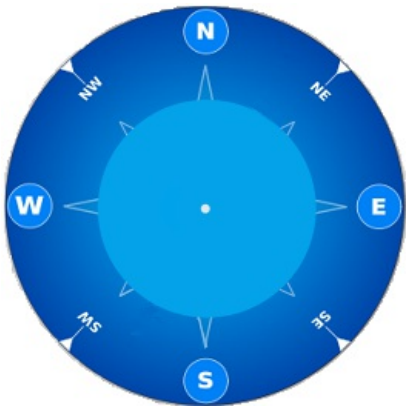
How to turn the N97 into a Magnetic Compass

Introduction

This code snippet shows how we can convert N97 to a compass. This application can be used to find magnetic north direction by the built in magnetic north sensor. If you rotate the device the compass is rotated to indicate the north.



Magnetic north sensor

N97 and some other devices have built in magnetometer. That magnetometer is used to find the direction. The direction change is feed to rotate the bitmap.

```
TInt i = 0;
CSensrvChannelFinder* finder = CSensrvChannelFinder::NewLC();
RSensrvChannelInfoList channelList;
CleanupClosePushL( channelList );
TSensrvChannelInfo Channelinfo;
finder->FindChannelsL( channelList, Channelinfo );

for (i = 0; i<channelList.Count(); i++)
 {
    Channelinfo = channelList[i];
    switch(Channelinfo.iChannelType)
        {
        case KSensrvChannelTypeIdMagneticNorthData:
         iBoolMagSupported = ETrue;
            break;
        default:
         iBoolMagSupported = EFalse;
            break;
        }
    if(iBoolMagSupported == (TInt)ETrue)
     {
     break;
     }
    }

   if(iBoolMagSupported == (TInt)EFalse)
    {
    User::Leave(KErrNotSupported);
    }

   if ( channelList.Count() > 0 )
       {
```


(C) Copyright Nokia 2014. All rights reserved.

```
        Channelinfo = channelList[i];
        iMagneticNorthSensor = CSensrvChannel::NewL( channelList[ i ] );
        iMagneticNorthSensor->OpenChannelL();
        iMagneticNorthSensor->StartDataListeningL( this, 1, 1, 0 );
        }
    channelList.Close();
    CleanupStack::Pop( &channelList );
    CleanupStack::PopAndDestroy( finder );
```

Bitmap rotation

```
Following code shows how to rotate bitmap.

CFbsBitmap *CImageConverterEngine::Bitmap_Rotate(TInt Angle)
{

 Angle = 360 -Angle;
    TReal CosA = 0.0;
    TReal SinA = 0.0;

    if(!iOriginalBitmap)
      {
      return NULL;
      }

    TBitmapUtil BaseBmpUtil(iOriginalBitmap);

    TBitmapUtil RotateBmpUtil(iRotatedBitmap);

    BaseBmpUtil.Begin(TPoint(0,0));

    RotateBmpUtil.Begin(TPoint(0,0), BaseBmpUtil);
      RotateBmpUtil.Begin(TPoint(0,0));

    TReal aAngle = (KPi/180.0)* (TReal)Angle;

    TInt ret = Math::Cos(CosA, aAngle);
      if(ret != KErrNone)
       {
       return NULL;
       }

     ret = Math::Sin(SinA, aAngle);
      if(ret != KErrNone)
       {
       return NULL;
       }

    TInt xC = iSize.iWidth/2;
    TInt yC = iSize.iHeight/2;

    for(TInt new_y=0;new_y<iSize.iHeight;new_y++)
    {
        for(TInt new_x=0;new_x<iSize.iWidth;new_x++)
        {
```


(C) Copyright Nokia 2014. All rights reserved.

```
            if( (new_x-xC)*(new_x-xC) + (new_y-yC)*(new_y-yC) <= (xC*yC))
   {
    TInt x = xC + (TInt)(((new_x - xC)*CosA) + ((new_y-yC)*SinA)); //new x pixel bitmap
    TInt y = yC + (TInt)(-((new_x-xC)*SinA) + ((new_y-yC)*CosA)); // new y pixel bitmap

    BaseBmpUtil.SetPos(TPoint(x,y));
    RotateBmpUtil.SetPos(TPoint(new_x,new_y));
    RotateBmpUtil.SetPixel(BaseBmpUtil);
   }
        }
    }
    BaseBmpUtil.End();
    RotateBmpUtil.End();

    return iRotatedBitmap;
}
```

Example Applications

The capability required for this app is ReadDeviceData. Example application was tested with N97. Can be found from following link: File:PhoneCompass.zip

(C) Copyright Nokia 2014. All rights reserved.