# Lab 1 Report

Timothy Dee, Brent Barth

October 8, 2015

## Abstract

This report contains a description and analysis of the findings durring Lab 1 of CprE 458, Real Time Systems. Contained herein is all relevent 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

$$\begin{aligned} \frac{V_{out} - V_{-}}{R2} - \frac{V_{-} - V_{in}}{R1} &= 0 \\ R1(V_{out} - V_{-}) &= R2(V_{-} - V_{in}) \\ V_{out} &= \frac{R1V_{-} + R2V_{-} - R2Vin}{R1} \\ V_{-} &= 0 \\ V_{out} &= \frac{-V_{in}R2}{R1} \\ \frac{V_{out}}{V_{in}} &= \frac{-R2}{R1} \end{aligned} \tag{1}$$

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

Equation 1 demonstrates example equation.

Listing 1: High Performance Mode

```
PMbutton2.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View v) {

      OPERATIONmessage("[High Performance Mode]
          ###########################################");
      //TODO Please program for High Performance Mode here (done)

      DATAname = "1300000"; // Setting up the minimum frequency 1300 Mhz
      DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
      ChangeCPUinfor(CPUname, DATAname, DATAaddress);
      DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
      DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
      ChangeCPUinfor(CPUname, DATAname, DATAaddress);

      CPUname = "High Power";
      DATAname = "current min frequency";
      DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
      ReadCPUinfor(CPUname, DATAname, DATAaddress);
      DATAname = "current MAX frequency";
      DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
      ReadCPUinfor(CPUname, DATAname, DATAaddress);

      OPERATIONmessage("[High Performance Mode]
          ###########################################");


    }
   });
```

Listing 2: Dynamic Frequency Scaling Mode I

```
private void setDFS_1(double cpu_load){
   // determine if I should go to a power mode
   if(cpu_load < .2){
     // go to low performance mode
     PMbutton1.performClick();
   }else if(cpu_load > .9){
     //go to high performance mode
     PMbutton2.performClick();
   }else{
     // set the frequency range between 51 Mhz and 1.3 Ghz
     DATAname = "51000"; // Setting up the minimum frequency 51 Mhz
     DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
     ChangeCPUinfor(CPUname, DATAname, DATAaddress);
     DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
     DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
     ChangeCPUinfor(CPUname, DATAname, DATAaddress);

     CPUname = "Dynamic Mode";
     DATAname = "current min frequency";
     DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
     ReadCPUinfor(CPUname, DATAname, DATAaddress);
     DATAname = "current MAX frequency";
     DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
     ReadCPUinfor(CPUname, DATAname, DATAaddress);
   }
  }
```

Listing 3: Dynamic Frequency Scaling Mode II

```
private void setDFS_2(){
   // get the battery level
   IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
   Intent batteryStatus = this.registerReceiver(null, ifilter);

   int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
   int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);

   float battery_percent = level / (float)scale;

   // get the wireless radio state
   ConnectivityManager connectivityManager = (ConnectivityManager)
      this.getSystemService(Context.CONNECTIVITY_SERVICE);

   NetworkInfo network_info = connectivityManager.getActiveNetworkInfo();
   boolean is_wireless_on = (network_info != null);

   // get the charging state
   int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
   boolean is_charging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
      status == BatteryManager.BATTERY_STATUS_FULL;

   //(continued on next page)
```

Listing 4: Dynamic Frequency Scaling Mode II (continued)

```
// determine if we are charging
   if(is_charging){
    // if we are charging we don't care about power usage.... set to high performance mode
    PMbutton2.performClick();
   }else if(battery_percent < .3) {
    // we want to conserve energy because we are almost out of it, set to low performance mode
    PMbutton1.performClick();
   }else{
      // if we're not charging and not low battery we have some decisions to make
      // if the wireless radio is on we are likely doing something online.
      // If we are doing something we will like a more responsive device
      // so allow the processor to vary between two fairly high frequency states
      if (is_wireless_on) {
       // set to fairly high processing state
       // set the frequency range between 700 Mhz and 1.3 Ghz
       DATAname = "700000"; // Setting up the minimum frequency 51 Mhz
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
       ChangeCPUinfor(CPUname, DATAname, DATAaddress);
       DATAname = "1300000"; // Setting up the maximum frequency at 1300 MHz
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
       ChangeCPUinfor(CPUname, DATAname, DATAaddress);

       CPUname = "Dynamic Mode";
       DATAname = "current min frequency";
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
       ReadCPUinfor(CPUname, DATAname, DATAaddress);
       DATAname = "current MAX frequency";
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
       ReadCPUinfor(CPUname, DATAname, DATAaddress);
      } else {
       // set to lower processing state
       // set the frequency range between 51 Mhz and 700 Mhz
       DATAname = "51000"; // Setting up the minimum frequency 51 Mhz
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
       ChangeCPUinfor(CPUname, DATAname, DATAaddress);
       DATAname = "700000"; // Setting up the maximum frequency at 1300 MHz
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
       ChangeCPUinfor(CPUname, DATAname, DATAaddress);

       CPUname = "Dynamic Mode";
       DATAname = "current min frequency";
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
       ReadCPUinfor(CPUname, DATAname, DATAaddress);
       DATAname = "current MAX frequency";
       DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
       ReadCPUinfor(CPUname, DATAname, DATAaddress);
      }
   }
  }
```

Listing 5: Dynamic Frequency Scaling Mode Mixed

```
private void setDFS_Mixed(double cpu_load) {
   // get the battery level
   IntentFilter ifilter = new IntentFilter(Intent.ACTION_BATTERY_CHANGED);
   Intent batteryStatus = this.registerReceiver(null, ifilter);

   int level = batteryStatus.getIntExtra(BatteryManager.EXTRA_LEVEL, -1);
   int scale = batteryStatus.getIntExtra(BatteryManager.EXTRA_SCALE, -1);

   float battery_percent = level / (float)scale;

   // get the charging state
   int status = batteryStatus.getIntExtra(BatteryManager.EXTRA_STATUS, -1);
   boolean is_charging = status == BatteryManager.BATTERY_STATUS_CHARGING ||
      status == BatteryManager.BATTERY_STATUS_FULL;

   // determine if we are charging
   if(is_charging){
    // if we are charging we don’t care about power usage.... set to high performance mode
    PMbutton2.performClick();
   }else {
    // vary the processor frequency based on the battery level.
    // the higher the battery level, the higher the frequency.
    // anywhere from 100000 khz to 1300000 khz
    double step_size = 100.0/13.0;
    int load_fraction = new Double(Math.ceil(battery_percent / step_size)).intValue();

    // load_fraction is at greatest 1
    int high_frequency = 100000 * Integer.valueOf(load_fraction * 13);
    int low_frequency = high_frequency;

    // set to fairly high processing state
    // set the frequency range between 700 Mhz and 1.3 Ghz
    DATAname = String.valueOf(low_frequency); // Setting up the minimum frequency at low frequency
    DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
    ChangeCPUinfor(CPUname, DATAname, DATAaddress);
    DATAname = String.valueOf(high_frequency); // Setting up the maximum frequency at high frequency
    DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
    ChangeCPUinfor(CPUname, DATAname, DATAaddress);

    CPUname = "Dynamic Mode";
    DATAname = "current min frequency";
    DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq";
    ReadCPUinfor(CPUname, DATAname, DATAaddress);
    DATAname = "current MAX frequency";
    DATAaddress = "/sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq";
    ReadCPUinfor(CPUname, DATAname, DATAaddress);
   }
  }
```