Volume 20 Number 1 ISSN 1074-3197 (printed) ISSN 1074-8059 (digital) March 2013

The Compendium*

Journal of the
North American Sundial Society

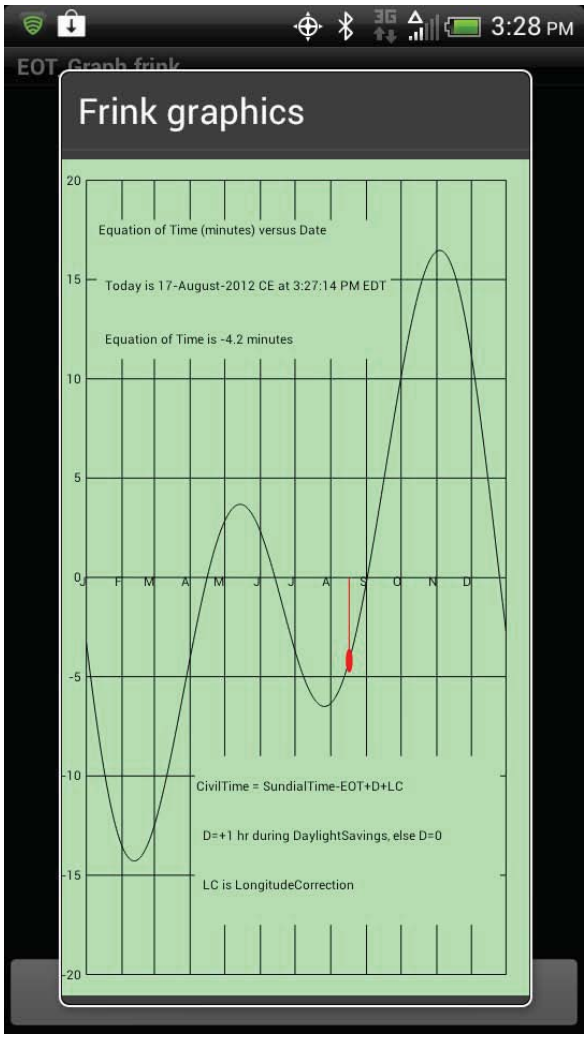
Every moment of light and dark is a miracle.

–Walt Whitman

* **Compendium**... "giving the sense and substance of the topic within small compass." In dialing, a compendium is a single instrument incorporating a variety of dial types and ancillary tools.

An Open-Source App(lication) For Displaying The Equation of Time On Android-Based Platforms

Donald L Snyder (St. Louis MO)



Screenshot showing EOT to be -4.2 minutes on August 17, 2012, at 3:27 PM EDT

Frink is the implementation by Alan Eliason of a scientific programming language that runs on PCs and Android-based smart phones and tablets. It can be downloaded at no cost from the website http://futureboy.us/frinkdocs/. Documentation for *Frink* is available on this website as are listings of many sample programs written in the *Frink* language. Included among the samples are programs implementing some expressions found in the book *Astronomical Algorithms* by Jean Meeus and other programs that dialists might find of interest, such as a program to predict solar alignments with the "infinite" hallway of MIT.

I have written a program (*i.e.*, an "app") to run on my Android phone for displaying the equation of time. Shown in Fig. 1 is a screen shot of my Android phone taken during the 2012 Annual Meeting of the North American Sundial Society in Asheville, NC. A listing of my program is in the Appendix. The equation of time is evaluated using the approximation FE1 of H. O. Ramp [1]. Here are steps you can take to get this program running on your own Android platform.

Step 1. Download *Frink* onto your Android platform. Instructions for doing this are on the *Frink* website given above. *Frink* will then appear among the applications on your device. All the features of *Frink* can then be used to perform scientific calculations and to run programs you or others have written.

Step 2. Store the program on your device. This can be accomplished by entering the lines of code directly into *Frink*, but this is a very tedious way to do it. Instead, you can enter the lines of code into a text file on your PC and then load that file into your Android device by connecting the device via a USB cable with your device in disk-drive mode, but this is still somewhat tedious. It is easier to download the a file having the code and then load that into your Android device via a USB cable. Download and save the file EOT.zip from http://dls-website.com/documents/EOT_program. This zip file contains: 1, a text file, EOT.txt, having the program in the Appendix; and 2, a *Frink* file, EOT_Graph.frink, that can be executed by *Frink*.

That's it. Run *Frink*, load the EOT program, and run it.

Reference: [1] Herbert O. Ramp , "Equation of Time – Comparison Of Approximating Formulae," *The Compendium – Journal of the North American Sundial Society*, 18(1):18-20, March 2011.

Donald L. Snyder, 7385 Pershing Ave., Unit #2, St. Louis MO 63130 dls@ee.wustl.edu

Appendix. EOT program listing

```
// Calculate Equation of Time using approx. FE1
// noted by H.O. Ramp in The Compendium  3/2011
// Created by: D. L. Snyder   27 February 2011
// Start .....
fmt = ### d-MMMM-yyyy 'CE at' h:mm:ss a z ###
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
// In the line below change 2011 to the current year
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]
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]
//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", 1],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
```