .0.4.3.20 double **Zxtar::GetACCY** ( )

Method to get Y-axis Acceleration

Definition at line 176 of file Zxtar.cpp.

#### .0.4.3.21 double **Zxtar::GetACCYFrameAvarage** ( )

Method to get the Y-axis data frame avarage

Definition at line 346 of file Zxtar.cpp.

#### .0.4.3.22 double **Zxtar::GetACCYFrameStdDeviation** ( )

Method to get the Y-axis data frame standard deviation

Definition at line 367 of file Zxtar.cpp.

#### .0.4.3.23 double **Zxtar::GetACCZ** ( )

Method to get Z-axis Acceleration

Definition at line 181 of file Zxtar.cpp.

#### .0.4.3.24 double **Zxtar::GetACCZFrameAvarage** ( )

Method to get the Z-axis data frame avarage

Definition at line 418 of file Zxtar.cpp.

#### .0.4.3.25 double **Zxtar::GetACCZFrameStdDeviation** ( )

Method to get the Z-axis data frame standard deviation

Definition at line 445 of file Zxtar.cpp.

#### .0.4.3.26 int **Zxtar::Kbhit** ( void )

Method for the keyboard hit

Definition at line 70 of file Zxtar.cpp.

#### .0.4.3.27 void **Zxtar::printTime** ( double *duration* ) `[static]`

Method to print the timer

Definition at line 218 of file Zxtar.cpp.

#### .0.4.3.28 void **Zxtar::SetFrameWord** ( unsigned int *framevalue* )

Method to set the frame word of the acceleration readings

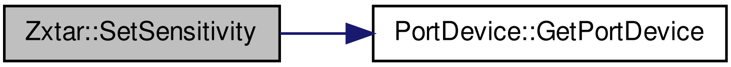
Definition at line 226 of file Zxtar.cpp.

#### .0.4.3.29 void **Zxtar::SetSensitivity** ( int *gSensitivity* )

Method to select the g-range for the accelerometer Sensor Board BY SENDING A g0 g1 or g2 you are able to change the SENSITIVITY g0 = 8g ; g1 = 2g ; g2 = 4g

Definition at line 96 of file Zxtar.cpp.

Here is the call graph for this function:



#### .0.4.3.30 void **Zxtar::startTimer** ( )

Method to start the timer

Definition at line 197 of file Zxtar.cpp.

#### .0.4.3.31 double **Zxtar::stopTimer** ( )

Method to stop the timer

Definition at line 202 of file Zxtar.cpp.

### .0.4.4 Member Data Documentation

#### .0.4.4.1 PortDevice Zxtar::ZSTAR

Data Member for the use fo the **PortDevice** (p. 6) class with the sensor board

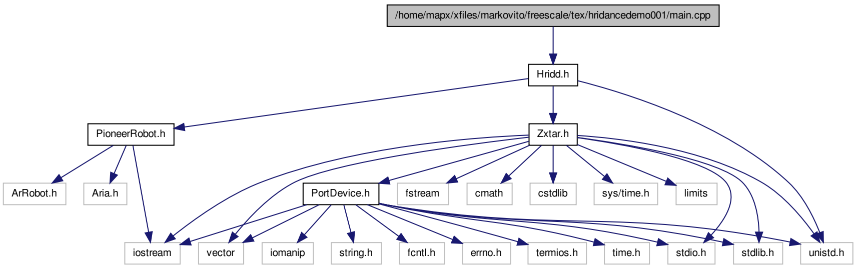
Definition at line 88 of file Zxtar.h.

The documentation for this class was generated from the following files:

- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**Zxtar.h**
- /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/**Zxtar.cpp**

## .0.5 /home/mapx/xfiles/markovito/freescale/tex/hridancedemo001/main.cpp File - Reference

`#include "Hridd.h"` Include dependency graph for main.cpp:



### Functions

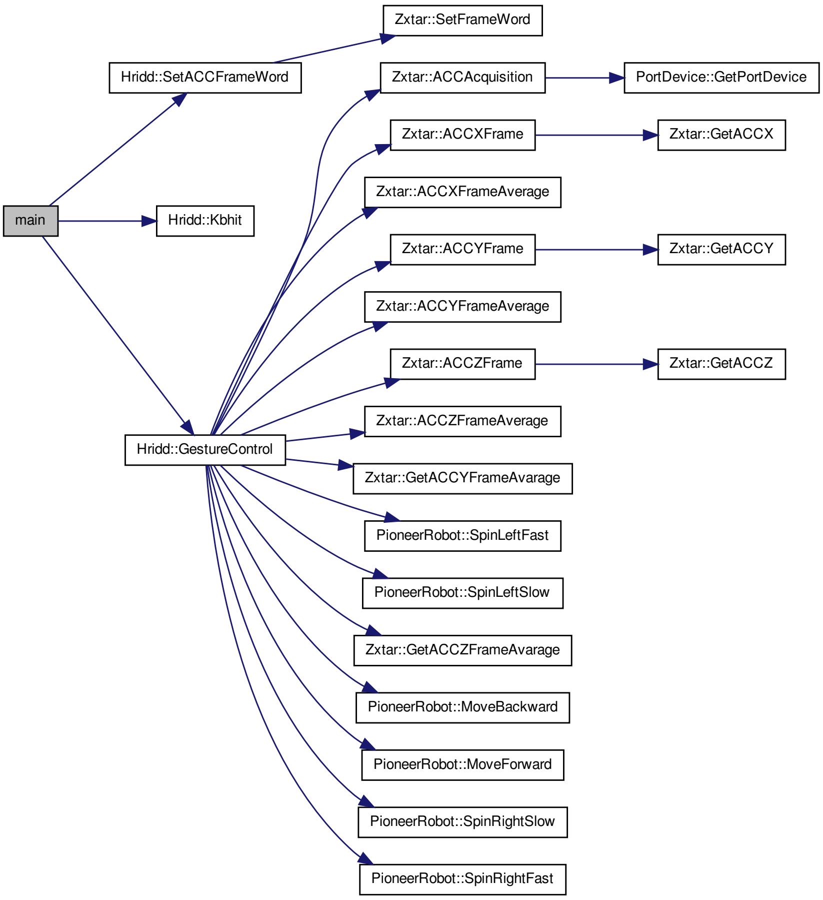
- int **main** ()

### .0.5.1 Function Documentation

#### .0.5.1.1 int main ( )

Definition at line 3 of file main.cpp.

Here is the call graph for this function:


Zxtar::SetFrameWord
Hridd::SetACCFrameWord
Zxtar::ACCAcquisition
PortDevice::GetPortDevice
Zxtar::ACCXFrame
Zxtar::GetACCX
Zxtar::ACCXFrameAverage
main
Hridd::Kbhit
Zxtar::ACCYFrame
Zxtar::GetACCY
Zxtar::ACCYFrameAverage
Zxtar::ACCZFrame
Zxtar::GetACCZ
Zxtar::ACCZFrameAverage
Hridd::GestureControl
Zxtar::GetACCYFrameAvarage
PioneerRobot::SpinLeftFast
PioneerRobot::SpinLeftSlow
Zxtar::GetACCZFrameAvarage
PioneerRobot::MoveBackward
PioneerRobot::MoveForward
PioneerRobot::SpinRightSlow
PioneerRobot::SpinRightFast