Axel-tilt

1.7.0.11

by Teodor Krastev for Imperial College London

Chapter 1

Axel tilt Introduction

Axel tilt is an application to control the tilt of a platform the navigator experiment sits on. Tilting the platform simulates acceleration of the experiment by using a component of earth acceleration: the resulting acceleration is proportional to the angle of the tilt. The actual mechanical control is implemented via two step motors with micro steps moving one side of the platform up and down. The software provides:

- Means to adjust initial horizontal orientation of the platform, which is saved and retrieved after start.
- Tilting manually the platform to a target tilt/acceleration or gradually.
- Tilting the platform following pre-programmed patterns of behaviour in repeated manner.
- The application can be connected to another application Axel Show in order to report its current position on demand

Chapter 2

Namespace Index

2.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 Axel_tilt Namespace Reference

Classes

- class App

  Interaction logic for App.xaml
- class Horizontal

  Intitial (horizontal) position & default speed
- class MainWindow

  Interaction logic for MainWindow.xaml
- class Motor

  Motor abstraction presenting the motor controller into the tilt platform funtionality
- class Tilt

  Class to be used as abstraction of controlling the tilt of the platform using motors abstractions

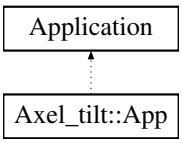
Chapter 7

Class Documentation

7.1 Axel_tilt::App Class Reference

Interaction logic for App.xaml

Inheritance diagram for Axel_tilt::App:



7.1.1 Detailed Description

Interaction logic for App.xaml

Definition at line 14 of file App.xaml.cs.

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

- App.xaml.cs

7.2 Axel_tilt.Horizontal Class Reference

Intitial (horizontal) position & default speed

Properties

- double posA `[get, set]`
- double posB `[get, set]`
- double speed `[get, set]`
- double OffsetDodging `[get, set]`

7.2.1 Detailed Description

Intitial (horizontal) position & default speed

Definition at line 21 of file Tilt.cs.

7.2.2 Property Documentation

7.2.2.1 OffsetDodging

```
double Axel_tilt.Horizontal.OffsetDodging  [get], [set]
```

Definition at line 26 of file Tilt.cs.

7.2.2.2 posA

```
double Axel_tilt.Horizontal.posA  [get], [set]
```

Definition at line 23 of file Tilt.cs.

7.2.2.3 posB

```
double Axel_tilt.Horizontal.posB  [get], [set]
```

Definition at line 24 of file Tilt.cs.

7.2.2.4 speed

```
double Axel_tilt.Horizontal.speed  [get], [set]
```

Definition at line 25 of file Tilt.cs.

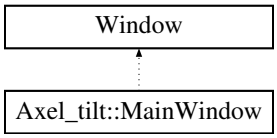
The documentation for this class was generated from the following file:

- Tilt.cs

7.3 Axel_tilt::MainWindow Class Reference

Interaction logic for MainWindow.xaml

Inheritance diagram for Axel_tilt::MainWindow:



7.3.1 Detailed Description

Interaction logic for MainWindow.xaml

Definition at line 29 of file MainWindow.xaml.cs.

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

- MainWindow.xaml.cs

7.4 Axel_tilt.Motor Class Reference

Motor abstraction presenting the motor controller into the tilt platform funtionality

Public Member Functions

- Motor (string Name="")
  Class constructor
- char letter ()
- void Open ()
  Open the device controller
- bool InRange (double dist)
  Check for distance range validity
- bool Home (bool wait=true)
  Go to Home position
- bool Zero (bool setHoriz=false)
  Set corrent position to be Zero
- bool Stop ()
  Stop whatever has been executed
- bool Wait2stop ()
  Wrapper for "command_wait_for_stop" controller command
- status_t Status ()
  Get controller status
- List< string > ListStatus ()
  The same status in easy to print (log) form

- double GetPosition ()

  Get current position [mm]
- bool SetSpeed (double speed)

  Set movement speed [mm/s]
- bool SetBacklash (bool bl)

  Set backlash compensation ON/OFF ON - more accurate position in back direction, can be very slow for small steps OFF - Forward is fine (calibration is valid); backward is not accurate, but useful for scanning if only the change matters
- double steps2dist (int steps, int usteps)

  Convert from step:usteps to distance [mm]
- int[ ] dist2steps (double dist)

  Reverse to the upper method -> from distance to steps:usteps
- bool MoveD (double dist, bool wait=true)

  Goto (move) to specific position in [mm]

Properties

- int devIdx `[get]`
- double horizOffset `[get]`

7.4.1 Detailed Description

Motor abstraction presenting the motor controller into the tilt platform funtionality

Definition at line 32 of file Tilt.cs.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 Motor()

```
Axel_tilt.Motor.Motor (
            string Name = "" )
```

Class constructor

Parameters

Name	

Definition at line 50 of file Tilt.cs.

7.4.3 Member Function Documentation

7.4.3.1 dist2steps()

```
int [] Axel_tilt.Motor.dist2steps (
            double dist )
```

Reverse to the upper method -> from distance to steps:usteps

Parameters

dist	[mm]

Returns

[steps, usteps]

Definition at line 267 of file Tilt.cs.

7.4.3.2 GetPosition()

```
double Axel_tilt.Motor.GetPosition ( )
```

Get current position [mm]

Returns

Definition at line 174 of file Tilt.cs.

7.4.3.3 Home()

```
bool Axel_tilt.Motor.Home (
            bool wait = true )
```

Go to Home position

Parameters

wait	Synchronious or Asynchro... execution

Returns

Definition at line 89 of file Tilt.cs.

7.4.3.4 InRange()

```
bool Axel_tilt.Motor.InRange (
            double dist )
```

Check for distance range validity

Parameters

dist	

Returns

Definition at line 79 of file Tilt.cs.

7.4.3.5 letter()

```
char Axel_tilt.Motor.letter ( )
```

Definition at line 59 of file Tilt.cs.

7.4.3.6 ListStatus()

```
List<string> Axel_tilt.Motor.ListStatus ( )
```

The same status in easy to print (log) form

Returns

Definition at line 156 of file Tilt.cs.

7.4.3.7 MoveD()

```
bool Axel_tilt.Motor.MoveD (
            double dist,
            bool wait = true )
```

Goto (move) to specific position in [mm]

Parameters

dist	[mm]
wait	Synchronious or Asynchro... execution

Returns

OK

Definition at line 298 of file Tilt.cs.

7.4.3.8 Open()

```
void Axel_tilt.Motor.Open ( )
```

Open the device controller

Definition at line 67 of file Tilt.cs.

7.4.3.9 SetBacklash()

```
bool Axel_tilt.Motor.SetBacklash (
            bool bl )
```

Set backlash compensation ON/OFF ON - more accurate position in back direction, can be very slow for small steps OFF - Forward is fine (calibration is valid); backward is not accurate, but useful for scanning if only the change matters

Parameters

bl	Backlash compensation

Returns

OK

Definition at line 212 of file Tilt.cs.

7.4.3.10 SetSpeed()

```
bool Axel_tilt.Motor.SetSpeed (
            double speed )
```

Set movement speed [mm/s]

Parameters

speed	[mm/s]

Returns

OK

Definition at line 188 of file Tilt.cs.

7.4.3.11 Status()

```
status_t Axel_tilt.Motor.Status ( )
```

Get controller status

Returns

Definition at line 143 of file Tilt.cs.

7.4.3.12 steps2dist()

```
double Axel_tilt.Motor.steps2dist (
            int steps,
            int usteps )
```

Convert from step:usteps to distance [mm]

Parameters

steps	steps
usteps	microsteps

Returns

[mm]

Definition at line 257 of file Tilt.cs.

7.4.3.13 Stop()

```
bool Axel_tilt.Motor.Stop ( )
```

Stop whatever has been executed

Returns

OK

Definition at line 119 of file Tilt.cs.

7.4.3.14 Wait2stop()

```
bool Axel_tilt.Motor.Wait2stop ( )
```

Wrapper for "command_wait_for_stop" controller command

Returns

OK

Definition at line 131 of file Tilt.cs.

7.4.3.15 Zero()

```
bool Axel_tilt.Motor.Zero (
            bool setHoriz = false )
```

Set corrent position to be Zero

Parameters

setHoriz	

Returns

OK

Definition at line 104 of file Tilt.cs.

7.4.4 Property Documentation

7.4.4.1 devIdx

```
int Axel_tilt.Motor.devIdx  [get]
```

Definition at line 35 of file Tilt.cs.

7.4.4.2 horizOffset

```
double Axel_tilt.Motor.horizOffset  [get]
```

Definition at line 36 of file Tilt.cs.

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

- Tilt.cs

7.5 Axel_tilt.Tilt Class Reference

Class to be used as abstraction of controlling the tilt of the platform using motors abstractions

Public Member Functions

- Tilt ()
  Class contructor
- void HomeAndZero ()
  Go home and set to be a zero position
- void SetHorizontal (double posA, double posB)
  Initialization of initial horizontal position
- void Stop ()
  Stop both motors
- void Close ()
  Close controller for both motors
- double dist2tilt (double dist)
  Convert from dist[mm] to tilt[mrad]
- double tilt2dist (double tilt)
- double accel2tilt (double accel)
  Convert acceleration [mg] to tilt [mrad]
- double tilt2accel (double tilt)
  Convert tilt [mrad] to acceleration [mg]
- bool MoveDist (double dist, bool wait=true)
  Move to new position in [mm]
- bool MoveAccel (double accel, bool wait=true)
  Move to new acceleration
- void Wait4Stop ()
  Wait the last movement to be concluded
- bool SetSpeed (double speed=-1)

Set speed of movement [mm/s]

- bool SetBacklash (bool bl)

  Set both mottors backlash
- double GetPosition ()

  Get the tilt position in mm
- double GetAccel ()

  Get the tilt position in mg
- void DoEvents ()
- object ExitFrame (object f)
- delegate void EndHandler (bool userCancel)
- delegate void LogHandler (string txt)
- delegate void MoveHandler (Point target)
- void SingleMove (double fromPos, double toPos, double time)

  start async movement to toPos, with a speed so to take time
- void NextStep (object sender, EventArgs e)

  Execute next step (move) in pattern movement
- void MoveInPattern (double[,] ptrn, double period, double ampl, double offset)

  Move in pattern: first pair (0, init.pos) second - (time1, second.pos) ...etc. [time,ampl] in [s,mg] units

Public Attributes

- Motor mA
- bool AutoBacklash = true
- Horizontal horizontal
- Stopwatch sw = new Stopwatch()
- bool request2Stop = false
- DispatcherTimer dTimer

Static Public Attributes

- const double tilt_arm = 510.003
- const double MemsCorr_A = 1.071
- const double MemsCorr_B = -8.034
- const double minSpeed = 0.1

Protected Member Functions

- void EndEvent (bool userCancel)
- void LogEvent (string txt)
- void MoveEvent (Point target)

Properties

- bool busy `[get]`
- bool MemsCorr `[get, set]`
- double accelSpeed `[get, set]`

  Get/Set acceleration speed [mg/s]

Events

- EndHandler OnEnd
- LogHandler OnLog
- MoveHandler OnMove

7.5.1 Detailed Description

Class to be used as abstraction of controlling the tilt of the platform using motors abstractions

Definition at line 310 of file Tilt.cs.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 Tilt()

```
Axel_tilt.Tilt.Tilt ( )
```

Class contructor

Definition at line 329 of file Tilt.cs.

7.5.3 Member Function Documentation

7.5.3.1 accel2tilt()

```
double Axel_tilt.Tilt.accel2tilt (
            double accel )
```

Convert acceleration [mg] to tilt [mrad]

Parameters

accel	[mg]

Returns

[mrad]

Definition at line 596 of file Tilt.cs.

7.5.3.2 Close()

```
void Axel_tilt.Tilt.Close ( )
```

Close controller for both motors

Definition at line 560 of file Tilt.cs.

7.5.3.3 dist2tilt()

```
double Axel_tilt.Tilt.dist2tilt (
            double dist )
```

Convert from dist[mm] to tilt[mrad]

Parameters

dist	[mm]

Returns

[mrad]

Definition at line 576 of file Tilt.cs.

7.5.3.4 DoEvents()

```
void Axel_tilt.Tilt.DoEvents ( )
```

Definition at line 731 of file Tilt.cs.

7.5.3.5 EndEvent()

```
void Axel_tilt.Tilt.EndEvent (
            bool userCancel )  [protected]
```

Definition at line 746 of file Tilt.cs.

7.5.3.6 EndHandler()

```
delegate void Axel_tilt.Tilt.EndHandler (
            bool userCancel )
```

7.5.3.7 ExitFrame()

```
object Axel_tilt.Tilt.ExitFrame (
            object f )
```

Definition at line 738 of file Tilt.cs.

7.5.3.8 GetAccel()

```
double Axel_tilt.Tilt.GetAccel ( )
```

Get the tilt position in mg

Returns

[mg]

Definition at line 725 of file Tilt.cs.

7.5.3.9 GetPosition()

```
double Axel_tilt.Tilt.GetPosition ( )
```

Get the tilt position in mm

Returns

[mm]

Definition at line 716 of file Tilt.cs.

7.5.3.10 HomeAndZero()

```
void Axel_tilt.Tilt.HomeAndZero ( )
```

Go home and set to be a zero position

Definition at line 525 of file Tilt.cs.

7.5.3.11 LogEvent()

```
void Axel_tilt.Tilt.LogEvent (
            string txt )  [protected]
```

Definition at line 753 of file Tilt.cs.

7.5.3.12 LogHandler()

```
delegate void Axel_tilt.Tilt.LogHandler (
            string txt )
```

7.5.3.13 MoveAccel()

```
bool Axel_tilt.Tilt.MoveAccel (
            double accel,
            bool wait = true )
```

Move to new acceleration

Parameters

accel	[mg];
wait	for manual oper -> true; for moving patterns -> false

Returns

Definition at line 662 of file Tilt.cs.

7.5.3.14 MoveDist()

```
bool Axel_tilt.Tilt.MoveDist (
            double dist,
            bool wait = true )
```

Move to new position in [mm]

Parameters

dist	[mm]
wait	Sychronious/Asychronious call

Returns

Definition at line 646 of file Tilt.cs.

7.5.3.15 MoveEvent()

```
void Axel_tilt.Tilt.MoveEvent (
            Point target )  [protected]
```

Definition at line 760 of file Tilt.cs.

7.5.3.16 MoveHandler()

```
delegate void Axel_tilt.Tilt.MoveHandler (
            Point target )
```

7.5.3.17 MoveInPattern()

```
void Axel_tilt.Tilt.MoveInPattern (
            double ptrn[,],
            double period,
            double ampl,
            double offset )
```

Move in pattern: first pair (0, init.pos) second - (time1, second.pos) ...etc. [time,ampl] in [s,mg] units

Parameters

ptrn	
period	Defines horizontal scale
ampl	Defines vertical scale
offset	Vertical shift

Definition at line 814 of file Tilt.cs.

7.5.3.18 NextStep()

```
void Axel_tilt.Tilt.NextStep (
            object sender,
            EventArgs e )
```

Execute next step (move) in pattern movement

Parameters

sender	
e	

Definition at line 787 of file Tilt.cs.

7.5.3.19 SetBacklash()

```
bool Axel_tilt.Tilt.SetBacklash (
            bool bl )
```

Set both mottors backlash

Parameters

bl	

Returns

Definition at line 707 of file Tilt.cs.

7.5.3.20 SetHorizontal()

```
void Axel_tilt.Tilt.SetHorizontal (
            double posA,
            double posB )
```

Initialization of initial horizontal position

Parameters

posA	
posB	

Definition at line 540 of file Tilt.cs.

7.5.3.21 SetSpeed()

```
bool Axel_tilt.Tilt.SetSpeed (
            double speed = -1 )
```

Set speed of movement [mm/s]

Parameters

speed	

Returns

Definition at line 692 of file Tilt.cs.

7.5.3.22 SingleMove()

```
void Axel_tilt.Tilt.SingleMove (
            double fromPos,
            double toPos,
            double time )
```

start async movement to toPos, with a speed so to take time

Parameters

fromPos	[mg]
toPos	[mg]
time	[s]

Definition at line 771 of file Tilt.cs.

7.5.3.23 Stop()

```
void Axel_tilt.Tilt.Stop ( )
```

Stop both motors

Definition at line 552 of file Tilt.cs.

7.5.3.24 tilt2accel()

```
double Axel_tilt.Tilt.tilt2accel (
            double tilt )
```

Convert tilt [mrad] to acceleration [mg]

Parameters

tilt	[mrad]

Returns

[mg]

Definition at line 609 of file Tilt.cs.

7.5.3.25 tilt2dist()

```
double Axel_tilt.Tilt.tilt2dist (
            double tilt )
```

Parameters

tilt	

Returns

Definition at line 586 of file Tilt.cs.

7.5.3.26 Wait4Stop()

```
void Axel_tilt.Tilt.Wait4Stop ( )
```

Wait the last movement to be concluded

Definition at line 671 of file Tilt.cs.

7.5.4 Member Data Documentation

7.5.4.1 AutoBacklash

```
bool Axel_tilt.Tilt.AutoBacklash = true
```

Definition at line 313 of file Tilt.cs.

7.5.4.2 dTimer

```
DispatcherTimer Axel_tilt.Tilt.dTimer
```

Definition at line 781 of file Tilt.cs.

7.5.4.3 horizontal

```
Horizontal Axel_tilt.Tilt.horizontal
```

Definition at line 323 of file Tilt.cs.

7.5.4.4 mA

```
Motor Axel_tilt.Tilt.mA
```

Definition at line 312 of file Tilt.cs.

7.5.4.5 MemsCorr_A

```
const double Axel_tilt.Tilt.MemsCorr_A = 1.071  [static]
```

Definition at line 320 of file Tilt.cs.

7.5.4.6 MemsCorr_B

```
const double Axel_tilt.Tilt.MemsCorr_B = -8.034  [static]
```

Definition at line 321 of file Tilt.cs.

7.5.4.7 minSpeed

```
const double Axel_tilt.Tilt.minSpeed = 0.1  [static]
```

Definition at line 322 of file Tilt.cs.

7.5.4.8 request2Stop

```
bool Axel_tilt.Tilt.request2Stop = false
```

Definition at line 777 of file Tilt.cs.

7.5.4.9 sw

```
Stopwatch Axel_tilt.Tilt.sw = new Stopwatch()
```

Definition at line 324 of file Tilt.cs.

7.5.4.10 tilt_arm

```
const double Axel_tilt.Tilt.tilt_arm = 510.003  [static]
```

Definition at line 319 of file Tilt.cs.

7.5.5 Property Documentation

7.5.5.1 accelSpeed

```
double Axel_tilt.Tilt.accelSpeed  [get], [set]
```

Get/Set acceleration speed [mg/s]

Definition at line 621 of file Tilt.cs.

7.5.5.2 busy

```
bool Axel_tilt.Tilt.busy  [get]
```

Definition at line 314 of file Tilt.cs.

7.5.5.3 MemsCorr

```
bool Axel_tilt.Tilt.MemsCorr  [get], [set]
```

Definition at line 315 of file Tilt.cs.

7.5.6 Event Documentation

7.5.6.1 OnEnd

`EndHandler Axel_tilt.Tilt.OnEnd`

Definition at line 745 of file Tilt.cs.

7.5.6.2 OnLog

`LogHandler Axel_tilt.Tilt.OnLog`

Definition at line 752 of file Tilt.cs.

7.5.6.3 OnMove

`MoveHandler Axel_tilt.Tilt.OnMove`

Definition at line 759 of file Tilt.cs.

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

- Tilt.cs

Chapter 8

File Documentation

8.1 App.xaml.cs File Reference

Classes

- class Axel_tilt::App

  Interaction logic for App.xaml

Namespaces

- Axel_tilt

8.2 MainWindow.xaml.cs File Reference

Classes

- class Axel_tilt::MainWindow

  Interaction logic for MainWindow.xaml

Namespaces

- Axel_tilt

8.3 README.md File Reference

8.4 Tilt.cs File Reference

Classes

- class Axel_tilt.Horizontal

  Intitial (horizontal) position & default speed
- class Axel_tilt.Motor

  Motor abstraction presenting the motor controller into the tilt platform funtionality
- class Axel_tilt.Tilt

  Class to be used as abstraction of controlling the tilt of the platform using motors abstractions

Namespaces

- Axel_tilt

Index