

```

#
# Local sidereal clock GUI
# TB 14/10/05
#
# Displays Greenwich Sidereal Time and Local Sidereal Time for another (user-
selected) location
#

from Tkinter import *
# for Python 3
# import _thread
# for Python 2.7
import thread
import time
import Pmw

def find_lst(tel_long_deg):
    sid = find_sid()
    (sid_hour,sid_min,sid_sec) = sid.split()
    sid_dhour = float(sid_hour) + float(sid_min)/60.0 + float(sid_sec)/3600.
    lst_hours = sid_dhour - tel_long_deg/15.
    if lst_hours > 24.0:
        lst_hours = lst_hours - 24.0
    if lst_hours < 0.0:
        lst_hours = lst_hours + 24.0

    lst_hr = int(lst_hours)
    lst_min = int((lst_hours - lst_hr)*60.)
    lst_sec = int((lst_hours - lst_hr - lst_min/60.)*3600.)
    return "%2.2d %2.2d %2.2d" % (lst_hr, lst_min, lst_sec)

def find_sid():
    #
    # formula from Duffett-Smith's ``Practical Astronomy'' see Page 16-20
    #
    xxx = time.time()
    (year,month,mday,hours,mins,sec,week,jd,d)=time.gmtime(xxx)
    ut_hours = hours + (mins + sec/60.)/60.
    mday = mday + ut_hours/24.

    if month <= 2 :
        year = year - 1
        month = month + 12

```

```

a = int(year/100.)
b = 2. - a + int(a/4.)

if year < 0 :
    c = int((365.2500*year)-0.7500)
else:
    c = int(365.2500*year)

d = int(30.600100*(month+1))
jd = b + c + d + int(mday) + 1720994.500

s = jd - 2451545.000
t = s/36525.000
t0 = 6.697374558 + (2400.051336*t) + (0.000025862*(t*t));
t0 = (t0 - int(t0/24.)*24)
if t0 < 0.0: t0 = t0 + 24.
ut = 1.002737909*ut_hours
tmp = int((ut + t0)/24.)
gst = ut + t0 - tmp*24.
gst_hour = int(gst)
gst_min = int((gst - gst_hour)*60.)
gst_sec = int((gst - gst_hour - gst_min/60.)*3600.)
return "%2.2d %2.2d %2.2d" % (gst_hour, gst_min, gst_sec)

```

```

LOCATIONS=[[ 'Durham',1.583],
             ['De Kalb', 88.743],
             ['New York',74.],
             ['Belfast',5.],
             ['Sydney',-151.],
             ['Kathmandu',-85.],
             ['Eurodisney',2.77]]

```

```

class Local_Sidereal_Clock:

```

```

    def __init__(self):
        self.root=Tk()
        self.root.title("Local Sidereal Clock")

        Label(self.root,text="Location:").grid(row=0,column=1,padx=5,sticky='ew')
        Label(self.root,text="Time:").grid(row=0,column=2,padx=5,sticky='ew')

        Label(self.root,text="GMST").grid(row=1,column=0,padx=5,pady=5,sticky='ew')

```

```
Label(self.root,text="LAST").grid(row=2,column=0,padx=5,pady=5,sticky='ew')
```

```
Label(self.root,text="Greenwich",anchor=W).grid(row=1,column=1,padx=5,pady=5,sticky='ew')
```

```
locs=[]
```

```
for l in LOCATIONS:
```

```
    locs.append(l[0])
```

```
self.loc=Pmw.ComboBox(self.root,
```

```
                        scrolledlist_items=locs,
```

```
                        selectioncommand=self.set_location,
```

```
                        listheight=19*len(LOCATIONS),
```

```
                        entry_width=12)
```

```
self.loc.selectitem('De Kalb')
```

```
self.set_location('De Kalb')
```

```
self.loc.grid(row=2,column=1,padx=5,pady=5)
```

```
self.GMST=Pmw.EntryField(self.root,entry_width=10,entry_justify=CENTER)
```

```
self.GMST.grid(row=1,column=2,padx=5,pady=5)
```

```
self.LAST=Pmw.EntryField(self.root,entry_width=10,entry_justify=CENTER)
```

```
self.LAST.grid(row=2,column=2,padx=5,pady=5)
```

```
thread.start_new_thread(self.update_thread,())
```

```
self.root.mainloop()
```

```
def update_thread(self):    # constantly calculate times and update display
```

```
    while 1:
```

```
        gmst=find_lst(0.)
```

```
        self.GMST.setvalue(gmst)
```

```
        last=find_lst(self.longitude)
```

```
        self.LAST.setvalue(last)
```

```
        time.sleep(0.5)
```

```
def set_location(self,loc):    # set longitude for new location
```

```
    for l in LOCATIONS:
```

```
        if l[0]==loc:
```

```
            self.longitude=l[1]
```

Local_Sidereal_Clock()