D. Snyder Feb. 27, 2011

**Equation of Time Calculated and Displayed on a Cell Phone**

Following is a Frink program to calculate and display the *Equation of Time* on an Android-based cell phone. The equation approximating the Equation of Time in minutes is (FE1 from Ref. 2):

$$EoT = \begin{cases} (-107.0605\sin B - 428.6697\cos B + 596.1009\sin 2B - 2.0898\cos 2B \\ \qquad +4.4173\sin 3B + 19.2776\cos 3B - 12.7338\sin 4B)/60 \end{cases}$$

where

$$B = 360dy/365.2422 - 80.535132,$$

in which *dy* is the "day number" for the date of interest (eg. *dy* = 1 on Jan. 1, *dy* = 2 on Jan. 2, dy = 32 on Feb. 1, etc.)

References

1. Frink documentation http://futureboy.us/frinkdocs/frinkframe.html
2. Herbert O. Ramp, Equation of Time – Comparison of Approximating Formulae , *Compendium of the North American Sundial Society*, Vol. 18, No. 1, pp. 20-22, March 2011.

**Frink program to calculate the *Equation of Time* and display it:**

```
// Calculate Equation of Time
// using approximation FE1 noted by H. O. Ramp
// in The Compendium of the North American Sundial Society, March 2011
// Created by: D. L. Snyder  27 February 2011
// Start .....
// define date format
        fmt = ### d-MMMM-yyyy 'CE at' h:mm:ss a z ###
// define constants
        c1 := 360.0/365.2422  degrees
        c2 := 80.535132      degrees
        e1 := -107.0605      seconds
        e2 := -428.6697      seconds
        e3 := +596.1009      seconds
        e4 := -2.0898        seconds
        e5 := +4.4173        seconds
        e6 := +19.2776       seconds
        e7 := -12.7338       seconds
// start calculations
        TodayNumber = (now[] - #2011-01-01#) / (1.0 days)
        B = c1*int[TodayNumber] - c2
```

```
        EOT = e1*sin[B] + e2*cos[B] + e3*sin[2*B] +
                e4*cos[2*B] + e5*sin[3*B] + e6*cos[3*B] +
                e7*sin[4*B]
//println["Today's Number is " + int[TodayNumber]]
        b[n] := c1*n - c2
        eot[b] := e1*sin[b] + e2*cos[b] + e3*sin[2*b] +
                e4*cos[2*b] + e5*sin[3*b] + e6*cos[3*b] +
                e7*sin[4*b]
//make display graphics
g = new graphics
        g.backgroundColor[0.7,0.9,0.7]
        g.color[0.7,0.9,0.7]
        g.drawRectSides[-10,20 minutes,375,-20 minutes]
        g.color[0,0,0]
        g.drawRectSides[1,20 minutes,365,-20 minutes]
        g.color[0,0,0]
//make vert coords (day 1 each month)
        g.line[1,0.0 seconds,365,1.0 seconds]
        g.line[32,-20.0 minutes,32,20.0 minutes]
        g.line[60,-20.0 minutes,60,20.0 minutes]
        g.line[91,-20.0 minutes,91,20.0 minutes]
        g.line[121,-20.0 minutes,121,20.0 minutes]
        g.line[152,-20.0 minutes,152,20.0 minutes]
        g.line[182,-20.0 minutes,182,20.0 minutes]
        g.line[213,-20.0 minutes,213,20.0 minutes]
        g.line[244,-20.0 minutes,244,20.0 minutes]
        g.line[274,-20.0 minutes,274,20.0 minutes]
        g.line[305,-20.0 minutes,305,20.0 minutes]
        g.line[335,-20.0 minutes,335,20.0 minutes]
//make horiz coords (5 minute intervals)
        g.line[1,-15 minutes,365,-15 minutes]
        g.line[1,-10 minutes,365,-10 minutes]
        g.line[1,-5 minutes,365,-5 minutes]
        g.line[1,5 minutes,365,5 minutes]
        g.line[1,10 minutes,365,10 minutes]
        g.line[1,15 minutes,365,15 minutes]
//make vert labels (months)
        x = 3
 //label offset
        g.text["J",1,0 minutes,"right","top"]
        g.text["F",32,0 minutes,"right","top"]
        g.text["M",60,0 minutes,"right","top"]
        g.text["A",91,0 minutes,"right","top"]
        g.text["M",121,0 minutes,"right","top"]
        g.text["J",152,0 minutes,"right","top"]
        g.text["J",182,0 minutes,"right","top"]
        g.text["A",213,0 minutes,"right","top"]
        g.text["S",244,0 minutes,"right","top"]
```

```
        g.text["O",274,0 minutes,"right","top"]
        g.text["N",305,0 minutes,"right","top"]
        g.text["D",335,0 minutes,"right","top"]
//make horiz labels (5 minute intervals)
        d = -3
  //horiz displacement
        g.text["20",d,-20 minutes,"right","center"]
        g.text["15",d,-15 minutes,"right","center"]
        g.text["10",d,-10 minutes,"right","center"]
        g.text["5",d,-5 minutes,"right","center"]
        g.text["0",d,0 minutes,"right","center"]
        g.text["-5",d,5 minutes,"right","center"]
        g.text["-10",d,10 minutes,"right","center"]
        g.text["-15",d,15 minutes,"right","center"]
        g.text["-20",d,20 minutes,"right","center"]
//draw EOT graph
        g.color[0,0,0]
        for n=1 to 364
                {g.line[n,-eot[b[n]],n+1,-eot[b[n+1]]]}
//add EOT at today's date
        g.color[1,0,0]
        g.line[int[TodayNumber],0 minutes,int[TodayNumber],-EOT]
        g.fillEllipseCenter[int[TodayNumber],-EOT,6,1.2 minutes]
//add results boxes
        g.color[0.7,0.9,0.7]
        g.fillRectSides[10,-11 minutes,265,-18 minutes]
        g.color[0,0,0]
        ThisDay = now[] -> fmt
        g.text["Equation of Time (minutes) versus Date", 12,-17.5 minutes,"left", "center"]
        g.text["  Today is " + ThisDay, 12, -14.7 minutes,"left", "center"]
        g.text["  Equation of Time is " + format[EOT, "minutes", 3],12, -12.0 minutes, "left",
"center"]
        g.color[0.7,0.9,0.7]
        g.fillRectSides[95,9 minutes,360,17.5 minutes]
        g.color[0,0,0]
        g.text["CivilTime = SundialTime-EOT+D+LC",96,10.5 minutes,"left","center"]
        g.text[" D=+1 hr during DaylightSavings, else D=0",99,13.0 minutes,"left","center"]
        g.text[" LC is LongitudeCorrection",99,15.5 minutes,"left","center"]
        g.show[]
//..... End
```

**Screen shots of an Android-based cell phone running this Frink program for two dates:**