```
# GUI to control RPI GPIO
# 18.5.17
# by Julian Rogers
# gpio_45 equivalent to "GPIO CONTROLLER 1.2"
# development for "old pi" used as control for back berdoom heating
# includes control of radiator valve, aux heater, water heater and rail plus a spare
# (basically, the hardware controls four relays and has a temperature sensor input)
# (could easily be extended for more I2C sensors)

# four channels of on off control each with two ons and offs
# thermostat function on any one of the channels
# thermostat over rides Advance setting - to over ride, select System "0"
# delay in operation of thermostat privided to avoid chatter near target temperature
# "total_delay"  variable determines the delay...
# which is between total_delay and 2 X total_delay secs
# this may result in the system seeming to be unresponsive (but it's not!)
# (an alternative would have been to build in a thermostsat hysteresis function)
# Advance function is available on any channel but see above...
# Advance does not over ride the themostat if the thermostat says off, it means it!
# all set values saved to file
# now with master on/off

# Apologies...
# too many global variables!
# not enough oop etc etc!
# unsatisfactory order of functions, Tkinter Labels, Option Menus, Buttons etc...
# a bit mixed up!
# needs a lot of stuff for clarity and elegance but it seems to work!!
# which is probably the main thing.


#import GUI and GPIO
from tkinter import *

import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)


GPIO.setup(4, GPIO.OUT)
GPIO.setup(17, GPIO.OUT)
GPIO.setup(27, GPIO.OUT)
GPIO.setup(22, GPIO.OUT)

GPIO.output(4, False)
GPIO.output(17, False)
GPIO.output(27, False)
GPIO.output(22, False)

#import time stuff and I2C
import datetime
import time
import smbus
bus = smbus.SMBus(1)

# create the root window
root = Tk()

# modify the window
root.title("mr-r.co.uk's IOT CONTROLLER for Raspberry Pi")
#root.geometry("700x430")
root.configure(bg = "gray")
#root.attributes('-fullscreen', True)   #eliminates the title bar

w, h = root.winfo_screenwidth(), root.winfo_screenheight()
root.geometry("%dx%d+0+0" % (w, h))

# create a frame
app = Frame(root)
app.configure(bg = "gray")
app.grid()

global advance1
advance1 = False

global advance2
advance2 = False

global advance3
advance3 = False

global advance4
advance4 = False

global all_on
all_on = True

global total_mins

global old_save_list
```

```
global new_select, old_select, new_temp, old_temp, temperature

global flag_on1, flag_on2, flag_on3, flag_on4

global flag_off1, flag_off2, flag_off3, flag_off4

global ok
# ok is used with error reporting
ok = True

# these flags are used to ensure that Advance is cancelled
# after just one period of Advance
flag_on1 = False
flag_on2 = False
flag_on3 = False
flag_on4 = False

flag_off1 = False
flag_off2 = False
flag_off3 = False
flag_off4 = False

#-----------------------------------------------------------------

# simulate on/off hysteresis

global delay, total_delay
delay = 0
total_delay = 10

#-----------------------------------------------------------------------
temperature = 18 # testing purposes
#-----------------------------------------------------------------

# reset error message
def ok():
    global ok
    ok = True

#-------------------------------------------------------------------------------

def m_switch():
    global all_on
    all_on = not all_on




#-----------------------------------------------------------------------------------------

temperature_lab = Label(app, bg = "grey", fg = "maroon", font = ("Arial", 14), text = "Temp")
temperature_lab.grid(row = 9, column = 8)

temp_but = Button(app, text = str(temperature), font = ("Arial", 14), fg = "maroon", bg = "grey")
temp_but.grid(row = 10 , column = 8)

status_lab = Label(app, bg = "grey", fg = "maroon", font = ("Arial", 14), text = "Status")
status_lab.grid(row = 9, column = 11)

status_but = Button(app, font = ("Arial", 15), fg = "maroon", bg = "grey", text = " OK ", command = ok)
status_but.grid(row = 10, column = 11)

on_off_but = Button(app, font = ("Arial", 15), fg = "blue" , bg = "red", text = " ON", command = m_switch)
on_off_but.grid(row = 10, column = 1)

on_off_lab = Label(app, bg = "grey", fg = "maroon", font = ("Arial", 14), text = "Master")
on_off_lab.grid(row = 9, column = 1)

#-------------------------------------------------------------------------------------------------

# get on/off data from file

try:
    data_file_2 = open("data_2.txt" , "r")
    old_save_list = data_file_2.readlines()
    data_file_2.close()

except IOError:
    old_save_list = ["0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0"]    #default values
    status_but.config(text = "E1")
    ok = False

def advance_not_1():
    global advance1
    advance1 = not advance1

def advance_not_2():
    global advance2
    advance2 = not advance2

def advance_not_3():
```

```
    global advance3
    advance3 = not advance3

def advance_not_4():
    global advance4
    advance4 = not advance4




#-------------------------------------------------------------------------------------------

title_lab = Label(app, text = "GPIO CONTROLLER 1.2", font = ("Arial Bold", 20), fg = "blue", bg = "gray")
title_lab.grid(row = 1, column = 1, columnspan = 10, sticky = W)

on_hr_1_lab = Label(app, bg = "gray", fg = "maroon", text = "ON hrs", font = ("Arial Bold", 12))
on_hr_1_lab.grid(row = 2, column = 2)

on_min_1_lab = Label(app, bg = "gray", fg = "maroon", text = "ON min", font = ("Arial Bold", 12))
on_min_1_lab.grid(row = 2, column = 3)

off_hr_1_lab = Label(app, bg = "gray", fg = "maroon", text = "OFF hrs", font = ("Arial Bold", 12))
off_hr_1_lab.grid(row = 2, column = 5)

off_min_1_lab = Label(app, bg = "gray", fg = "maroon", text = "OFF min", font = ("Arial Bold", 12))
off_min_1_lab.grid(row = 2, column = 6)

on_hr_2_lab = Label(app, bg = "gray", fg = "maroon", text = "ON hrs", font = ("Arial Bold", 12))
on_hr_2_lab.grid(row = 2, column = 8)

on_min_2_lab = Label(app, bg = "gray", fg = "maroon", text = "ON min", font = ("Arial Bold", 12))
on_min_2_lab.grid(row = 2, column = 9)

off_hr_2_lab = Label(app, bg = "gray", fg = "maroon", text = "OFF hrs", font = ("Arial Bold", 12))
off_hr_2_lab.grid(row = 2, column = 11)

off_min_2_lab = Label(app, bg = "gray", fg = "maroon", text = "OFF min", font = ("Arial Bold", 12))
off_min_2_lab.grid(row = 2, column = 12)


#-----------------------------------------------------------------------------------------------

sys_1_but = Button(app, text = "Sys 1", font = ("Arial", 16), fg = "maroon", bg = "light blue", command = advance_not_1)
sys_1_but.grid(row = 3, column = 1)

sys_2_but = Button(app, text = "Sys 2", font = ("Arial", 16), fg = "maroon", bg = "light blue", command = advance_not_2)
sys_2_but.grid(row = 4, column = 1)

sys_3_but = Button(app, text = "Sys 3", font = ("Arial", 16), fg = "maroon", bg = "light blue", command = advance_not_3)
sys_3_but.grid(row = 5, column = 1)

sys_4_but = Button(app, text = "Sys 4", font = ("Arial", 16), fg = "maroon", bg = "light blue", command = advance_not_4)
sys_4_but.grid(row = 6, column = 1)



#-------------------------------------------------------------------------

# option menues to set minutes and hours

var1a = StringVar(app) # initial value hours
var2a = StringVar(app) # initial value minutes
var3a = StringVar(app) # initial value hours
var4a = StringVar(app) # initial value minutes
var5a = StringVar(app) # initial value hours
var6a = StringVar(app) # initial value minutes
var7a = StringVar(app) # initial value hours
var8a = StringVar(app) # initial value minutes

var1b = StringVar(app) # initial value hours
var2b = StringVar(app) # initial value minutes
var3b = StringVar(app) # initial value hours
var4b = StringVar(app) # initial value minutes
var5b = StringVar(app) # initial value hours
var6b = StringVar(app) # initial value minutes
var7b = StringVar(app) # initial value hours
var8b = StringVar(app) # initial value minutes

var1c = StringVar(app) # initial value hours
var2c = StringVar(app) # initial value minutes
var3c = StringVar(app) # initial value hours
var4c = StringVar(app) # initial value minutes
var5c = StringVar(app) # initial value hours
var6c = StringVar(app) # initial value minutes
var7c = StringVar(app) # initial value hours
var8c = StringVar(app) # initial value minutes

var1d = StringVar(app) # initial value hours
var2d = StringVar(app) # initial value minutes
var3d = StringVar(app) # initial value hours
```

```
var4d = StringVar(app) # initial value minutes
var5d = StringVar(app) # initial value hours
var6d = StringVar(app) # initial value minutes
var7d = StringVar(app) # initial value hours
var8d = StringVar(app) # initial value minutes

var_temp = StringVar(app) # value deg C for thermostat
var_select =StringVar(app) # number of system selected for thermostat

#-----------------------------------------------------------------

# get thermosta settings and system number for thermostat operation from file
try:
    thermo_file = open("thermostat.txt", "r")
    var_temp.set(thermo_file.read())
    thermo_file.close()

except IOError:
    var_temp.set("18")   #default value is 18 degrees
    status_but.config(text = "E2")
    ok = False

try:
    select_file = open("system_select.txt", "r")
    var_select.set(select_file.read())
    select_file.close()

except IOError:
    var_select.set("1")   #default value is 1
    status_but.config(text = "E3")
    ok = False

# update variables
old_temp = int(var_temp.get())
old_select = int(var_select.get())


#-------------------------------------------------------------------------------------------

def format_times(on_off_times):

    mins_in_file = int(on_off_times[0])
    hours_in_file = mins_in_file // 60
    mins_in_file = mins_in_file % 60
    hours_string = str(hours_in_file)
    mins_string = str(mins_in_file)

    if len(hours_string) == 1:
        hours_string = "0" + hours_string

    if len(mins_string) == 1:
        mins_string = "0" + mins_string


    var1a.set(hours_string)
    var2a.set(mins_string)



    mins_in_file = int(on_off_times[1])
    hours_in_file = mins_in_file // 60
    mins_in_file = mins_in_file % 60
    hours_string = str(hours_in_file)
    mins_string = str(mins_in_file)

    if len(hours_string) == 1:
        hours_string = "0" + hours_string

    if len(mins_string) == 1:
        mins_string = "0" + mins_string


    var3a.set(hours_string)
    var4a.set(mins_string)

    mins_in_file = int(on_off_times[2])
    hours_in_file = mins_in_file // 60
    mins_in_file = mins_in_file % 60
    hours_string = str(hours_in_file)
    mins_string = str(mins_in_file)

    if len(hours_string) == 1:
        hours_string = "0" + hours_string

    if len(mins_string) == 1:
        mins_string = "0" + mins_string

    var5a.set(hours_string)
    var6a.set(mins_string)

    mins_in_file = int(on_off_times[3])
```

```
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var7a.set(hours_string)
var8a.set(mins_string)

#-----------------------------------

mins_in_file = int(on_off_times[4])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var1b.set(hours_string)
var2b.set(mins_string)

mins_in_file = int(on_off_times[5])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var3b.set(hours_string )
var4b.set(mins_string)

mins_in_file = int(on_off_times[6])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var5b.set(hours_string)
var6b.set(mins_string)

mins_in_file = int(on_off_times[7])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var7b.set(hours_string)
var8b.set(mins_string)

#----------------------------------

mins_in_file = int(on_off_times[8])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var1c.set(hours_string)
```

```
var2c.set(mins_string)

mins_in_file = int(on_off_times[9])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var3c.set(hours_string)
var4c.set(mins_string)

mins_in_file = int(on_off_times[10])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var5c.set(hours_string)
var6c.set(mins_string)

mins_in_file = int(on_off_times[11])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var7c.set(hours_string)
var8c.set(mins_string)

#---------------------------------

mins_in_file = int(on_off_times[12])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60

hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string
var1d.set(hours_string )
var2d.set(mins_string)

mins_in_file = int(on_off_times[13])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string

var3d.set(hours_string)
var4d.set(mins_string)

mins_in_file = int(on_off_times[14])
hours_in_file = mins_in_file // 60
mins_in_file = mins_in_file % 60
hours_string = str(hours_in_file)
mins_string = str(mins_in_file)

if len(hours_string) == 1:
    hours_string = "0" + hours_string

if len(mins_string) == 1:
    mins_string = "0" + mins_string
```

```
    var5d.set(hours_string)
    var6d.set(mins_string)

    mins_in_file = int(on_off_times[15])
    hours_in_file = mins_in_file // 60
    mins_in_file = mins_in_file % 60
    hours_string = str(hours_in_file)
    mins_string = str(mins_in_file)

    if len(hours_string) == 1:
        hours_string = "0" + hours_string

    if len(mins_string) == 1:
        mins_string = "0" + mins_string

    var7d.set(hours_string)
    var8d.set(mins_string)

#-------------------------------------------------------------------------------------------

# get on / off data from file - this is in minutes

try:
    data_file_2 = open("data_2.txt" , "r")
    on_off_times = data_file_2.readlines()
    data_file_2.close()

except IOError:
    on_off_times = ["0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0"]    #default values
    status_but.config(text = "ERR1")



# format data with leading zeros where required and insert into appropriate
# option menus

format_times(on_off_times)



#----------------------------------



hour_opt_men1a = OptionMenu(app,var1a,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men1a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men1a.grid(row = 3, column = 2, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men2a = OptionMenu(app,var2a,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men2a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men2a.grid(row = 3, column = 3, columnspan = 1, sticky = W)

fill_lab1 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab1.grid(row = 3, column = 4)

hour_opt_men3a = OptionMenu(app,var3a,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men3a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men3a.grid(row = 3, column = 5, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men4a = OptionMenu(app,var4a,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men4a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men4a.grid(row = 3, column = 6, columnspan = 1, sticky = W)

fill_lab2 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab2.grid(row = 3, column = 7)

hour_opt_men5a = OptionMenu(app,var5a,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men5a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men5a.grid(row = 3, column = 8, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men6a = OptionMenu(app,var6a,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men6a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men6a.grid(row = 3, column = 9, columnspan = 1, sticky = W)

fill_lab3 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab3.grid(row = 3, column = 10)
```

```
hour_opt_men7a = OptionMenu(app,var7a,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men7a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men7a.grid(row = 3, column = 11, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men8a = OptionMenu(app,var8a,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men8a.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men8a.grid(row = 3, column = 12, columnspan = 1, sticky = W)


#-----------------------------------------------------------------------------------------------


hour_opt_men1b = OptionMenu(app,var1b,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men1b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men1b.grid(row = 4, column = 2, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men2b = OptionMenu(app,var2b,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men2b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men2b.grid(row = 4, column = 3, columnspan = 1, sticky = W)

fill_lab4 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab4.grid(row = 4, column = 4)

hour_opt_men3b = OptionMenu(app,var3b,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men3b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men3b.grid(row = 4, column = 5, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men4b = OptionMenu(app,var4b,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men4b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men4b.grid(row = 4, column = 6, columnspan = 1, sticky = W)

fill_lab5 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab5.grid(row = 4, column = 7)

hour_opt_men5b = OptionMenu(app,var5b,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men5b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men5b.grid(row = 4, column = 8, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men6b = OptionMenu(app,var6b,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men6b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men6b.grid(row = 4, column = 9, columnspan = 1, sticky = W)

fill_lab6 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab6.grid(row = 4, column = 10)

hour_opt_men7b = OptionMenu(app,var7b,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men7b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men7b.grid(row = 4, column = 11, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men8b = OptionMenu(app,var8b,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men8b.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men8b.grid(row = 4, column = 12, columnspan = 1, sticky = W)


#-----------------------------------------------------------------------------------------------


hour_opt_men1c = OptionMenu(app,var1c,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men1c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men1c.grid(row = 5, column = 2, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!
```

```
min_opt_men2c = OptionMenu(app,var2c,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men2c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men2c.grid(row = 5, column = 3, columnspan = 1, sticky = W)

fill_lab7 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab7.grid(row = 5, column = 4)

hour_opt_men3c = OptionMenu(app,var3c,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men3c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men3c.grid(row = 5, column = 5, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men4c = OptionMenu(app,var4c,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men4c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men4c.grid(row = 5, column = 6, columnspan = 1, sticky = W)

fill_lab8 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab8.grid(row = 5, column = 7)

hour_opt_men5c = OptionMenu(app,var5c,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men5c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men5c.grid(row = 5, column = 8, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men6c = OptionMenu(app,var6c,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men6c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men6c.grid(row = 5, column = 9, columnspan = 1, sticky = W)

fill_lab9 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab9.grid(row = 5, column = 10)

hour_opt_men7c = OptionMenu(app,var7c,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men7c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men7c.grid(row = 5, column = 11, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men8c = OptionMenu(app,var8c,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men8c.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men8c.grid(row = 5, column = 12, columnspan = 1, sticky = W)

#-----------------------------------------------------------------------------------------------------------------

hour_opt_men1d = OptionMenu(app,var1d,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men1d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men1d.grid(row = 6, column = 2, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men2d = OptionMenu(app,var2d,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men2d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men2d.grid(row = 6, column = 3, columnspan = 1, sticky = W)

fill_lab10 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab10.grid(row = 6, column = 4)

hour_opt_men3d = OptionMenu(app,var3d,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men3d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men3d.grid(row = 6, column = 5, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men4d = OptionMenu(app,var4d,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men4d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men4d.grid(row = 6, column = 6, columnspan = 1, sticky = W)

fill_lab11 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab11.grid(row = 6, column = 7)

hour_opt_men5d = OptionMenu(app,var5d,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men5d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
```

```
hour_opt_men5d.grid(row = 6, column = 8, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men6d = OptionMenu(app,var6d,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men6d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men6d.grid(row = 6, column = 9, columnspan = 1, sticky = W)

fill_lab12 = Label(app, text = "X", font = ("Arial", 16), fg = "grey", bg = "grey")
fill_lab12.grid(row = 6, column = 10)

hour_opt_men7d = OptionMenu(app,var7d,"00","01","02","03","04","05","06","07","08","09","10",\
                                "11","12","13","14","15","16","17","18","19","20","21","22",\
                                "23")
hour_opt_men7d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
hour_opt_men7d.grid(row = 6, column = 11, sticky = W)

# minutes only multiples of 5 as 60 options do not fit on the Pi screen!

min_opt_men8d = OptionMenu(app,var8d,"00","05","10","15","20","25","30","35","40","45","50","55")

min_opt_men8d.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
min_opt_men8d.grid(row = 6, column = 12, columnspan = 1, sticky = W)


#-----------------------------------------------------------------------------------------

def read_temperature():
    #some more error handling needed here?
    global temperature, ok

    try:
        data = bus.read_i2c_block_data(0x48,0)
        msb = data[0]
        lsb = data[1]
        temperature = (((msb << 8) | lsb) >> 4) * 0.0625
        temperature = round(temperature,1)
        temp_but.config(text = str(temperature))
    except OSError:
        temperature = -1
        ok = False

    if temperature < 0 or temperature > 35:
        temperature = 0
        status_but.config(text = "E4")
        temp_but.config(text = "err")

#-------------------------------------------------------------


def get_time():
    time_lab.config(text = "Comp time " + time.strftime('%H:%M'))
    now = datetime.datetime.now()
    hour = int(now.hour)
    mins = int(now.minute)
    global total_mins
    total_mins = hour * 60 + mins


#-----------------------------------------------------------
#-----------------------------------------------------------


def get_timed_sys():

    get_time()


    global advance1, advance2, advance3, advance4
    global all_on
    global total_mins
    global delay, total_delay

    global flag_on1, flag_on2, flag_on3, flag_on4
    global flag_off1, flag_off2, flag_off3, flag_off4

    global temperature



    set_mins_1a = int(old_save_list[0])
    set_mins_2a = int(old_save_list[1])
    set_mins_3a = int(old_save_list[2])
    set_mins_4a = int(old_save_list[3])
    set_mins_1b = int(old_save_list[4])
    set_mins_2b = int(old_save_list[5])
    set_mins_3b = int(old_save_list[6])
    set_mins_4b = int(old_save_list[7])
    set_mins_1c = int(old_save_list[8])
    set_mins_2c = int(old_save_list[9])
```

```
set_mins_3c = int(old_save_list[10])
set_mins_4c = int(old_save_list[11])
set_mins_1d = int(old_save_list[12])
set_mins_2d = int(old_save_list[13])
set_mins_3d = int(old_save_list[14])
set_mins_4d = int(old_save_list[15])


#------------




#----------------------sys 1 start
#off

if total_mins < set_mins_1a and advance1 == False:
    sys_1_but.config(bg = "light blue")
    GPIO.output(4, False)
    flag_off1 = True
#----------------------

if total_mins < set_mins_1a and advance1 == True:

    if int(var_select.get()) != 1:
        sys_1_but.config(bg = "yellow")
        if all_on:
            GPIO.output(4, True)
        else:
            GPIO.output(4, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "yellow")
            if all_on:
                GPIO.output(4, True)
            else:
                GPIO.output(4, False)



    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "light pink")
            GPIO.output(4, False)



    flag_off1 = True
#--------------------------
#on

if total_mins >= set_mins_1a and total_mins < set_mins_2a and advance1 == False:

    if int(var_select.get()) != 1:
        sys_1_but.config(bg = "yellow")
        if all_on:
            GPIO.output(4, True)
        else:
            GPIO.output(4, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "yellow")
            if all_on:
                GPIO.output(4, True)
            else:
                GPIO.output(4, False)



    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "light pink")
            GPIO.output(4, False)



    flag_on1 = True
#----------------

if total_mins >= set_mins_1a and total_mins < set_mins_2a and advance1 == True:
    sys_1_but.config(bg = "light blue")
```

```
    GPIO.output(4, False)
    flag_on1 = True
#-----------------
#off

if total_mins >= set_mins_2a and total_mins < set_mins_3a and advance1 == False:
    sys_1_but.config(bg = "light blue")
    GPIO.output(4, False)
    flag_off1 = True
#----------------

if total_mins >= set_mins_2a and total_mins < set_mins_3a and advance1 == True:

    if int(var_select.get()) != 1:
        sys_1_but.config(bg = "yellow")
        if all_on:
            GPIO.output(4, True)
        else:
            GPIO.output(4, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "yellow")
            if all_on:
                GPIO.output(4, True)
            else:
                GPIO.output(4, False)



    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "light pink")
            GPIO.output(4, False)



    flag_off1 = True
#---------------
#on

if total_mins >= set_mins_3a and total_mins < set_mins_4a and advance1 == False:

    if int(var_select.get()) != 1:
        sys_1_but.config(bg = "yellow")
        if all_on:
            GPIO.output(4, True)
        else:
            GPIO.output(4, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "yellow")
            if all_on:
                GPIO.output(4, True)
            else:
                GPIO.output(4, False)



    elif temperature >= int(var_temp.get()):
        tflag_off = True
        if delay == total_delay:
            delay = 0
            sys_1_but.config(bg = "light pink")
            GPIO.output(4, False)



    flag_on1 = True
#-----------------

if total_mins >= set_mins_3a and total_mins < set_mins_4a and advance1 == True:
    sys_1_but.config(bg = "light blue")
    GPIO.output(4, False)
    flag_on1 = True
#-----------------
#off
if total_mins >= set_mins_4a and advance1 == False:
    sys_1_but.config(bg = "light blue")
    GPIO.output(4, False)

    flag_off1 = True
#---------------
```

```
    if total_mins >= set_mins_4a and advance1 == True:

        if int(var_select.get()) != 1:
            sys_1_but.config(bg = "yellow")
            if all_on:
                GPIO.output(4, True)
            else:
                GPIO.output(4, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_1_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(4, True)
                else:
                    GPIO.output(4, False)



        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_1_but.config(bg = "light pink")
                GPIO.output(4, False)



        flag_off1 = True
#-------------------------------------------------------------------------
    #-----------sys 2 start
    #off

    if total_mins < set_mins_1b and advance2 == False:
        sys_2_but.config(bg = "light blue")
        GPIO.output(17, False)
        flag_off2 = True
    #---------------------

    if total_mins < set_mins_1b and advance2 == True:

        if int(var_select.get()) != 2:
            sys_2_but.config(bg = "yellow")
            if all_on:
                GPIO.output(17, True)
            else:
                GPIO.output(17, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_2_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(17, True)
                else:
                    GPIO.output(17, False)


        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_2_but.config(bg = "light pink")
                GPIO.output(17, False)



        flag_off2 = True
    #---------------------
    #on

    if total_mins >= set_mins_1b and total_mins < set_mins_2b and advance2 == False:

        if int(var_select.get()) != 2:
            sys_2_but.config(bg = "yellow")
            if all_on:
                GPIO.output(17, True)
            else:
                GPIO.output(17, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_2_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(17, True)
                else:
```

```
                GPIO.output(17, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_2_but.config(bg = "light pink")
            GPIO.output(17, False)



    flag_on2 = True
#---------------------

if total_mins >= set_mins_1b and total_mins < set_mins_2b and advance2 == True:
    sys_2_but.config(bg = "light blue")
    GPIO.output(17, False)
    flag_on2 = True
#---------------------
#off

if total_mins >= set_mins_2b and total_mins < set_mins_3b and advance2 == False:
    sys_2_but.config(bg = "light blue")
    GPIO.output(17, False)
    flag_off2 = True
#---------------------

if total_mins >= set_mins_2b and total_mins < set_mins_3b and advance2 == True:

    if int(var_select.get()) != 2:
        sys_2_but.config(bg = "yellow")
        if all_on:
            GPIO.output(17, True)
        else:
            GPIO.output(17, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_2_but.config(bg = "yellow")
            if all_on:
                GPIO.output(17, True)
            else:
                GPIO.output(17, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_2_but.config(bg = "light pink")
            GPIO.output(17, False)



    flag_off2 = True
#---------------------
#on

if total_mins >= set_mins_3b and total_mins < set_mins_4b and advance2 == False:

    if int(var_select.get()) != 2:
        sys_2_but.config(bg = "yellow")
        if all_on:
            GPIO.output(17, True)
        else:
            GPIO.output(17, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_2_but.config(bg = "yellow")
            if all_on:
                GPIO.output(17, True)
            else:
                GPIO.output(17, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_2_but.config(bg = "light pink")
            GPIO.output(17, False)



    flag_on2 = True
```

```
    #-----------------

    if total_mins >= set_mins_3b and total_mins < set_mins_4b and advance2 == True:
        sys_2_but.config(bg = "light blue")
        GPIO.output(17, False)
        flag_on2 = True
    #-----------------
    #off

    if total_mins >= set_mins_4b and advance2 == False:
        sys_2_but.config(bg = "light blue")
        GPIO.output(17, False)
        flag_off2 = True
    #-----------------

    if total_mins >= set_mins_4b and advance2 == True:

        if int(var_select.get()) != 2:
            sys_2_but.config(bg = "yellow")
            if all_on:
                GPIO.output(17, True)
            else:
                GPIO.output(17, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_2_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(17, True)
                else:
                    GPIO.output(17, False)


        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_2_but.config(bg = "light pink")
                GPIO.output(17, False)



        flag_off2 = True

#-------------------------------------------------------------------------------
    #------------sys 3 start
    #off

    if total_mins < set_mins_1c and advance3 == False:
        sys_3_but.config(bg = "light blue")
        GPIO.output(27, False)
        flag_off3 = True
    #-----------------

    if total_mins < set_mins_1c and advance3 == True:

        if int(var_select.get()) != 3:
            sys_3_but.config(bg = "yellow")
            if all_on:
                GPIO.output(27, True)
            else:
                GPIO.output(27, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_3_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(27, True)
                else:
                    GPIO.output(27, False)


        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_3_but.config(bg = "light pink")
                GPIO.output(27, False)



        flag_off3 = True
    #-----------------
    #on
    if total_mins >= set_mins_1c and total_mins < set_mins_2c and advance3 == False:

        if int(var_select.get()) != 3:
```

```
        sys_3_but.config(bg = "yellow")
        if all_on:
            GPIO.output(27, True)
        else:
            GPIO.output(27, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_3_but.config(bg = "yellow")
            if all_on:
                GPIO.output(27, True)
            else:
                GPIO.output(27, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_3_but.config(bg = "light pink")
            GPIO.output(27, False)


    flag_on3 = True
#------------------

if total_mins >= set_mins_1c and total_mins < set_mins_2c and advance3 == True:
    sys_3_but.config(bg = "light blue")
    GPIO.output(27, False)
    flag_on3 = True
#-----------------
#off
if total_mins >= set_mins_2c and total_mins < set_mins_3c and advance3 == False:
    sys_3_but.config(bg = "light blue")
    GPIO.output(27, False)
    flag_off3 = True
#-----------------

if total_mins >= set_mins_2c and total_mins < set_mins_3c and advance3 == True:

    if int(var_select.get()) != 3:
        sys_3_but.config(bg = "yellow")
        if all_on:
            GPIO.output(27, True)
        else:
            GPIO.output(27, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_3_but.config(bg = "yellow")
            if all_on:
                GPIO.output(27, True)
            else:
                GPIO.output(27, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_3_but.config(bg = "light pink")
            GPIO.output(27, False)



    flag_off3 = True
#-------------------
#on
if total_mins >= set_mins_3c and total_mins < set_mins_4c and advance3 == False:

    if int(var_select.get()) != 3:
        sys_3_but.config(bg = "yellow")
        if all_on:
            GPIO.output(27, True)
        else:
            GPIO.output(27, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_3_but.config(bg = "yellow")
            if all_on:
                GPIO.output(27, True)
            else:
                GPIO.output(27, False)
```

```
        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_3_but.config(bg = "light pink")
                GPIO.output(27, False)



        flag_on3 = True
    #----------------

    if total_mins >= set_mins_3c and total_mins < set_mins_4c and advance3 == True:
        sys_3_but.config(bg = "light blue")
        GPIO.output(27, False)
        flag_on3 = True
    #----------------
    #off
    if total_mins >= set_mins_4c and advance3 == False:
        sys_3_but.config(bg = "light blue")
        GPIO.output(27, False)
        flag_off3 = True
    #----------------

    if total_mins >= set_mins_4c and advance3 == True:

        if int(var_select.get()) != 3:
            sys_3_but.config(bg = "yellow")
            if all_on:
                GPIO.output(27, True)
            else:
                GPIO.output(27, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_3_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(27, True)
                else:
                    GPIO.output(27, False)


        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_3_but.config(bg = "light pink")
                GPIO.output(27, False)



        flag_off3 = True
#-------------------------------------------------------------------------------
    #---------sys 4 start
    #off

    if total_mins < set_mins_1d and advance4 == False:
        sys_4_but.config(bg = "light blue")
        GPIO.output(22, False)
        flag_off4 = True
    #--------------

    if total_mins < set_mins_1d and advance4 == True:

        if int(var_select.get()) != 4:
            sys_4_but.config(bg = "yellow")
            if all_on:
                GPIO.output(22, True)
            else:
                GPIO.output(22, False)

        elif temperature < int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_4_but.config(bg = "yellow")
                if all_on:
                    GPIO.output(22, True)
                else:
                    GPIO.output(22, False)


        elif temperature >= int(var_temp.get()):

            if delay == total_delay:
                delay = 0
                sys_4_but.config(bg = "light pink")
                GPIO.output(22, False)
```

```
    flag_off4 = True
#--------------
#on

if total_mins >= set_mins_1d and total_mins < set_mins_2d and advance4 == False:

    if int(var_select.get()) != 4:
        sys_4_but.config(bg = "yellow")
        if all_on:
            GPIO.output(22, True)
        else:
            GPIO.output(22, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "yellow")
            if all_on:
                GPIO.output(22, True)
            else:
                GPIO.output(22, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "light pink")
            GPIO.output(22, False)



    flag_on4 = True
#---------------

if total_mins >= set_mins_1d and total_mins < set_mins_2d and advance4 == True:
    sys_4_but.config(bg = "light blue")
    GPIO.output(22, False)
    flag_on4 = True
#-----------------
#off

if total_mins >= set_mins_2d and total_mins < set_mins_3d and advance4 == False:
    sys_4_but.config(bg = "light blue")
    GPIO.output(22, False)
    flag_off4 = True
#-----------------


if total_mins >= set_mins_2d and total_mins < set_mins_3d and advance4 == True:

    if int(var_select.get()) != 4:
        sys_4_but.config(bg = "yellow")
        if all_on:
            GPIO.output(22, True)
        else:
            GPIO.output(22, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "yellow")
            if all_on:
                GPIO.output(22, True)
            else:
                GPIO.output(22, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "light pink")
            GPIO.output(22, False)



    flag_off4 = True
#-----------------
#on

if total_mins >= set_mins_3d and total_mins < set_mins_4d and advance4 == False:

    if int(var_select.get()) != 4:
        sys_4_but.config(bg = "yellow")
        if all_on:
```

```
            GPIO.output(22, True)
        else:
            GPIO.output(22, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "yellow")
            if all_on:
                GPIO.output(22, True)
            else:
                GPIO.output(22, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "light pink")
            GPIO.output(22, False)



    flag_on4 = True
#--------------


if total_mins >= set_mins_3d and total_mins < set_mins_4d and advance4 == True:
    sys_4_but.config(bg = "light blue")
    GPIO.output(22, False)
    flag_on4 = True
#--------------
#off

if total_mins >= set_mins_4d and advance4 == False:
    sys_4_but.config(bg = "light blue")
    GPIO.output(22, False)
    flag_off4 = True
#--------------

if total_mins >= set_mins_4d and advance4 == True:

    if int(var_select.get()) != 4:
        sys_4_but.config(bg = "yellow")
        if all_on:
            GPIO.output(22, True)
        else:
            GPIO.output(22, False)

    elif temperature < int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "yellow")
            if all_on:
                GPIO.output(22, True)
            else:
                GPIO.output(22, False)


    elif temperature >= int(var_temp.get()):

        if delay == total_delay:
            delay = 0
            sys_4_but.config(bg = "light pink")
            GPIO.output(22, False)



    flag_off4 = True
#--------------------


if flag_on1 == True and flag_off1 == True:
    advance1 = False
    flag_on1 = False
    flag_off1 = False

if flag_on2 == True and flag_off2 == True:
    advance2 = False
    flag_on2 = False
    flag_off2 = False

if flag_on3 == True and flag_off3 == True:
    advance1 = False
    flag_on3 = False
    flag_off3 = False

if flag_on4 == True and flag_off4 == True:
    advance1 = False
```

```
        flag_on4 = False
        flag_off4 = False


    # delay avoids chatter at thermostat threshold
    delay = delay +1

    if delay > total_delay:
        delay = 0


#-----------------------------------------------------------------
#-----------------------------------------------------------------


def save_times():

    global old_save_list, new_select, old_select, new_temp, old_temp


    set_mins_1a = str(int(var2a.get()) + int(var1a.get()) * 60)
    set_mins_2a = str(int(var4a.get()) + int(var3a.get()) * 60)
    set_mins_3a = str(int(var6a.get()) + int(var5a.get()) * 60)
    set_mins_4a = str(int(var8a.get()) + int(var7a.get()) * 60)
    set_mins_1b = str(int(var2b.get()) + int(var1b.get()) * 60)
    set_mins_2b = str(int(var4b.get()) + int(var3b.get()) * 60)
    set_mins_3b = str(int(var6b.get()) + int(var5b.get()) * 60)
    set_mins_4b = str(int(var8b.get()) + int(var7b.get()) * 60)
    set_mins_1c = str(int(var2c.get()) + int(var1c.get()) * 60)
    set_mins_2c = str(int(var4c.get()) + int(var3c.get()) * 60)
    set_mins_3c = str(int(var6c.get()) + int(var5c.get()) * 60)
    set_mins_4c = str(int(var8c.get()) + int(var7c.get()) * 60)
    set_mins_1d = str(int(var2d.get()) + int(var1d.get()) * 60)
    set_mins_2d = str(int(var4d.get()) + int(var3d.get()) * 60)
    set_mins_3d = str(int(var6d.get()) + int(var5d.get()) * 60)
    set_mins_4d = str(int(var8d.get()) + int(var7d.get()) * 60)

    new_select = int(var_select.get())
    new_temp = int(var_temp.get())

    # here is some error checking - can't end before starting!

    if int(set_mins_2a) < int(set_mins_1a):
        set_mins_2a = set_mins_1a
        hour_opt_men3a.config(bg = "pink")
        min_opt_men4a.config(bg = "pink")
    if int(set_mins_2a) > int(set_mins_1a):
        hour_opt_men3a.config(bg = "grey")
        min_opt_men4a.config(bg = "grey")

    if int(set_mins_3a) < int(set_mins_2a):
        set_mins_3a = set_mins_2a
        hour_opt_men5a.config(bg = "pink")
        min_opt_men6a.config(bg = "pink")
    if int(set_mins_3a) > int(set_mins_2a):
        hour_opt_men5a.config(bg = "grey")
        min_opt_men6a.config(bg = "grey")

    if int(set_mins_4a) < int(set_mins_3a):
        set_mins_4a = set_mins_3a
        hour_opt_men7a.config(bg = "pink")
        min_opt_men8a.config(bg = "pink")
    if int(set_mins_4a) > int(set_mins_3a):
        hour_opt_men7a.config(bg = "grey")
        min_opt_men8a.config(bg = "grey")

    if int(set_mins_2b) < int(set_mins_1b):
        set_mins_2b = set_mins_1b
        hour_opt_men3b.config(bg = "pink")
        min_opt_men4b.config(bg = "pink")
    if int(set_mins_2b) > int(set_mins_1b):
        hour_opt_men3b.config(bg = "grey")
        min_opt_men4b.config(bg = "grey")

    if int(set_mins_3b) < int(set_mins_2b):
        set_mins_3b = set_mins_2b
        hour_opt_men5b.config(bg = "pink")
        min_opt_men6b.config(bg = "pink")
    if int(set_mins_3b) > int(set_mins_2b):
        hour_opt_men5b.config(bg = "grey")
        min_opt_men6b.config(bg = "grey")

    if int(set_mins_4b) < int(set_mins_3b):
        set_mins_4b = set_mins_3b
        hour_opt_men7b.config(bg = "pink")
        min_opt_men8b.config(bg = "pink")
    if int(set_mins_4b) > int(set_mins_3b):
        hour_opt_men7b.config(bg = "grey")
        min_opt_men8b.config(bg = "grey")

    if int(set_mins_2c) < int(set_mins_1c):
```

```
    set_mins_2c = set_mins_1c
    hour_opt_men3c.config(bg = "pink")
    min_opt_men4c.config(bg = "pink")
if int(set_mins_2c) > int(set_mins_1c):
    hour_opt_men3b.config(bg = "grey")
    min_opt_men4c.config(bg = "grey")

if int(set_mins_3c) < int(set_mins_2c):
    set_mins_3c = set_mins_2c
    hour_opt_men5c.config(bg = "pink")
    min_opt_men6c.config(bg = "pink")
if int(set_mins_3c) > int(set_mins_2c):
    hour_opt_men5c.config(bg = "grey")
    min_opt_men6c.config(bg = "grey")

if int(set_mins_4c) < int(set_mins_3c):
    set_mins_4c = set_mins_3c
    hour_opt_men7c.config(bg = "pink")
    min_opt_men8c.config(bg = "pink")
if int(set_mins_4c) > int(set_mins_3c):
    hour_opt_men7c.config(bg = "grey")
    min_opt_men8c.config(bg = "grey")

if int(set_mins_2d) < int(set_mins_1d):
    set_mins_2d = set_mins_1d
    hour_opt_men3d.config(bg = "pink")
    min_opt_men4d.config(bg = "pink")
if int(set_mins_2d) > int(set_mins_1d):
    hour_opt_men3d.config(bg = "grey")
    min_opt_men4d.config(bg = "grey")


if int(set_mins_3d) < int(set_mins_2d):
    set_mins_3d = set_mins_2d
    hour_opt_men5d.config(bg = "pink")
    min_opt_men6d.config(bg = "pink")
if int(set_mins_3d) > int(set_mins_2d):
    hour_opt_men5d.config(bg = "grey")
    min_opt_men6d.config(bg = "grey")

if int(set_mins_4d) < int(set_mins_3d):
    set_mins_4d = set_mins_3d
    hour_opt_men7d.config(bg = "pink")
    min_opt_men8d.config(bg = "pink")
if int(set_mins_4d) > int(set_mins_3d):
    hour_opt_men7d.config(bg = "grey")
    min_opt_men8d.config(bg = "grey")



save_list = [set_mins_1a, set_mins_2a, set_mins_3a, set_mins_4a, set_mins_1b, set_mins_2b, set_mins_3b, set_mins_4b, \
             set_mins_1c, set_mins_2c, set_mins_3c, set_mins_4c, set_mins_1d, set_mins_2d, set_mins_3d, set_mins_4d]

format_times(save_list)

if old_save_list != save_list:

    global advance1, advance2, advance3, advance4
    advance1 = False
    advance2 = False
    advance3 = False
    advance4 = False


    # some error trapping here might be a good idea

    data_file = open("data_2.txt", "w")


    for item in save_list:
        data_file.write("%s\n" %item)

    data_file.close
    format_times(save_list)



old_save_list = save_list

if new_temp != old_temp:
    temp_file = open("thermostat.txt", "w")
    temp_file.write(str(new_temp))
    temp_file.close()

    old_temp = new_temp


if new_select != old_select:
    temp_file = open("system_select.txt", "w")
    temp_file.write(str(new_select))
    temp_file.close()
```

```
        old_select = new_select


#-------------------------------------------------------------------------------

stat_lab = Label(app, bg = "grey", fg = "maroon", font = ("Arial", 14), text = "Stat")
stat_lab.grid(row = 9, column = 2)

select_lab = Label(app, bg = "grey", fg = "maroon", font = ("Arial", 14), text = "Sel Sys")
select_lab.grid(row = 9, column = 5)

#-------------------------------------------------------------------------------

temp_opt_men = OptionMenu(app,var_temp, "14","15","16","17","18","19","20","21","22","23")

temp_opt_men.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
temp_opt_men.grid(row = 10, column = 2, columnspan = 1, sticky = W)

select_opt_men = OptionMenu(app,var_select, "0", "1","2","3","4")

select_opt_men.config(font = ("Arial", 14), fg = "black", bg = "grey", width = 2)
select_opt_men.grid(row = 10, column = 5, columnspan = 1, sticky = W)

#-------------------------------------------------------------------------------

advance_lab1 = Label(app, bg = "gray", fg = "gray", text = "  A", font = ("Arial", 18))
advance_lab1.grid(row = 3, column = 13)

advance_lab2 = Label(app, bg = "gray", fg = "gray", text = "  A", font = ("Arial", 18))
advance_lab2.grid(row = 4, column = 13)

advance_lab3 = Label(app, bg = "gray", fg = "gray", text = "  A", font = ("Arial", 18))
advance_lab3.grid(row = 5, column = 13)

advance_lab4 = Label(app, bg = "gray", fg = "gray", text = "  A", font = ("Arial", 18))
advance_lab4.grid(row = 6, column = 13)

blank_lab2 = Label(app, bg = "gray")
blank_lab2.grid(row = 8, column = 1)

#-------------------------------------------------------------------------------

time_lab = Label(app , text = "Comp time...", font = ("Arial", 16), fg = "maroon", bg = "gray")
time_lab.grid(row = 1, column = 8, columnspan = 4, sticky = W)



#-------------------------------------------------------------------------------------------
get_time()

#-------------------------------------------------------------------------

# calls a function after given time
def after(self, ms, func = None, *args):
    """call function after a given time"""

#---------------------------------------------------------------------
# updates screen every 1 seconds
def task():

    # shows when system is advanced
    global advance1, advance2, advance3, advance4, all_on, ok


    if advance1 == True:
        advance_lab1.config(fg = "maroon")

    else:
        advance_lab1.config(fg = "gray")

    if advance2 == True:
        advance_lab2.config(fg = "maroon")

    else:
        advance_lab2.config(fg = "gray")

    if advance3 == True:
        advance_lab3.config(fg = "maroon")

    else:
        advance_lab3.config(fg = "gray")

    if advance4 == True:
        advance_lab4.config(fg = "maroon")

    else:
        advance_lab4.config(fg = "gray")

    if all_on:
```

```
        on_off_but.config(text = " ON", fg = "blue", bg = "red")

    else:
        on_off_but.config(text = "OFF", fg = "maroon", bg = "light blue")


    get_timed_sys()

    read_temperature()

    save_times()

    if ok:
        status_but.config(text = " OK ")

    root.after(1000, task)
#---------------------------------------------------------------------

# calls the screen update every 1 seconds
root.after(1000, task)

#-------------------------------------------------------------------------------------------
# kick off the window's event-loop
root.mainloop()
```