from signal import signal, SIGTERM, SIGHUP, pause

from rpi\_lcd import LCD

import time

import datetime

from datetime import datetime as dt2

import subprocess

#from pytz import timezone

import pytz

lcd = LCD()

def execute\_unix(inputcommand):

p = subprocess.Popen(inputcommand, stdout=subprocess.PIPE, shell=True)

(output, err) = p.communicate()

return output

def safe\_exit(signum, frame):

exit(1)

while True:

t = time.time()

### time\_current = time.strftime("%H:%M:%S", time.gmtime(t))

my\_datetime = dt2.now(datetime.timezone.utc)

### my\_datetime\_cst = my\_datetime.astimezone(pytz.timezone('US/Central')).strftime('%Y-%m-%d %H:%M:%S %Z%z')

my\_datetime\_cst\_1 = my\_datetime.astimezone(pytz.timezone('US/Central')).strftime('%B %d')

my\_datetime\_cst\_2 = my\_datetime.astimezone(pytz.timezone('US/Central')).strftime('%H:%M:%S')

print(my\_datetime)

print(my\_datetime\_cst\_1)

print(my\_datetime\_cst\_2)

try:

# Display date and time in LCD

lcd.text(my\_datetime\_cst\_1, 1)

lcd.text(my\_datetime\_cst\_2, 2)

# Create Phonic Messages

msg\_1 = 'espeak -ven+m3 -k5 -s150 --punct="<characters>" "%s" 2>>/dev/null' % my\_datetime\_cst\_1 #speak aloud

msg\_2 = 'espeak -ven+m3 -k5 -s150 --punct="<characters>" "%s" 2>>/dev/null' % my\_datetime\_cst\_2 #speak aloud

# Execute Phonetic Messages

execute\_unix(msg\_1)

execute\_unix(msg\_2)

signal(SIGTERM, safe\_exit)

signal(SIGHUP, safe\_exit)

#time.sleep()

pause()

except KeyboardInterrupt:

pass

finally:

lcd.clear()