

# Full Source Codes of **Sensors Set**

Sensor Programming for Windows Phone

● Sei Yakushiji



FREE AND NO SUPPORT ONLINE BOOK

# Table of Contents

1. Introduction .....	3
1.1 The Phone's sensors .....	3
1.2 Features of "Sensors Set" .....	5
2. MainPage .....	8
2.1 Layout of "Menu" page .....	8
2.2 Programming of "Menu" page .....	10
3. Accelerometer .....	12
3.1 Layout of "Accelerometer" page .....	12
3.2 Programming of "Accelerometer" page .....	17
4. Gyroscope .....	24
4.1 Layout of "Gyroscope" page .....	24
4.2 Programming of "Gyroscope" page .....	33
5. Compass .....	43
5.1 Layout of "Compass" page .....	43
5.2 Programming of "Compass" page .....	49
6. Combined Motion API .....	56
6.1 Layout of "Combined Motion" page .....	56
6.2 Programming of "Combined Motion" page .....	64
7. Features of "今日も歩こう！" .....	71
7.1 How to use the application .....	71
8. MainPage .....	73
8.1 Layout of "Menu" page .....	73
8.2 Programming of "Menu" page .....	75
9. Recording data .....	78
9.1 Layout of "RecordData" page .....	78
9.2 Programming of "RecordData" page .....	81

10. The list of data .....	90
10.1 Layout of "ShowDataList" page .....	90
10.2 Programming of "ShowDataList" page .....	92
11. More information for the developers .....	96
11.1 Visual Studio project files .....	96
11.2 Sensor applications .....	96
11.3 Online books and article .....	98
Copyright .....	99
Author and Publisher .....	99

*The applications in this document are Silverlight applications written with Visual Basic, developed via those tools.  
Windows 7 Professional (32 bit) SP1 (Japanese), Microsoft Visual Studio 2010 Ultimate SP1 (English), Windows Phone SDK 7.1 (English).  
All graphics were created by Microoft Expression Design.*

# 1. Introduction

## 1.1 The Phone's sensors

To develop the applications, it is important for us to use APIs and the sensors.

A mobile device consists of various sensors, touch screen sensor, photo sensor, CMOS image sensor, GPS sensor, pressure sensor.

Programming for such a device is exactly to process data from the sensor at a given time.

Windows Phone supports 4 sensors, accelerometer, compass, gyroscope, and combined motion API.

We, developer can use the classes in Microsoft.Devices.Sensors namespace. (●fig 1)

●fig 1 The information from 4 sensors.

### Properties of AccelerometerReading Structure

Timestamp	timestamp	DateTimeOffset	
Acceleration (Vector3)	X	X value in gravitational units	Double
	Y	Y value in gravitational units	Double
	Z	Z value in gravitational units	Double

### Properties of CompassReading Structure

Timestamp	timestamp	DateTimeOffset
HeadingAccuracy	the accuracy of the compass heading readings (the degrees)	Double
MagnetometerReading	the raw magnetometer reading (microteslas)	Vector3
MagneticHeading	the heading measured from the Earth's magnetic north (the degrees)	Double
TrueHeading	the heading from the Earth's geographic north (the degrees)	Double

### Properties of GyroscopeReading Structure

Timestamp	timestamp	DateTimeOffset
RotationRate	the rotational velocity around each axis (rad/sec)	Vector3

### Properties of MotionReading Structure

Timestamp	timestamp	DateTimeOffset
Attitude	the attitude of the device (radians)	AttitudeReading
DeviceAcceleration	the linear acceleration of the device (gravitational units)	Vector3
DeviceRotationRate	the rotational velocity of the device (rad/sec)	Vector3
Gravity	the gravity vector	Vector3

### Properties of AttitudeReading Structure

Timestamp	timestamp	DateTimeOffset
Pitch	the pitch of the attitude reading (radians)	Single
Roll	the roll of the attitude reading (radians)	Single
Yaw	the yaw of the attitude reading (radians)	Single
Quaternion	the quaternion representation	Quaternion
RotationMatrix	the matrix representation	Matrix

## Accelerometer

The Information of an accelerometer indicated the magnitude and direction of forces when the device moves.

## Compass

It provides the orientation of the device on a map, or the magnetic field of the earth. The sensor accuracy is depending on the device, calibration is needed if the value of headingAccuracy is less than 15.

## Gyroscope

It provides the information about the rotational velocity per second around 3 axes of the device.

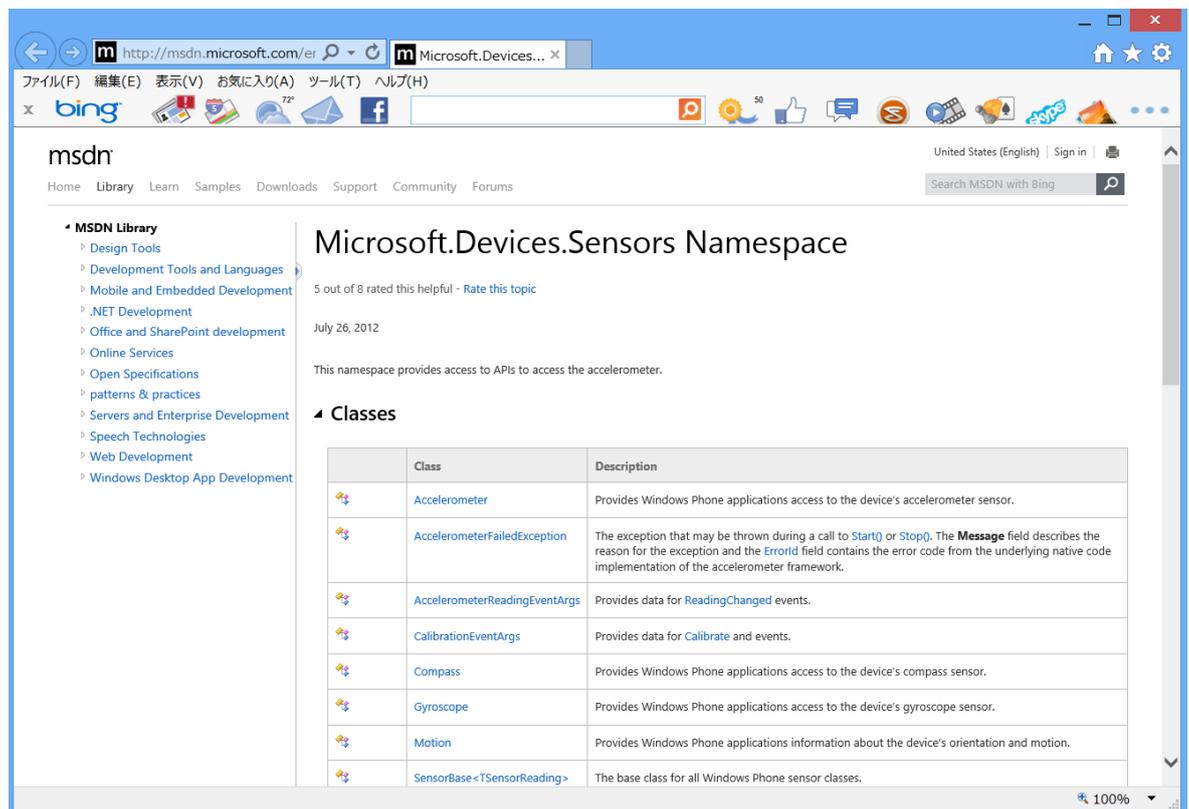
## The Combined Motion API

It provides the information of the attitude, the linear acceleration, the rotational velocity, the gravity, and a timestamp that represent the movement of the device.

You can get more information of Microsoft.Devices.Sensors Namespace in "msdn Library". (●fig 2)

●fig 2 msdn Library, Microsoft.Devices.Sensors Namespace

[http://msdn.microsoft.com/en-us/library/microsoft.devices.sensors\(VS.92\).aspx](http://msdn.microsoft.com/en-us/library/microsoft.devices.sensors(VS.92).aspx)



The screenshot shows a web browser window displaying the MSDN Library page for the Microsoft.Devices.Sensors Namespace. The page title is "Microsoft.Devices.Sensors Namespace" and it includes a rating of "5 out of 8 rated this helpful". The page content includes a description: "This namespace provides access to APIs to access the accelerometer." and a list of classes with their descriptions.

Class	Description
Accelerometer	Provides Windows Phone applications access to the device's accelerometer sensor.
AccelerometerFailedException	The exception that may be thrown during a call to Start() or Stop(). The Message field describes the reason for the exception and the ErrorId field contains the error code from the underlying native code implementation of the accelerometer framework.
AccelerometerReadingEventArgs	Provides data for ReadingChanged events.
CalibrationEventArgs	Provides data for Calibrate and events.
Compass	Provides Windows Phone applications access to the device's compass sensor.
Gyroscope	Provides Windows Phone applications access to the device's gyroscope sensor.
Motion	Provides Windows Phone applications information about the device's orientation and motion.
SensorBase<TSensorReading>	The base class for all Windows Phone sensor classes.

## 1.2 Features of "Sensors Set"



● fig 4 Sensors Set Ver.0.8 (En)

Windows Phone supports accelerometer, gyroscope, compass, and the combined motion API.

"Sensors Set Ver.0.8" is for the developers to use these sensors in Windows Phone application. (● fig 4)

For each sensor, you can set timespan (milliseconds) and decimals of accuracy (integer / one decimal / two decimals).

And you can receive various data like this:

### **Accelerometer**

- (1) The values of acceleration on the X, Y, and Z axes.
- (2) The angles.
- (3) An object moves in a circle when you move your device.

### **Gyroscope**

- (1) The first update time and the last update time.
- (2) The values and the angles of the rotational acceleration.
- (3) The values and the angles of the cumulative rotation.
- (4) The bar charts to represent the angles of the rotational acceleration and cumulative rotation.
- (5) A counter is displayed when the values of the rotational acceleration is or over the value that you have set in accuracy for the counter.

A gyroscope is to measure rotational velocity. If you need the attitude of the device, use "Combined Motion API".

### **Compass**

- (1) The raw magnetometer.
- (2) The heading accuracy of the compass.
- (3) A current axis to measure magnetometer (using accelerometer).
- (4) The heading in degrees from the Earth's magnetic north.
- (5) The heading in degrees from the Earth's geographic north.

(6) The hands rotate in degrees.

### **Combined Motion API**

- (1) The linear acceleration on the X, Y, and Z axes.
- (2) The rotational velocity.
- (3) The gravity.
- (4) The attitude (pitch, roll, and yaw) of the device.
- (5) The angles of pitch, roll, and yaw.
- (6) The hands point the angles.

If you reset values, go back to Main Page and choose a sensor again.

Tap '?' button on the Main Page and wait a moment, Help page opens. If you need the Visual Studio project file, see page 96 in this book.

This application is developed for Windows Phone that supports these sensors.

本書で取り上げているアプリは、日本国内で公開中の、「センサー計測セット Ver. 0.8」の英語版です。搭載機能は、日本語版と同じですが、日本語版の XAML とヘルプを英語化し、コードの品質を向上させたものです。

このアプリは、Windows Phone に搭載されている、加速度センサー、ジャイロスコープセンサー、コンパスセンサー、複合モーションによって、各種データを計測するツールです。Windows Phone センサーアプリ開発者のために作られました。

操作は簡単で、「開始」ボタンと「停止」ボタンをタップするだけです。計測間隔、精度（整数、小数点以下 1 桁、小数点以下 2 桁）、ジャイロスコープについては検出精度を設定できます。

このツールは、次のデータを計測／表示します。

#### **加速度センサー**

- (1) X, Y, Z の各軸の加速度
- (2) X, Y, Z の各軸の傾斜角度
- (3) X, Y の値に応じて、2次元平面上のオブジェクトの動きを視覚的に表示

#### **ジャイロスコープセンサー**

- (1) 計測開始時間、終了時間
- (2) X, Y, Z の各軸の角速度および、角度
- (3) 累積値および、累積角度

- (4) X, Y, Z の計測角度と累積角度を棒グラフで表示
- (5) 計測値および累積値が「検出」設定の数値を超えた時にカウントして、カウンターを表示

### コンパスセンサー

- (1) X, Y, Z の各軸の磁気強度
- (2) 方位測定値の精度
- (3) 計測軸（加速度センサー併用）
- (4) 地磁気上の角度
- (5) 地図上の角度
- (6) 地磁気上の角度および地図上の角度を、視覚的に表示

### 複合モーション

- (1) 重力加速度を除く加速度
- (2) 角速度
- (3) 重力方向
- (4) ピッチ、ロール、ヨーの計測値
- (5) ピッチ ロール ヨーの角度
- (6) ピッチ ロール ヨーの角度を、時計回りで視覚的に表示

※設定や値をリセットしたい時は、メインページに戻って、メニューを再選択してください。  
※本アプリは、アプリ公開時点で国内販売されている Windows Phone 「au IS12T」を標準として開発しています。この機種は、各センサーをサポートしています。センサーをサポートしていない機種では、使用できる機能が限られます。

#### CAUTION

This application uses accelerometer, gyroscope, compass, and combined motion API of Windows Phone.

Please note that the developer without any warranty for the result of this application.

Be careful when you shake your device. Don't use where the device would slip and fall to the floor.

## 2. MainPage

### 2.1 Layout of "Menu" page

Choose a sensor that you use and tap its name. (●fig 5)

#### **Accelerometer**

You can get the acceleration of the device.

#### **Gyroscope**

You can get the rotational velocity of the device.

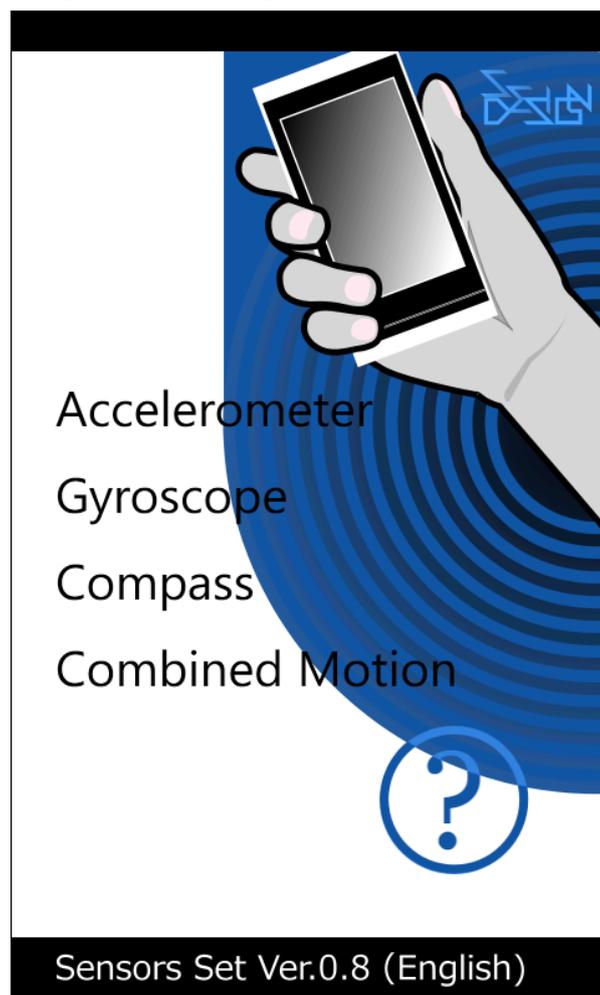
#### **Compass**

You can get the information about the magnetic field around the device.

#### **Combined Motion API**

You can get the information about the orientation and movement of the device.

●fig 5 "Menu" Page



The XAML markup code is like this. (●list 1)

●list 1 A part of MainPage.xaml

```
<phone:PhoneApplicationPage
    x:Class="SensorsSet.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="White">
        <!--<Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions-->

        <!--TitlePanel contains the name of the application and page title-->
        <!--<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
            <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION"
Style="{StaticResource PhoneTextNormalStyle}"/>
            <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
        </StackPanel-->

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">
            <Image Height="768" HorizontalAlignment="Left" Margin="0,0,0,0"
Name="Image1" Stretch="Fill" VerticalAlignment="Top" Width="480"
Source="/SensorsSet/component/Image/MainPageBackE.png" />
            <TextBlock Height="65" HorizontalAlignment="Left" Margin="35,260,0,0"
Name="AccelerometerButton" Text="Accelerometer" VerticalAlignment="Top"

```

```

Width="340" FontSize="40" FontWeight="Normal" Foreground="Black" />
    <TextBlock Height="65" HorizontalAlignment="Left" Margin="35,330,0,0"
Name="GyroscopeButton" Text="Gyroscope" VerticalAlignment="Top" Width="340"
FontSize="40" FontWeight="Normal" Foreground="Black" />
    <TextBlock Height="65" HorizontalAlignment="Left" Margin="35,400,0,0"
Name="CompassButton" Text="Compass" VerticalAlignment="Top" Width="340"
FontSize="40" FontWeight="Normal" Foreground="Black" />
    <TextBlock Height="65" HorizontalAlignment="Left" Margin="35,470,0,0"
Name="MotionButton" Text="Combined Motion" VerticalAlignment="Top" Width="400"
FontSize="40" FontWeight="Normal" Foreground="Black" />
    <Image Height="120" HorizontalAlignment="Left" Margin="295,545,0,0"
Name="HelpImage" Stretch="Fill" VerticalAlignment="Top" Width="120"
Source="/SensorsSet;component/Image/HelpIcon.png" />
    </Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->

```

## 2.2 Programming of "Menu" page

10

Visual Basic code is like this. (●list 2)

●list 2 MainPage.xaml.vb

Option Strict On

Partial Public Class MainPage

Inherits PhoneApplicationPage

' Constructor

Public Sub New()

InitializeComponent()

End Sub

Private Sub AccelerometerButton\_Tap(sender As Object, e As  
System.Windows.Input.GestureEventArgs) Handles AccelerometerButton.Tap

NavigationService.Navigate(New Uri("/AccelerometerPage1.xaml?Name=",  
UriKind.Relative))

End Sub

```
Private Sub GyroscopeButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles GyroscopeButton.Tap
    NavigationService.Navigate(New Uri("/GyroscopePage1.xaml?Name=",
UriKind.Relative))
End Sub
```

```
Private Sub CompassButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles CompassButton.Tap
    NavigationService.Navigate(New Uri("/CompassPage1.xaml?Name=",
UriKind.Relative))
End Sub
```

```
Private Sub MotionButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles MotionButton.Tap
    NavigationService.Navigate(New Uri("/MotionPage1.xaml?Name=",
UriKind.Relative))
End Sub
```

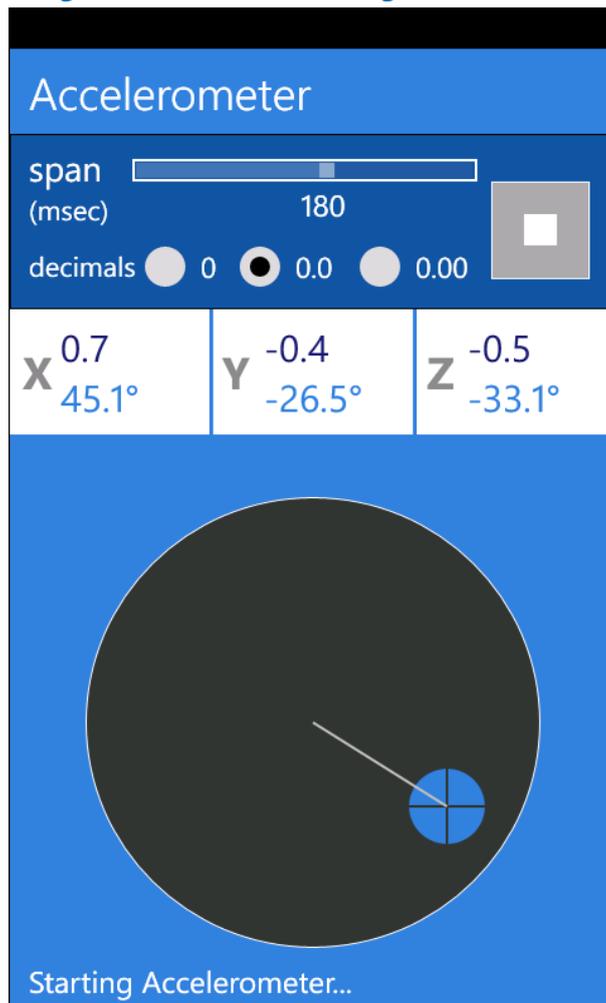
```
Private Sub HelpImage_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles HelpImage.Tap
    NavigationService.Navigate(New Uri("/SensorSetHelp.xaml?Name=",
UriKind.Relative))
End Sub
End Class
```

## 3. Accelerometer

### 3.1 Layout of "Accelerometer" page

Accelerometer provides the information of an accelerometer when the device moves. (●  
fig 6)

●fig 6 "Accelerometer" Page



You can get various values from the sensor.

#### (1) The values

The values of the acceleration on the X, Y, and Z axes in gravitational units.

✘ `Microsoft.Devices.Sensors namespace / AccelerometerReading / Acceleration properties.`

#### (2) The angles

The angles of the X, Y, and Z axes.

### (3) The moving image

When you move the device, an object moves in a circle.

### (4) The status of accelerometer

The XAML markup code is like this. (●list 3)

#### ●list 3 A part of AccelerometerPage1.xaml

```
<phone:PhoneApplicationPage
    x:Class="SensorsSet.AccelerometerPage1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
    mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
    shell:SystemTray.IsVisible="True">

    <!--LayoutRoot is the root grid where all page content is placed-->
    <Grid x:Name="LayoutRoot" Background="#FF3080E2">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
        </Grid.RowDefinitions>

        <!--TitlePanel contains the name of the application and page title-->
        <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="15,10,0,10">
            <TextBlock x:Name="PageTitle" Text="Accelerometer" Margin="0,0,0,0"
Style="{StaticResource PhoneTextTitle1Style}" FontSize="36" Foreground="White" />
        </StackPanel>

        <!--ContentPanel - place additional content here-->
        <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">
```

```
<!--●●設定 (背景) -->
<Rectangle Height="140" HorizontalAlignment="Left"
Name="SettingsBack" Stroke="Black" StrokeThickness="1" VerticalAlignment="Top"
Width="480" Fill="#FF1353A1"/>
```

```
<!--●間隔-->
<TextBlock Height="50" HorizontalAlignment="Left" Margin="15,8,0,0"
Name="TimeSpanLabel" Text="span" VerticalAlignment="Top" Width="70"
FontSize="28" Foreground="White" />
```

```
<TextBlock Height="30" HorizontalAlignment="Left" Margin="15,45,0,0"
Name="TimeSpanLabe2" Text="(msec)" VerticalAlignment="Top" Width="80"
FontSize="22" Foreground="White" />
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="93,40,0,0"
Name="TimeSpanTextBlock" Text="30" VerticalAlignment="Top" Width="60"
FontSize="24" TextAlignment="Center" Foreground="White" />
```

```
<Rectangle x:Name="SliderBorder" Height="18" Margin="97,20,0,0"
Stroke="White" VerticalAlignment="Top" Width="273" HorizontalAlignment="Left"
StrokeThickness="2"/>
```

```
<Slider Height="85" HorizontalAlignment="Left" Margin="88,0,0,0"
Name="TimespanSlider" VerticalAlignment="Top" Width="290"
Orientation="Horizontal" Value="3" Maximum="30" Minimum="1" LargeChange="1"
SmallChange="1" />
```

```
<!--●精度-->
<TextBlock Height="30" HorizontalAlignment="Left" Margin="15,90,0,0"
Name="RadioLabel1" Text="decimals" VerticalAlignment="Top" Width="90"
FontSize="22" Foreground="White" />
```

```
<RadioButton Content="0" Height="72" HorizontalAlignment="Left"
Margin="95,70,0,0" Name="RadioButton1" VerticalAlignment="Top" Width="80"
IsChecked="True" Foreground="White" Background="Gainsboro" />
```

```
<RadioButton Content="0.0" Height="72" HorizontalAlignment="Left"
Margin="170,70,0,0" Name="RadioButton10" VerticalAlignment="Top" Width="100"
Foreground="White" Background="Gainsboro" />
```

```
<RadioButton Content="0.00" Height="72" HorizontalAlignment="Left"
Margin="265,70,0,0" Name="RadioButton100" VerticalAlignment="Top" Width="110"
Foreground="White" Background="Gainsboro" />
```

```
<!--●ボタン-->
<Button Style="{StaticResource StopButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StopButton"
VerticalAlignment="Top" Width="78" IsEnabled="False" Visibility="Collapsed" />
```

```
<Button Style="{StaticResource StartButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StartButton"
VerticalAlignment="Top" Width="78" />
```

```
<!--●計測結果(背景とラベル)-->
```

```
<Rectangle Height="100" HorizontalAlignment="Left" Margin="0,140,0,0"
Name="RectangleX" VerticalAlignment="Top" Width="158" Fill="White" />
```

```
<Rectangle Height="100" HorizontalAlignment="Left"
Margin="161,140,0,0" Name="RectangleY" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<Rectangle Height="100" HorizontalAlignment="Left"
Margin="322,140,0,0" Name="RectangleZ" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="10,165,0,0"
Name="LabelX" Text="X" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="168,165,0,0"
Name="LabelY" Text="Y" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="330,165,0,0"
Name="LabelZ" Text="Z" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />
```

```
<!--●計測結果(Raw データ)-->
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="40,150,0,0"
Name="ResultX" Text="value" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="MidnightBlue"/>
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="201,150,0,0"
Name="ResultY" Text="value" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="MidnightBlue" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="362,150,0,0"
Name="ResultZ" Text="value" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="MidnightBlue" />
```

```
<!--●計測結果(角度)-->
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="40,190,0,0"
Name="AngleX" Text="angle" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="#FF3080E2" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="201,190,0,0"
Name="AngleY" Text="angle" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="#FF3080E2" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="362,190,0,0"
Name="AngleZ" Text="angle" VerticalAlignment="Top" Width="110" FontSize="30"
Foreground="#FF3080E2" />
```

```
<!--●状態-->
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="15,660,0,0"
Name="StatusTextBlock" Text="Status" VerticalAlignment="Top" Width="460"
FontSize="24" Foreground="White" />
```

```
<!--●位置表示-->
```

```
<Ellipse Height="360" HorizontalAlignment="Left" Margin="60,290,0,0"
Name="Ellipse1" Stroke="White" StrokeThickness="1" VerticalAlignment="Top"
Width="360" Fill="#FF333333" />
```

```
<Canvas Height="60" HorizontalAlignment="Left" Margin="210,440,0,0"
Name="Canvas1" VerticalAlignment="Top" Width="60">
```

```
<Image Height="60" HorizontalAlignment="Left" Margin="0,0,0,0"
Name="Image1" Stretch="Fill" VerticalAlignment="Top" Width="60"
Source="/SensorsSet;component/Image/AccelerometerCircle.png" />
```

```
</Canvas>
```

```
<!--●軌跡表示-->
```

```
<Line x:Name="MyMoveLine" X1="240" Y1="470" X2="240" Y2="470"
Stroke="silver" StrokeThickness="2"></Line>
```

```
<!--●0° の場合の軸表示-->
```

```
<Line x:Name="MyAxisX" X1="60" Y1="470" X2="420" Y2="470"
Stroke="RoyalBlue" StrokeThickness="3" Visibility="Collapsed" ></Line>
```

```
<Line x:Name="MyAxisY" X1="240" Y1="290" X2="240" Y2="650"
Stroke="OrangeRed" StrokeThickness="3" Visibility="Collapsed"></Line>
```

```
<Ellipse Height="360" HorizontalAlignment="Left" Margin="60,290,0,0"
Name="MyAxisZ" Stroke="LimeGreen" StrokeThickness="3" VerticalAlignment="Top"
Width="360" Visibility="Collapsed"/>
```

```
</Grid>
```

```
</Grid>
```

```
<!--Sample code showing usage of ApplicationBar-->
```

## 3.2 Programming of “Accelerometer” page

Visual Basic code is like this. (●list 4)

### ●list 4 AccelerometerPage1.xaml.vb

```
Option Strict On
```

```
Imports Microsoft.Devices.Sensors
```

```
Imports Microsoft.Xna.Framework
```

```
Partial Public Class AccelerometerPage1
```

```
    Inherits PhoneApplicationPage
```

```
    Public Sub New()
```

```
        InitializeComponent()
```

```
    End Sub
```

```
    Dim MyAccelerometer As Accelerometer '加速度センサー
```

```
    Dim MyTimeSpan As Integer = 30 '計測間隔。デフォルトは 30 ミリ秒とする
```

```
    Dim MyAccuracy As String = "F0" '精度。整数、小数点以下 1 桁、小数点以下 2 桁から選択
```

```
    Dim MyAreaLeft, MyAreaTop As Double 'オブジェクトを動かす台紙となる楕円の中心点の座標値
```

```
    Dim MajorAxis, MinorAxis As Double '楕円の長径、短径（その子にオブジェクトの画像を含む Canvas の可動範囲）
```

```
    Dim MyImageWidth, MyImageHeight As Double '移動させるオブジェクトの幅、高さ
```

```
'■このページがロードされた時の処理
```

```
Private Sub AccelerometerPage1_Loaded(sender As Object, e As System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

```
    '●加速度センサーがサポートされていない機種への対応
```

```
    If Accelerometer.IsSupported = False Then
```

```
        MessageBox.Show("Device does not support accelerometer.")
```

```
    '「開始」「停止」ボタンどちらも使用不可とする
```

```
    StartButton.IsEnabled = False
```

```
    StartButton.Opacity = 0.3
```

```

        StopButton.IsEnabled = False
    Exit Sub
End If

```

'画面の幅や高さを利用する場合の記述方法

'Dim ScreenWidth, ScreenHeight As Double '実機の画面サイズ

'ScreenWidth = Application.Current.Host.Content.ActualWidth

'ScreenHeight = Application.Current.Host.Content.ActualHeight - 32 '※800 が

取得される

'●オブジェクトの幅と高さ

```

        MyImageWidth =
CDBl(Image1.GetValue(FrameworkElement.ActualWidthProperty))
        MyImageHeight =
CDBl(Image1.GetValue(FrameworkElement.ActualHeightProperty))

```

'●楕円の長軸と短軸の半径

```

        MajorAxis = CDBl(Ellipse1.GetValue(FrameworkElement.ActualWidthProperty)) /
2
        MinorAxis = CDBl(Ellipse1.GetValue(FrameworkElement.ActualHeightProperty))
/ 2

```

'●楕円の中心点の座標値

MyAreaLeft = Ellipse1.Margin.Left + MajorAxis

MyAreaTop = Ellipse1.Margin.Top + MinorAxis

End Sub

'■精度設定

'計測値を表示する際の桁数を設定

```

Private Sub RadioButton1_Checked(sender As Object, e As
System.Windows.RoutedEventArgs) Handles RadioButton1.Checked
    MyAccuracy = "F0"
End Sub

```

```

Private Sub RadioButton10_Checked(sender As Object, e As
System.Windows.RoutedEventArgs) Handles RadioButton10.Checked
    MyAccuracy = "F1"
End Sub

```

```

Private Sub RadioButton100_Checked(sender As Object, e As
System.Windows.RoutedEventArgs) Handles RadioButton100.Checked

```

```
MyAccuracy = "F2"  
End Sub
```

'■計測間隔設定

'スライダーの値は 1~30 で 1 刻み設定 (デフォルトは 3)、計測には 10 を乗算した値を使用、単位はミリ秒

```
Private Sub TimespanSlider_ValueChanged(sender As Object, e As  
System.Windows.RoutedPropertyChangedEventArgs(Of Double)) Handles  
TimespanSlider.ValueChanged
```

```
MyTimeSpan = CInt(TimespanSlider.Value) * 10
```

```
TimeSpanTextBlock.Text = CStr(MyTimeSpan) '設定値を表示
```

```
TimeSpanTextBlock.Margin = New Thickness(78 + TimespanSlider.Value * 8, 40,  
0, 0) '設定値の TextBlock をスライダーに合わせて移動
```

```
End Sub
```

'■「開始」ボタンがタップされた時の処理

```
Private Sub StartButton_Tap(sender As Object, e As  
System.Windows.Input.GestureEventArgs) Handles StartButton.Tap
```

```
'「開始」ボタンは非表示にして「停止」ボタンを表示
```

```
StartButton.Visibility = Windows.Visibility.Collapsed
```

```
StartButton.IsEnabled = False
```

```
StopButton.Visibility = Windows.Visibility.Visible
```

```
StopButton.IsEnabled = True
```

```
'間隔設定スライダーは使用不可とする
```

```
TimespanSlider.IsEnabled = False
```

```
TimespanSlider.Opacity = 0.5
```

```
'加速度オブジェクトが有効な場合は、停止させる
```

'開始と停止を 2 個のボタンで切り替えているため、開始ボタンをタップした時は常にオブジェクトは空。

'これは開発者が登壇した技術イベント COD2012 での説明用であり、この条件分岐は、実際は不要。

```
'開始と停止をボタン 1 個で切り替える場合は必要。
```

```
If MyAccelerometer IsNot Nothing AndAlso MyAccelerometer.IsDataValid Then  
MyAccelerometer.Stop()
```

```
MessageBox.Show("Accelerometer stopped.")
```

```

Else
    If MyAccelerometer Is Nothing Then '加速度オブジェクトが空の場合の処理
        MyAccelerometer = New Accelerometer '加速度センサーの新しいインスタンスを生成
        MyAccelerometer.TimeBetweenUpdates =
            TimeSpan.FromMilliseconds(MyTimeSpan) '間隔設定スライダーで指定した 10 ミリ秒～
            300 ミリ秒単位で更新
            '加速度センサーの値の更新処理を実行
            AddHandler MyAccelerometer.CurrentValueChanged, Sub(senderValue
            As Object, eValue As SensorReadingEventArgs(Of AccelerometerReading))

        Deployment.Current.Dispatcher.BeginInvoke(Sub()
            CurrentValueChanged(eValue.SensorReading))

            End Sub

        End If
    Try
        '加速度センサーのセンシングを開始。状態に「計測中」と表示
        StatusTextBlock.Text = "Starting Accelerometer..."
        MyAccelerometer.Start()
    Catch
        'センシングが失敗した場合の処理。状態に、メッセージを表示
        StatusTextBlock.Text = "Unable to start accelerometer."
    Exit Sub
    End Try
End If
End Sub

'■センシングが実行された時の処理
Private Sub CurrentValueChanged(myReading As AccelerometerReading)
    'StatusTextBlock.Text = "Getting data..."

    '●Raw データの取得
    'AccelerometerReading.Acceleration プロパティで Raw データを取得して変数
    に代入
    'デバイスの X, Y, Z の各軸の加速度（重力単位）
    Dim myAcceleration = myReading.Acceleration

    '加速度 Raw データを 3 軸に分解
    Dim XValue As Double = myAcceleration.X

```

```
Dim YValue As Double = myAcceleration.Y
Dim ZValue As Double = myAcceleration.Z
```

```
'誤差を防ぐため上下をカット。
'角度に変換する際のパラメータエラーを防ぐ
```

```
If myAcceleration.X > 1 Then
    XValue = 1
Elseif myAcceleration.X < -1 Then
    XValue = -1
End If
```

```
If myAcceleration.Y > 1 Then
    YValue = 1
Elseif myAcceleration.Y < -1 Then
    YValue = -1
End If
```

```
If myAcceleration.Z > 1 Then
    ZValue = 1
Elseif myAcceleration.Z < -1 Then
    ZValue = -1
End If
```

```
'●Raw データを角度に変換
```

```
Dim MyAngleX, MyAngleY, MyAngleZ As Double '3軸の角度を代入する変数
'ラジオボタンで設定した桁数に応じて表示桁数を決定
```

```
'Raw データを3軸各々の TextBlock に表示
```

```
ResultX.Text = CStr(XValue.ToString(MyAccuracy))
```

```
ResultY.Text = CStr(YValue.ToString(MyAccuracy))
```

```
ResultZ.Text = CStr(ZValue.ToString(MyAccuracy))
```

```
'Raw データから3軸各々の角度を算出
```

```
MyAngleX = CDbI((Math.Asin(XValue) * 180 / Math.PI).ToString(MyAccuracy))
```

```
MyAngleY = CDbI((Math.Asin(YValue) * 180 / Math.PI).ToString(MyAccuracy))
```

```
MyAngleZ = CDbI((Math.Asin(ZValue) * 180 / Math.PI).ToString(MyAccuracy))
```

```
'算出した3軸各々の角度を TextBlock 内に単位を付けて表示
```

```
AngleX.Text = CStr(MyAngleX) & "°"
```

```
AngleY.Text = CStr(MyAngleY) & "°"
```

```
AngleZ.Text = CStr(MyAngleZ) & "°"
```

'●オブジェクトの移動処理

'Canvas の可動範囲と角度から、水平垂直の移動距離を計算

```
Dim MyMoveX As Integer = CInt(XValue * (MajorAxis - MyImageWidth / 2))
```

```
Dim MyMoveY As Integer = CInt(YValue * (MinorAxis - MyImageHeight / 2))
```

'楕円の中心点とオブジェクトの画像の幅高さから、Canvas の表示位置の座標をもとめる

```
Dim MyXPos As Integer = CInt(MyAreaLeft - MyImageWidth / 2 + MyMoveX)
```

```
Dim MyYPos As Integer = CInt(MyAreaTop - MyImageHeight / 2 - MyMoveY)
```

'オブジェクトの親要素である Canvas に座標を適用して移動させる

```
Canvas1.Margin = New Thickness(MyXPos, MyYPos, 0, 0)
```

'●ラインを引く処理

'終点座標を指定してラインを引く

```
MyMoveLine.X2 = CInt(MyAreaLeft + MyMoveX)
```

```
MyMoveLine.Y2 = CInt(MyAreaTop - MyMoveY)
```

'●水平時の表示

'各軸が水平の時、色付きの線を表示、水平でない時は非表示

```
If MyAngleX = 0 Then
```

```
    MyAxisX.Visibility = Windows.Visibility.Visible
```

```
Else
```

```
    MyAxisX.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
If MyAngleY = 0 Then
```

```
    MyAxisY.Visibility = Windows.Visibility.Visible
```

```
Else
```

```
    MyAxisY.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
If MyAngleZ = 0 Then
```

```
    MyAxisZ.Visibility = Windows.Visibility.Visible
```

```
Else
```

```
    MyAxisZ.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
End Sub
```

'■「停止」ボタンがタップされた時の処理

```
Private Sub StopButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StopButton.Tap
```

'加速度センサーを停止して、各種設定を初期化する

Try

MyAccelerometer.Stop()

MyAccelerometer = Nothing

StopButton.Visibility = Windows.Visibility.Collapsed

StopButton.IsEnabled = False

StartButton.Visibility = Windows.Visibility.Visible

StartButton.IsEnabled = True

TimespanSlider.IsEnabled = True

TimespanSlider.Opacity = 1.0

StatusTextBlock.Text = "Accelerometer stopped."

Catch

StatusTextBlock.Text = "Error stopping accelerometer."

End Try

End Sub

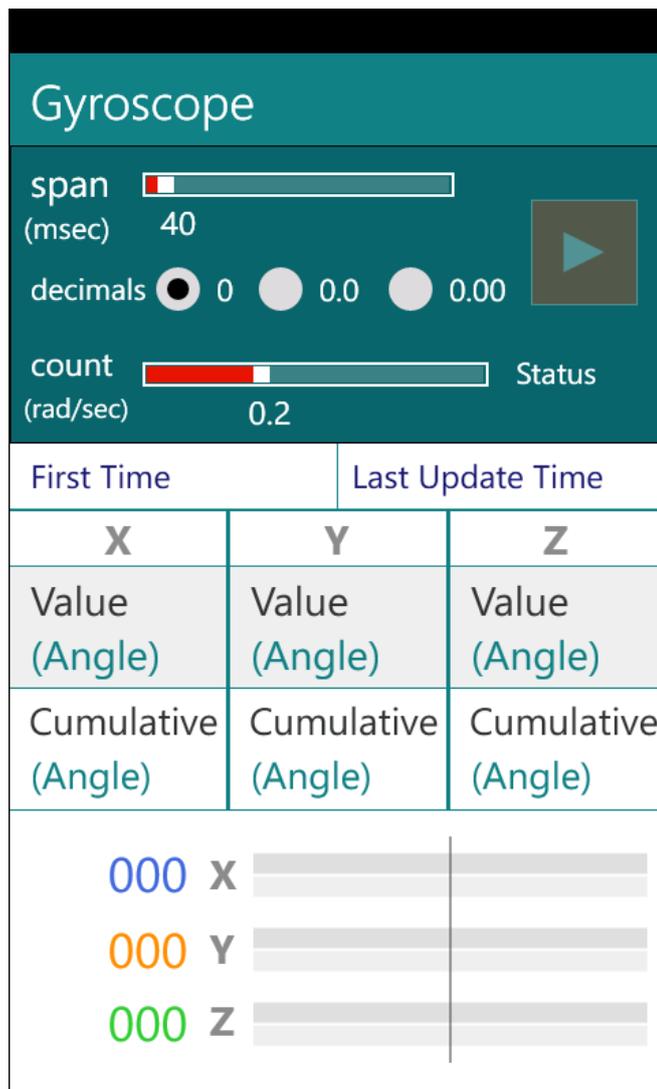
End Class

## 4. Gyroscope

### 4.1 Layout of "Gyroscope" page

Gyroscope provides the information around 3 axes of the device. (●fig 6)

●fig 6 "Gyroscope" Page



You can get various values from the sensor.

#### (1) The first update time

The first update time when the current value is changed.

※you can get a timestamp that the gyroscope reading was taken.

#### (2) The last update time

The current updated time to calculate the amount of rotation.

### (3) The rotational acceleration

The values of the rotational acceleration around the X, Y, and Z axes. (radians per second)

✧ [Microsoft.Devices.Sensors namespace / GyroscopeReading / RotationRate Property](#)

### (4) The angles

The angles of the rotational acceleration around the X, Y, and Z axes.

### (5) The cumulative rotation

The values of the cumulative rotation around the X, Y, and Z axes.

### (6) The angles

The angles of the cumulative rotation around the X, Y, and Z axes.

### (7) The bar charts

The bar charts to represent the angles. The upper bar represents the rotational acceleration. The lower bar represents the cumulative rotation.

If the value is 360 or over, the red rectangles will be displayed on the edge of the bars.

### (8) The counter

A counter is displayed when the values of the rotational acceleration is or over the value that you have set in accuracy for the counter slider.

25

### (9) The status of gyroscope

The XAML markup code is like this. (● [list 5](#))

#### ● [list 5](#) A part of GyroscopePage1.xaml

```
<phone:PhoneApplicationPage
    x:Class="SensorsSet.GyroscopePage1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
    Foreground="{StaticResource PhoneForegroundBrush}"
    SupportedOrientations="Portrait" Orientation="Portrait"
```

```

mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
shell:SystemTray.IsVisible="True">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="#FF0f8187">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*" />
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="15,10,0,10">
        <TextBlock x:Name="PageTitle" Text="Gyroscope" Margin="0,0,0,0"
Style="{StaticResource PhoneTextTitle1Style}" FontSize="36" Foreground="White" />
    </StackPanel>

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">

        <!--●●設定 (背景) -->
        <Rectangle Height="220" HorizontalAlignment="Left"
Name="SettingsBack" Stroke="Black" StrokeThickness="1" VerticalAlignment="Top"
Width="480" Fill="#FF0A656A"/>

        <!--●間隔-->
        <TextBlock Height="40" HorizontalAlignment="Left" Margin="15,8,0,0"
Name="TimeSpanLabel" Text="span" VerticalAlignment="Top" Width="70"
FontSize="28" Foreground="White" />
        <TextBlock Height="30" HorizontalAlignment="Left" Margin="10,45,0,0"
Name="TimeSpanLabe2" Text="(msec)" VerticalAlignment="Top" Width="80"
FontSize="22" Foreground="White" />
        <TextBlock Height="40" HorizontalAlignment="Left" Margin="93,40,0,0"
Name="TimeSpanTextBlock" Text="40" VerticalAlignment="Top" Width="60"
FontSize="24" TextAlignment="Center" Foreground="White" />
        <Rectangle x:Name="SliderBorder" Height="18" Margin="97,20,0,0"
Stroke="White" VerticalAlignment="Top" Width="228" HorizontalAlignment="Left"
StrokeThickness="2"/>
        <Slider Height="85" HorizontalAlignment="Left" Margin="88,0,0,0"
Name="TimespanSlider" VerticalAlignment="Top" Width="245"
Orientation="Horizontal" Value="4" Maximum="30" Minimum="3" LargeChange="1"
SmallChange="1" />

```

```

<!--●精度-->
<TextBlock Height="30" HorizontalAlignment="Left" Margin="15,90,0,0"
Name="RadioLabel1" Text="decimals" VerticalAlignment="Top" Width="90"
FontSize="22" Foreground="White" />
<RadioButton Content="0" Height="72" HorizontalAlignment="Left"
Margin="95,70,0,0" Name="RadioButton1" VerticalAlignment="Top" Width="80"
IsChecked="True" Foreground="White" Background="Gainsboro" />
<RadioButton Content="0.0" Height="72" HorizontalAlignment="Left"
Margin="170,70,0,0" Name="RadioButton10" VerticalAlignment="Top" Width="100"
Foreground="White" Background="Gainsboro" />
<RadioButton Content="0.00" Height="72" HorizontalAlignment="Left"
Margin="265,70,0,0" Name="RadioButton100" VerticalAlignment="Top" Width="110"
Foreground="White" Background="Gainsboro" />

```

```

<!--●検出-->
<TextBlock Height="40" HorizontalAlignment="Left" Margin="15,144,0,0"
Name="CounterLabel" Text="count" VerticalAlignment="Top" Width="80"
FontSize="24" Foreground="White" />
<TextBlock Height="30" HorizontalAlignment="Left" Margin="10,179,0,0"
Name="CounterLabe2" Text="(rad/sec)" VerticalAlignment="Top" Width="80"
Foreground="White" />
<TextBlock Height="40" HorizontalAlignment="Left" Margin="160,180,0,0"
Name="CounterTextBlock" Text="0.2" VerticalAlignment="Top" Width="60"
FontSize="24" TextAlignment="Center" Foreground="White" />
<Rectangle x:Name="SliderBorder2" Height="18" Margin="97,160,0,0"
Stroke="White" VerticalAlignment="Top" Width="253" HorizontalAlignment="Left"
StrokeThickness="2"/>
<Slider Height="85" HorizontalAlignment="Left" Margin="88,140,0,0"
Name="CounterSlider" VerticalAlignment="Top" Width="270" Orientation="Horizontal"
Value="4" Maximum="10" Minimum="1" LargeChange="1" SmallChange="1" />

```

```

<!--●ボタン-->
<Button Style="{StaticResource StopButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,40,0,0" Name="StopButton"
VerticalAlignment="Top" Width="78" IsEnabled="False" Visibility="Collapsed" />
<Button Style="{StaticResource StartButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,40,0,0" Name="StartButton"
VerticalAlignment="Top" Width="78" />

```

```

<!--●状態-->
<TextBlock Height="50" HorizontalAlignment="Left" Margin="370,152,0,0"
Name="StatusTextBlock" Text="Status" VerticalAlignment="Top" Width="100"
FontSize="22" TextAlignment="Left" Foreground="White" />

<!--●時間(背景)-->
<Rectangle Height="48" HorizontalAlignment="Left" Margin="0,220,0,0"
Name="StartTimeBack" VerticalAlignment="Top" Width="239" Fill="White" />
<Rectangle Height="48" HorizontalAlignment="Left"
Margin="240,220,0,0" Name="StopTimeBack" VerticalAlignment="Top" Width="239"
Fill="White" />

<TextBlock Height="40" HorizontalAlignment="Left" Margin="15,227,0,0"
Name="StartTime" Text="First Time " VerticalAlignment="Top" Width="220"
FontSize="24" Foreground="MidnightBlue"/>
<TextBlock Height="40" HorizontalAlignment="Left" Margin="250,227,0,0"
Name="StopTime" Text="Last Update Time" VerticalAlignment="Top" Width="220"
FontSize="24" Foreground="MidnightBlue"/>

<!--●計測結果(ラベル)-->
<Rectangle Height="40" HorizontalAlignment="Left" Margin="0,270,0,0"
Name="AxisX" VerticalAlignment="Top" Width="158" Fill="White" />
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="161,270,0,0" Name="AxisY" VerticalAlignment="Top" Width="158" Fill="White"
/>
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="322,270,0,0" Name="AxisZ" VerticalAlignment="Top" Width="158" Fill="White"
/>

<TextBlock Height="40" HorizontalAlignment="Left" Margin="0,270,0,0"
Name="LabelX" Text="X" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
<TextBlock Height="40" HorizontalAlignment="Left" Margin="160,270,0,0"
Name="LabelY" Text="Y" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
<TextBlock Height="40" HorizontalAlignment="Left" Margin="320,270,0,0"
Name="LabelZ" Text="Z" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>

```

<!--●計測結果(背景)-->

```
<Rectangle Height="89" HorizontalAlignment="Left" Margin="0,311,0,0"
Name="RectangleX" VerticalAlignment="Top" Width="158" Fill="#FFFEDEDED" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="161,311,0,0" Name="RectangleY" VerticalAlignment="Top" Width="158"
Fill="#FFFEDEDED" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="322,311,0,0" Name="RectangleZ" VerticalAlignment="Top" Width="158"
Fill="#FFFEDEDED" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left" Margin="0,401,0,0"
Name="TotalRectangleX" VerticalAlignment="Top" Width="158" Fill="#FFFEFEFE" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="161,401,0,0" Name="TotalRectangleY" VerticalAlignment="Top" Width="158"
Fill="#FFFEFEFE" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="322,401,0,0" Name="TotalRectangleZ" VerticalAlignment="Top" Width="158"
Fill="#FFFEFEFE" />
```

29

<!--●計測結果(計測値 Raw データ)-->

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,315,0,0"
Name="ValueX" Text="Value" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333"/>
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,315,0,0"
Name="ValueY" Text="Value" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,315,0,0"
Name="ValueZ" Text="Value" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333" />
```

<!--●計測結果(計測角度)-->

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,355,0,0"
Name="AngleX" Text="(Angle)" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF0f8187" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,355,0,0"
Name="AngleY" Text="(Angle)" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF0f8187" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,355,0,0"
Name="AngleZ" Text="(Angle)" VerticalAlignment="Top" Width="140" FontSize="30"
```

```

Foreground="#FF0f8187" />
  <!--●計測結果(累積値 Raw データ)-->
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="14,405,0,0"
Name="CumulativeValueX" Text="Cumulative" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333"/>
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="175,405,0,0"
Name="CumulativeValueY" Text="Cumulative" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333" />
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="336,405,0,0"
Name="CumulativeValueZ" Text="Cumulative" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333" />

  <!--●計測結果(累積角度)-->
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="15,445,0,0"
Name="CumulativeAngleX" Text="(Angle)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="176,445,0,0"
Name="CumulativeAngleY" Text="(Angle)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="337,445,0,0"
Name="CumulativeAngleZ" Text="(Angle)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />

  <!--●計測結果表示 (軸ラベル)-->
  <Rectangle Height="210" HorizontalAlignment="Left" Margin="0,491,0,0"
Name="GraphBack" VerticalAlignment="Top" Width="480" Fill="White" />
  <TextBlock Height="40" HorizontalAlignment="Left" Margin="146,517,0,0"
Name="CounterLabelX" Text="X" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Left" />
  <TextBlock Height="40" HorizontalAlignment="Left" Margin="146,572,0,0"
Name="CounterLabelY" Text="Y" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Left" />
  <TextBlock Height="40" HorizontalAlignment="Left" Margin="146,625,0,0"
Name="CounterLabelZ" Text="Z" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Left" />

  <!--●計測結果表示 (カウンター、検出と累計の棒グラフ)-->

  <!--Xカウンター-->
  <TextBlock Height="48" HorizontalAlignment="Left" Margin="15,512,0,0"
Name="CounterX" Text="000" VerticalAlignment="Top" Width="115"

```

```

Foreground="RoyalBlue" TextAlignment="Right" FontSize="36" />
  <!--X 背景-->
    <Line x:Name="ValueXLineBack" X1="178" Y1="530" X2="466" Y2="530"
Stroke="#FFDDDDDD" StrokeThickness="15"></Line>
    <Line x:Name="CumulativeXLineBack" X1="178" Y1="547" X2="466"
Y2="547" Stroke="#FFEEEEEE" StrokeThickness="15"></Line>
  <!--X 値-->
    <Line x:Name="ValueXLine" X1="322" Y1="530" X2="322" Y2="530"
Stroke="RoyalBlue" StrokeThickness="15"></Line>
    <Line x:Name="CumulativeXLine" X1="322" Y1="547" X2="322" Y2="547"
Stroke="CornflowerBlue" StrokeThickness="15"></Line>
  <!--X 振り切れ-->
    <Line x:Name="ValueXMax" X1="451" Y1="530" X2="466" Y2="530"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>
    <Line x:Name="CumulativeXMax" X1="451" Y1="547" X2="466" Y2="547"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>
    <Line x:Name="ValueXMin" X1="178" Y1="530" X2="193" Y2="530"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>
    <Line x:Name="CumulativeXMin" X1="178" Y1="547" X2="193" Y2="547"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>

  <!--Y カウンター-->
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="15,568,0,0"
Name="CounterY" Text="000" VerticalAlignment="Top" Width="115"
Foreground="DarkOrange" TextAlignment="Right" FontSize="36" />
  <!--Y 背景-->
    <Line x:Name="ValueYLineBack" X1="178" Y1="585" X2="466" Y2="585"
Stroke="#FFDDDDDD" StrokeThickness="15"></Line>
    <Line x:Name="CumulativeYLineBack" X1="178" Y1="602" X2="466"
Y2="602" Stroke="#FFEEEEEE" StrokeThickness="15"></Line>
  <!--Y 値-->
    <Line x:Name="ValueYLine" X1="322" Y1="585" X2="322" Y2="585"
Stroke="DarkOrange" StrokeThickness="15"></Line>
    <Line x:Name="CumulativeYLine" X1="322" Y1="602" X2="322" Y2="602"
Stroke="Orange" StrokeThickness="15"></Line>
  <!--Y 振り切れ-->
    <Line x:Name="ValueYMax" X1="451" Y1="585" X2="466" Y2="585"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>
    <Line x:Name="CumulativeYMax" X1="451" Y1="602" X2="466" Y2="602"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"></Line>
    <Line x:Name="ValueYMin" X1="178" Y1="585" X2="193" Y2="585"

```

```

Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>
    <Line x:Name="CumulativeYMin" X1="178" Y1="602" X2="193" Y2="602"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>

    <!--Z カウンター-->
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="15,622,0,0"
Name="CounterZ" Text="000" VerticalAlignment="Top" Width="115"
Foreground="LimeGreen" TextAlignment="Right" FontSize="36" />
    <!--Z 背景-->
    <Line x:Name="ValueZLineBack" X1="178" Y1="640" X2="466" Y2="640"
Stroke="#FFDDDDDD" StrokeThickness="15"> </Line>
    <Line x:Name="CumulativeZLineBack" X1="178" Y1="657" X2="466"
Y2="657" Stroke="#FFEEEEEE" StrokeThickness="15"> </Line>
    <!--Z 値-->
    <Line x:Name="ValueZLine" X1="322" Y1="640" X2="322" Y2="640"
Stroke="LimeGreen" StrokeThickness="15"> </Line>
    <Line x:Name="CumulativeZLine" X1="322" Y1="657" X2="322" Y2="657"
Stroke="MediumAquaMarine" StrokeThickness="15"> </Line>
    <!--Z 振り切れ-->
    <Line x:Name="ValueZMax" X1="451" Y1="640" X2="466" Y2="640"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>
    <Line x:Name="CumulativeZMax" X1="451" Y1="657" X2="466" Y2="657"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>
    <Line x:Name="ValueZMin" X1="178" Y1="657" X2="193" Y2="657"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>
    <Line x:Name="CumulativeZMin" X1="178" Y1="640" X2="193" Y2="640"
Stroke="OrangeRed" StrokeThickness="15" Visibility="Collapsed"> </Line>

    <!--原点 0-->
    <Line x:Name="OriginLine" X1="322" Y1="510" X2="322" Y2="677"
Stroke="#FF333333" StrokeThickness="1"> </Line>

    </Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->

```

## 4.2 Programming of “Gyroscope” page

Visual Basic code is like this (●list 6)

### ●list 6 GyroscopePage1.xaml.vb

Option Strict On

Imports Microsoft.Devices.Sensors

Imports Microsoft.Xna.Framework

'Vector3 データを扱うには、Microsoft.Xna.Framework の参照の追加が必要

Partial Public Class GyroscopePage1

Inherits PhoneApplicationPage

Public Sub New()

InitializeComponent()

End Sub

Dim MyGyro As Gyroscope 'ジャイロスコープセンサー

Dim MyTimeSpan As Integer = 40 '計測間隔。デフォルトは 40 ミリ秒とする

Dim MyAccuracy As String = "F0" '精度。整数、小数点以下 1 桁、小数点以下 2 桁から選択

Dim MyCurrentRotationRate As Vector3 = Vector3.Zero 'Vector3 型計測 Raw データ

Dim MyCumulativeRotation As Vector3 = Vector3.Zero 'Vector3 型累積 Raw データ

Dim MyUpdateTime As DateTimeOffset = DateTimeOffset.MinValue '計測開始時刻

Dim MyLastUpdateTimespan As TimeSpan '計測時間

'検出値設定用のスライダーの値、1～10 まで 1 刻み、0.05 を乗算した値を利用

Dim CompareValue As Integer = 4

Dim MyCountX As Integer = 0 'カウンターの値

Dim MyCountY As Integer = 0

Dim MyCountZ As Integer = 0

Dim CounterLevel As Double = 5 'デフォルトの比較値。検出設定が 4 で計測間隔が 40 の時、1 秒単位の値

'■このページがロードされた時の処理

Private Sub GyroscopePage1\_Loaded(sender As Object, e As System.Windows.RoutedEventArgs) Handles MyBase.Loaded

'●ジャイロスコープがサポートされていない機種への対応

If Gyroscope.IsSupported = False Then

    MessageBox.Show("Device does not support gyroscope.")

    '「開始」「停止」ボタンどちらも使用不可とする

    StartButton.IsEnabled = False

    StartButton.Opacity = 0.3

    StopButton.IsEnabled = False

    Exit Sub

End If

End Sub

'■精度設定

'計測値を表示する際の桁数を設定

Private Sub RadioButton1\_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton1.Checked

    MyAccuracy = "F0"

End Sub

Private Sub RadioButton10\_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton10.Checked

    MyAccuracy = "F1"

End Sub

Private Sub RadioButton100\_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton100.Checked

    MyAccuracy = "F2"

End Sub

'■間隔設定

Private Sub TimespanSlider\_ValueChanged(sender As Object, e As System.Windows.RoutedPropertyChangedEventArgs(Of Double)) Handles TimespanSlider.ValueChanged

    'スライダーの値は3~30で1刻み設定(デフォルトは4)、計測には10を乗算した値を使用、単位はミリ秒

    MyTimeSpan = CInt(TimespanSlider.Value) \* 10

    TimeSpanTextBlock.Text = CStr(MyTimeSpan) '設定値を表示

    TimeSpanTextBlock.Margin = New Thickness(65 + TimespanSlider.Value \* 7, 40, 0, 0) '設定値のTextBlockをスライダーに合わせて移動

End Sub

#### '■ カウンターの基準設定

```
Private Sub CounterSlider_ValueChanged(sender As Object, e As System.Windows.RoutedPropertyChangedEventArgs(Of Double)) Handles CounterSlider.ValueChanged
```

```
    CompareValue = CInt(CounterSlider.Value)
    CounterTextBlock.Text = CStr(CompareValue * 0.05)
    CounterTextBlock.Margin = New Thickness(58 + CompareValue * 26, 180, 0, 0)
```

'センサーから逐次取得の RotationRate の値 (rad/sec) と比較するため単位を揃える

```
'スライダーで 10 段階設定した値を検出値 (0.05~0.5) として設定し
'この設定値に (1000/計測間隔) を乗算して、1 秒あたりの検出値とする
    CounterLevel = CompareValue * 0.05 * 1000 / MyTimeSpan
```

```
End Sub
```

#### '■ 「開始」 ボタンがタップされた時の処理

```
Private Sub StartButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StartButton.Tap
```

```
'「開始」 ボタンは非表示にして「停止」 ボタンを表示
    StartButton.Visibility = Windows.Visibility.Collapsed
    StartButton.IsEnabled = False
    StopButton.Visibility = Windows.Visibility.Visible
    StopButton.IsEnabled = True
```

```
'間隔設定スライダー、カウンター設定スライダーは使用不可とする
    TimespanSlider.IsEnabled = False
    TimespanSlider.Opacity = 0.5
    CounterSlider.IsEnabled = False
    CounterSlider.Opacity = 0.5
```

```
'ジャイロスコープオブジェクトが有効な場合は、停止させる
```

'開始と停止を 2 個のボタンで切り替えているため、開始ボタンをタップした時は常にオブジェクトは空。

```
'開始と停止をボタン 1 個で切り替える場合は必要。
```

```
'If MyGyro IsNot Nothing AndAlso MyGyro.IsValid Then
```

```
'    MyGyro.Stop()
```

```
'    MessageBox.Show("Gyroscope stopped.")
```

```
'Else
```

' If MyGyro Is Nothing Then 'ジャイロスコープオブジェクトが空の場合の  
処理

MyGyro = New Gyroscope 'ジャイロスコープの新しいインスタンスを生成

MyGyro.TimeBetweenUpdates = TimeSpan.FromMilliseconds(MyTimeSpan) '間  
隔設定スライダーで指定した 30 ミリ秒~300 ミリ秒単位で更新

'ジャイロスコープの値の更新処理を実行

AddHandler MyGyro.CurrentValueChanged, Sub(senderValue As Object, eValue  
As SensorReadingEventArgs(Of GyroscopeReading))

Deployment.Current.Dispatcher.BeginInvoke(Sub() UpdateUI(eValue.SensorReading))

End Sub

'End If

Try

'ジャイロスコープのセンシングを開始。状態に「計測中」を表示  
StatusTextBlock.Text = "Starting..."

MyGyro.Start()

Catch

'センシングが失敗した場合の処理。状態に、メッセージを表示  
MessageBox.Show("Unable to start gyroscope.")

Exit Sub

End Try

'End If

End Sub

'■センシングが実行された時の処理

Private Sub UpdateUI(MyReading As GyroscopeReading)

'StatusTextBlock.Text = "Getting data..."

'●開始時間、終了時間表示、累積データ計算に必要な時間の取得

If MyUpdateTime = DateTimeOffset.MinValue Then '初回計測時かどうかを判別

MyUpdateTime = MyReading.Timestamp '初回計測時の時刻を取得

StartTime.Text = CStr(MyUpdateTime.ToString("yyyy/MM/dd/HH:mm:ss")) '

「開始時間」欄に計測開始時間を表示

Else '2 回目以降の計測

MyCurrentRotationRate = MyReading.RotationRate 'センサーからの値を逐  
次取得。単位は rad/sec (後述の **A** で利用)

MyLastUpdateTimespan = MyReading.Timestamp - MyUpdateTime '前回計  
測時からの経過時間

'取得された、単位 rad/sec の値に秒数の合計を乗算して累積角度 rad を算

出（後述の **B** で利用）

```
MyCumulativeRotation = MyCumulativeRotation + MyCurrentRotationRate  
* CSng(MyLastUpdateTimespan.TotalSeconds)
```

```
MyUpdateTime = MyReading.Timestamp 'データ取得時刻を現在に設定  
StopTime.Text = CStr(MyUpdateTime.ToString("yyyy/MM/dd/HH:mm:ss")) '  
「終了時間」欄に現在時間を表示  
End If
```

'●Raw データの取得、角度への変換、TextBlock への表示

' (A) 逐次取得回転速度の値を 3 軸に分解

'デバイスの X, Y, Z の各軸の回転速度 (rad/sec)

'GyroscopeReading 構造体 RotationRate プロパティで取得される値

```
Dim MyValueX As Single = MyCurrentRotationRate.X
```

```
Dim MyValueY As Single = MyCurrentRotationRate.Y
```

```
Dim MyValueZ As Single = MyCurrentRotationRate.Z
```

'3 軸に分解した回転速度を角度に変換

```
Dim MyAngleX As Double = MathHelper.ToDegrees(MyValueX)
```

```
Dim MyAngleY As Double = MathHelper.ToDegrees(MyValueY)
```

```
Dim MyAngleZ As Double = MathHelper.ToDegrees(MyValueZ)
```

'計測値を表示

'ラジオボタンで設定した桁数に応じて表示桁数を決定

```
ValueX.Text = CStr(MyValueX.ToString(MyAccuracy))
```

```
ValueY.Text = CStr(MyValueY.ToString(MyAccuracy))
```

```
ValueZ.Text = CStr(MyValueZ.ToString(MyAccuracy))
```

'カウンターの加算と表示

'計測値を「■カウンターの基準設定」で設定したレベルと比較

'3 軸の値がそれぞれ設定値以上であればカウントする

```
If Math.Abs(MyValueX) >= CounterLevel Then
```

```
MyCountX = MyCountX + 1
```

```
CounterX.Text = CStr(MyCountX)
```

```
End If
```

```
If Math.Abs(MyValueY) >= CounterLevel Then
```

```
MyCountY = MyCountY + 1
```

```
CounterY.Text = CStr(MyCountY)
```

```
End If
```

```

If Math.Abs(MyValueZ) >= CounterLevel Then
    MyCountZ = MyCountZ + 1
    CounterZ.Text = CStr(MyCountZ)
End If

```

```

'回転角度を表示
AngleX.Text = CStr(MyAngleX.ToString(MyAccuracy)) & ""
AngleY.Text = CStr(MyAngleY.ToString(MyAccuracy)) & ""
AngleZ.Text = CStr(MyAngleZ.ToString(MyAccuracy)) & ""

```

```

'●累積データの取得、角度への変換、TextBlock への表示
' (B) 累積された X, Y, Z の各軸の回転速度 (rad/sec)
Dim MyCumulativeX As Single = MyCumulativeRotation.X
Dim MyCumulativeY As Single = MyCumulativeRotation.Y
Dim MyCumulativeZ As Single = MyCumulativeRotation.Z

```

```

'累積角度
'3 軸に分解した累積値を角度に変換
Dim MyCumulativeAngleX As Double = MathHelper.ToDegrees(MyCumulativeX)
Dim MyCumulativeAngleY As Double = MathHelper.ToDegrees(MyCumulativeY)
Dim MyCumulativeAngleZ As Double = MathHelper.ToDegrees(MyCumulativeZ)

```

```

'累積値を表示
'ラジオボタンで設定した桁数に応じて表示桁数を決定
CumulativeValueX.Text = CStr(MyCumulativeX.ToString(MyAccuracy))
CumulativeValueY.Text = CStr(MyCumulativeY.ToString(MyAccuracy))
CumulativeValueZ.Text = CStr(MyCumulativeZ.ToString(MyAccuracy))

```

```

'累積角度を表示
CumulativeAngleX.Text = CStr(MyCumulativeAngleX.ToString(MyAccuracy)) &
""
CumulativeAngleY.Text = CStr(MyCumulativeAngleY.ToString(MyAccuracy)) &
""
CumulativeAngleZ.Text = CStr(MyCumulativeAngleZ.ToString(MyAccuracy)) &
""

```

```

'●棒グラフの描画処理

```

```

'●計測値のグラフ

```

```

'計測角度から、グラフの罫線の X の終点座標値を計算。始点は 322、プラスマ
イナス方向にそれぞれ 144 幅

```

```

'グラフの最大幅に 360 度を割り当て、振りきれた場合は、別処理とする

```

```
Dim GraphValueX As Double = 322 + MyAngleX * 144 / 360
Dim GraphValueY As Double = 322 + MyAngleY * 144 / 360
Dim GraphValueZ As Double = 322 + MyAngleZ * 144 / 360
Dim GraphValueXLine, GraphValueYLine, GraphValueZLine As Double 'グラフの
X の終点座標値を代入する変数
```

'360 度以上の時 X 座標は 466 以上、マイナス 360 度以下の場合 X 座標は 178 以下になる

'プラスマイナス 360 度以内の表示とし、それ以上とそれ以下の場合グラフの背景からはみ出さないようにして赤のストッパーを表示する

'プラスマイナス 360 度以内の場合は、赤のストッパーは非表示とする

```
If GraphValueX >= 466 Then
    GraphValueXLine = 466
    ValueXMax.Visibility = Windows.Visibility.Visible
    ValueXMin.Visibility = Windows.Visibility.Collapsed
Elseif GraphValueX <= 178 Then
    GraphValueXLine = 178
    ValueXMax.Visibility = Windows.Visibility.Collapsed
    ValueXMin.Visibility = Windows.Visibility.Visible
Elseif GraphValueX > 178 And GraphValueX < 466 Then
    GraphValueXLine = GraphValueX
    ValueXMax.Visibility = Windows.Visibility.Collapsed
    ValueXMin.Visibility = Windows.Visibility.Collapsed
End If
```

```
If GraphValueY >= 466 Then
    GraphValueYLine = 466
    ValueYMax.Visibility = Windows.Visibility.Visible
    ValueYMin.Visibility = Windows.Visibility.Collapsed
Elseif GraphValueY <= 178 Then
    GraphValueYLine = 178
    ValueYMax.Visibility = Windows.Visibility.Collapsed
    ValueYMin.Visibility = Windows.Visibility.Visible
Elseif GraphValueY > 178 And GraphValueY < 466 Then
    GraphValueYLine = GraphValueY
    ValueYMax.Visibility = Windows.Visibility.Collapsed
    ValueYMin.Visibility = Windows.Visibility.Collapsed
End If
```

```
If GraphValueZ >= 466 Then
    GraphValueZLine = 466
```

```

ValueZMax.Visibility = Windows.Visibility.Visible
ValueZMin.Visibility = Windows.Visibility.Collapsed
Elseif GraphValueZ <= 178 Then
    GraphValueZLine = 178
    ValueZMax.Visibility = Windows.Visibility.Collapsed
    ValueZMin.Visibility = Windows.Visibility.Visible
Elseif GraphValueZ > 178 And GraphValueZ < 466 Then
    GraphValueZLine = GraphValueZ
    ValueZMax.Visibility = Windows.Visibility.Collapsed
    ValueZMin.Visibility = Windows.Visibility.Collapsed
End If

```

'計測値のグラフを表示

```

ValueXLine.X2 = GraphValueXLine
ValueYLine.X2 = GraphValueYLine
ValueZLine.X2 = GraphValueZLine

```

'●累積値のグラフ

'累積角度から、グラフの罫線の X の終点座標値を計算。始点は 322、プラスマイナス方向にそれぞれ 144 幅

'グラフの最大幅に 360 度を割り当て、振りきれた場合は、別処理とする

```
Dim GraphCumulativeX As Double = 322 + MyCumulativeAngleX * 144 / 360
```

```
Dim GraphCumulativeY As Double = 322 + MyCumulativeAngleY * 144 / 360
```

```
Dim GraphCumulativeZ As Double = 322 + MyCumulativeAngleZ * 144 / 360
```

```
Dim GraphCumulativeXLine, GraphCumulativeYLine, GraphCumulativeZLine As
```

Double 'グラフの X の終点座標値を代入する変数

'360 度以上の時 X 座標は 466 以上、マイナス 360 度以下の場合 X 座標は 178 以下になる

'プラスマイナス 360 度以内の表示とし、それ以上とそれ以下の場合グラフの背景からはみ出さないように赤のストッパーを表示する

'プラスマイナス 360 度以内の場合は、赤のストッパーは非表示とする

```

If GraphCumulativeX >= 466 Then
    GraphCumulativeXLine = 466
    CumulativeXMax.Visibility = Windows.Visibility.Visible
    CumulativeXMin.Visibility = Windows.Visibility.Collapsed
Elseif GraphCumulativeX <= 178 Then
    GraphCumulativeXLine = 178
    CumulativeXMax.Visibility = Windows.Visibility.Collapsed
    CumulativeXMin.Visibility = Windows.Visibility.Visible

```

```
Else
    GraphCumulativeXLine = GraphCumulativeX
    CumulativeXMax.Visibility = Windows.Visibility.Collapsed
    CumulativeXMin.Visibility = Windows.Visibility.Collapsed
End If
```

```
If GraphCumulativeY >= 466 Then
    GraphCumulativeYLine = 466
    CumulativeYMax.Visibility = Windows.Visibility.Visible
    CumulativeYMin.Visibility = Windows.Visibility.Collapsed
```

```
Elseif GraphCumulativeY <= 178 Then
    GraphCumulativeYLine = 178
    CumulativeYMax.Visibility = Windows.Visibility.Collapsed
    CumulativeYMin.Visibility = Windows.Visibility.Visible
```

```
Else
    GraphCumulativeYLine = GraphCumulativeY
    CumulativeYMax.Visibility = Windows.Visibility.Collapsed
    CumulativeYMin.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

```
If GraphCumulativeZ >= 466 Then
    GraphCumulativeZLine = 466
    CumulativeZMax.Visibility = Windows.Visibility.Visible
    CumulativeZMin.Visibility = Windows.Visibility.Collapsed
```

```
Elseif GraphCumulativeZ <= 178 Then
    GraphCumulativeZLine = 178
    CumulativeZMax.Visibility = Windows.Visibility.Collapsed
    CumulativeZMin.Visibility = Windows.Visibility.Visible
```

```
Else
    GraphCumulativeZLine = GraphCumulativeZ
    CumulativeZMax.Visibility = Windows.Visibility.Collapsed
    CumulativeZMin.Visibility = Windows.Visibility.Collapsed
```

```
End If
```

'累積値のグラフを表示

```
CumulativeXLine.X2 = GraphCumulativeXLine
```

```
CumulativeYLine.X2 = GraphCumulativeYLine
```

```
CumulativeZLine.X2 = GraphCumulativeZLine
```

```
End Sub
```

'■ 「停止」 ボタンがタップされた時の処理

```
Private Sub StopButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StopButton.Tap
    'ジャイロスコープセンサーを停止して、各種設定を初期化する
    Try
        MyGyro.Stop()
        MyGyro = Nothing

        StopButton.Visibility = Windows.Visibility.Collapsed
        StopButton.IsEnabled = False
        StartButton.Visibility = Windows.Visibility.Visible
        StartButton.IsEnabled = True

        TimespanSlider.IsEnabled = True
        TimespanSlider.Opacity = 1.0
        CounterSlider.IsEnabled = True
        CounterSlider.Opacity = 1.0

        StatusTextBlock.Text = "Stopped."

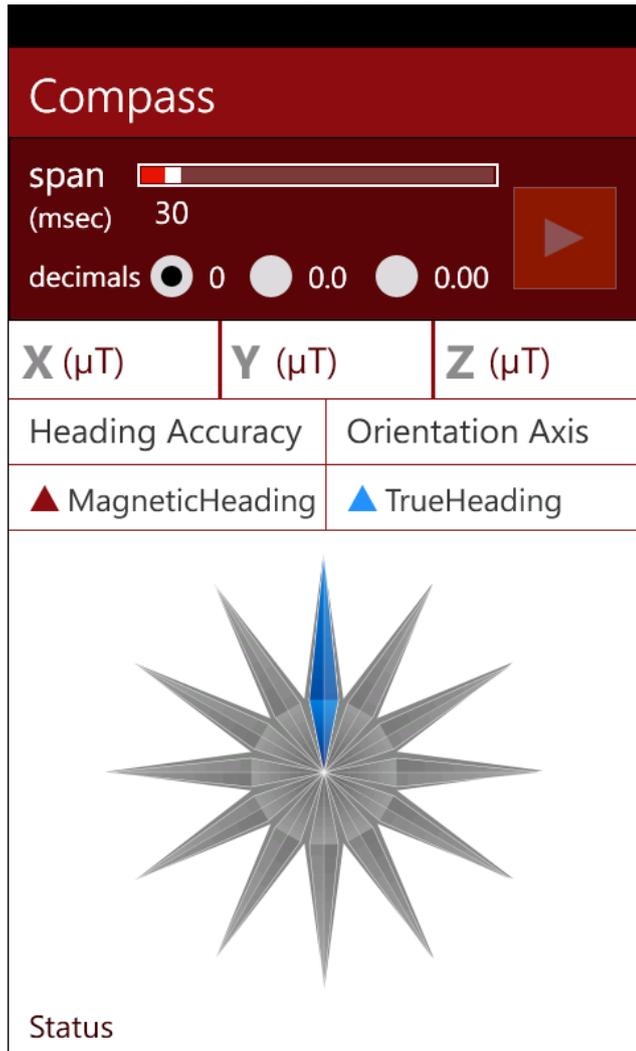
    Catch
        MessageBox.Show("Error stopping gyroscope.")
    End Try
End Sub
End Class
```

## 5. Compass

### 5.1 Layout of "Compass" page

Compass provides the orientation of the device on a map, or the magnetic field of the earth. (●fig 7)

●fig 7 "Compass" Page



You can get various values from the sensor.

#### (1) The raw magnetometer

The raw magnetometer on the X, Y, and Z axes (microteslas)

✂ [Microsoft.Devices.Sensors namespace](#) / [CompassReading Structure](#) / [MagnetometerReading property](#)

#### (2) The heading accuracy

If the heading accuracy is 15 or over, the compass calibration dialog opens.

✂ *Microsoft.Devices.Sensors Namespace / CompassReading Structure / HeadingAccuracy Property*

### (3) A current axis

It is a current axis to measure magnetometer using accelerometer.

### (4) The heading

The heading in degrees from the Earth's magnetic north.

✂ *Microsoft.Devices.Sensors Namespace / CompassReading Structure / MagneticHeading Property*

The heading in degrees from the Earth's geographic north.

✂ *Microsoft.Devices.Sensors Namespace / CompassReading Structure / TrueHeading Property*

### (5) The hands

The hands rotate in degrees from the Earth's magnetic north and geographic north.

### (6) The status of compass

The XAML markup code is like this. (●list 7)

●list 7 A part of CompassPage1.xaml

```
<phone:PhoneApplicationPage
  x:Class="SensorsSet.CompassPage1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  FontFamily="{StaticResource PhoneFontFamilyNormal}"
  FontSize="{StaticResource PhoneFontSizeNormal}"
  Foreground="{StaticResource PhoneForegroundBrush}"
  SupportedOrientations="Portrait" Orientation="Portrait"
  mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
  shell:SystemTray.IsVisible="True">

  <!--LayoutRoot is the root grid where all page content is placed-->
```

```

<Grid x:Name="LayoutRoot" Background="#FF890E0E">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
    <RowDefinition Height="*/>
  </Grid.RowDefinitions>

  <!--TitlePanel contains the name of the application and page title-->
  <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="15,10,0,10">
    <TextBlock x:Name="PageTitle" Text="Compass" Margin="0,0,0,0"
Style="{StaticResource PhoneTextTitle1Style}" FontSize="36" Foreground="White" />
  </StackPanel>

  <!--ContentPanel - place additional content here-->
  <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">

    <!--●●設定 (背景) -->
    <Rectangle Height="140" HorizontalAlignment="Left"
Name="SettingsBack" Stroke="Black" StrokeThickness="1" VerticalAlignment="Top"
Width="480" Fill="#FF5A0505"/>

    <!--●間隔-->
    <TextBlock Height="50" HorizontalAlignment="Left" Margin="15,8,0,0"
Name="TimeSpanLabel" Text="span" VerticalAlignment="Top" Width="70"
FontSize="28" Foreground="White" />
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="15,45,0,0"
Name="TimeSpanLabe2" Text="(msec)" VerticalAlignment="Top" Width="80"
FontSize="22" Foreground="White" />
    <TextBlock Height="40" HorizontalAlignment="Left" Margin="93,40,0,0"
Name="TimeSpanTextBlock" Text="30" VerticalAlignment="Top" Width="60"
FontSize="24" TextAlignment="Center" Foreground="White" />
    <Rectangle x:Name="SliderBorder" Height="18" Margin="97,20,0,0"
Stroke="White" VerticalAlignment="Top" Width="273" HorizontalAlignment="Left"
StrokeThickness="2"/>
    <Slider Height="85" HorizontalAlignment="Left" Margin="88,0,0,0"
Name="TimespanSlider" VerticalAlignment="Top" Width="290"
Orientation="Horizontal" Value="3" Maximum="30" Minimum="1" LargeChange="1"
SmallChange="10" />

    <!--●精度-->
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="15,90,0,0"
Name="RadioLabel1" Text="decimals" VerticalAlignment="Top" Width="90"

```

```

FontSize="22" Foreground="White" />
    <RadioButton Content="0" Height="72" HorizontalAlignment="Left"
Margin="95,70,0,0" Name="RadioButton1" VerticalAlignment="Top" Width="85"
IsChecked="True" Foreground="White" Background="Gainsboro" />
    <RadioButton Content="0.0" Height="72" HorizontalAlignment="Left"
Margin="170,70,0,0" Name="RadioButton10" VerticalAlignment="Top" Width="100"
Foreground="White" Background="Gainsboro" />
    <RadioButton Content="0.00" Height="72" HorizontalAlignment="Left"
Margin="265,70,0,0" Name="RadioButton100" VerticalAlignment="Top" Width="110"
Foreground="White" Background="Gainsboro" />

<!--●ボタン-->
    <Button Style="{StaticResource StopButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StopButton"
VerticalAlignment="Top" Width="78" IsEnabled="False" Visibility="Collapsed" />
    <Button Style="{StaticResource StartButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StartButton"
VerticalAlignment="Top" Width="78" />

<!--●計測結果(背景とラベル)-->
    <Rectangle Height="59" HorizontalAlignment="Left" Margin="0,140,0,0"
Name="MagnetometerBackX" VerticalAlignment="Top" Width="158" Fill="White" />
    <Rectangle Height="59" HorizontalAlignment="Left"
Margin="161,140,0,0" Name="MagnetometerBackY" VerticalAlignment="Top"
Width="158" Fill="White" />
    <Rectangle Height="59" HorizontalAlignment="Left"
Margin="322,140,0,0" Name="MagnetometerBackZ" VerticalAlignment="Top"
Width="158" Fill="White" />
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="10,145,0,0"
Name="MagnetometerLabelX" Text="X" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="168,145,0,0"
Name="MagnetometerLabelY" Text="Y" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="330,145,0,0"
Name="MagnetometerLabelZ" Text="Z" VerticalAlignment="Top" Width="30"
Foreground="#FF888888" FontSize="36" FontWeight="Bold" />

<!--●計測結果(磁気強度)-->
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="40,150,0,0"
Name="MagnetometerX" Text="(μT)" VerticalAlignment="Top" Width="115"

```

```
FontSize="28" Foreground="#FF5A0505"/>
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="201,150,0,0"
Name="MagnetometerY" Text="(μT)" VerticalAlignment="Top" Width="115"
FontSize="28" Foreground="#FF5A0505" />
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="362,150,0,0"
Name="MagnetometerZ" Text="(μT)" VerticalAlignment="Top" Width="115"
FontSize="28" Foreground="#FF5A0505" />
```

```
<!--●計測状況-->
    <Rectangle Height="49" HorizontalAlignment="Left" Margin="0,200,0,0"
Name="HeadingAccuracyBack" VerticalAlignment="Top" Width="239" Fill="White" />
    <Rectangle Height="49" HorizontalAlignment="Left"
Margin="240,200,0,0" Name="OrientationBack" VerticalAlignment="Top" Width="239"
Fill="White" />
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="15,205,0,0"
Name="HeadingAccuracyValue" Text="Heading Accuracy" VerticalAlignment="Top"
Width="220" FontSize="26" Foreground="#FF333333"/>
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="255,205,0,0"
Name="OrientationAxis" Text="Orientation Axis" VerticalAlignment="Top" Width="220"
FontSize="26" Foreground="#FF333333"/>
```

```
<!--●計測状況-->
    <Rectangle Height="49" HorizontalAlignment="Left" Margin="0,250,0,0"
Name="MagneticHeadingBack" VerticalAlignment="Top" Width="239" Fill="White" />
    <Rectangle Height="49" HorizontalAlignment="Left"
Margin="240,250,0,0" Name="TrueHeadingBack" VerticalAlignment="Top" Width="239"
Fill="White" />
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="15,255,0,0"
Name="MagneticHandColor" Text=" ▲ " VerticalAlignment="Top" Width="30"
FontSize="28" Foreground="#FF890E0E"/>
    <TextBlock Height="42" HorizontalAlignment="Left" Margin="44,259,0,0"
Name="MagneticHeadingAngle" Text="MagneticHeading" VerticalAlignment="Top"
Width="190" FontSize="24" Foreground="#FF333333"/>
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="255,255,0,0"
Name="TrueHandColor" Text="▲" VerticalAlignment="Top" Width="30" FontSize="28"
Foreground="DodgerBlue"/>
    <TextBlock Height="42" HorizontalAlignment="Left" Margin="284,259,0,0"
Name="TrueHeadingAngle" Text="TrueHeading" VerticalAlignment="Top" Width="190"
FontSize="24" Foreground="#FF333333"/>
```

```
<!--●計測結果表示 (方角)-->
<Rectangle Height="400" HorizontalAlignment="Left" Margin="0,300,0,0"
Name="CompassHandBack" VerticalAlignment="Top" Width="480" Fill="White" />
```

```
<!--●計測結果表示 (コンパス)-->
<Canvas Height="332" HorizontalAlignment="Left" Margin="72,318,0,0"
Name="Canvas1" VerticalAlignment="Top" Width="332">
    <Image Height="332" Canvas.Left="0" Canvas.Top="0" Name="Angle"
Stretch="Fill" Width="332" Source="/SensorsSet;component/Image/Angle.png" />
    <Image Height="332" Canvas.Left="0" Canvas.Top="0"
Name="MagneticHand" Stretch="Fill" Width="332"
Source="/SensorsSet;component/Image/MagneticHand.png" >
        <Image.Projection>
            <PlaneProjection x:Name="MyMagneticHandProjection" />
        </Image.Projection>
    </Image>
    <Image Height="332" Canvas.Left="0" Canvas.Top="0"
Name="TrueHand" Stretch="Fill" Width="332"
Source="/SensorsSet;component/Image/TrueHand.png" >
        <Image.Projection>
            <PlaneProjection x:Name="MyTrueHandProjection" />
        </Image.Projection>
    </Image>
</Canvas>
```

```
<!--●キャリブレーションの注意画面-->
<Image Height="354" HorizontalAlignment="Left" Margin="0,300,0,0"
Name="ImageMessage" Stretch="Fill" VerticalAlignment="Top" Width="480"
Source="/SensorsSet;component/Image/CompassMessage.png"
Visibility="Collapsed"/>
```

```
<!--●状態-->
<TextBlock Height="40" HorizontalAlignment="Left" Margin="15,660,0,0"
Name="StatusTextBlock" Text="Status" VerticalAlignment="Top" Width="460"
FontSize="24" Foreground="#FF370202" />
```

```
</Grid>
</Grid>
```

```
<!--Sample code showing usage of ApplicationBar-->
```

## 5.2 Programming of “Compass” page

Visual Basic code is like this (●list 8)

### ●list 8 CompassPage1.xaml.vb

```
Option Strict On
```

```
Imports Microsoft.Devices.Sensors
```

```
Imports Microsoft.Xna.Framework
```

```
Partial Public Class CompassPage1
```

```
    Inherits PhoneApplicationPage
```

```
    Public Sub New()
```

```
        InitializeComponent()
```

```
    End Sub
```

```
    Dim MyCompass As Compass 'コンパスセンサー
```

```
    Dim MyAccelerometer As Accelerometer '加速度センサー
```

```
    Dim MyTimeSpan As Integer = 30 '計測間隔。デフォルトは 30 ミリ秒とする
```

```
    Dim MyAccuracy As String = "F0" '精度。整数、小数点以下 1 桁、小数点以下 2 桁から選択
```

```
'■このページがロードされた時の処理
```

```
    Private Sub CompassPage1_Loaded(sender As Object, e As System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

```
        '●コンパスセンサーがサポートされていない機種への対応
```

```
        If Compass.IsSupported = False Then
```

```
            MessageBox.Show("Device does not support compass.")
```

```
            '「開始」「停止」ボタンどちらも使用不可とする
```

```
            'Device does not support compass.
```

```
            StartButton.IsEnabled = False
```

```
            StartButton.Opacity = 0.3
```

```
            StopButton.IsEnabled = False
```

```
            Exit Sub
```

```
        End If
```

'●加速度センサーがサポートされていない機種への対応

```
If Accelerometer.IsSupported = False Then
```

```
    MessageBox.Show("Compass and accelerometer is used on this page.  
Device does not support accelerometer.")
```

```
    ' 「開始」「停止」 ボタンどちらも使用不可とする
```

```
    StartButton.IsEnabled = False
```

```
    StartButton.Opacity = 0.3
```

```
    StopButton.IsEnabled = False
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

'■精度設定

'計測値を表示する際の桁数を設定

```
Private Sub RadioButton1_Checked(sender As Object, e As  
System.Windows.RoutedEventArgs) Handles RadioButton1.Checked
```

```
    MyAccuracy = "F0"
```

```
End Sub
```

```
Private Sub RadioButton10_Checked(sender As Object, e As  
System.Windows.RoutedEventArgs) Handles RadioButton10.Checked
```

```
    MyAccuracy = "F1"
```

```
End Sub
```

```
Private Sub RadioButton100_Checked(sender As Object, e As  
System.Windows.RoutedEventArgs) Handles RadioButton100.Checked
```

```
    MyAccuracy = "F2"
```

```
End Sub
```

'■間隔設定

'スライダーの値は 1~30 で 1 刻み設定 (デフォルトは 3)、計測には 10 を乗算した  
値を使用、単位はミリ秒

```
Private Sub TimeSpanSlider_ValueChanged(sender As Object, e As  
System.Windows.RoutedPropertyChangedEventArgs(Of Double)) Handles  
TimeSpanSlider.ValueChanged
```

```
    MyTimeSpan = CInt(TimeSpanSlider.Value) * 10
```

```
    TimeSpanTextBlock.Text = CStr(MyTimeSpan) '設定値を表示
```

```
    TimeSpanTextBlock.Margin = New Thickness(78 + TimeSpanSlider.Value * 8, 40,  
0, 0) '設定値の TextBlock をスライダーに合わせて移動
```

End Sub

'■「開始」ボタンがタップされた時の処理

```
Private Sub StartButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StartButton.Tap
```

'「開始」ボタンは非表示にして「停止」ボタンを表示

```
StartButton.Visibility = Windows.Visibility.Collapsed
```

```
StartButton.IsEnabled = False
```

```
StopButton.Visibility = Windows.Visibility.Visible
```

```
StopButton.IsEnabled = True
```

'間隔設定スライダーは使用不可とする

```
TimespanSlider.IsEnabled = False
```

```
TimespanSlider.Opacity = 0.5
```

'コンパスセンサーオブジェクトが有効な場合は、停止させる

'開始と停止を2個のボタンで切り替えているため、開始ボタンをタップした時は常にオブジェクトは空。

'技術イベント COD2012 での説明用であり、この条件分岐は実際は不要。

'開始と停止をボタン1個で切り替える場合は必要。

```
If MyCompass IsNot Nothing AndAlso MyCompass.IsDataValid Then
```

```
MyCompass.Stop()
```

```
MessageBox.Show("Compass stopped.")
```

```
Else
```

If MyCompass Is Nothing Then 'コンパスセンサーオブジェクトが空の場合の処理

MyCompass = New Compass 'コンパスセンサーの新しいインスタンスを生成

MyAccelerometer = New Accelerometer '加速度センサーの新しいインスタンスを生成

```
MyCompass.TimeBetweenUpdates = TimeSpan.FromMilliseconds(MyTimeSpan) '間隔設定スライダーで指定した 10 ミリ秒〜300 ミリ秒単位で更新
```

'コンパスセンサーの値の更新処理を実行

```
AddHandler MyCompass.CurrentValueChanged, Sub(senderValue As Object, eValue As SensorReadingEventArgs(Of CompassReading))
```

```
Deployment.Current.Dispatcher.BeginInvoke(Sub() UpdateUI(eValue.SensorReading))
```

End Sub

End If

Try

'コンパスセンサーと加速度センサーのセンシングを開始。状態に「計測中」と表示

StatusTextBlock.Text = "Starting compass and accelerometer..."

MyCompass.Start() 'コンパスセンサー取得開始

MyAccelerometer.Start() '加速度センサー取得開始

Catch

'センシングが失敗した場合の処理。状態に、メッセージを表示

StatusTextBlock.Text = "Unable to start compass or accelerometer."

Exit Sub

End Try

End If

End Sub

'■センシングが実行された時の処理

Private Sub UpdateUI(MyReading As CompassReading)

'StatusTextBlock.Text = "Getting data..."

'磁気強度

'コンパスセンサーによって X, Y, Z 軸に対する検出された磁場の Raw データ (マイクロテスラ単位)

Dim MagnetMeterX = MyReading.MagnetometerReading.X '3 軸に分解

Dim MagnetMeterY = MyReading.MagnetometerReading.Y

Dim MagnetMeterZ = MyReading.MagnetometerReading.Z

'方位測定精度。CompassReading 構造体 HeadingAccuracy プロパティで取得される値

Dim MyheadingAccuracy = Math.Abs(MyReading.HeadingAccuracy)

HeadingAccuracyValue.Text = "Accuracy : " & MyheadingAccuracy.ToString

'地磁気上の角度。地球の磁北から時計回りに測定された方位(度数)

'CompassReading 構造体 MagneticHeading プロパティで取得される値

Dim MyMagneticAngle As Double = MyReading.MagneticHeading

'地図上の角度。地球の地理上の北から時計回りに測定された方位(度数)

'CompassReading 構造体 TrueHeading プロパティで取得される値

Dim MyTrueAngle As Double = MyReading.TrueHeading

```
'2種類の角度をグラフィカルに表示する時の針の回転角度
Dim MyMagneticHandAngle As Integer = 0 '地磁気上の針の角度
Dim MyTrueHandAngle As Integer = 0 '地図上の針の角度
```

'●数値表示

'ラジオボタンで設定した桁数に応じて表示桁数を決定

```
MagnetometerX.Text = MagnetMeterX.ToString(MyAccuracy)
```

```
MagnetometerY.Text = MagnetMeterY.ToString(MyAccuracy)
```

```
MagnetometerZ.Text = MagnetMeterZ.ToString(MyAccuracy)
```

```
MagneticHeadingAngle.Text = "Magnet : " &
```

```
MyMagneticAngle.ToString(MyAccuracy) & "° "
```

```
TrueHeadingAngle.Text = "True : " & MyTrueAngle.ToString(MyAccuracy) & "° "
```

'●グラフィックの表示

```
MyMagneticHandAngle = CInt(MyMagneticAngle) '地磁気上の針の角度
```

```
MyTrueHandAngle = CInt(MyTrueAngle) '地図上の針の角度
```

'●方位測定精度が 15 より小さい時の処理。

'IS12T はキャリブレーション済みで常に「10」、他機種の場合は「9」に変更してテスト

```
If MyheadingAccuracy < 15 Then
```

```
    ImageMessage.Visibility = Windows.Visibility.Collapsed
```

```
'地磁気上の磁石回転。時計回りで表示するため 360 より角度を減算
```

```
MyMagneticHandProjection.RotationZ = 360 - MyMagneticHandAngle
```

```
'地図上の磁石回転。時計回りで表示するため 360 より角度を減算
```

```
MyTrueHandProjection.RotationZ = 360 - MyTrueHandAngle
```

```
'方角を計測している軸を表示
```

```
Dim MyAccelerationZ = MyAccelerometer.CurrentValue.Acceleration.Z
```

```
Dim MyAccelerationY = MyAccelerometer.CurrentValue.Acceleration.Y
```

```
'計測に使われている軸を表示
```

'Cos 45° = 0.7 につき、Math.Cos(Math.PI / 4)より小さければ、加速度の Z の絶対値が 0.7 より小さい状態

'Sin 315° = -0.7 であるから、Math.Sin(7 \* Math.PI / 4)より小さい、つまり加速度の Y の値が -0.7 より小さい状態の時は、Z 軸により計測

```
'それ以外の時は Y 軸で計測
```

```
If Math.Abs(MyAccelerationZ) <= Math.Cos(Math.PI / 4) And
MyAccelerationY < Math.Sin(7 * Math.PI / 4) Then
```

```
    OrientationAxis.Text = "Z axis"
```

```

        OrientationAxis.Foreground = New SolidColorBrush(Colors.Green)
    Else
        OrientationAxis.Text = "Y axis"
        OrientationAxis.Foreground = New SolidColorBrush(Colors.Orange)
    End If
Else
    '● キャリブレーションの精度によっては、メッセージ画面を表示
    ImageMessage.Visibility = Windows.Visibility.Visible
End If
End Sub

'■ 「停止」 ボタンがタップされた時の処理
Private Sub StopButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles StopButton.Tap
    '● 「START」 ボタンを表示し、コンパスと加速度センサーを停止。
    Try
        MyCompass.Stop()
        MyCompass = Nothing

        MyAccelerometer.Stop()

        StopButton.Visibility = Windows.Visibility.Collapsed
        StopButton.IsEnabled = False
        StartButton.Visibility = Windows.Visibility.Visible
        StartButton.IsEnabled = True

        TimespanSlider.IsEnabled = True
        TimespanSlider.Opacity = 1.0
        StatusTextBlock.Text = "Compass and accelerometer stopped."

    Catch
        StatusTextBlock.Text = "Error stopping Compass or accelerometer."

    End Try
End Sub
End Class

```

## Revise

Revise the codes of "Compasspage1.xaml.vb".

【Line 127】 測定精度 > Accuracy

【Line146】 地磁気 > Magnet

【Line147】 地図 > True

This bug was reported by the user.

*Good app. Just ONE problem. When I started the compass the text changed from English to Japanese. (2012/08/29)*

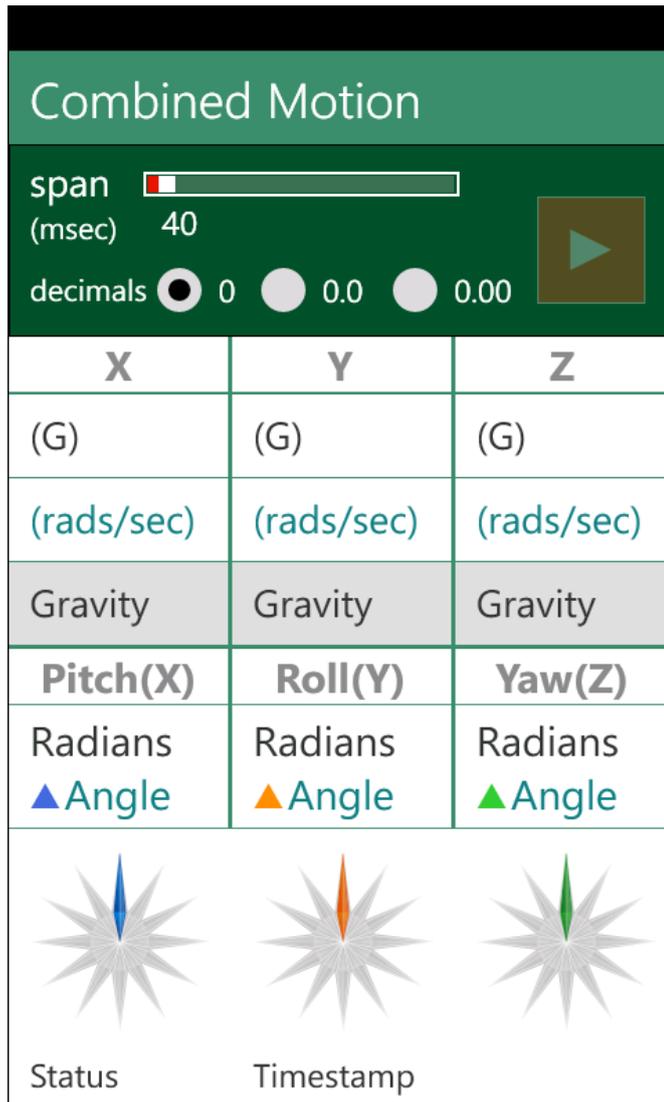
Thank you, Portuguese user.

## 6. Combined Motion API

### 6.1 Layout of "Combined Motion" page

This API provides the various data about the orientation and movement of the device. (● fig 8)

● fig 8 "Combined Motion" Page



You can get various values from the sensor.

#### (1) The acceleration

The linear acceleration on the X, Y, and Z axes in gravitational units.

✂ [Microsoft.Devices.Sensors Namespace / MotionReading Structure / DeviceAcceleration Property](#)

## (2) The device rotation

It is the rotational velocity in radians per second.

✂ *Microsoft.Devices.Sensors Namespace / MotionReading Structure / DeviceRotationRate Property*

## (3) The gravity

The gravity on the X, Y, and Z axes.

The value is from -1 to 1.

✂ *Microsoft.Devices.Sensors Namespace / MotionReading Structure / Gravity Property.*

## (4) The attitude

It is the attitude of the device in radians.

"Pitch" is the movement around an X axis. "Roll" is the movement around a Y axis. And "yaw" is the movement around a Z axis.

✂ *Microsoft.Devices.Sensors Namespace / MotionReading Structure / Attitude Property*

## (5) The angles

The angles of pitch, roll, and yaw.

## (6) The hands

The hands point the angles of pitch, roll, and yaw.

## (7) Status

It is the status of combined Motion API.

The XAML markup code is like this. (●list 9)

### ●list 9 A part of MotionPage1.xaml

```
<phone:PhoneApplicationPage
    x:Class="SensorsSet.MotionPage1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    FontFamily="{StaticResource PhoneFontFamilyNormal}"
    FontSize="{StaticResource PhoneFontSizeNormal}"
```

```

Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
shell:SystemTray.IsVisible="True">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="#FF388F67">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="15,10,0,10">
        <TextBlock x:Name="PageTitle" Text="Combined Motion" Margin="0,0,0,0"
Style="{StaticResource PhoneTextTitle1Style}" FontSize="36" Foreground="White" />
    </StackPanel>

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">

        <!--●●設定 (背景) -->
        <Rectangle Height="140" HorizontalAlignment="Left"
Name="SettingsBack" Stroke="Black" StrokeThickness="1" VerticalAlignment="Top"
Width="480" Fill="#FF04502A"/>

        <!--●間隔-->
        <TextBlock Height="50" HorizontalAlignment="Left" Margin="15,8,0,0"
Name="TimeSpanLabel" Text="span" VerticalAlignment="Top" Width="90"
FontSize="28" Foreground="White" />
        <TextBlock Height="30" HorizontalAlignment="Left" Margin="15,45,0,0"
Name="TimeSpanLabe2" Text="(msec)" VerticalAlignment="Top" Width="80"
FontSize="22" Foreground="White" />
        <TextBlock Height="40" HorizontalAlignment="Left" Margin="93,40,0,0"
Name="TimeSpanTextBlock" Text="40" VerticalAlignment="Top" Width="60"
FontSize="24" TextAlignment="Center" Foreground="White" />
        <Rectangle x:Name="SliderBorder" Height="18" Margin="97,20,0,0"
Stroke="White" VerticalAlignment="Top" Width="228" HorizontalAlignment="Left"
StrokeThickness="2"/>
        <Slider Height="85" HorizontalAlignment="Left" Margin="88,0,0,0"
Name="TimespanSlider" VerticalAlignment="Top" Width="245"

```

```
Orientation="Horizontal" Value="4" Maximum="30" Minimum="3" LargeChange="10"
SmallChange="10" />
```

```
<!--●精度-->
```

```
<TextBlock Height="30" HorizontalAlignment="Left" Margin="15,90,0,0"
Name="RadioLabel1" Text="decimals" VerticalAlignment="Top" Width="90"
FontSize="22" Foreground="White" />
```

```
<RadioButton Content="0" Height="72" HorizontalAlignment="Left"
Margin="95,70,0,0" Name="RadioButton1" VerticalAlignment="Top" Width="85"
IsChecked="True" Foreground="White" Background="Gainsboro" />
```

```
<RadioButton Content="0.0" Height="72" HorizontalAlignment="Left"
Margin="170,70,0,0" Name="RadioButton10" VerticalAlignment="Top" Width="100"
Foreground="White" Background="Gainsboro" />
```

```
<RadioButton Content="0.00" Height="72" HorizontalAlignment="Left"
Margin="265,70,0,0" Name="RadioButton100" VerticalAlignment="Top" Width="110"
Foreground="White" Background="Gainsboro" />
```

```
<!--●ボタン-->
```

```
<Button Style="{StaticResource StopButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StopButton"
VerticalAlignment="Top" Width="78" IsEnabled="False" Visibility="Collapsed" />
```

```
<Button Style="{StaticResource StartButtonStyle}" Height="78"
HorizontalAlignment="Left" Margin="381,38,0,0" Name="StartButton"
VerticalAlignment="Top" Width="78" />
```

```
<!--●計測結果(軸 ラベル)-->
```

```
<Rectangle Height="40" HorizontalAlignment="Left" Margin="0,140,0,0"
Name="XLabelBak" VerticalAlignment="Top" Width="158" Fill="White" />
```

```
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="161,140,0,0" Name="YLabelBak" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="322,140,0,0" Name="ZLabelBak" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="0,140,0,0"
Name="XLabel" Text="X" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="160,140,0,0"
Name="YLabel" Text="Y" VerticalAlignment="Top" Width="158"
```

```
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="320,140,0,0"
Name="ZLabel" Text="Z" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<!--●計測結果(Device Acceleration 背景)-->
```

```
<Rectangle Height="60" HorizontalAlignment="Left" Margin="0,182,0,0"
Name="DeviceAccelerationXBack" VerticalAlignment="Top" Width="158" Fill="White"
/>
```

```
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="161,182,0,0" Name="DeviceAccelerationYBack" VerticalAlignment="Top"
Width="158" Fill="White" />
```

```
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="322,182,0,0" Name="DeviceAccelerationZBack" VerticalAlignment="Top"
Width="158" Fill="White" />
```

```
<!--●計測結果(Device Acceleration)-->
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,192,0,0"
Name="DeviceAccelerationX" Text="(G)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333"/>
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,192,0,0"
Name="DeviceAccelerationY" Text="(G)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,192,0,0"
Name="DeviceAccelerationZ" Text="(G)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF333333" />
```

```
<!--●計測結果(Device RotationRate 背景)-->
```

```
<Rectangle Height="60" HorizontalAlignment="Left" Margin="0,243,0,0"
Name="DeviceRotationRateXBack" VerticalAlignment="Top" Width="158" Fill="White"
/>
```

```
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="161,243,0,0" Name="DeviceRotationRateYBack" VerticalAlignment="Top"
Width="158" Fill="White" />
```

```
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="322,243,0,0" Name="DeviceRotationRateZBack" VerticalAlignment="Top"
Width="158" Fill="White" />
```

```
<!--●計測結果(Device RotationRate)-->
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,253,0,0"
Name="DeviceRotationRateX" Text="(rads/sec)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,253,0,0"
Name="DeviceRotationRateY" Text="(rads/sec)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,253,0,0"
Name="DeviceRotationRateZ" Text="(rads/sec)" VerticalAlignment="Top" Width="140"
FontSize="28" Foreground="#FF0f8187" />
```

```
<!--●計測結果(Device RotationRate 背景)-->
<Rectangle Height="60" HorizontalAlignment="Left" Margin="0,304,0,0"
Name="GravityXBack" VerticalAlignment="Top" Width="158" Fill="#FFDDDDDD" />
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="161,304,0,0" Name="GravityYBack" VerticalAlignment="Top" Width="158"
Fill="#FFDDDDDD" />
<Rectangle Height="60" HorizontalAlignment="Left"
Margin="322,304,0,0" Name="GravityZBack" VerticalAlignment="Top" Width="158"
Fill="#FFDDDDDD" />
```

```
<!--●計測結果(Device RotationRate)-->
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,314,0,0"
Name="GravityX" Text="Gravity" VerticalAlignment="Top" Width="140" FontSize="28"
Foreground="#FF333333"/>
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,314,0,0"
Name="GravityY" Text="Gravity" VerticalAlignment="Top" Width="140" FontSize="28"
Foreground="#FF333333" />
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,314,0,0"
Name="GravityZ" Text="Gravity" VerticalAlignment="Top" Width="140" FontSize="28"
Foreground="#FF333333" />
```

```
<!--●計測結果(Attitude ラベル)-->
<Rectangle Height="40" HorizontalAlignment="Left" Margin="0,367,0,0"
Name="PitchLabelBak" VerticalAlignment="Top" Width="158" Fill="White" />
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="161,367,0,0" Name="RollLabelBak" VerticalAlignment="Top" Width="158"
Fill="White" />
<Rectangle Height="40" HorizontalAlignment="Left"
Margin="322,367,0,0" Name="YawLabelBak" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="0,367,0,0"
Name="PitchLabel" Text="Pitch(X)" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="160,367,0,0"
Name="RollLabel" Text="Roll(Y)" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<TextBlock Height="40" HorizontalAlignment="Left" Margin="320,367,0,0"
Name="YawLabel" Text="Yaw(Z)" VerticalAlignment="Top" Width="158"
Foreground="#FF888888" FontSize="30" FontWeight="Bold" TextAlignment="Center"
/>
```

```
<!--●計測結果(Attitude 背景)-->
```

```
<Rectangle Height="89" HorizontalAlignment="Left" Margin="0,408,0,0"
Name="PitchValueBak" VerticalAlignment="Top" Width="158" Fill="White" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="161,408,0,0" Name="RollValueBack" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<Rectangle Height="89" HorizontalAlignment="Left"
Margin="322,408,0,0" Name="YawValueBack" VerticalAlignment="Top" Width="158"
Fill="White" />
```

```
<!--●計測結果(Attitude ラジアン)-->
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="15,413,0,0"
Name="PitchRad" Text="Radians" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333"/>
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="176,413,0,0"
Name="RollRad" Text="Radians" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333" />
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="337,413,0,0"
Name="YawRad" Text="Radians" VerticalAlignment="Top" Width="140" FontSize="30"
Foreground="#FF333333" />
```

```
<!--●計測結果(Attitude 角度)-->
```

```
<TextBlock Height="45" HorizontalAlignment="Left" Margin="15,455,0,0"
Name="PitchHandImage" Text="▲" VerticalAlignment="Top" Width="30" FontSize="26"
Foreground="RoyalBlue"/>
```

```
<TextBlock Height="48" HorizontalAlignment="Left" Margin="40,452,0,0"
Name="PitchAngle" Text="Angle" VerticalAlignment="Top" Width="115" FontSize="30"
```

```

Foreground="#FF0f8187" />
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="176,455,0,0"
Name="RollHandImage" Text="▲" VerticalAlignment="Top" Width="30" FontSize="26"
Foreground="DarkOrange"/>
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="201,452,0,0"
Name="RollAngle" Text="Angle" VerticalAlignment="Top" Width="115" FontSize="30"
Foreground="#FF0f8187" />
    <TextBlock Height="45" HorizontalAlignment="Left" Margin="337,455,0,0"
Name="YawHandImage" Text="▲" VerticalAlignment="Top" Width="30" FontSize="26"
Foreground="LimeGreen"/>
    <TextBlock Height="48" HorizontalAlignment="Left" Margin="362,452,0,0"
Name="YawAngle" Text="Angle" VerticalAlignment="Top" Width="115" FontSize="30"
Foreground="#FF0f8187" />

<!--●計測結果表示 (図形)-->
    <Rectangle Height="203" HorizontalAlignment="Left" Margin="0,498,0,0"
Name="GraphBack" VerticalAlignment="Top" Width="480" Fill="White" />
    <Image Height="130" HorizontalAlignment="Left" Margin="15,515,0,0"
Name="PitchBack" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/PitchRollYawAngle.png" />
    <Image Height="130" HorizontalAlignment="Left" Margin="176,515,0,0"
Name="RollBack" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/PitchRollYawAngle.png" />
    <Image Height="130" HorizontalAlignment="Left" Margin="337,515,0,0"
Name="YawBack" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/PitchRollYawAngle.png" />

    <Image Height="130" HorizontalAlignment="Left" Margin="15,515,0,0"
Name="PitchHand" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/PitchHand.png">
    <Image.Projection>
        <PlaneProjection x:Name="MyPitchHandProjection" />
    </Image.Projection>
</Image>
    <Image Height="130" HorizontalAlignment="Left" Margin="176,515,0,0"
Name="RollHand" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/RollHand.png">
    <Image.Projection>
        <PlaneProjection x:Name="MyRollHandProjection" />
    </Image.Projection>
</Image>

```

```

        <Image Height="130" HorizontalAlignment="Left" Margin="337,515,0,0"
Name="YawHand" Stretch="Fill" VerticalAlignment="Top" Width="130"
Source="/SensorsSet;component/Image/YawHand.png">
        <Image.Projection>
            <PlaneProjection x:Name="MyYawHandProjection" />
        </Image.Projection>
    </Image>

    <!--●状態-->
    <TextBlock Height="40" HorizontalAlignment="Left" Margin="15,660,0,0"
Name="StatusTextBlock" Text="Status" VerticalAlignment="Top" Width="130"
FontSize="24" Foreground="#FF333333" />
    <TextBlock Height="40" HorizontalAlignment="Left" Margin="176,660,0,0"
Name="TimeStampTextBlock" Text="Timestamp" VerticalAlignment="Top" Width="291"
FontSize="24" Foreground="#FF333333"/>

    </Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->

```

## 6.2 Programming of “Combined Motion” page

Visual Basic code is like this. (●list 10)

●list 10 MotionPage1.xaml.vb

Option Strict On

Imports Microsoft.Devices.Sensors

Imports Microsoft.Xna.Framework

Partial Public Class MotionPage1

Inherits PhoneApplicationPage

Public Sub New()

InitializeComponent()

End Sub

Dim MyMotion As Motion 'モーショセンサー

```
Dim MyTimeSpan As Integer = 40 '計測間隔。デフォルトは 40 ミリ秒とする
Dim MyAccuracy As String = "F0" '精度。整数、小数点以下 1 桁、小数点以下 2 桁から選択
```

```
Dim MyPitch, MyRoll, MyYaw As Single 'Pitch、Roll、Yaw の計測 Raw データ (rad)
Dim MyAttitudeX, MyAttitudeY, MyAttitudeZ As Single 'デバイスの角度データ
Dim MyDeviceAccerlation As Vector3 = Vector3.Zero '加速度 (G)。Vedtor3 型計測 Raw データ
```

```
Dim MyDeviceRotationRate As Vector3 = Vector3.Zero '角速度 (rad/sec) Vedtor3 型計測 Raw データ
```

```
Dim MyGravity As Vector3 = Vector3.Zero '重力の方向ベクトル。Vedtor3 型計測 Raw データ
```

'■このページがロードされた時の処理

```
Private Sub MotionPage1_Loaded(sender As Object, e As System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

'●モーションがサポートされていない機種への対応

```
If Motion.IsSupported = False Then
```

```
    MessageBox.Show("Device does not support the Combined Motion API.")
```

```
    '「開始」「停止」ボタンどちらも使用不可とする
```

```
    StartButton.IsEnabled = False
```

```
    StartButton.Opacity = 0.3
```

```
    StopButton.IsEnabled = False
```

```
    Exit Sub
```

```
End If
```

'●モーションはサポートしていてもジャイロスコープがサポートされていない機種 (Lumia800) への対応

```
If Gyroscope.IsSupported = False Then
```

```
    MessageBox.Show("Device does not support gyroscope in Combined Motion API.")
```

```
    '「開始」「停止」ボタンどちらも使用不可とする
```

```
    StartButton.IsEnabled = False
```

```
    StartButton.Opacity = 0.3
```

```
    StopButton.IsEnabled = False
```

```
    Exit Sub
```

```
End If
```

```
End Sub
```

### '■精度設定

'計測値を表示する際の桁数を設定

```
Private Sub RadioButton1_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton1.Checked
    MyAccuracy = "F0"
End Sub
```

```
Private Sub RadioButton10_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton10.Checked
    MyAccuracy = "F1"
End Sub
```

```
Private Sub RadioButton100_Checked(sender As Object, e As System.Windows.RoutedEventArgs) Handles RadioButton100.Checked
    MyAccuracy = "F2"
End Sub
```

### '■間隔設定

```
Private Sub TimespanSlider_ValueChanged(sender As Object, e As System.Windows.RoutedPropertyChangedEventArgs(Of Double)) Handles TimespanSlider.ValueChanged
```

'スライダーの値は 3～30 で 1 刻み設定 (デフォルトは 4)、計測には 10 を乗算した値を使用、単位はミリ秒

```
MyTimeSpan = CInt(TimespanSlider.Value) * 10
```

```
TimeSpanTextBlock.Text = CStr(MyTimeSpan) '設定値を表示
```

```
TimeSpanTextBlock.Margin = New Thickness(65 + TimespanSlider.Value * 7, 40, 0, 0) '設定値の TextBlock をスライダーに合わせて移動
```

```
End Sub
```

### '■「開始」ボタンがタップされた時の処理

```
Private Sub StartButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StartButton.Tap
```

'「開始」ボタンは非表示にして「停止」ボタンを表示

```
StartButton.Visibility = Windows.Visibility.Collapsed
```

```
StartButton.IsEnabled = False
```

```
StopButton.Visibility = Windows.Visibility.Visible
```

```
StopButton.IsEnabled = True
```

```

'間隔設定スライダーは使用不可とする
TimespanSlider.IsEnabled = False
TimespanSlider.Opacity = 0.5
'モーションオブジェクトが有効な場合は、停止させる
'開始と停止を 2 個のボタンで切り替えているため、開始ボタンをタップした時
は常にオブジェクトは空。
'技術イベント COD2012 での説明用であり、この条件分岐は、実際は不要。
'開始と停止をボタン 1 個で切り替える場合は必要。
If MyMotion IsNot Nothing AndAlso MyMotion.IsDataValid Then
    MyMotion.Stop()
    MessageBox.Show("Combined Motion stopped.")

Else
    If MyMotion Is Nothing Then 'モーションオブジェクトが空の場合の処理

        MyMotion = New Motion 'モーションの新しいインスタンスを生成
        MyMotion.TimeBetweenUpdates =
TimeSpan.FromMilliseconds(MyTimeSpan) '間隔設定スライダーで指定した 30 ミリ秒～
300 ミリ秒単位で更新
        'モーションの値の更新処理を実行
        AddHandler MyMotion.CurrentValueChanged, Sub(senderValue As
Object, eValue As SensorReadingEventArgs(Of MotionReading))

Deployment.Current.Dispatcher.BeginInvoke(Sub()
CurrentValueChanged(eValue.SensorReading))

End Sub

End If

Try
    'モーションのセンシングを開始。状態に「計測中」を表示
    StatusTextBlock.Text = "Starting.."

    MyMotion.Start()
Catch
    'センシングが失敗した場合の処理。状態に、メッセージを表示
    MessageBox.Show("Unable to start Combined Motion.")
Exit Sub
End Try
End If
End Sub

```

'■センシングが実行された時の処理

Private Sub CurrentValueChanged(MyReading As MotionReading)

'StatusTextBlock.Text = "Getting data..."

'Accelerometer クラスで取得される生データから重力加速度を除いた加速度

'MotionReading 構造体 Device Acceleration プロパティで取得される線形加速度 (G)

MyDeviceAccerlation = MyReading.DeviceAcceleration

'デバイスの回転速度

'MotionReading 構造体 DeviceRotationRate プロパティで取得される角速度 (rad/sec)

MyDeviceRotationRate = MyReading.DeviceRotationRate

'重力の方向

'MotionReading 構造体 Gravity プロパティで取得される重力の方向ベクトル。  
-1~1。

'加速度センサーに同じだが精度が高い

MyGravity = MyReading.Gravity

'デバイスの姿勢 (ピッチ、ロール、ヨー)

'MotionReading 構造体 Attitude プロパティで取得される、Pitch、Roll、Yaw の値 (rad)

MyPitch = MyReading.Attitude.Pitch

MyRoll = MyReading.Attitude.Roll

MyYaw = MyReading.Attitude.Yaw

'●グラフィックの表示

'デバイスの姿勢 (ピッチ、ロール、ヨー) を角度に変換

MyAttitudeX = MathHelper.ToDegrees(MyPitch)

MyAttitudeY = MathHelper.ToDegrees(MyRoll)

MyAttitudeZ = MathHelper.ToDegrees(MyYaw) '北を 0 として Yaw の角度が比較的正確。

'もとめた角度を指定して、3 軸それぞれの図形を時計回りに回転

MyPitchHandProjection.RotationZ = 360 - MyAttitudeX

MyRollHandProjection.RotationZ = 360 - MyAttitudeY

MyYawHandProjection.RotationZ = 360 - MyAttitudeZ

'●数値表示

'ラジオボタンで設定した桁数に応じて表示桁数を決定

```

'加速度の Raw データを 3 軸に分解
DeviceAccelerationX.Text = CStr(MyDeviceAcceralation.X.ToString(MyAccuracy))
& "G"
DeviceAccelerationY.Text = CStr(MyDeviceAcceralation.Y.ToString(MyAccuracy))
& "G"
DeviceAccelerationZ.Text = CStr(MyDeviceAcceralation.Z.ToString(MyAccuracy))
& "G"
'回転速度の Raw データを 3 軸に分解
DeviceRotationRateX.Text =
CStr(MyDeviceRotationRate.X.ToString(MyAccuracy))
DeviceRotationRateY.Text =
CStr(MyDeviceRotationRate.Y.ToString(MyAccuracy))
DeviceRotationRateZ.Text =
CStr(MyDeviceRotationRate.Z.ToString(MyAccuracy))
'重力の方向の Raw データを 3 軸に分解
GravityX.Text = CStr(MyGravity.X.ToString(MyAccuracy))
GravityY.Text = CStr(MyGravity.Y.ToString(MyAccuracy))
GravityZ.Text = CStr(MyGravity.Z.ToString(MyAccuracy))
'ピッチ、ロール、ヨーの Raw データ
PitchRad.Text = CStr(MyPitch.ToString(MyAccuracy))
RollRad.Text = CStr(MyRoll.ToString(MyAccuracy))
YawRad.Text = CStr(MyYaw.ToString(MyAccuracy))
'前述のグラフィックの表示用に計算した、ピッチ、ロール、ヨーの角度データ
PitchAngle.Text = CStr(MyAttitudeX.ToString(MyAccuracy)) & ""
RollAngle.Text = CStr(MyAttitudeY.ToString(MyAccuracy)) & ""
YawAngle.Text = CStr(MyAttitudeZ.ToString(MyAccuracy)) & ""

'現在時刻の表示
TimeStampTextBlock.Text = CStr(MyReading.Timestamp.ToString)

```

End Sub

'■ 「停止」 ボタンがタップされた時の処理

```

Private Sub StopButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles StopButton.Tap
'複合モーションを停止して、各種設定を初期化する
Try
MyMotion.Stop()
MyMotion = Nothing

StopButton.Visibility = Windows.Visibility.Collapsed

```

```
StopButton.IsEnabled = False
StartButton.Visibility = Windows.Visibility.Visible
StartButton.IsEnabled = True
TimespanSlider.IsEnabled = True
TimespanSlider.Opacity = 1.0

StatusTextBlock.Text = "Stopped."

Catch
    MessageBox.Show("Error stopping Combined Motion")

End Try
End Sub
End Class
```

## 7. Features of “今日も歩こう！”

### 7.1 How to use the application

前述のようなセンサーの機能は、いろいろなアプリ開発に利用できます。  
ここでは、複合モーション API の加速度データと角速度データの利用方法を紹介します。



「今日も歩こう！ Ver.0.8」は、終日 PC の前から動かないアプリ開発者が、歩く習慣をつけるためのツールです。（●fig 9）

計測データは XML ファイルとして保存し、リスト表示してソート可能にします。

●fig 9 今日も歩こう！ Ver.0.8 (日)

#### 機能と操作

##### (1) 計測

メインページの「肝臓」をタップして計測ページに移動します。

体重を半角数字で入力し、（必要であれば）スライダーで負荷を設定します。

「開始」ボタンをタップして歩くと、おおよその歩数と、消費カロリーが表示されます。  
歩き終わったら「誤動作防止シール」上をダブルタップしてから、「停止」ボタンをタップします。

##### (2) 保存

スライダーでの設定値、開始時刻、終了時刻、体重、歩数、消費カロリー、の各データが実機内に保存されます。

※歩くことなく保存すると、体重記録ツールになります。

##### (3) 表示

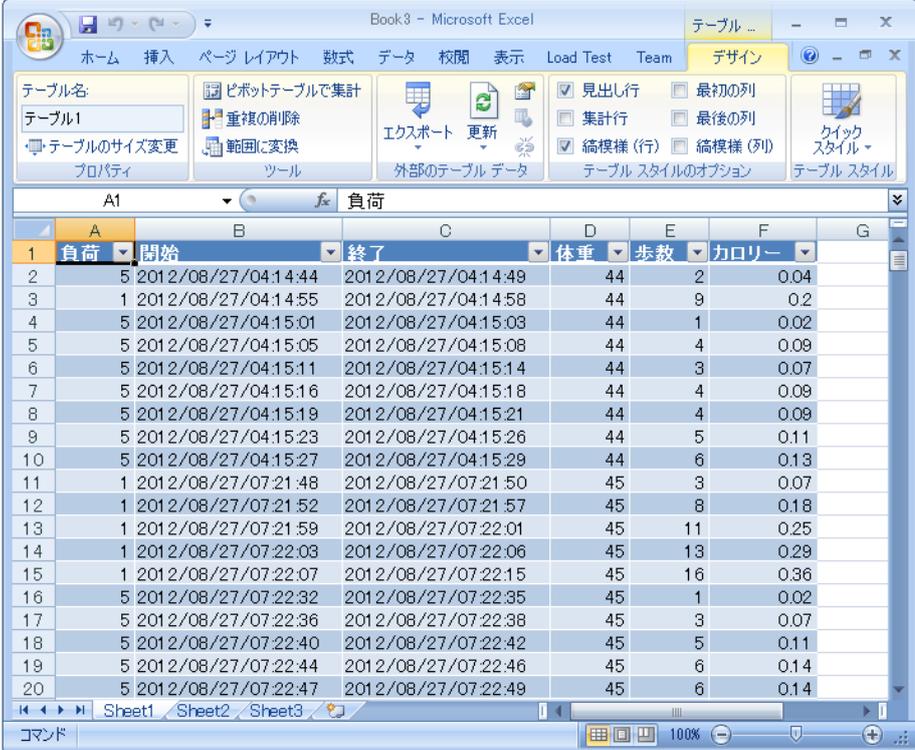
測定日、歩数、体重のデータを、記録時間の昇順あるいは降順で一覧表示できます。

#### 記録データの利用

データは、Windows Phone 実機内にインストールされている Walking プロジェクトの、

MyWalkingData フォルダ内の「MyWalkingData.xml」というファイルに保存されます。CodePlex からダウンロードできる「[Windows Phone Power Tools](http://wptools.codeplex.com/)」( <http://wptools.codeplex.com/> )を利用して、このファイルを PC 上に取得すれば、Microsoft Excel で直接開いて利用できます。(●fig 10)

●fig 10 記録データの Excel での表示例



	A	B	C	D	E	F	G
1	負荷	開始	終了	体重	歩数	カロリー	
2	5	2012/08/27/04:14:44	2012/08/27/04:14:49	44	2	0.04	
3	1	2012/08/27/04:14:55	2012/08/27/04:14:58	44	9	0.2	
4	5	2012/08/27/04:15:01	2012/08/27/04:15:03	44	1	0.02	
5	5	2012/08/27/04:15:05	2012/08/27/04:15:08	44	4	0.09	
6	5	2012/08/27/04:15:11	2012/08/27/04:15:14	44	3	0.07	
7	5	2012/08/27/04:15:16	2012/08/27/04:15:18	44	4	0.09	
8	5	2012/08/27/04:15:19	2012/08/27/04:15:21	44	4	0.09	
9	5	2012/08/27/04:15:23	2012/08/27/04:15:26	44	5	0.11	
10	5	2012/08/27/04:15:27	2012/08/27/04:15:29	44	6	0.13	
11	1	2012/08/27/07:21:48	2012/08/27/07:21:50	45	3	0.07	
12	1	2012/08/27/07:21:52	2012/08/27/07:21:57	45	8	0.18	
13	1	2012/08/27/07:21:59	2012/08/27/07:22:01	45	11	0.25	
14	1	2012/08/27/07:22:03	2012/08/27/07:22:06	45	13	0.29	
15	1	2012/08/27/07:22:07	2012/08/27/07:22:15	45	16	0.36	
16	5	2012/08/27/07:22:32	2012/08/27/07:22:35	45	1	0.02	
17	5	2012/08/27/07:22:36	2012/08/27/07:22:38	45	3	0.07	
18	5	2012/08/27/07:22:40	2012/08/27/07:22:42	45	5	0.11	
19	5	2012/08/27/07:22:44	2012/08/27/07:22:46	45	6	0.14	
20	5	2012/08/27/07:22:47	2012/08/27/07:22:49	45	6	0.14	

### 注意事項

このツールは、あくまで「毎日一歩からでも歩く」習慣を付けるためのものです。位置情報サービスは使っておらず、歩数計並みの精度を保証するものではありません。

### 制限事項

このツールでは、複合モーション API の加速度データと角速度データを使っているため、加速度センサーとジャイロスコープセンサーの両方が搭載されている機種でなければ動作しません。

au IS12T は両方に対応しています。

※詳しい使い方については、アプリをダウンロードし、メインページの「？」アイコンをタップして表示されるヘルプを参照してください。

## 8. MainPage

### 8.1 Layout of "Menu" page

Before walking, tap "Liver" button on "Main" page. (●fig 11)

●fig 11 "今日も歩こう!"メインページ



The XAML markup code is like this. (●list 11)

●list 11 A part of MainPage.xaml

```
<phone:PhoneApplicationPage
  x:Class="Walking.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
  FontFamily="{StaticResource PhoneFontFamilyNormal}"
  FontSize="{StaticResource PhoneFontSizeNormal}"
```

```

Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
shell:SystemTray.IsVisible="True">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="#03C800">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <!--<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
        <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION"
Style="{StaticResource PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel-->

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">
        <Image Height="480" HorizontalAlignment="Left" Margin="0,75,0,0"
Name="MainBack" Stretch="Fill" VerticalAlignment="Top" Width="480"
Source="/Walking;component/Image/Back480.png" />
        <Image Height="103" HorizontalAlignment="Left" Margin="148,335,0,0"
Name="Logolcon" Stretch="Fill" VerticalAlignment="Top" Width="120"
Source="/Walking;component/Image/LogoLiver.png" />
        <Image Height="100" HorizontalAlignment="Left" Margin="60,580,0,0"
Name="HelpIcon" Stretch="Fill" VerticalAlignment="Top" Width="100"
Source="/Walking;component/Image/Help.png" />
        <Button Opacity="0.3" Content=" 削 除 " Style="{StaticResource
DeleteFileIconStyle}" Height="100" HorizontalAlignment="Left" Margin="190,580,0,0"
Name="DeleteFileIcon" VerticalAlignment="Top" Width="100" IsEnabled="False" />
        <Button Opacity="0.3" Content=" 表 示 " Style="{StaticResource
ShowDataListIconStyle}" Height="100" HorizontalAlignment="Left"
Margin="320,580,0,0" Name="ShowDataListIcon" VerticalAlignment="Top" Width="100"
IsEnabled="False" />
        <Image Height="40" HorizontalAlignment="Left" Margin="370,30,0,0"
Name="SSDLogo" Stretch="Fill" VerticalAlignment="Top" Width="80"
Source="/Walking;component/Image/New_SeinDesignLOGO_W80.png" />

```

```

        </Grid>
    </Grid>

    <!--Sample code showing usage of ApplicationBar-->

```

## 8.2 Programming of “Menu” page

Visual Basic code is like this (●list 12)

### ●list 12 MainPage.xaml.vb

```
Option Strict On
```

```
Imports System.IO.IsolatedStorage
```

```
Imports System.IO
```

```
Imports System.Xml.Linq
```

```
Partial Public Class MainPage
```

```
    Inherits PhoneApplicationPage
```

```
    ' Constructor
```

```
    Public Sub New()
```

```
        InitializeComponent()
```

```
    End Sub
```

```
    Dim MyStrage As IsolatedStorageFile
```

```
    Private Sub MainPage_Loaded(sender As Object, e As
System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

```
        ' 「MyWalkingData」フォルダがなければ生成しておく
```

```
        MyStrage = IsolatedStorageFile.GetUserStoreForApplication
```

```
        If MyStrage.DirectoryExists("MyWalkingData") = False Then
```

```
            MyStrage.CreateDirectory("MyWalkingData")
```

```
        End If
```

```
        'ファイルがあれば削除ボタンと表示ボタンを使用可とする
```

```
        Dim MyFilePath As String = Path.Combine("MyWalkingData", "MyWalkingData"
& ".xml")
```

```
        If MyStrage.FileExists(MyFilePath) = True Then
```

```
            DeleteFileIcon.Opacity = 1.0
```

```
            DeleteFileIcon.IsEnabled = True
```

```

        ShowDataListIcon.Opacity = 1.0
        ShowDataListIcon.IsEnabled = True
    Else
        DeleteFileIcon.Opacity = 0.3
        DeleteFileIcon.IsEnabled = False

        ShowDataListIcon.Opacity = 0.3
        ShowDataListIcon.IsEnabled = False
    End If
End Sub

Private Sub DeleteFileIcon_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles DeleteFileIcon.Tap
    Dim CheckRemove As MessageBoxResult = MessageBox.Show("ほんとうに削除
してもよいですか？この操作を実行すると、データはすべて失われます。","ファイルを
削除しますか？", MessageBoxButton.OKCancel)

    Select Case CheckRemove
        Case MessageBoxResult.OK

            Dim DelMyXml As XElement
            'MyStrage = IsolatedStorageFile.GetUserStoreForApplication
            Dim MyFilePath As String = Path.Combine("MyWalkingData",
"MyWalkingData" & ".xml")
            If MyStrage.DirectoryExists("MyWalkingData") = True And
MyStrage.FileExists(MyFilePath) = True Then
                Dim MyStream As IsolatedStorageFileStream =
MyStrage.OpenFile(MyFilePath, FileMode.Open, FileAccess.Read)
                Using MyReader As StreamReader = New
StreamReader(MyStream, System.Text.Encoding.UTF8)
                    Dim ReadDeleteXmlDoc As String = MyReader.ReadToEnd
                    DelMyXml = XElement.Parse(ReadDeleteXmlDoc)
                    MyStream.Close()
                End Using

                MyStrage.DeleteFile(MyFilePath)
                MessageBox.Show("ファイルを削除しました。")
                DeleteFileIcon.Opacity = 0.3
                DeleteFileIcon.IsEnabled = False
            End If
        Case Else
            'Do Nothing
    End Select
End Sub

```

```

        ShowDataListIcon.Opacity = 0.3
        ShowDataListIcon.IsEnabled = False
    Else
        MessageBox.Show("ファイルはありません。")
    Exit Sub
End If

Exit Select
Case Else
    Exit Sub
End Select
End Sub

Private Sub Logolcon_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles Logolcon.Tap
    NavigationService.Navigate(New Uri("/RecordData.xaml?Name=",
UriKind.Relative))
End Sub

Private Sub HelpIcon_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles HelpIcon.Tap
    NavigationService.Navigate(New Uri("/WalkingHelp.xaml?Name=",
UriKind.Relative))
End Sub

Private Sub ShowDataListIcon_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles ShowDataListIcon.Tap
    NavigationService.Navigate(New Uri("/ShowDataList.xaml?Name=",
UriKind.Relative))
End Sub
End Class

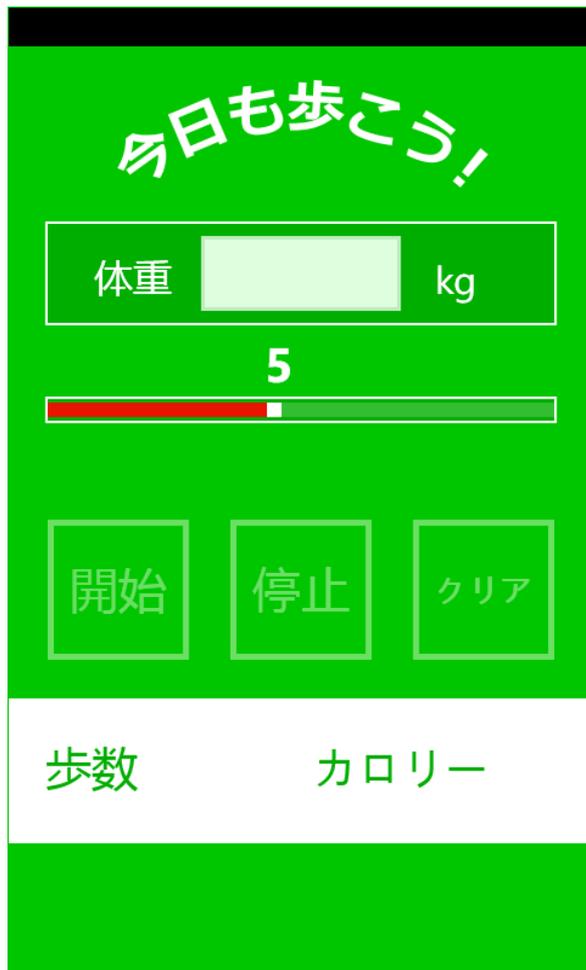
```

## 9. Recording data

### 9.1 Layout of "RecordData" page

When you shake your device, value is changed and added. (●fig 12)

●fig 12 Recoding data page of "今日も歩こう!"



The XAML markup code is like this. (●list 13)

●list 13 A part of RecordData.xaml

```
<phone:PhoneApplicationPage
    x:Class="Walking.RecordData"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
```

```

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
shell:SystemTray.IsVisible="True">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="#03c800">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <!--<StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28">
        <TextBlock x:Name="ApplicationTitle" Text="MY APPLICATION"
Style="{StaticResource PhoneTextNormalStyle}"/>
        <TextBlock x:Name="PageTitle" Text="page name" Margin="9,-7,0,0"
Style="{StaticResource PhoneTextTitle1Style}"/>
    </StackPanel-->

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">
        <Image Height="140" HorizontalAlignment="Left" Margin="0,0,0,0"
Name="LogoType" Stretch="Fill" VerticalAlignment="Top" Width="480"
Source="/Walking;component/Image/WhiteLogo.png" />

        <!-- ■ 体重 -->
        <Rectangle x:Name="WeightBack" Fill="#FF00AD00" Height="86"
Margin="30,145,30,0" Stroke="White" StrokeThickness="2" VerticalAlignment="Top"/>

        <TextBlock Height="59" HorizontalAlignment="Left" Margin="70,168,0,0"
x:Name="WeightLabel1" Text=" 体 重 " VerticalAlignment="Top" Width="91"
FontSize="32" Foreground="White" />
        <TextBox Height="86" HorizontalAlignment="Left" Margin="146,145,0,0"
x:Name="WeightTextBox1" Text="" VerticalAlignment="Top" Width="188" FontSize="40"
Background="#FFDAFFDA" Foreground="#FF007902" FontWeight="Bold"
TextAlignment="Center" />
        <TextBlock Height="48" HorizontalAlignment="Left" Margin="350,170,0,0"

```

```
x:Name="WeightLabel2" Text="kg" VerticalAlignment="Top" Width="60" FontSize="32"
Foreground="White" />
```

```
<!-- ■ 負荷設定スライダー -->
```

```
<Rectangle Height="22" HorizontalAlignment="Left" Margin="30,290,0,0"
x:Name="SliderBack" Stroke="White" StrokeThickness="2" VerticalAlignment="Top"
Width="420" Fill="#FF00AD00" />
```

```
<TextBlock Height="46" HorizontalAlignment="Left" Margin="172,235,0,0"
x:Name="MyLoadTextBlock" Text="5" VerticalAlignment="Top" Width="100"
FontSize="40" TextAlignment="Center" FontWeight="Bold" Foreground="White"/>
```

```
<Slider Height="90" HorizontalAlignment="Left" Margin="20,270,0,0"
x:Name="Slider1" VerticalAlignment="Top" Width="440" Value="5" Minimum="1" />
```

```
<!-- ■ 歩数計ボタン -->
```

```
<Button Content=" 開始 " Height="140" HorizontalAlignment="Left"
Margin="20,380,0,0" x:Name="StartButton" VerticalAlignment="Top" Width="140"
IsEnabled="True" Foreground="White" Background="#FF00AD00" BorderBrush="White"
BorderThickness="5" FontSize="40" />
```

```
<Button Content=" 停止 " Height="140" HorizontalAlignment="Left"
Margin="170,380,0,0" x:Name="StopButton" VerticalAlignment="Top" Width="140"
IsEnabled="False" Foreground="White" Background="#FF00AD00" BorderBrush="White"
BorderThickness="5" FontSize="40" />
```

```
<Button Content=" クリア " Height="140" HorizontalAlignment="Left"
Margin="320,380,0,0" x:Name="ClearButton" VerticalAlignment="Top" Width="140"
IsEnabled="False" Foreground="White" Background="#FF00AD00" BorderBrush="White"
BorderThickness="5" FontSize="26" FontWeight="Bold" />
```

```
<!-- ■ 結果 -->
```

```
<Rectangle Height="120" HorizontalAlignment="Left" Margin="0,540,0,0"
x:Name="ResultBack" Stroke="{x:Null}" StrokeThickness="1" VerticalAlignment="Top"
Width="480" Fill="White" />
```

```
<TextBlock Height="70" Width="210" Margin="30,570,250,0"
x:Name="MyStepTextBlock" Text=" 歩数 " VerticalAlignment="Top" FontSize="38"
Foreground="#FF00AD00" />
```

```
<TextBlock Height="70" Width="210" Margin="250,572,30,0"
x:Name="CalTextBlock" Text=" カロリー " VerticalAlignment="Top" FontSize="36"
Foreground="#FF00AD00" />
```

```
<!-- ■ タイムスタンプ -->
```

```
<TextBlock Height="50" HorizontalAlignment="Left" Margin="30,710,0,0"
x:Name="TimeStampTextBlock" Text="" VerticalAlignment="Top" Width="420"
```

```
FontSize="30" Visibility="Visible" Foreground="White" />
```

```
<!-- ■ 誤動作防止シール -->  
<Image x:Name="SealImage" VerticalAlignment="Top" Height="420"  
Margin="0,120,0,0" Width="420" Source="/Walking;component/Image/Seal.png"  
Visibility="Collapsed" />
```

```
<!-- ■ 以下不使用（開発時確認用） -->  
<!-- 状態 -->  
<TextBlock Height="44" HorizontalAlignment="Left" Margin="30,670,0,0"  
x:Name="StatusTextBlock" Text=" 状態 " VerticalAlignment="Top" Width="420"  
FontSize="26" Visibility="Collapsed" />  
<!-- DeviceAcceleration -->  
<TextBlock Height="50" HorizontalAlignment="Left" Margin="404,518,0,0"  
x:Name="XGTextBlock" Text="X(G)" VerticalAlignment="Top" Width="52" FontSize="16"  
Visibility="Collapsed" />  
<TextBlock Height="50" HorizontalAlignment="Left" Margin="404,583,0,0"  
x:Name="YGTextBlock" Text="Y(G)" VerticalAlignment="Top" Width="46" FontSize="16"  
Visibility="Collapsed" />  
<!-- 比較対象値、平均値、取得値 -->  
<TextBlock Height="44" HorizontalAlignment="Left" Margin="210,718,0,0"  
x:Name="CheckTextBlock" Text=" 開発確認用データ " VerticalAlignment="Top"  
Width="247" FontSize="22" Visibility="Collapsed" />
```

```
</Grid>  
</Grid>
```

```
<!-- Sample code showing usage of ApplicationBar -->
```

81

## 9.2 Programming of "RecordData" page

Visual Basic code is like this. (●list 14)

●list 14 RecordData.xaml.vb

```
Option Strict On  
Imports Microsoft.Devices.Sensors  
Imports Microsoft.Xna.Framework  
Imports System.IO.IsolatedStorage
```

```
Imports System.IO
Imports System.Text
Imports System.Windows.Media.Imaging
Imports Microsoft.Expression.Interactivity.Layout
Imports System.Xml.Linq
```

```
Partial Public Class RecordData
```

```
    Inherits PhoneApplicationPage
```

```
    Public Sub New()
```

```
        InitializeComponent()
```

```
    End Sub
```

```
    '■データ保存用変数
```

```
    Dim MyStorage As IsolatedStorageFile
```

```
    Dim MyFilePath As String
```

```
    '■この計測ページがロードされた時の処理
```

```
    Private Sub MainPage_Loaded(sender As Object, e As  
System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

```
        '端末がモーションセンサーをサポートしていない場合の対応
```

```
        If Motion.IsSupported = False Then
```

```
            MessageBox.Show("この端末はモーションセンサーをサポートしていません。")
```

```
            '「開始」ボタンを使用不可とする
```

```
            StartButton.IsEnabled = False
```

```
            Exit Sub
```

```
        End If
```

```
        'モーションはサポートしていてもジャイロスコープがサポートされていない機種（Lumia800）への対応
```

```
        If Gyroscope.IsSupported = False Then
```

```
            MessageBox.Show("この端末はジャイロスコープをサポートしていません。")
```

```
            '「開始」ボタンを使用不可とする
```

```
            StartButton.IsEnabled = False
```

```
            Exit Sub
```

```
        End If
```

'「MyWalkingData」フォルダがなければ生成しておく（MainPage で作成済みだが書いている）

```
MyStorage = IsolatedStorageFile.GetUserStoreForApplication
```

```
If MyStorage.DirectoryExists("MyWalkingData") = False Then
```

```
    MyStorage.CreateDirectory("MyWalkingData")
```

```
End If
```

'MyWalkingData.xml 中の体重を取得して表示する

```
MyFilePath = Path.Combine("MyWalkingData", "MyWalkingData" & ".xml")
```

'ファイルが存在する場合の処理。読み込んで体重をあらかじめ表示し、ユーザーの入力の手間を省く

```
If MyStorage.FileExists(MyFilePath) = True Then
```

```
    Dim MyStream As IsolatedStorageFileStream =  
MyStorage.OpenFile(MyFilePath, FileMode.Open, FileAccess.Read)
```

```
    Dim Reader As StreamReader = New StreamReader(MyStream,  
System.Text.Encoding.UTF8)
```

```
    Dim ReadXmlDoc As String = Reader.ReadToEnd
```

```
    Dim ItemDoc As XDocument = XDocument.Parse(ReadXmlDoc)
```

```
    Reader.Close()
```

```
    MyStream.Close()
```

```
    Dim MyDataQuery As IEnumerable(Of XElement) = From c In  
ItemDoc.Descendants("記録") Select c
```

```
    For Each MyResult In MyDataQuery
```

```
        WeightTextBox1.Text = MyResult.Attribute("体重").Value
```

```
    Next
```

```
Else
```

```
    WeightTextBox1.Text = String.Empty
```

```
End If
```

```
End Sub
```

```
Dim MyMotion As Motion 'モーションセンサーオブジェクト
```

```
Dim MyDeviceAccerlation As Vector3 = Vector3.Zero '加速度
```

```
Dim MyDeviceRotationRate As Vector3 = Vector3.Zero '角速度
```

```
Dim MyCount As Integer = 0 '歩数
```

```
Dim MyDataA As Double '取得データ。加速度。
```

```
Dim MyDataR As Double '取得データ。角速度（ラジアン/秒）。
```

Dim MySpanMSecond As Integer = 240 'データ取得時間間隔。一歩 50cm として一歩を歩く場合 360msec が標準。それ以下にする

Dim MyCompareData As Double = 0.4 '判断用データ。初期値はスライダーがデフォルトの 5 の場合の値。

Dim MyStandardData As Double = 0.4 '暫定平均値。初期値はスライダーがデフォルトの 5 の場合の値。

Dim MyLoad As Integer = 5 'スライダーのデフォルト。整数値

Dim MyLoadData As Double = 0.4 '負荷データ。初期値はスライダーがデフォルトの 5 の場合の値。

Dim MyLimitData As Double = 0.087 'カウントの可否を決めるデータ。初期値はスライダーがデフォルトの 5 の場合の値

Dim MyWeight As Double '体重

Dim MyCalPerWeight As Double '体重 1kg あたりの消費カロリー

Dim MyCal As Double '総カロリー

Dim MyStartDateTime As String '歩行開始日時

'■ スライダーの負荷が変わった時の処理

```
Private Sub Slider1_ValueChanged(sender As Object, e As System.Windows.RoutedEventArgs) Handles Slider1.ValueChanged
```

```
    MyLoad = CInt(Slider1.Value) 'スライダーの値を取得
```

```
    MyLoadTextBlock.Text = CStr(MyLoad) '取得した値を表示
```

```
    MyLoadTextBlock.Margin = New Thickness(Slider1.Value * 40 - 30, 235, 0, 0)
```

```
    MyLoadData = (0.8 + MyLoad) * 0.07 '1~10 まで 10 段階で 0.126~0.756 の値が入る。加速度センサーで比較する
```

```
    'スライダーの 1 目盛りにつき 1 度の角速度により、水平垂直の細かい動きでのカウントを制限する
```

```
    MyLimitData = MyLoad * Math.PI / 180
```

```
    '計測間隔=(身長-100)/100/((3.3+(MyLoad*0.3))/60/60)だが、逆に負荷をかけて 192~300 の範囲とする
```

```
    MySpanMSecond = 180 + (MyLoad) * 12
```

```
End Sub
```

'■ 「開始」 ボタンがタップされた時の処理

```
Private Sub StartButton_Tap(sender As Object, e As System.Windows.Input.GestureEventArgs) Handles StartButton.Tap
```

```
    '● 体重の入力値の検証
```

```
    If WeightTextBox1.Text = String.Empty Then
```

```
        MessageBox.Show("体重を入力してください。")
```

```
    Exit Sub
```

```

Else
    Dim WeightNum As Integer =
Encoding.GetEncoding("utf-8").GetByteCount(WeightTextBox1.Text) '全角半角の検証
    Dim WeightLen As Integer = WeightTextBox1.Text.Length
    If WeightNum = WeightLen AndAlso
System.Text.RegularExpressions.Regex.IsMatch(WeightTextBox1.Text,
"^[1-9]¥d*|0)(¥.¥d+)?([1-9]¥d*|0)?$",
System.Text.RegularExpressions.RegexOptions.IgnoreCase) Then
        MyWeight = CDbI(WeightTextBox1.Text)
    Else
        MessageBox.Show("体重は半角数字で入力してください。")
        Exit Sub
    End If
End If

```

'●歩数計の処理を開始するため、各ボタンを非表示にする

```

StartButton.IsEnabled = False
ClearButton.IsEnabled = False
Slider1.IsEnabled = False
WeightTextBox1.IsEnabled = False

```

'●複合モーションの加速度センサー、ジャイロセンサーでのセンシング開始

```

If MyMotion IsNot Nothing AndAlso MyMotion.IsDataValid Then 'センサーオブ
ジェクトがある場合の処理

```

```

    MyMotion.Stop()
    MessageBox.Show("モーションセンサーを停止しました。")

```

```

Else

```

```

    If MyMotion Is Nothing Then 'モーションオブジェクトが空の場合の処理
        MyMotion = New Motion 'モーションの新しいインスタンスを生成
        MyMotion.TimeBetweenUpdates =
TimeSpan.FromMilliseconds(MySpanMSecond) 'センサーデータの更新間隔を指定

```

```

        AddHandler MyMotion.CurrentValueChanged, Sub(senderValue As
Object, eValue As SensorReadingEventArgs(Of MotionReading))

```

```

Deployment.Current.Dispatcher.BeginInvoke(Sub()
CurrentValueChanged(eValue.SensorReading))

```

```

End Sub

```

```

End If

```

```

Try
    MyCalPerWeight = 300 / 60 * MyWeight / 10000
    '体重 60kg1万歩で300KCalとした場合の入力体重に対する一步あたりの消費カロリー
    MyStartDateTime = DateTime.Now.ToString("yyyy/MM/dd/HH:mm:ss")
    '歩行開始時間
    'Starting Motion API...
    SeallImage.Visibility = Windows.Visibility.Visible '誤動作防止シールを表示
    MyMotion.Start() 'モーションセンサーでのデータ取得を開始
Catch
    MessageBox.Show("モーションセンサーを開始できません。")
Exit Sub
End Try
End If
End Sub

```

'■ センサーの値が変化した時の処理

```
Private Sub CurrentValueChanged(MyReading As MotionReading)
```

```

    TimeStampTextBlock.Text = CStr(MyReading.Timestamp.ToString) '時刻表示
    'Accelerometer クラスで取得される Raw データから重力加速度を除いた加速度
    'MotionReading 構造体 Device Acceleration プロパティで取得される線形加速度 (G)
    MyDeviceAcceralation = MyReading.DeviceAcceleration
    'デバイスの回転速度
    'MotionReading 構造体 DeviceRotationRate プロパティで取得される角速度 (rad/sec)
    MyDeviceRotationRate = MyReading.DeviceRotationRate
    '■ 少しの動きでもカウントする代わりに、誤動作を制限する
    '端末の向きにより、利用する計測値を変更。X、Y の値が同じ場合は縦向き優先。
    'ジャイロを利用して制限値以下をカウントしない (多少の傾きではカウントしない)
    If Math.Abs(MyReading.Gravity.X) > Math.Abs(MyReading.Gravity.Y) Then
        MyDataA = Math.Abs(MyDeviceAcceralation.X) '実機が横の場合
        MyDataR = Math.Abs(MyDeviceRotationRate.Y) * MySpanMSecond / 1000
    * 2 '計測間隔での角速度、横の場合は振りにくいので2倍にする

```

```
Elseif Math.Abs(MyReading.Gravity.Y) >= Math.Abs(MyReading.Gravity.X) Then
    MyDataA = Math.Abs(MyDeviceAcceralation.Y) '実機が縦の場合
    MyDataR = Math.Abs(MyDeviceRotationRate.X) * MySpanMSecond / 1000 '
計測間隔での角速度
End If
```

'●ジャイロのデータが小さい場合は評価しない  
 '計測値が比較値以上の場合にカウント  
 '比較値は、スライダーで設定した負荷データの2倍と、この比較値（初回のみ  
 0.4）、最新の計測値の平均値とする  
 '屋外散歩用に負荷を設定して、絨毯の室内での歩行時から計測を開始しても、  
 徐々にカウントが合うようにする

```
If MyDataR > MyLimitData AndAlso MyDataA >= MyCompareData Then
    MyCount = MyCount + 1
    MyCompareData = (MyLoadData * 2 + MyStandardData + MyDataA) / 4
    MyStandardData = MyCompareData

    MyCal = CDbI((MyCalPerWeight * MyCount).ToString("f2")) '歩数を乗算し
    て消費カロリーを算出
    CalTextBlock.Text = CStr(MyCal) & "KCal" '消費カロリーを表示
End If
```

```
MyStepTextBlock.Text = CStr(MyCount) & "歩" '歩数を表示
End Sub
```

'■誤動作防止シールがダブルタップされた時の処理

```
Private Sub SeallImage_DoubleTap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles SeallImage.DoubleTap
    SeallImage.Visibility = Windows.Visibility.Collapsed '誤動作防止の解除
    StopButton.IsEnabled = True '「停止」ボタンを使用可にする
End Sub
```

'■「停止」ボタンがタップされた時の処理

```
Private Sub StopButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles StopButton.Tap
    Try
        MyMotion.Stop() 'モーションのセンシングを停止
        MyMotion = Nothing
        StopButton.IsEnabled = False '「停止」ボタンを使用不可にする

        DataSave() 'データの保存処理を実行
```

```

Catch
    MessageBox.Show("センサーの停止に失敗しました。")
Exit Sub
End Try

```

```
End Sub
```

'■ 「クリア」 ボタンがタップされた時の処理

```

Private Sub ClearButton_Tap1(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles ClearButton.Tap
    MyCount = 0 '歩数を初期化
    MyCal = 0.0 '消費カロリーを初期化
    CalTextBlock.Text = "カロリー"
    MyStepTextBlock.Text = "歩数"
    StartButton.IsEnabled = True
    StopButton.IsEnabled = False
    ClearButton.IsEnabled = False
    Slider1.Value = 5
    MyDataA = 0 '計測データを初期化
End Sub

```

'■ データ保存処理（「停止」 ボタンがタップされた時）

```

Sub DataSave()
    Dim MyRecordDateTime As String = String.Empty
    Try
        MyStorage = IsolatedStorageFile.GetUserStoreForApplication
        If MyStorage.DirectoryExists("MyWalkingData") = False Then
            MyStorage.CreateDirectory("MyWalkingData")
        End If

        MyFilePath = Path.Combine("MyWalkingData", "MyWalkingData" & ".xml")
        MyRecordDateTime = DateTime.Now.ToString("yyyy/MM/dd/HH:mm:ss")

        If MyStorage.FileExists(MyFilePath) = False Then
            Dim MyXmlDoc As XDocument = <?xml version="1.0"
encoding="utf-8"?>
                <歩数記録>
                    <記録 負荷 =<%=
CStr(MyLoad) %> 開始=<%= MyStartDateTime %> 終了=<%= MyRecordDateTime %>
体重=<%= CStr(MyWeight) %> 歩数 =<%= CStr(MyCount) %> カロリー =<%=
CStr(MyCal) %>></記録>

```

```

        </歩数記録>
        Using MyStream As IsolatedStorageFileStream = New
IsolatedStorageFileStream(MyFilePath, FileMode.Create, FileAccess.Write, MyStorage)
        MyXmlDoc.Save(MyStream)
        MessageBox.Show("歩数を記録しました。")
    End Using
Else

    Dim AddStream As IsolatedStorageFileStream =
MyStorage.OpenFile(MyFilePath, FileMode.Open, FileAccess.Read)
    Dim Reader As StreamReader = New StreamReader(AddStream)
    Dim ReadXmlDoc As String = Reader.ReadToEnd
    Dim AddDoc As XElement = XElement.Parse(ReadXmlDoc)
    Dim AddXml As XElement = <記録 負荷=<%= CStr(MyLoad) %> 開始
=<%= MyStartDateTime %> 終了=<%= MyRecordDateTime %> 体重=<%=
CStr(MyWeight) %> 歩数=<%= CStr(MyCount) %> カロリー=<%= CStr(MyCal) %>></
記録>

    AddDoc.Add(AddXml)
    AddStream.Dispose()

    Using stream As IsolatedStorageFileStream = New
IsolatedStorageFileStream(MyFilePath, FileMode.Create, FileAccess.Write, MyStorage)
        If MyStorage.FileExists(MyFilePath) = True Then
            AddDoc.Save(stream)
            Reader.Dispose()
            MessageBox.Show("歩数を記録しました。")
        End If
    End Using
End If

StartButton.IsEnabled = True
ClearButton.IsEnabled = True
Slider1.IsEnabled = True
WeightTextBox1.IsEnabled = True
Catch
Exit Sub
End Try
End Sub
End Class

```

## 10. The list of data

### 10.1 Layout of "ShowDataList" page

You can check your recorded data on this page. (●fig 13)

●fig 13 The list of data



The XAML markup code is like this. (●list 15)

●list 15 A part of ShowDataList.xaml

```
<phone:PhoneApplicationPage
    x:Class="Walking.ShowDataList"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
```

```

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
mc:Ignorable="d" d:DesignHeight="768" d:DesignWidth="480"
shell:SystemTray.IsVisible="True">

<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="white">
    <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*/>
    </Grid.RowDefinitions>

    <!--TitlePanel contains the name of the application and page title-->
    <StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="0,0,0,0">
        <Image Height="140" HorizontalAlignment="Left" Margin="0,0,0,0"
Name="HeloBackground" Stretch="Fill" VerticalAlignment="Top" Width="480"
Source="/Walking;component/Image/WhiteLogo.png" />
    </StackPanel>

    <!--ContentPanel - place additional content here-->
    <Grid x:Name="ContentPanel" Grid.Row="1" Margin="0,0,0,0">
        <Button Height="60" HorizontalAlignment="Left" Margin="150,15,0,0"
Name="SortAscendingButton" VerticalAlignment="Top" Width="60"
Style="{StaticResource SortAscendingButtonStyle}" />
        <Button Height="60" HorizontalAlignment="Left" Margin="270,15,0,0"
Name="SortDescendingButton" VerticalAlignment="Top" Width="60"
Style="{StaticResource SortDescendingButtonStyle}" />

        <ListBox Name="MyWalkingListBox" Margin="0,90,0,15" FontSize="24">
            <ListBox.ItemTemplate>
                <DataTemplate>
                    <StackPanel Orientation="vertical" Name="record">
                        <TextBlock Name="MyStartDateBlock" Text="{Binding
計測日}" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="20,0,0,0"
Foreground="#FF00AD00" FontSize="26" Width="140" Height="35" />
                        <TextBlock Name="MyCountBlock" Text="{Binding 歩
数}" VerticalAlignment="Top" HorizontalAlignment="Left" Margin="170,-35,0,0"

```

```

Foreground="#FF000000"      FontSize="28"      Width="150"      Height="35"
TextAlignment="Right" />
        <TextBlock Name="MyWeightBlock" Text="{Binding 体
重 }"      VerticalAlignment="Top"      HorizontalAlignment="Left"      Margin="350,-35,0,0"
Foreground="#FF00AD00" FontSize="26" Width="120" Height="35" />

        <Rectangle      Height="1"      HorizontalAlignment="Left"
Margin="0,5,0,15"      Name="ListBorder"      VerticalAlignment="Top"      Width="480"
Fill="#FF03c800" />
    </StackPanel>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>
</Grid>
</Grid>

<!--Sample code showing usage of ApplicationBar-->

```

## 10.2 Programming of "ShowDataList" page

92

Visual Basic code is like this. (●list 16)

●list 16 ShowDataList.xaml.vb

```

Option Strict On
Imports System.IO.IsolatedStorage
Imports System.IO
Imports System.Xml.Linq

Partial Public Class ShowDataList
    Inherits PhoneApplicationPage

    Public Sub New()
        InitializeComponent()
    End Sub

    Public Class MyData
        Property 計測日 As String
        Property 歩数 As String
    End Class

```

```
Property 体重 As String
End Class
```

```
'■ソート処理対象のXML木を格納する変数を宣言
Dim MyWalkingDoc As XDocument
```

```
Private Sub ShowDataList_Loaded(sender As Object, e As
System.Windows.RoutedEventArgs) Handles MyBase.Loaded
```

```
'デフォルトは昇順表示なので、昇順ボタンを使用不可としておく
```

```
SortAscendingButton.Opacity = 0.3
```

```
SortAscendingButton.IsEnabled = False
```

```
'パラメータから渡された年月に拡張子.xmlを付けて処理対象ファイルとする
```

```
Dim MyStorage As IsolatedStorageFile =
IsolatedStorageFile.GetUserStoreForApplication
```

```
Dim MyFilePath As String = Path.Combine("MyWalkingData", "MyWalkingData"
& ".xml")
```

```
'ファイルが存在する場合の処理。ファイルのXML木を読み込み、昇順ソート
表示を実行する
```

```
If MyStorage.DirectoryExists("MyWalkingData") = True And
MyStorage.FileExists(MyFilePath) = True Then
```

```
Dim MyStream As IsolatedStorageFileStream =
MyStorage.OpenFile(MyFilePath, FileMode.Open, FileAccess.Read)
```

```
Dim MyReader As StreamReader = New StreamReader(MyStream,
System.Text.Encoding.UTF8)
```

```
Dim MyReadXmlDoc As String = MyReader.ReadToEnd
```

```
MyWalkingDoc = XDocument.Parse(MyReadXmlDoc)
```

```
MyReader.Close()
```

```
MyStream.Close()
```

```
SortAscending()
```

```
Else
```

```
Exit Sub
```

```
End If
```

```
End Sub
```

```
'■デフォルトの昇順ソート表示の処理
```

```
Private Sub SortAscending()
```

'このファイルの冒頭に記述している MyData クラスを使い、新規リストを生成  
Dim MyWalkingList As New List(Of MyData)

'登録順、すなわち文書順に「記録」要素を取得

```
Dim MyDataQuery As IEnumerable(Of XElement) = From c In  
MyWalkingDoc.Descendants("記録") Order By c.@開始 Ascending
```

'「記録」要素の「開始」「体重」「歩数」属性値を取得

```
For Each MyResult In MyDataQuery
```

```
Dim MyStartDateTime As String = MyResult.@開始
```

```
Dim MyStartDate As String = MyStartDateTime.Substring(0, 10)
```

```
Dim MyWeight As String = MyResult.@体重
```

```
Dim MyCount As String = MyResult.@歩数
```

'生成したリストに、各々取得したデータを追加

```
MyWalkingList.Add(New MyData With {.計測日 = MyStartDate, .体重 =  
MyWeight & "kg", .歩数 = MyCount})
```

```
Next
```

'ListBox にデータを適用

```
MyWalkingListBox.ItemsSource = MyWalkingList
```

```
End Sub
```

'■降順ソート表示の処理

```
Private Sub SortDescending()
```

'このファイルの冒頭に記述している MyTitleData クラスを使い、新規リストを  
生成

```
Dim MyWalkingList As New List(Of MyData)
```

'登録日時の逆、すなわち逆文書順に「記録」要素を取得

```
Dim MyDataQuery As IEnumerable(Of XElement) = From c In  
MyWalkingDoc.Descendants("記録") Order By c.@開始 Descending
```

'「記録」要素の「開始」「体重」「歩数」属性値を取得

```
For Each MyResult In MyDataQuery
```

```
Dim MyStartDateTime As String = MyResult.@開始
```

```
Dim MyStartDate As String = MyStartDateTime.Substring(0, 10)
```

```
Dim MyWeight As String = MyResult.@体重
```

```
Dim MyCount As String = MyResult.@歩数
```

```

        '生成したリストに、各々取得したデータを追加
        MyWalkingList.Add(New MyData With {.計測日 = MyStartDate, .体重 =
MyWeight & "kg", .歩数 = MyCount})
        Next
        'ListBox にデータを適用
        MyWalkingListBox.ItemsSource = MyWalkingList
    End Sub

```

```

'■ 「昇順ソート」 ボタンがタップされた時の処理
Private Sub SortAscendingButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles SortAscendingButton.Tap
    ' 「昇順ソート」 ボタンを使用不可とし、「降順ソート」 ボタンを使用可とする
    SortAscendingButton.Opacity = 0.3
    SortAscendingButton.IsEnabled = False

    SortDescendingButton.Opacity = 1.0
    SortDescendingButton.IsEnabled = True

    '昇順ソート表示の処理を実行
    SortAscending()
End Sub

```

```

'■ 「降順ソート」 ボタンがタップされた時の処理
Private Sub SortDescendingButton_Tap(sender As Object, e As
System.Windows.Input.GestureEventArgs) Handles SortDescendingButton.Tap
    ' 「降順ソート」 ボタンを使用不可とし、「昇順ソート」 ボタンを使用可とする
    SortDescendingButton.Opacity = 0.3
    SortDescendingButton.IsEnabled = False

    SortAscendingButton.Opacity = 1.0
    SortAscendingButton.IsEnabled = True

    '降順ソート表示の処理を実行
    SortDescending()
End Sub
End Class

```

## 11. More Information for the developers

### 11.1 Visual Studio project files

If you need the codes of this tool, you can download the Microsoft Visual Studio 2010 project files.

[Sensors Set Ver.0.8 Codes Download Page](#)

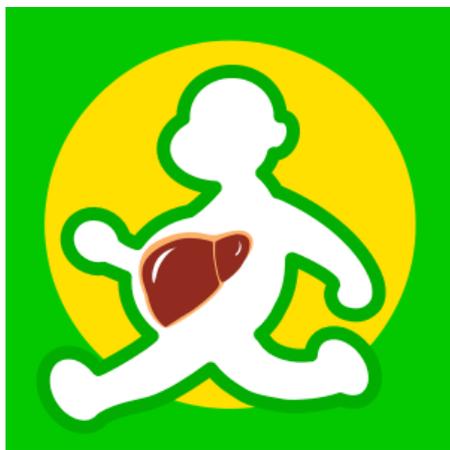
[今日も歩こう！ Ver.0.8 Codes](#) (zip file)

### 11.2 Sensor Applications

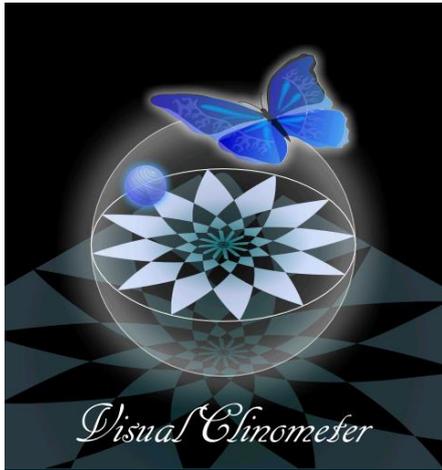


[Sensor Set \(English\)](#)

[Sensor Set センサー計測セット\(Japanese\)](#)



[今日も歩こう！（日）](#)



[Visual Clinometer \(English\)](#)  
[ビジュアル傾斜計\(Japanese\)](#)



[Sound Clinometer \(English\)](#)

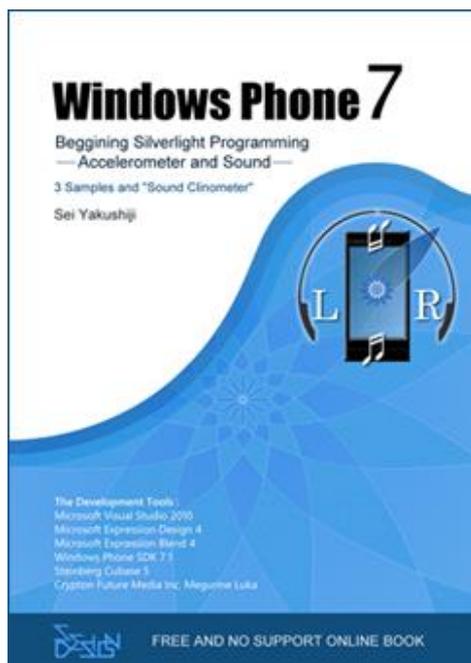
## 11.3 Online books and article

### Windows Phone 7, Beginning Silverlight Programming ～Accelerometer and Sound～

81 pages pdf book. (29 June 2012)

This book is written about an accelerometer with the sound.

[Download the Book](#)



98

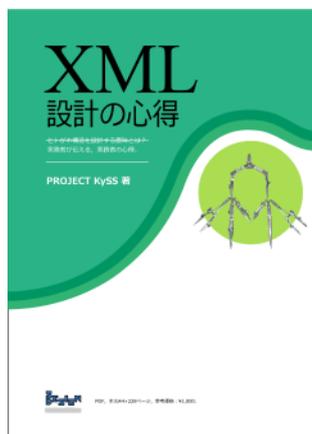
IMPRESS BUSINESS MEDIA CORPORATION "Software Developer's Think IT"

[「加速度センサーやジャイロスコープセンサーを使った、3つのセンサーサンプル」](#)

### The Essence of XML Design

228 pages pdf book. (24 March 2010)

[Download the Book](#)



## Copyright

Trademarked names may appear in this book, Rather than use a trademark symbol with every occurrence of a trademarked name.

I use these names only in an editorial fashion and to the benefit of the trademark owner, with no intension of infringement of the trademark.

本書に掲載されている登録商標または商標は、本書を制作する目的でのみ記載しており、その商標権を侵害する意図のあるものではありません。

Microsoft, Visual Studio, Visual Basic, Expression Studio, Silverlight は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。その他、本書に記載されている会社名、システム名、製品名は、一般に各社の商標または登録商標です。なお本文中には、TM、OR マークは明記しておりません。

Please note that the developer without any warranty for the result of this application.

本書掲載のアプリを実行にすることによって生じる結果につきましては、筆者はその一切の責任を負いかねます。あらかじめご了承ください。

**Sei Yakushiji**, author

A graphic designer, a song writer, a columnist and a pioneer of XML design in Japan.

Microsoft MVP for Windows Server System - XML (Oct 2003-Sep 2008),

Data Platform Development (Oct 2008-Sep 2010),

Development Platforms - Client App Dev (Oct 2010-Sep 2013)

Independent label "SeiSeinDesign" <http://www.SeinDesign.net/>

100

**Full Source Codes of Sensors Set - Sensor Programming for Windows Phone –**

Published in 2013/06/29



[SeiSeinDesign](http://www.SeinDesign.net/)

<http://www.SeinDesign.net/>

Copyright c 2013 SeiSeinDesign All rights reserved.