```python

#
# This programme relies upon being called at some frequency defined by a cr
# It will read the temperature from any number of DS18B20 sensors and write
# to a file whose name depends upon the year and month. Although it could b
# sophisticated by only writing when the temperature has changed, my expect
# it will be run about every 10 minutes (144 entries per day) and really th
# wouldn't be worthwhile. It also makes plotting the graph easier.
#
import subprocess
import os
import glob
import time
from datetime import datetime

#
# file_name is the place where the readings will be stored.
#
global file_name
#
base_dir = '/sys/bus/w1/devices/'
starttime = time.time()
filename = datetime.now().strftime("%Y%m")
file_name = '/home/pi/Documents/Python_2_Projects/t'+ filename + '.csv'
#
# base_dir defines where to find the devices to be read
#



#
#This section is a copy of the standard ways of reading the sensors
#
def read_temp_raw(device_file):
        f = open(device_file, 'r')
        lines = f.readlines()
        f.close()
        return lines
#
# Added modification to read_temp to deal with the circumstances where temp
# cannot be converted to a float number (e.g. 0)
#
def f_float(x):
        try:
                return float(x)
        except (ValueError, TypeError):
                return 0.0
#
# Read the given device and return the temperature in degrees C
#
def read_temp(device_file):
        lines = read_temp_raw(device_file)
        while lines[0].strip()[-3:] != 'YES':
                print ('failed', str(device_file))
                time.sleep(0.2)
                lines = read_temp_raw(device_file)
        equals_pos = lines[1].find('t=')
        if equals_pos != -1:
                temp_string = lines[1][equals_pos+2:-3]
                temp_c = f_float(temp_string) / 10.0
                temp_f = temp_c * 9.0 / 5.0 + 32.0
                return temp_c

# --------------------------------------------------------
#
# The following opens the months output file, then for each of the DS18B20s
# the existence of a file 28* in the /devices/ directory, it reads the temp
# writes the value to the file. It then writes a newline to the file and cl

file_object = open(file_name, "a", 1)
print_time = str(datetime.now())[:-7]
file_object.write(print_time)
for device in glob.glob(base_dir + '28*'):
        device_file = device + '/w1_slave'
        temp_temp = read_temp(device_file)
        file_object.write(", ")
        file_object.write(temp_temp)

file_object.write("\n")
file_object.close()
#



```