|  |
| --- |
| Technical Milestone 1 |
| Dylan StoneApplied Research Project 1October 27, 2016 |

One of the most important parts of my project will be the differential temperature readings. I need to be able to read in the temperature in and out of the solar heating system, so I can know how much the air has been heated. Because this is very important to my project, it is my first milestone. To measure temperature I have used a Raspberry Pi and two, one wire bus temperature sensors. The temperature sensors use the power from the Raspberry Pi to operate and have their data outputs shorted together with a 4.7kΩ resistor. In this configuration I can call the address of either temperature and read its temperature with a resolution of 1/16th of a degree Celsius.

My program begins by calling a function and passing the address of the first sensor to check the temperature of the first sensor, then it will call the same function and pass the second address to get the second temperature reading. Readings take about a second, so my program will light an LED while it is taking temperature readings. The temperature reading functions return a temperature value which is saved to float variables. The program will then pass these two variables to a function that will light an LED if the first temperature is higher than the second, this is for demonstrational purposes only. Next the program will run a function that concatenates a string containing the time and date this measurement was taken, and prints the measurements and date to the console. The program then opens a comma separated values file and appends the same data that was printed to the console window in a way that can be easily read. Because I am appending to the file, it will never overwrite previous data. The program then waits a certain amount of time, by default I have it waiting 10 seconds. After it has delayed the program will start over from the beginning.

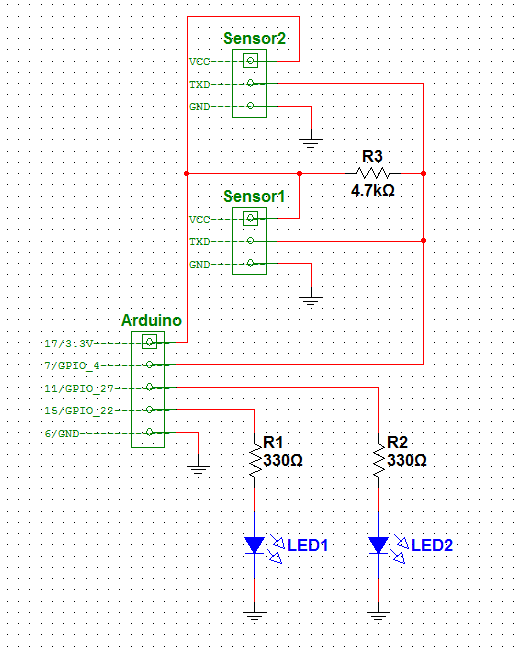
I plan to use a large portion of the code in my final project as I will need to take and log temperature measurements. This program will be expanded to include other measurements that I will need to take, and it will not need to print to the console window. I would like the program to be able to create graphs showing the efficiency of the solar heater over time, and this program is the first step to being able to do that. I have included the schematic and program code in this document. 

Figure 1

Machine generated alternative text:
E:\School\Project\Milestone 1\Program.py
Wednesday, October 26, 2016 6:31 PM
import
os
import
glob
import
time
import
RPi
.
GPIO
as
GPIO
os
.
system
(
'modprobe w1-gpio'
)
#sets up IO and temp readings
os
.
system
(
'modprobe w1-therm'
)
base_dir
=
'/sys/bus/w1/devices/'
#assigns variable to each temperature sensor 
address
device_file1
=
glob
.
glob
(
base_dir
+
'28-011592209dff'
)[
0
]+
'/w1_slave'
device_file2
=
glob
.
glob
(
base_dir
+
'28-0115922482ff'
)[
0
]+
'/w1_slave'
#flashes led on 
call                                            
def
flash
(
pin
,
state
,
wait
):
boo
=
not
state
GPIO
.
output
(
pin
,
state
)
time
.
sleep
(
wait
)
GPIO
.
output
(
pin
,
boo
)
#opens temperature sensor when passed address
def
read_temp_raw
(
d_file
):
f
=
open
(
d_file
,
'r'
)
lines
=
f
.
readlines
()
f
.
close
()
return
lines
#reads and returns the temperature
def
read_temp
(
d_file
):
lines
=
read_temp_raw
(
d_file
)
while
lines
[
0
].
strip
()[-
3
:]!=
'YES'
:
time
.
sleep
(
0.2
)
lines
=
read_temp_raw1
()
equals_pos
=
lines
[
1
].
find
(
't='
)
if
equals_pos
!=-
1
:
temp_string
=
lines
[
1
][
equals_pos
+
2
:]
temp_c
=
float
(
temp_string
)/
1000.0
return
temp_c
#function that lights an LED when temperature 1 
is higher than temperature 2
def
lighttemp
(
pin
,
t1
,
t2
):
if
t1
>
t2
:
GPIO
.
output
(
pin
,
1
)
else
:
GPIO
.
output
(
pin
,
0
)
def
results
():
#prints the results to the console
date
=(
str
((
time
.
localtime
()).
tm_hour
)+
":"
+
str
((
time
.
localtime
()).
tm_min
)+
":"
+
str
((
time
.
localtime
()).
tm_sec
)+
","
+
str
((
time
.
localtime
()).
tm_mday
)+
"/"
+
str
((
time
.
localtime
()).
tm_mon
)+
"/"
+
str
((
time
.
localtime
()).
tm_year
))
print
(
date
)
-1-


Machine generated alternative text:
E:\School\Project\Milestone 1\Program.py
Wednesday, October 26, 2016 6:31 PM
print
(
"Sersor 1 temperature:           "
,
round
(
temp1
,
3
),
"C"
)
print
(
"Sersor 2 temperature:           "
,
round
(
temp2
,
3
),
"C"
)
print
(
"Sensor temperature difference:  "
,
round
(
abs
(
temp1
-
temp2
),
3
),
"C\n\n"
)
return
date
while
1
:
GPIO
.
setmode
(
GPIO
.
BCM
)
#set up GPIO using BCM numbering
GPIO
.
setup
(
27
,
GPIO
.
OUT
)
GPIO
.
setup
(
22
,
GPIO
.
OUT
)
GPIO
.
output
(
27
,
1
)
#LED Lights while reading
temp1
=
read_temp
(
device_file1
)
flash
(
27
,
0
,
0.1
)
temp2
=
read_temp
(
device_file2
)
GPIO
.
output
(
27
,
0
)
lighttemp
(
22
,
temp1
,
temp2
)
date
=
results
()
#prints results and saves to file
with
open
(
'/home/pi/Desktop/Programs/Data1.csv'
,
'a'
)
as
f
:
tempdata
=
date
+
','
+
str
(
round
(
temp1
,
3
))+
','
+
str
(
round
(
temp2
,
3
))+
','
+
str
(
round
(
abs
(
temp1
-
temp2
),
3
))+
'\n'
f
.
write
(
tempdata
)
GPIO
.
cleanup
()
time
.
sleep
(
10
)
#delay to next read
-2-
