

Lab 1 Report

Timothy Dee, Brent Barth

October 8, 2015

Abstract

This report contains a description and analysis of the findings during Lab 1 of CprE 458, Real Time Systems. Contained herein is all relevant program code, and the results from having run this code.

1 Introduction

TODO

2 Part 1 - Code

2.1 High Performance Mode

TODO

2.2 Dynamic Frequency Scaling Mode I

TODO

2.3 Dynamic Frequency Scaling Mode II

TODO

2.4 Dynamic Frequency Scaling Mode Mixed

TODO

3 Part 2 - Data Analysis

3.1 Measuring energy consumption

TODO

3.2 Comparison and Analysis

TODO

4 Extra Part - Measuring Computation Time

TODO

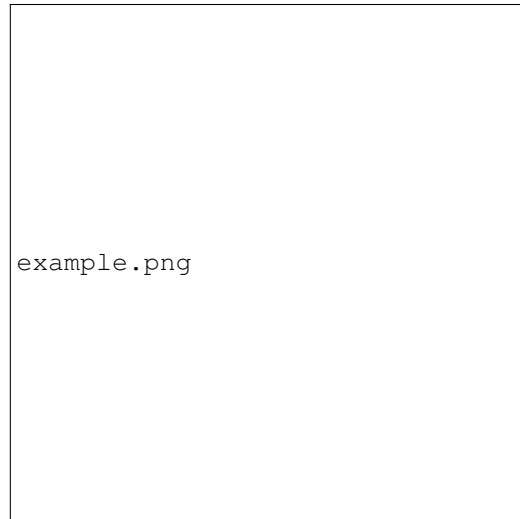


Figure 1: IR-Transmitter

$$\begin{aligned} \frac{V_{out} - V_-}{R2} - \frac{V_- - V_{in}}{R1} &= 0 \\ R1(V_{out} - V_-) &= R2(V_- - V_{in}) \\ V_{out} &= \frac{R1V_- + R2V_- - R2V_{in}}{R1} \\ V_- &= 0 \\ V_{out} &= \frac{-V_{in}R2}{R1} \\ \frac{V_{out}}{V_{in}} &= \frac{-R2}{R1} \end{aligned} \tag{1}$$

$$\frac{V_{out}}{V_{in}} = \frac{47000}{47} = 1000$$

Equation 1 demonstrates example equation.

Listing 1: High Performance Mode

```
1 PMbutton2.setOnClickListener(new View.OnClickListener() {
2
3     @Override
4     public void onClick(View v) {
5
6         OPERATIONmessage("[High Performance Mode]
7         #####");
8         //TODO Please program for High Performance Mode here (done)
9
10        DATAname = "1300000"; // Setting up the minimum frequency 1300 Mhz
11        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
12        ChangeCPUinfor(CPUname, DATAname, DATAaddress);
13        DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
14        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
15        ChangeCPUinfor(CPUname, DATAname, DATAaddress);
16
17        CPUname = "High Power";
18        DATAname = "current min frequency";
19        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
20        ReadCPUinfor(CPUname, DATAname, DATAaddress);
21        DATAname = "current MAX frequency";
22        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
23        ReadCPUinfor(CPUname, DATAname, DATAaddress);
24
25        OPERATIONmessage("[High Performance Mode]
26        #####");
27    }
28 });
```

Listing 2: Dynamic Frequency Scaling Mode I

```
1 private void setDFS_1(double cpu_load){
2     // determine if I should go to a power mode
3     if(cpu_load < .2){
4         // go to low performance mode
5         PMbutton1.performClick();
6     }else if(cpu_load > .9){
7         //go to high performance mode
8         PMbutton2.performClick();
9     }else{
10        // set the frequency range between 51 Mhz and 1.3 Ghz
11        DATAname = "51000"; // Setting up the minimum frequency 51 Mhz
12        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
13        ChangeCPUinfor(CPUname, DATAname, DATAaddress);
14        DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
15        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
16        ChangeCPUinfor(CPUname, DATAname, DATAaddress);
17
18        CPUname = "Dynamic Mode";
19        DATAname = "current min frequency";
20        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
21        ReadCPUinfor(CPUname, DATAname, DATAaddress);
22        DATAname = "current MAX frequency";
23        DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
24        ReadCPUinfor(CPUname, DATAname, DATAaddress);
25    }
26 }
```

Listing 3: Dynamic Frequency Scaling Mode II

```
1 private void setDFS_2(){
2     // get the battery level
3     IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
4     Intent batteryStatus = this.registerReceiver(null, ifilter);
5
6     int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
7     int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);
8
9     float battery_percent = level / (float)scale;
10
11    // get the wireless radio state
12    ConnectivityManager connectivityManager = (ConnectivityManager)
13        this.getSystemService(Context.CONNECTIVITY_SERVICE);
14
15    NetworkInfo network_info = connectivityManager.getActiveNetworkInfo();
16    boolean is_wireless_on = (network_info != null);
17
18    // get the charging state
19    int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
20    boolean is_charging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
21        status == BatteryManager.BATTERY_STATUS_FULL;
22
23    //(continued on next page)
```

Listing 4: Dynamic Frequency Scaling Mode II (continued)

```

1 // determine if we are charging
2 if(is_charging){
3     // if we are charging we don't care about power usage.... set to high performance
4     mode
5     PMbutton2.performClick();
6 }else if(battery_percent < .3) {
7     // we want to conserve energy because we are almost out of it, set to low
8     performance mode
9     PMbutton1.performClick();
10 }else{
11     // if we're not charging and not low battery we have some decisions to make
12     // if the wireless radio is on we are likely doing something online.
13     // If we are doing something we will like a more responsive device
14     // so allow the processor to vary between two fairly high frequency states
15     if (is_wireless_on) {
16         // set to fairly high processing state
17         // set the frequency range between 700 Mhz and 1.3 Ghz
18         DATAname = "700000"; // Setting up the minimum frequency 51 Mhz
19         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
20         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
21         DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
22         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
23         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
24
25         CPUname = "Dynamic Mode";
26         DATAname = "current min frequency";
27         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
28         ReadCPUinfor(CPUname, DATAname, DATAaddress);
29         DATAname = "current MAX frequency";
30         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
31         ReadCPUinfor(CPUname, DATAname, DATAaddress);
32     } else {
33         // set to lower processing state
34         // set the frequency range between 51 Mhz and 700 Mhz
35         DATAname = "51000"; // Setting up the minimum frequency 51 Mhz
36         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
37         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
38         DATAname = "700000"; // Setting up the maximum frequency at 1300 MHz
39         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
40         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
41
42         CPUname = "Dynamic Mode";
43         DATAname = "current min frequency";
44         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
45         ReadCPUinfor(CPUname, DATAname, DATAaddress);
46         DATAname = "current MAX frequency";
47         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
48         ReadCPUinfor(CPUname, DATAname, DATAaddress);
49     }
50 }
51 }

```

Listing 5: Dynamic Frequency Scaling Mode Mixed

```
1 private void setDFS_Mixed(double cpu_load) {
2     // get the battery level
3     IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
4     Intent batteryStatus = this.registerReceiver(null, ifilter);
5
6     int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
7     int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);
8
9     float battery_percent = level / (float)scale;
10
11     // get the charging state
12     int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
13     boolean is_charging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
14         status == BatteryManager.BATTERY_STATUS_FULL;
15
16     // determine if we are charging
17     if(is_charging) {
18         // if we are charging we don't care about power usage.... set to high performance
19         // mode
20         PMbutton2.performClick();
21     } else {
22         // vary the processor frequency based on the battery level.
23         // the higher the battery level, the higher the frequency.
24         // anywhere from 100000 khz to 1300000 khz
25         double step_size = 100.0/13.0;
26         int load_fraction = new Double(Math.ceil(battery_percent / step_size)).intValue();
27
28         // load_fraction is at greatest 1
29         int high_frequency = 100000 * Integer.valueOf(load_fraction * 13);
30         int low_frequency = high_frequency;
31
32         // set to fairly high processing state
33         // set the frequency range between 700 Mhz and 1.3 Ghz
34         DATAname = String.valueOf(low_frequency); // Setting up the minimum frequency at low
35         // frequency
36         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
37         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
38         DATAname = String.valueOf(high_frequency); // Setting up the maximum frequency at
39         // high frequency
40         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
41         ChangeCPUinfor(CPUname, DATAname, DATAaddress);
42
43         CPUname = "Dynamic Mode";
44         DATAname = "current min frequency";
45         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
46         ReadCPUinfor(CPUname, DATAname, DATAaddress);
47         DATAname = "current MAX frequency";
48         DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
49         ReadCPUinfor(CPUname, DATAname, DATAaddress);
50     }
51 }
```