


2차년도 기술문서

(과제명) 대규모 분산 에너지 저장장치 인프라의 안전한
자율운영 및 성능 평가를 위한 지능형 SW
프레임워크 개발

(과제번호) 2021-0-00077

- 결과물명 : BMS 에너지원 및 리소스 데이터
수집 모듈 개발 및 측정 데이터 분석
결과 보고서
- 작성일자 : 2022년 12월 8일

과학기술정보통신부 SW컴퓨팅산업원천기술개발사업
“기술문서”로 제출합니다.

수행기관	성명/직위	확인
슈어소프트테크(주)	심정민 / 이사	

정보통신기획평가원장 귀하

목차

1. 개요
2. BMS의 에너지원 및 시스템 리소스 수집 모듈 개발
 - 2.1 BMS 에너지원 측정 데이터
 - 2.2 BMS 시스템 리소스 측정 모듈 제작 및 삽입
 - 2.3 BMS 에너지원 및 시스템 리소스 측정 데이터 수집
3. BMS의 에너지원 및 시스템 리소스 데이터 시각화
 - 3.1 수집된 데이터 개요
 - 3.2 수집 데이터 전처리 및 시각화

<표 차례>

표 1 측정된 Rack 시험 데이터 범례

<그림 차례>

- 그림 1 MCU Max Runtime 측정 코드
그림 2 Stack Memory Usage 측정 코드
그림 3 MCU Reference Voltage 측정 코드
그림 4 MCU Temperature 측정 코드
그림 5 System Clock Register
그림 6 인셀 테스트베드 환경
그림 7 Rack BMS 테스트베드
그림 8 측정된 Rack 시험 데이터
그림 9 데이터 처리 전 시각화 그래프
그림 10 데이터 처리 후 시각화 그래프

1. 개요

배터리관리시스템(BMS)의 에너지원 데이터는 현재 테스트베드 및 운영 사이트에서 매 초마다 지속적으로 수집하고 있다. 따라서 다음 기술할 내용은 배터리관리시스템의 시스템 리소스 측면의 데이터를 측정하기 위한 모듈 개발 및 코드 탐침을 통한 시스템 리소스 데이터 측정 및 수집 과정을 다룬다.

2. BMS의 에너지원 및 시스템 리소스 수집 모듈 개발

2.1 BMS 에너지원 측정 데이터

BMS의 수행 기능 중에는 Battery Cell의 충/방전 정보 및 셀의 상태를 측정하는 기능과 이를 통하여 Battery Cell의 SoC를 계산하는 기능이 있다. 따라서 BMS 테스트베드를 통한 시험 수행 시 Rack BMS의 현재 전압 및 전류, 온도 데이터와 SoC 데이터, 셀 간의 전압 편차 및 온도 편차를 얻을 수 있다. 충/방전 시험을 통하여 다음에 해당 BMS의 시스템 리소스 측정 데이터와 함께 BMS의 측정 기능으로 얻은 에너지원 데이터를 확보하였다.

2.2 BMS 시스템 리소스 측정 모듈 개발 및 삽입

소프트웨어 코드는 의도치 않은 결함이 발생할 수 있으며 여러 산업분야에서 소프트웨어 결함으로 인한 크고 작은 사고가 여러 번 발생했었다. 따라서, BMS 테스트베드에서 측정할 데이터로 MCU Runtime, Stack Usage, MCU Vref, MCU Temperature 및 System Clock로 정하고 해당 리소스를 측정하기 위한 코드를 다음과 같이 테스트베드 BMS의 소프트웨어에 탐침하였다.

A. MCU Max Runtime

BMS를 포함한 임베디드 시스템은 설계 단계에서 명시한 동작 시간을 준수하도록 하는데, 소프트웨어 결함으로 인하여 Task의 지연이 발생한다면 기능 동작에 fault가 발생하여 BMS 시스템이 셀의 정보를 읽어들이거나 셀의 충/방전을 제어하는 데 문제가 생겨 ESS의 화재 사고가 발생할 수 있다. 이를 방지하기 위하여 동작 시간을 일정하기 유지하는 것이 중요하다.

임베디드 시스템은 하나의 클럭을 수행하기 위해서는 최소한 하나 이상의 클럭을 필요로 한다. 이 점을 이용하여 시작 지점으로부터 마지막 지점까지의 구동된 클럭 수를 계산하여 MCU의 동작 시간을 구할 수 있다. 해당 방법을 사용하여 다음과 같이 MCU의 동작 시간 계산 구문을 삽입하였다.

```
> /**...
uint32_t start_measure_runtime(void)
{
    start_timer();
    uint32_t start_time = get_timer();
    return start_time;
}

> /**...
uint32_t stop_measure_runtime(uint32_t start_time)
{
    uint32_t end_time = get_timer();
    uint32_t runtime = end_time - start_time;
    stop_timer();
    if(runtime > max_runtime){
        max_runtime = runtime;
    }
    return runtime;
}
```

그림 2 MCU Max Runtime 측정 코드

B. Stack Usage

메모리는 크게 코드 영역, 데이터 영역과 스택 영역으로 나뉜다. 코드 영역은 소프트웨어의 코드

가 저장되는 영역이고 데이터 영역은 소프트웨어의 전역 변수와 정적 변수가 저장되는 공간이다. 스택 영역은 return data address 등의 local 변수가 저장되는 공간이다.

스택 메모리는 다른 영역과 달리 크기가 고정적이지 않고 동작에 따라 가변적으로 변한다. 메모리 크기 자체가 크고 유동적으로 추가도 가능한 범용 시스템에 비해 임베디드 시스템은 크기가 매우 작은데다가 용량 확장도 불가능하다. 이는 소프트웨어 결함이 발생하여 비정상적으로 메모리 사용량이 늘어날 경우 BMS 등 임베디드 시스템에서는 치명적인 동작 오류가 발생할 수 있으며 따라서 메모리 영역의 관리가 더욱 중요해지며 주기적인 메모리 관리를 통하여 여유 공간을 확보해줘야 한다.

해당 메모리를 측정하기 위해서는 스택 메모리의 시작 주소 및 현재까지 사용한 영역의 주소를 구하여 해당 주소 위치의 차를 통하여 스택 메모리의 사이즈를 구할 수 있다. 해당 방식을 사용하여 테스트베드 MCU 동작 코드에 다음 구문을 삽입하였다.

```
uint8_t get_stack_use(void)
{
    uint32_t stack_pt = __get_SP();
    uint32_t stack_use = (stack_limit - stack_pt);
    uint8_t stack_percent = (uint32_t)(stack_use / stack_size * 100u);

    if(stack_percent > max_stackuse)
    {
        max_stackuse = stack_percent;
    }

    return (uint8_t)stack_percent;
}
```

그림 3 Stack Memory Usage 측정 코드

C. MCU VRef

임베디드 시스템의 공급 전력은 고정 값이 아니며 미세하게 공급 전압이 변동된다. 공급 전압이 불안정해지면 MCU 및 해당 공급 전력을 사용하는 주변 센서에도 영향을 미치며 ESS 시스템의 동작에 이상이 발생할 수 있다. 따라서 MCU의 공급 전력을 시스템 리소스 측정 요소로 선정하였다.

MCU의 공급 전압은 내부에서 별도로 측정되고 있으며 ADC를 통하여 해당 데이터의 측정값을 불러올 수 있다. 해당 기능을 사용하여 BMS 테스트베드에 다음과 같이 MCU의 공급 전압 측정 구문을 삽입하였다.

```
float get_reference_voltage(void)
{
    HAL_ADC_Start(&hadc1);
    while(HAL_ADC_PollForConversion(&hadc1, 1000) != HAL_OK);
    uint16_t adc_val = HAL_ADC_GetValue(&hadc1);
    float vref = (float)(1.21 * 4095) / adc_val;
    return vref;
}
```

그림 4 MCU Reference Voltage 측정 코드

D. MCU Temp

ESS의 배터리 충/방전 환경에서는 시스템의 발열이 발생할 가능성이 있으며 MCU가 동작하면서 자체 발열도 있다. 따라서 고온 환경이 발생하여 시스템 동작에 이상이 생기면 BMS 시스템의 셀 충/방전에도 영향을 끼칠 수 있다. 해당 조건 발생 여부와 발생 시 영향 확인을 위하여 MCU의 온도를 측정 요소로 선정하였다. MCU의 온도를 측정하기 위한 센서는 MCU 내부에 존재하며 해당 값은 ADC를 통하여 측정할 수 있다. 해당 기능을 사용하여 BMS 테스트베드에 다음과 같이 MCU의 온도 측정 구문을 삽입하였다.

```

float get_mcu_temp(void)
{
    HAL_ADC_Start(&hadc1);
    while(HAL_ADC_PollForConversion(&hadc1, 1000) != HAL_OK);
    uint16_t adc_val = HAL_ADC_GetValue(&hadc1);

    float temperature = adc_val * 3300;
    temperature /= 0xffff;           // Reading in mV
    temperature /= 1000.0f;          // Reading in Volts
    temperature -= 0.76f;            // Subtract the reference voltage at 25°C
    temperature /= 0.0025;          // Divide by slope 2.5mV
    temperature += 25.0;            // Add the 25°C

    return temperature;
}

```

그림 5 MCU Temperature 측정 코드

E. System Clock

System Clock은 시스템 동작 전체에 영향을 미친다. 해당 Clock을 기준으로 명령어를 수행하며 신호 입/출력에도 영향을 미친다. System Clock이 불안정해지면 ESS 시스템이 의도하지 않은 성능 이상 상황을 발생시킬 수 있다. 따라서 System Clock을 시스템 리소스 측정 요소로 포함하였다. 해당 데이터는 MCU의 레지스터를 통하여 데이터가 저장되어 있으며 해당 값을 측정할 수 있는 구문을 삽입하였다.

Registers 1		
Find: CYCLECOUNTER	Group: Current CPU Registers	
Name	Value	Access
R0	0x0000'0008	ReadWrite
R1	0x0000'000a	ReadWrite
R2	0x2000'0128	ReadWrite
R3	0x2000'0128	ReadWrite
R4	0x0800'2100	ReadWrite
R5	0xe000'1000	ReadWrite
R6	0x2000'00c4	ReadWrite
R7	0x0800'20a4	ReadWrite
R8	0x4002'0c00	ReadWrite
R9	0x2000'04f0	ReadWrite
R10	0x2000'0538	ReadWrite
R11	0x4002'0400	ReadWrite
R12	0x0000'0008	ReadWrite
# APSR	0x2000'0000	ReadWrite
# IPSR	0x0000'0000	ReadWrite
# EPSR	0x0100'0000	ReadWrite
PC	0x0800'1f60	ReadWrite
SP	0x2000'04f0	ReadWrite
LR	0x0800'01ad	ReadWrite
# PRIMASK	0x0000'0000	ReadWrite
# BASEPRI	0x0000'0000	ReadWrite
# BASEPRI_MAX	0x0000'0000	ReadWrite
# FAULTMASK	0x0000'0000	ReadWrite
# CONTROL	0x0000'0004	ReadWrite
CYCLECOUNTER	1'099'280'240	ReadOnly
CCTIMER1	1'099'280'240	ReadWrite
CCTIMER2	1'099'280'240	ReadWrite
CCSTEP	546'147	ReadOnly

그림 6 System Clock Register

2.3 BMS 에너지원 및 시스템 리소스 측정 데이터 수집



그림 7 인셀 테스트베드 환경

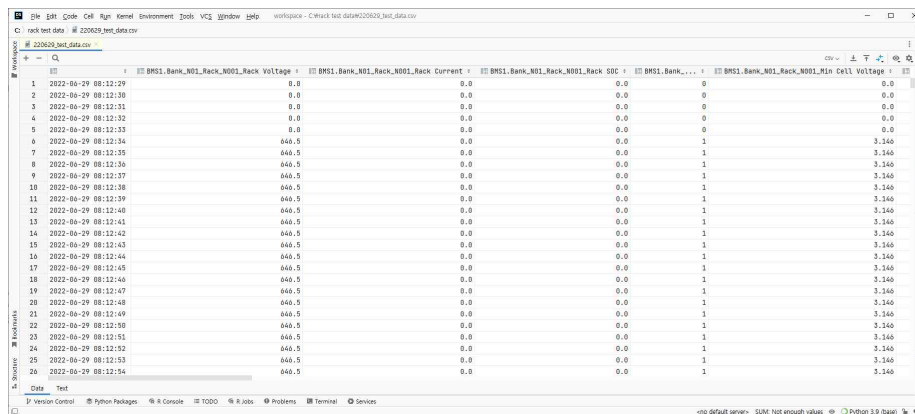


그림 8 Rack BMS 테스트베드

상기 문단에서 삽입한 시스템 리소스 측정 코드를 인셀 테스트베드 환경에 구축된 Rack BMS 내부 코드에 삽입하고 Rack BMS 충/방전 시험 시 시스템 리소스 데이터를 함께 측정하도록 하였다. 본 테스트베드에서 측정된 데이터는 1차적으로 KETI에서 가공을 수행하여 Google Cloud Storage에 적재된다.

3. BMS의 에너지원 및 시스템 리소스 데이터 시각화

3.1 수집된 데이터 개요

A screenshot of a data analysis software interface showing a table of test data. The table has columns for time, voltage, current, SOC, and min cell voltage. The data is organized into rows, with the first row being a header and the subsequent rows containing numerical values. The interface includes a menu bar at the top and a status bar at the bottom.

		BMS1.Bank_N01_Rack_N001_Rack Voltage	BMS1.Bank_N01_Rack_N001_Rack Current	BMS1.Bank_N01_Rack_N001_Rack SOC	BMS1.Bank_N01_Rack_N001_Min Cell Voltage
1	2022-04-29 08:12:29	0.0	0.0	0.0	0.0
2	2022-04-29 08:12:30	0.0	0.0	0.0	0.0
3	2022-04-29 08:12:31	0.0	0.0	0.0	0.0
4	2022-04-29 08:12:32	0.0	0.0	0.0	0.0
5	2022-04-29 08:12:33	0.0	0.0	0.0	0.0
6	2022-04-29 08:12:34	0.0	0.0	0.0	0.0
7	2022-04-29 08:12:35	0.0	0.0	0.0	0.0
8	2022-04-29 08:12:36	0.0	0.0	0.0	0.0
9	2022-04-29 08:12:37	0.0	0.0	0.0	0.0
10	2022-04-29 08:12:38	0.0	0.0	0.0	0.0
11	2022-04-29 08:12:39	0.0	0.0	0.0	0.0
12	2022-04-29 08:12:40	0.0	0.0	0.0	0.0
13	2022-04-29 08:12:41	0.0	0.0	0.0	0.0
14	2022-04-29 08:12:42	0.0	0.0	0.0	0.0
15	2022-04-29 08:12:43	0.0	0.0	0.0	0.0
16	2022-04-29 08:12:44	0.0	0.0	0.0	0.0
17	2022-04-29 08:12:45	0.0	0.0	0.0	0.0
18	2022-04-29 08:12:46	0.0	0.0	0.0	0.0
19	2022-04-29 08:12:47	0.0	0.0	0.0	0.0
20	2022-04-29 08:12:48	0.0	0.0	0.0	0.0
21	2022-04-29 08:12:49	0.0	0.0	0.0	0.0
22	2022-04-29 08:12:50	0.0	0.0	0.0	0.0
23	2022-04-29 08:12:51	0.0	0.0	0.0	0.0
24	2022-04-29 08:12:52	0.0	0.0	0.0	0.0
25	2022-04-29 08:12:53	0.0	0.0	0.0	0.0
26	2022-04-29 08:12:54	0.0	0.0	0.0	0.0

그림 9 측정된 Rack 시험 데이터

인셀社의 테스트베드에서 약 12시간동안 총 8회의 시험을 수행하였다. 해당 시험에서는 셀의 SoC 상태에 따른 충/방전을 수행하면서 에너지원 데이터 및 시스템 리소스 데이터를 측정하였다. 상기 이미지는 시험을 수행하면서 측정된 데이터들을 가리키며 다음 항목을 포함하고 있다.

항 목	설 명
Time	측정 시간
Rack Voltage	평균 전압
Rack Current	평균 전류
Rack SOC	State of Charge
Max Cell Temperature	셀 최대 온도
Max Cell Temperature Position	최대 온도 셀 위치
Min Cell Temperature	셀 최소 온도
Min Cell Temperature Position	최소 온도 셀 위치
Max Cell Voltage	셀 최대 전압
Max Cell Voltage Position	최대 전압 셀 위치
Min Cell Voltage	셀 최소 전압
Min Cell Voltage Position	최소 전압 셀 위치
MCU_stack_percent	스택 사용량 비율
MCU_VREF	MCU 공급 전압
MCU_TEMP	MCU 온도
Max_Runtime	MCU 최대 동작 시간

표 1 측정된 Rack 시험 데이터 범례

3.2 수집 데이터 전처리 및 시각화

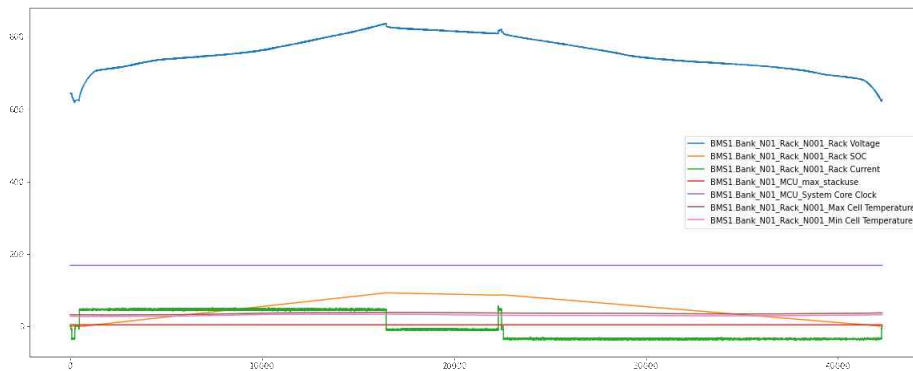


그림 10 데이터 처리 전 시각화 그래프

본 데이터를 도표화시키면 다음과 같다. 이와 같이 순수한 상태의 데이터는 양 데이터 간 값의 범위도 다르고 불필요한 데이터 범례도 있어 이를 해결해야 데이터의 시각화 및 분석이 가능해진다. 해당 전처리 과정은 시각화 및 데이터 연관성 분석을 위하여 불필요한 라벨 제거, 정규화를 수행하였다.

우선 시각화에 긍정적인 영향을 끼치지 않는 라벨을 제거한다. 본 데이터를 측정하면서 발생한 데이터 중 셀 간 에너지원 편차 데이터 발생 지점(Position) 관련 데이터와 당시의 시각을 측정하기 위한 Time 데이터가 있었다. 해당 데이터는 시각화하지 않을 데이터이므로 제거하는 과정을 수행한다.

정규화는 데이터의 범위 차이를 공통의 범위로 변경하는 과정이다. 각각 측정된 데이터는 서로 다른 데이터 범위를 가진다. 이로 인해 데이터 간 시각화 과정에서 측정 범위가 상대적으로 좁은 데이터의 시각화가 힘들어진다. 이를 해결하기 위하여 정규화 과정 중 min-max scaling 기법을 수행하였으며 해당 기법은 모든 데이터를 0 ~ 1의 데이터 영역으로 좁히게 하여 시각화 및 분석

을 수행하였을 때 각 데이터 간의 연관성을 쉽게 파악할 수 있게 해준다.

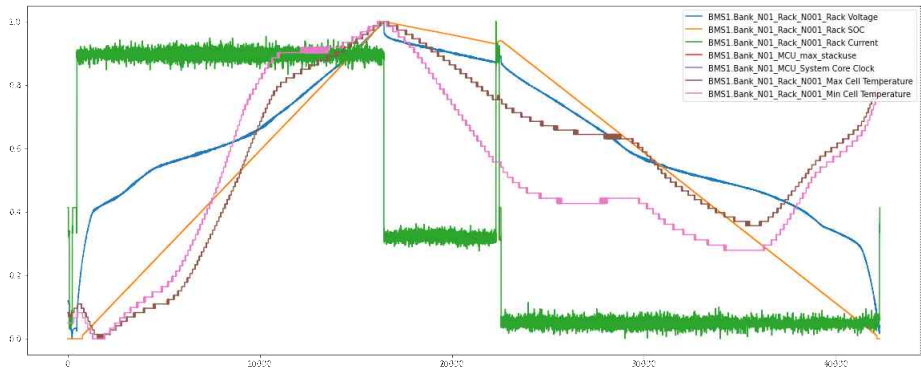


그림 11 데이터 처리 후 시각화 그래프

지금까지의 과정을 거친 후 데이터를 시각화하면 위 그래프가 출력된다.