

# Low Power Computing Assignment #1

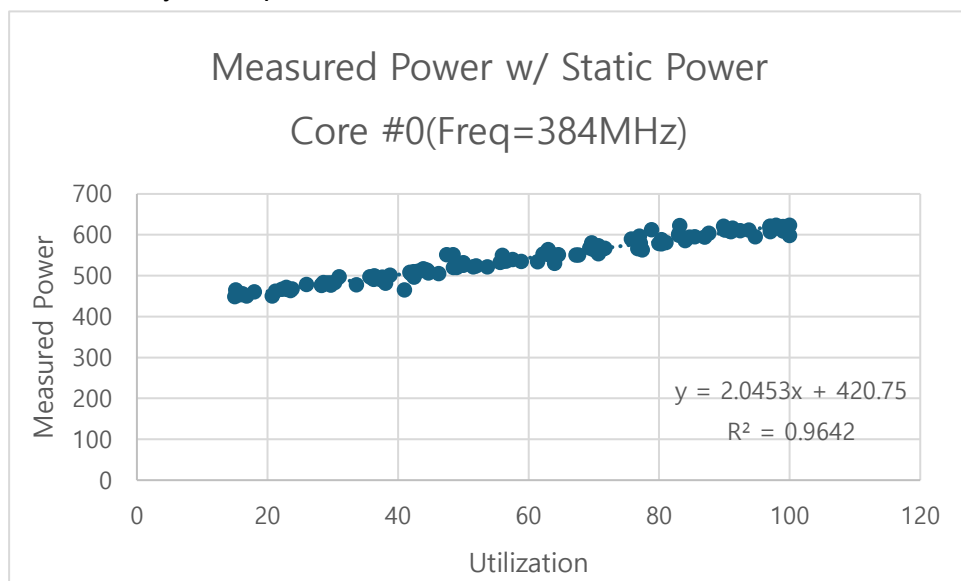
컴퓨터학과 2020320026 권도혁

## 1. Power Models

### A. CPU Power Model

CPU의 Power Model을 만들기 위해서 Core #0부터 Core #3까지 각각의 주어진 frequency에 따라 각각 선형 회귀법을 사용하여  $y = ax + b$ 의 식을 이끌어 낸 후 static power를 빼서 각 core의 dynamic power를 구했다.

우선, 각 core의 utilization에 따른 measured power의 관계를 관찰하기 위해 오름차순 정렬된 utilization을 x축에, 그에 따른 measured power를 y축에 두고 분산형 그래프를 구한 다음 선형 회귀법을 사용하여  $\beta = ax + b$ 의 식을 이끌어 내서  $\beta$ 를 구한 후,  $\beta$ 의 절편에 static power를 빼서 특정 frequency에서 core의 dynamic power를 구했다.



Core #0의 dynamic power는 Core #0의 각각의 frequency마다  $\beta$ 를 구한 후, 각각의  $\beta$ 마다 386.560959752475이라는 주어진 static power를 빼는 것으로 구할 수 있다.

Core1, 2, 3을 구하기 위해서는 주어진 수식을 사용하여 구한다.

$$P_{measured-core\#n} = P_{core0} + P_{core\#n} + P_{LCD}$$

우변을 보면, Core #0과 Core #n의 파워와 LCD의 파워를 더한 값인데, Core #0과 #n의 파워는 dynamic + static이기 때문에 우변을 다음과 같이 나타낼 수 있다.

$$P_{dynamic\#0} + P_{static\#0} + P_{dynamic\#n} + P_{static\#n} + P_{LCD}$$

여기서 Core #0과 #n의 static power와 P\_LCD를 더한 값은 static power 파일에 제공되었고, P\_dynamic#0은 core #n 파일에서 core #0의 utilization이 0이기 때문에 0이 됨을 알 수 있다. 따라서 P\_dynamic#n은 P\_measured-core#n에서 위의 값을 다 빼는 것으로 얻을 수 있다. 각 코어의 dynamic power는 다음과 같다.

$$P_{dynamic\#0} = P_{measured\#0} - 386.560959752475$$

$$P_{dynamic\#1} = P_{measured\#1} - 511.233218058036$$

$$P_{dynamic\#2} = P_{measured\#2} - 525.502686756302$$

$$P_{dynamic\#3} = P_{measured\#3} - 524.731190736221$$

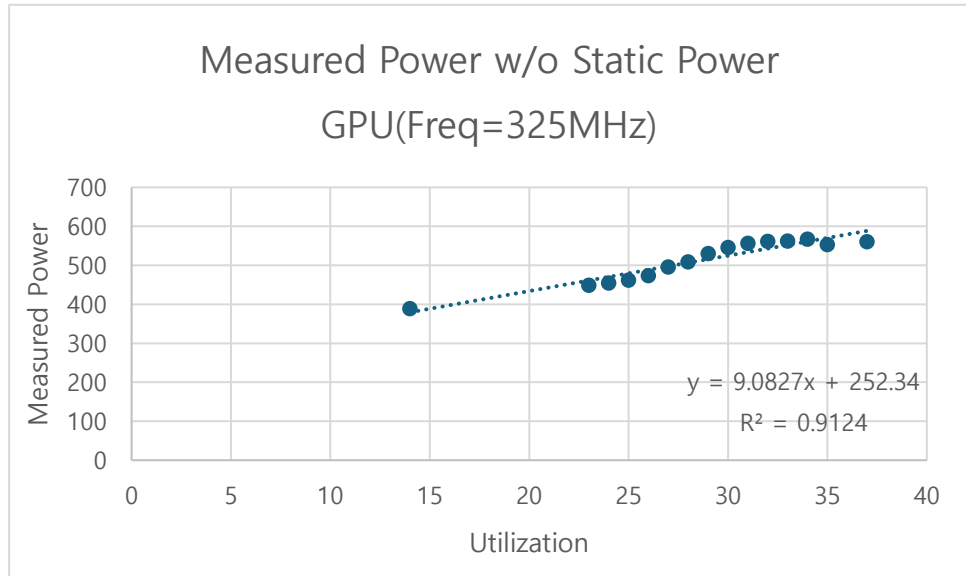
따라서 CPU의 Power Model은 다음과 같다.

$$P_{CPU} = \sum_{i=1}^n \{ \beta_{core\#i}(Freq_{core\#i}) \times Util_{core\#i} + P_{static\#i} \times S_{core\#i} \}$$

Freq\_core#i는 Core #i의 frequency, Util\_core#i는 Core #i의 utilization, P\_static#i는 Core #i의 static power, S\_core#i는 Core #i가 켜져 있을 때 1, 꺼져 있을 때 0이다. 또한 대괄호 안에서 덧셈 기호의 좌측에 있는 식이 Core #i의 dynamic power, 우측에 있는 식이 Core #i의 static power가 된다.

## B. GPU Power Model

GPU는 GPU utilization에 따른 measured power의 관계를 관찰했다. 각각의 frequency마다 GPU utilization을 10을 단위로 10~19, 20~29, 혹은 1을 단위로 1, 2, 3 이렇게 증가하면서 구간을 나누고 구간의 평균을 내어 해당 구간의 measured power의 평균을 계산한 다음 선형 회귀법을 사용하여  $\beta$ 를 계산하였다.



하지만 GPU Power Model의  $\beta$  값은 제공된 값을 사용하기로 했다.

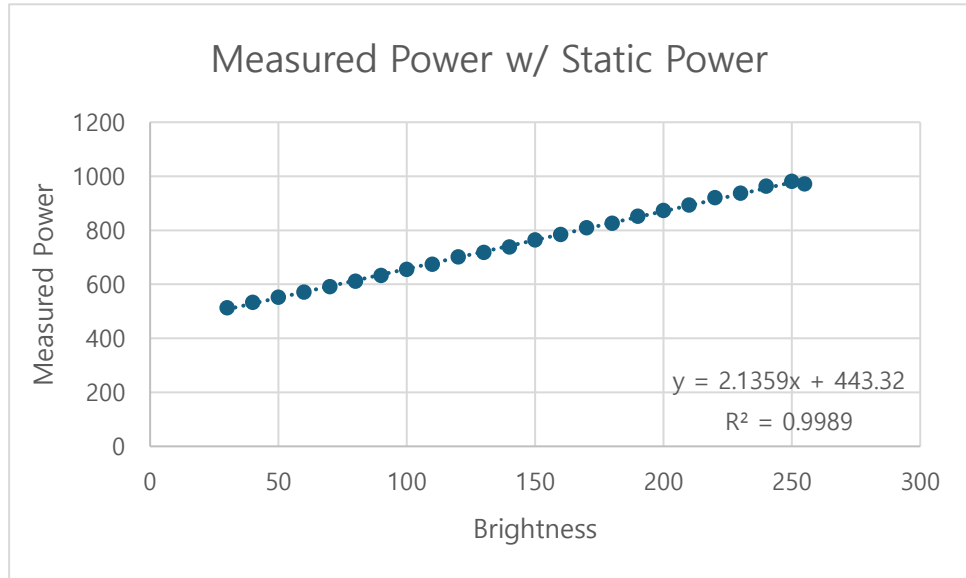
GPU Power Model은 다음과 같다.

$$P_{GPU} = (\beta_{freq} \times Util_{GPU} + \beta_{static}) \times S_{GPU}$$

$\beta_{freq}$ 은 GPU의 frequency,  $Util_{GPU}$ 는 GPU의 utilization,  $\beta_{static}$ 은 GPU의 static power,  $S_{GPU}$ 는 GPU가 켜져 있을 때 1, 꺼져 있을 때 0이다. 소괄호 안에서 덧셈 기호의 좌측의 식은 GPU의 dynamic power, 우측의 식은 GPU의 static power가 된다.

### C. Display Power Model

Display는  $\beta$ 를 구하기 위해 brightness에 따른 measured power의 관계를 관찰했다. Brightness마다 measured power의 평균을 구하고 선형 회귀법을 사용하였다.



$\beta$ 의 절편에서 386.560959752475(=  $P_{core\#0} + P_{LCD}$ )를 빼면 Display Power Model의  $\beta$ 를 얻을 수 있다. 또한 display의 최소 power인  $P_{LCD}$ (=329.8709598)를 더해줌으로써 Power Model을 구성할 수 있다. 따라서 Display Power Model은 다음과 같다.

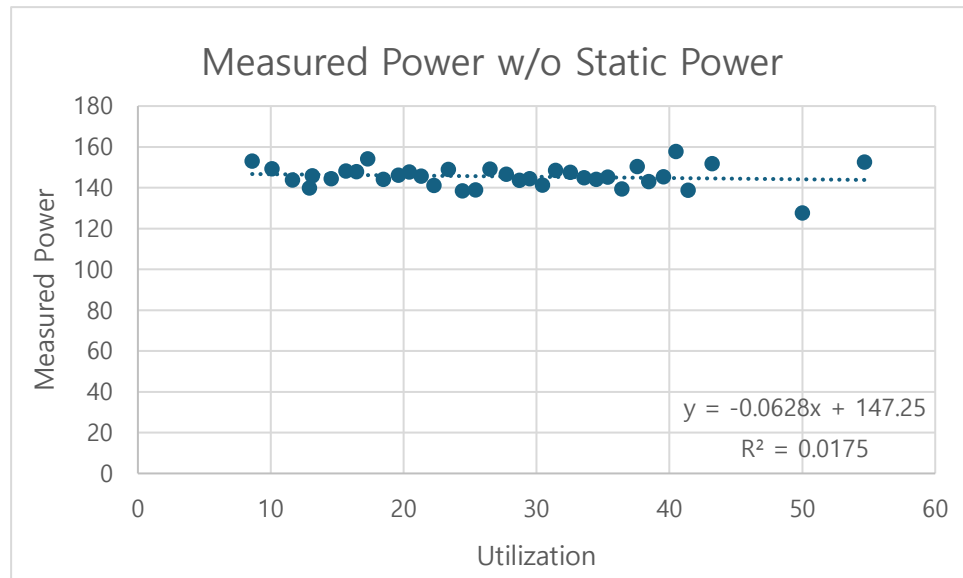
$$P_{Display} = (2.1359 \times brightness + 56.76 \times S_{lcd} + 329.8709598) \times S_{lcd}$$

$S_{LCD}$ 는 Display가 켜져 있으면 1, 꺼져 있으면 0이다.

### D. Audio Power Model

Audio는 Core #0의 utilization에 따른 Measured power의 관계를 관찰했다. Utilization의 구간을 정해서 구간의 평균을 내고 이에 해당하는 measured power의 평균을 사용하여 선형 회귀법을 적용한 결과, utilization에 따른 measured power의 비례 혹은 반비례 관계가 나타나지 않고 measured

power가 대체적으로 일정하게 나와서 measured power의 산술평균을  $\beta$ 로 사용하기로 했다.



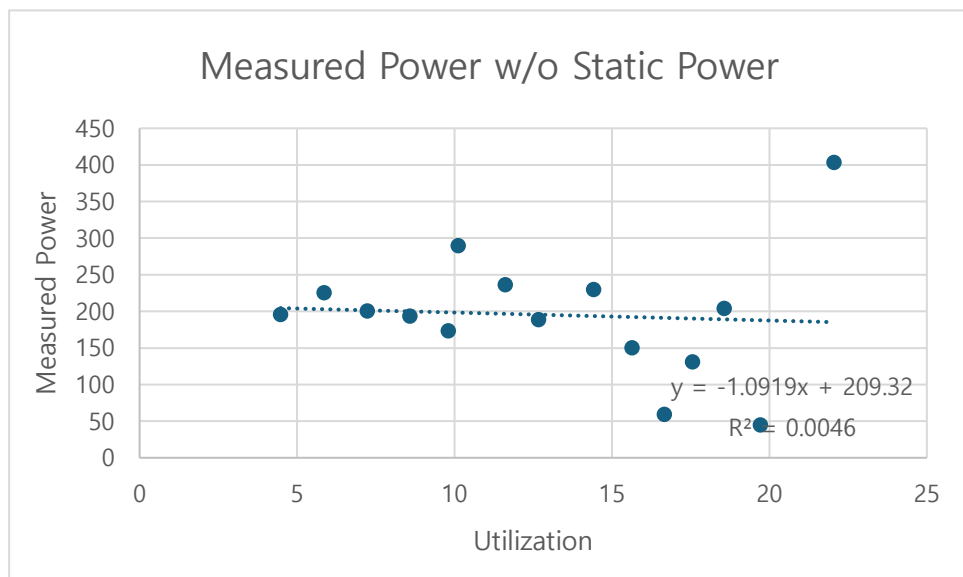
따라서 Audio Power Model은 다음과 같다.

$$P_{Audio} = 145.5304 \times S_{audio}$$

$S_{audio}$ 는 audio가 켜져 있으면 1, 꺼져 있으면 0이다.

#### E. GPS Power Model

GPS 또한 Core #0의 utilization에 따른 measured power의 관계를 보려고 했지만 Audio와 마찬가지로 관계성이 뚜렷하게 보이지 않아서 산술평균을  $\beta$ 로 사용하기로 했다.



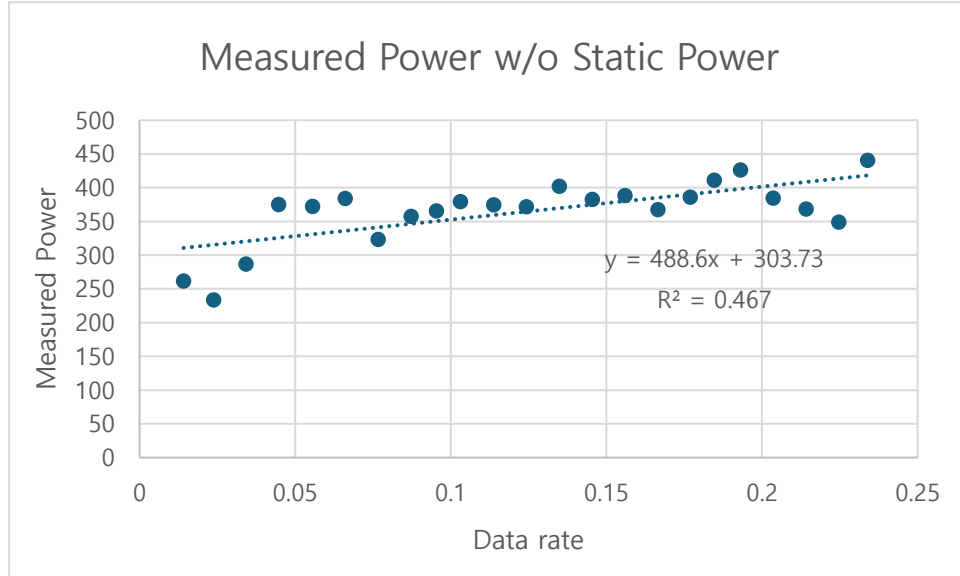
따라서 GPS Power Model은 다음과 같다.

$$P_{GPS} = 206.3579 \times S_{GPS}$$

$S_{GPS}$ 는 GPS가 켜져 있으면 1, 꺼져 있으면 0이다.

## F. Wi-Fi Power Model

Wi-Fi의 경우 일단 WiFi 모드가 low인 경우와 high인 경우 둘로 데이터를 나누고 각각의 모드에서 data rate에 따른 measured power의 관계를 관찰했다. 먼저 high 모드인 경우, Data rate를 0.01 단위로 증가할 때마다 data rate의 평균을 내고 그 사이 구간에 해당하는 measured power의 평균을 사용하여 선형 회귀법을 적용한 결과 어느 정도의 비례 관계를 확인하여 이를  $\beta$ 로 사용하기로 했다.



다음으로 low 모드인 경우, high 모드와 같은 방법으로 진행하려고 했지만 데이터의 양도 적고, 관계성도 불분명하며 대체적으로 비슷한 값을 보여주고 있기 때문에 measured power의 산술평균을  $\beta$ 로 사용하기로 했다.

따라서 Wi-Fi Power Model은 다음과 같다.

$$P_{Wi-Fi} = (488.6 * (data_{rate}) + 303.73 + (58 - 0.768 * channel_{rate}) * data_{rate}) * S_{wifi_h} + 162.0251 * S_{wifi_l}$$

$S_{wifi_h}$ 는 high 모드일 때 1, low 모드일 때 0이고,  $S_{wifi_l}$ 은 low 모드일 때 1, high 모드일 때 0이다.

## 2. Evaluation

위에서 구한 Power Model을 사용하여 아래 3개의 workload에 대한 power consumption을 계산했다. 계산을 위해 엑셀에서 사용자 지정 함수를 만들어서 사용하였다.

CPU Core의 frequency가 0이 아니면 해당 코어는 켜져 있다고 판단하고 dynamic과 static power를 계산했고, 나머지 유닛들은 모두 켜져 있다고 보고 power consumption을 계산했다.

사용자 정의 함수의 구현은 CPU의 경우, 각 core마다 사용자 정의 함수를 만

들어서 frequency와 utilization을 parameter로 입력해서 frequency에 따른 power model을 적용하여 core의 dynamic power를 계산했다. GPU의 경우, 각 frequency마다 사용자 정의 함수를 만들어서 utilization을 parameter로 입력해서 frequency에 따른 power model을 적용하여 dynamic + static power를 한꺼번에 계산했다.

Display는 brightness를 parameter로 입력 받는 사용자 정의 함수를 만들었고, Wi-Fi의 경우 3개의 workload 모두 low 모드이고 high 모드의 power model에서 사용되는 data rate와 channel rate의 값이 없기 때문에 사용자 지정 함수에서 모드를 입력 받아서 low 모드의  $\beta$ 값을 계산하도록 했다.

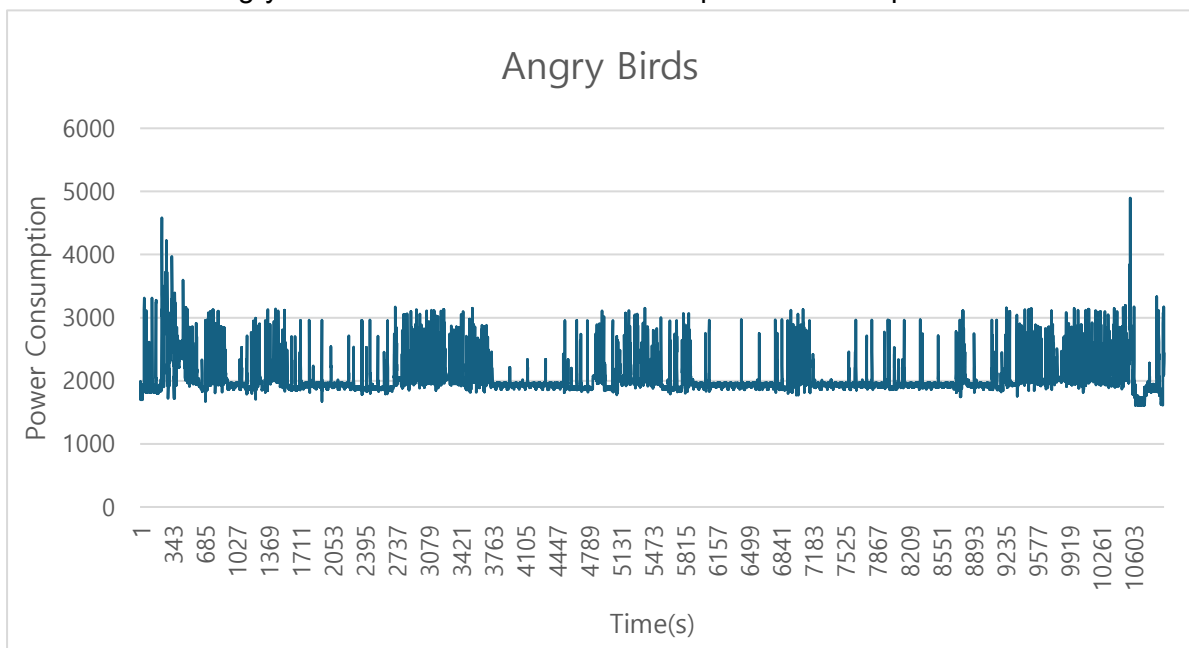
Audio, GPS의 경우 항상 켜져 있다고 보기 때문에 power model을 만들 때 계산한  $\beta$ 값을 사용했다.

Static power의 경우 core 1, 2, 3의 frequency를 입력 받아서 0보다 큰 frequency를 가진 core를 찾아서 어떤 core가 켜져 있을 때 어떤 static power인지 제공된 파일에 있는 값을 반환하게 하는 사용자 정의 함수를 만들었다.

이렇게 계산된 모든 값을 더한 값이 최종 power consumption이 된다.

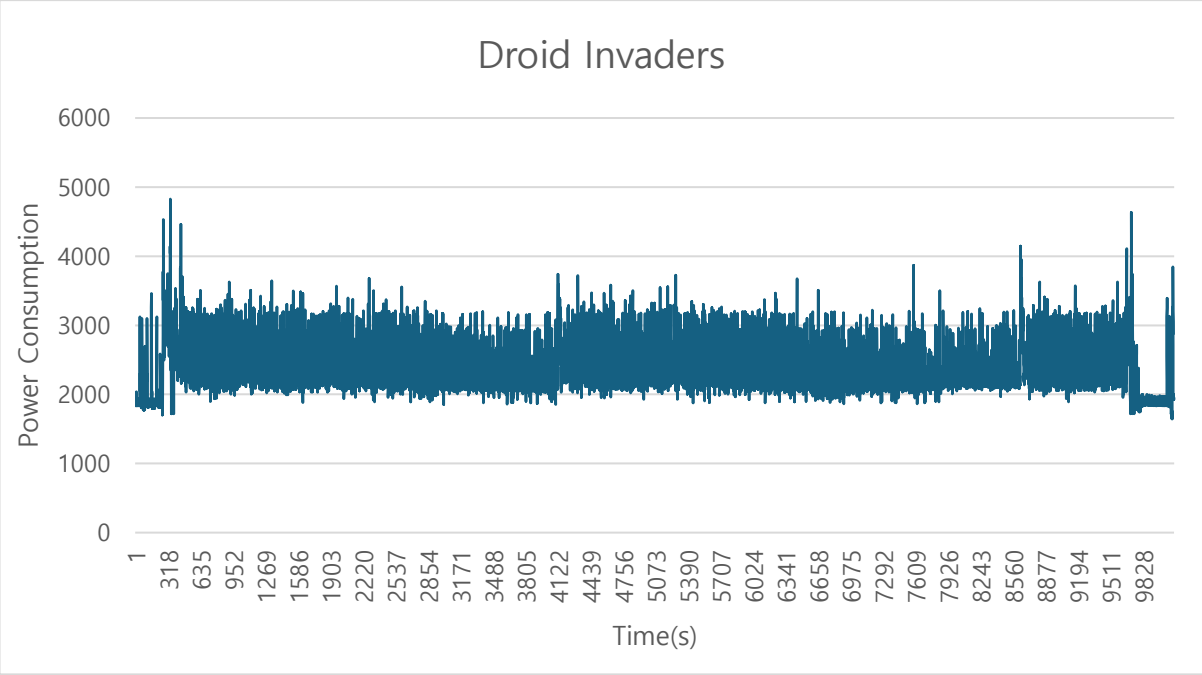
#### A. Angry Birds

Angry Birds workload의 시간에 따른 power consumption은 다음과 같다.



B. Droid Invaders

Droid Invaders workload의 시간에 따른 power consumption은 다음과 같다.



C. Pie3dDemo

Pie3dDemo workload의 시간에 따른 power consumption은 다음과 같다.

