


## 1차년도 기술문서

(과제명) 대규모 분산 에너지 저장장치 인프라의 안전한 자율운  
영 및 성능 평가를 위한 지능형 SW 프레임워크 개발  
(과제번호) 2021-0-00077

- 기술문서명 : 에너지저장장치의 BMS 리소스 관리
- 작성일자 : 2021년 12월 22일

과학기술정보통신부 SW컴퓨팅산업원천기술개발사업  
“기술문서”로 제출합니다.

수행기관	성명/직위	확인
슈어소프트테크(주)	심정민 / 이사	

정보통신기획평가원장 귀하

## 목 차

1. 에너지저장장치 ESS 구성요소 .....	3
2. 시스템의 자원관리를 통한 이상 감지 .....	4
3. 에너지저장장치의 위험요소 .....	5
4. BMS의 시스템 리소스 관리 .....	7

### <표 차례>

표 1 지락전류 변화에 따른 지락차단장치 동작시간.....	6
----------------------------------	---

### <그림 차례>

그림 1 .....	3
그림 2 .....	5
그림 3 .....	8
그림 4 .....	9

## 1. 에너지저장장치 ESS 구성요소

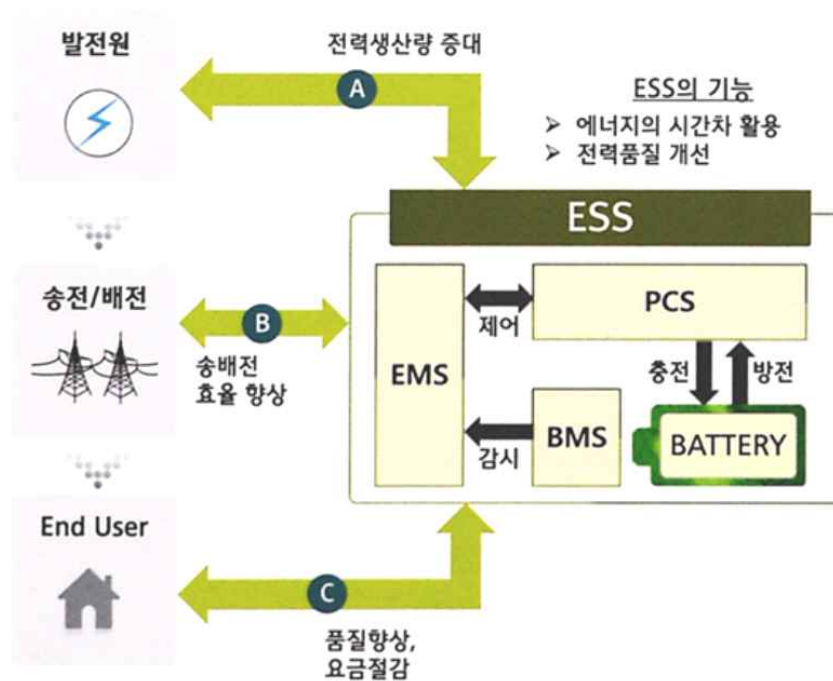


그림 1 한국에너지공단, ESS 시스템 구성

에너지 저장장치(ESS; Energy Storage System)는 전력을 효율적으로 사용하게 하는 설비로, 주파수 조정, 신재생에너지 출력 안정화, 전력피크 저감 등에 활용함으로써 전력품질 향상 및 전력수급 위기 대응이 가능한 설비임. ESS는 배터리(Battery), 전력변환장치(PCS; Power Conditioner System), 배터리관리장치(BMS; Battery Management System), 전력관리장치(Power Management System or Energy Management System) 등의 제반운영 시스템으로 구성됨. ESS의 종류는 ESS 구성요소 중 하나인 배터리(Battery)의 4가지 저장방식(물리적/화학적/전자기적/열적 저장방식)에 따라, 다양한 ESS가 존재함. - 출처 : 한국에너지공단 홈페이지

**배터리(Battery) :** 전력변환장치(PCS)로부터 받은 전기에너지를 직류(DC)저장(또는 충전)하거나, 저장되어 있는 전기에너지를 계통에 출력(또는 방전)하는 동작을 수행하는 장치임. 종류로는 한 번 소모 시 다시는 사용할 수 없는 1차 전지와 충전하여 여러 번 재사용이 가능한 2차 전지로 구분할 수 있음.

**배터리시스템(BMS) :** 배터리의 상태 데이터(전압, 전류, 온도 등) 실시간 데이터를 저장하고, 저장용량(SOC) 및 배터리 건강상태(SOH)를 계산하여, 배터리의 용량 및 수명을 예측하는 등의 전반적인 배터리의 상태 및 동작을 관리하고 감시하는 장치로, PCS, PMS와 통신하여 정보를 교환함. 배터리관리장치는 하드웨어적으로 크게 전자제어장치(ECU)와 셀 모듈(CM)로 구성되어 있으며, 소프트웨어적으로는 계측 알고리즘과 충전량 계산, 수명 예측, 셀 밸런싱 알고리즘 등이 사용됨. 배터리 안전을 위한 과충전, 과온, 과전류, 과전압 보호 기능 및 회로를 포함한다. 또한, 배터리관리장치에 사용되는 기술은 열관리 제어 기술과 배터리충전상태(SOC) 제어기술 등이 있음.

전력변환시스템(PCS, Power Conditioning System) : 배터리에 저장된 전기에너지를 상용의 전압, 주파수를 가진 전력으로 바꾸어 주거나(방전), 또는 그 반대로 상용의 전원, 주파수를 가진 전력을 직류(DC)로 변환하여 배터리를 충전하는 동작을 수행하는 장치임. 국내 PCS 제조 기업으로는 현대중공업, 효성, LS산전 등이 있다.

에너지관리시스템(EMS, Energy Management System / PMS, Power Management System) : 에너지 저장장치 내에서의 에너지 소비를 감시하고 규제하며, 전력 사용을 예측하여 필요한 조정을 할 수 있는 기능 등 전력을 관리하는 시스템으로, 전력변환장치(PCS)와 배터리 주변기기의 정보를 제공받아 에너지저장장치로 실시간 모니터링이 가능하여 어플리케이션의 연산결과를 반영, 배터리 관리장치(BMS)에 지시 및 관리하는 장치이다. 에너지 관리는 전력의 생산에서 소비에 이르는 전력계통, 즉 발전, 송전, 배전, 소비의 전 영역에 걸쳐 이루어지게 된다. 종류로는 건물용인 BEMS, 산업용(공장용)인 FEMS, 가정용인 HEMS, 지역용인 CEMS, 스마트 그리드용 EMS 등이 있음. EMS는 BMS와 PCS 외에도 온도와 습도에 민감한 배터리를 안정적으로 유지하기 위해 온습도계와 공조장비에서 상태데이터를 수집하고 역 전력 및 부하를 관리하기 위해 역 전력 계측기와 통신하게 되고, 지락을 사전에 예방하기 위해 GFD(지락검출기), IMD(절연감시장치)와도 연결된다.

## 2. 시스템의 자원관리를 통한 이상 감지

SW Code는 사용자의 의도치 않은 시스템 장애가 발생할 수 있다.

운행중인 장치에 장애가 발생할 경우 예기치 않은 손실과 사고가 발생 할수 있다. 시스템 리소스 모니터링의 목적은 빠른 장애 탐지, 시스템 다운 타임의 최소화, 시스템 파악, 시스템 자동화 등 많은 잇점이 있을 것이다.

소프트웨어의 올바른 실행, 특히 타이밍 관점에서의 올바른 실행은 점점 더 중요한 역할을 하고 있다. 따라서, 소프트웨어의 안전 검증에서 타이밍 분석은 각각 다른 환경에서 정확한 타이밍, 타스킹을 요구하는 시스템에서 반드시 검증되어야되는 하는 항목이라고 할 수 있다.

### ○ Stack 사용량 모니터링

컴파일시 고정되는 메모리영역

코드영역 : .hex나 .bin과 같은 프로그램 코드가 존재한다. (함수, 제어문, 상수)

데이터영역 : 전역 변수가 존재한다.(전역변수)

### ○ 실행 중 할당하는 메모리 영역

스택 영역 : 함수 호출 시 할당되며 종료시 반납되는 지역 변수가 존재한다.(지역변수)

힙 영역 : malloc(), calloc() 등으로 생성된 동적 변수가 존재. 이는 free()함수로 할당된 영역을 반납해줘야 한다.(동적할당)

○ 힙과 스택은 같은 공간을 공유한다. 힙 영역은 위 주소부터 할당이 되며, 스택 영역은 아래 주

소부터 할당 된다. 그렇기에 각 영역이 서로의 공간을 침범하는 일이 발생할 수 있다. 이를 각각 Heap Overflow, Stack Overflow라고 칭한다.

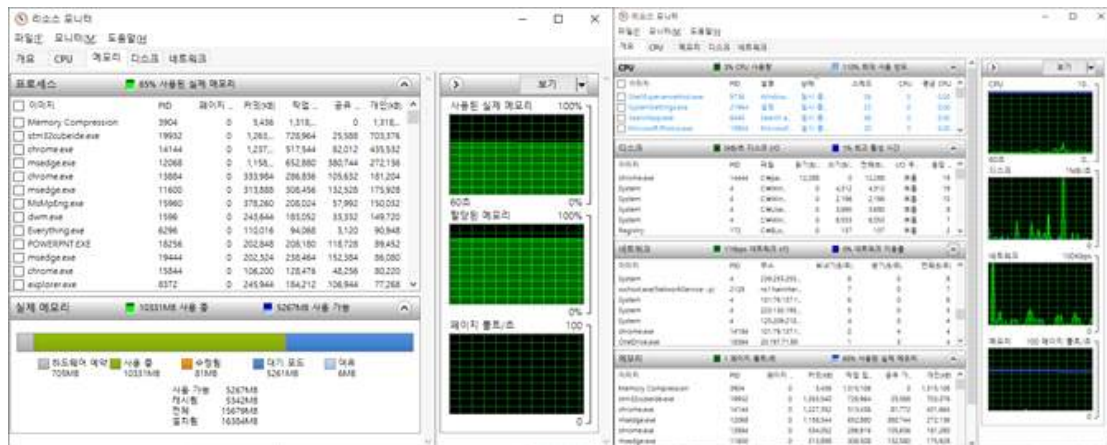


그림 2 윈도우의 리소스 모니터

#### [운영체제의 자원관리 기능과 프로세스 관리]

운영체제는 한정된 컴퓨터 시스템의 자원들을 보다 효율적으로 관리하고 운영한다. 운영체제의 자원 관리 기능은 대표적으로 4가지가 있다.

- A. 프로세서 관리 : 작업 시간을 할당하거나, 작업의 우선순위를 부여하여 효율적으로 실행되도록 프로세스를 관리하는 기능
- B. 메모리 관리 : 어떤 부분이 프로그램에 의해 사용되고 있는지를 관리, 프로그램에게 메모리 공간이 필요할 때 할당하고, 더 이상 필요하지 않게 될 때 회수하여 전체 기억 공간이 효율적으로 사용될 수 있도록 하는 기능
- C. 파일 관리 : 파일의 생성과 삭제, 변경, 유지 등의 관리를 담당하는 기능이며, 각 파일의 이름과 보조 기억 장치의 저장 영역 주소를 기억한다.
- D. 입출력 장치 관리

컴퓨터에 연결된 각종 주변기기를 포함하여 여러 가지 입출력 장치를 관리하고 제어한다. 입출력 장치의 스케줄링도 이루어진다.

### 3. 에너지저장장치의 위험요소

에너지저장장치의 구성요소에서 EMS는 ESS의 컨트롤 타워 역할을 하고 BMS는 배터리의 전압,전류로 충전상태를 모니터링하고 모듈내 각각의 셀간의 충방전 정도가 동일해지도록 조정을 한다(Cell balancing). EMS 통신과 무관하게 자체적으로도 과충전 방지를 하고, 내부 보호회로를 통해 과전류 및 단락시 외부 스위치를 차단하는 기능이 있다.

이렇듯 BMS는 에너지 관리시스템(EMS)이나 전력변환시스템인 PCS가 고장이 나더라도 높은 에너지 밀도로 화재가 발생하고 열폭주로 이어질수 있는 리튬이온 배터리의 최전방 수호병 역할을

하고 있다.

배터리와 직접 연결되어 배터리의 전압,전류,온도의 상태를 체크하고 있는 BMS의 상태정보를 보다 빨리 위험 상태가 EMS로 전송이되면 화재 예방 및 감지에 도움이 되리라고 예상된다.

2019년 ESS 화재 원인 조사에 따르면 일부 ESS의 PCS와 BMS 로그데이터를 분석하였고, 분석 결과 충전이 완료된 배터리가 대기상태 중 갑작스러운 전류 방향의 변화와 함께 셀전압과 셀온도의 편차가 급상승하였고, 이 후 화재로 이어진 것으로 파악되었다. ESS가 충전이 완료된 이후 방전을 하던 중 수차례의 Fault가 발생하였고, 배터리의 방전 전류가 갑자기 충전전류로 바뀌어 흐른 사실이 파악되었다.

“DC 지락<sup>1)</sup>에 의한 ESS시스템의 화재에 대한 연구”에서는 AC측과 DC측 지락실험을 실시하여 AC측 지락문제로는 배터리에 과전류가 발생하지 않는다는 사실을 확인하였으며 반면 배터리단(DC측)에서 지락이 발생할 경우, 직렬로 연결된 배터리에서 지락 발생위치에 따라서 지락 전류의 크기가 변한다는 사실을 확인하였다. 지락발생 지점을 기준으로 배터리의 전압에 차이가 있는 경우 배터리가 충분히 충전되어있음에도 불구하고 Grid 단에서 배터리쪽으로 높은 전류가 흐른다. 배터리 스트링 간의 전압차가 없는 경우 지락 발생시 순간적으로 과도 전류가 발생하나 전류가 크기가 작고 시간의 경과 후 곧 사라진다는 것을 알수 있었다. 이로 인해 이 논문에서는 DC 지락으로 인해 배터리 셀에 대전류가 흘러 배터리가 소손된 사고임을 추정하고 있다.

태양광발전설비는 비,바람,눈,자외선 등에 노출되어 있고 고양이,족제비,쥐 등 동물들이 수시로 다닐 수 있다. 이러한 환경속에서 수십년을 운영해야한다. 소방방재청 통계에 의하면, 최근 태양광발전설비 화재로 인해 소방서가 출동하는 건수가 연60여건이 넘고 있다.(최근 5년간 태양광 및 ESS 화재는 총 338건으로 355억원의 재산피해가 발생,국회의원 이주환의원,2020년) 경미한 사고까지 합하면, 더욱 많은 화재나 사고가 발생하는 것이다.

이러한 문제를 해결하고자 산업통상자원부는 판단기준 제54조를 개정하여 태양광발전설비에 지락 차단장치 설치를 의무화했고, 2021년 9월1일로 시행시기를 정했다.

지락전류의 급격한 변화	동작시간
$\Delta I_g = 30\text{mA/sec}$	<300ms
$\Delta I_g = 60\text{mA/sec}$	<150ms
$\Delta I_g = 150\text{mA/sec}$	<40ms

표 1 지락전류 변화에 따른 지락차단장치 동작시간

지락사고로 30mA가 급변하는 경우 0.3초, 60mA가 급변하는 경우 0.15초, 150mA가 급변하는 경우 0.04초 이내에 동작하여야 한다. 여기서 동작시간의 정의는 지락사고가 발생하는 순간부터 차단기가 동작하는 순간까지이다.

1) 지락 : 전류가 흐르는 상태에서 절연부분이 열화 또는 손상되어 충전부가 타물체와 접촉되어 대지로 전기가 흐르는 것, 일반적으로 누전(전로 이외에 흐르는 전류)과 같은 의미로 사용되나 저전압/저용량은 누전이라 칭하고 고전압/고용량은 지락이라 칭함

## 4. BMS의 시스템 리소스 관리

BMS의 기능은 계측, 계산, 제어, 통신의 기능이 있다. 전압/전류/온도 측정, SOC(State of Charge) 계산, SOH(State of Health) 계산, 배터리 셀 간의 전압을 균등하게 유지하는 기능, 배터리 온도 상승시 제어 기능, 랙(rack)간의 병렬운전 중 이상(전압,온도) 발생 시 제어 기능, 표준 통신프로토콜을 사용한 통신 기능(IEC-61850/mode bus)등이 있다.

BMS는 배터리의 전압/전류/온도를 측정하여 상태를 제어하는 장치로 배터리의 전압, 충전상태 등을 모니터링하고 모듈 내 단위 셀간의 충방전 정도가 동일해 지도록 조정하는 Cell balancing 뿐만 아니라 배터리의 안전을 위한 과충전 방지 등의 보호기능을 수행하고 보호회로를 통해 과전류 및 단락 시 외부 스위치를 차단하는 기능 및 EMS와 통신하는 역할을 한다. 기존의 Ni-Cd(니켈-카드뮴)이나 Ni-MH(니켈 수소) 전지는 안전성이 우수하여 보호회로가 필요하지 않았으나 리튬 배터리의 경우에는 열폭주, 수분 침투 및 단락 시 화재나 폭발 위험성이 있으므로 리튬이온 배터리에서는 BMS의 중요도가 높아졌다.

시스템의 성능 관리는 향후 발생할 수 있는 문제점을 발생전에 먼저 대응할 수 있고, 특정 영역에서 병목현상을 해결하여 자원의 활용도를 향상시킬수 있으며 운영되는 시스템의 적절한 응답시간을 보장할수 있는 신뢰도를 마련할 수 있다.

성능관리를 위한 대표적인 성능지표에는 응답시간, 시간당 처리량, 자원사용량 등이 있다.

프로파일링(profiling) 또는 성능분석은 프로그램의 시간 복잡도 및 메모리, 특정 명령어 이용, 함수 호출의 주기와 빈도 등을 측정하는 동적 프로그램의 한 형태이다. 프로파일링 정보는 프로그램 최적화를 보조하기 위해 사용된다. 프로파일링은 프로파일러(profiler)라는 도구를 사용하여 프로그램 소스 코드나 이진 실행 파일을 계측 분리함으로써 수행한다.

일반 소프트웨어와 달리 임베디드 소프트웨어의 경우 타겟에서 실행되면서 CPU나 메모리와 같은 자원을 얼마나 소비하고 어느정도 성능을 발휘하는지 파악하는 것이 중요하다.

코드 자체를 분석하는 "정적 프로파일링" 방법과 실제 타겟에서 동작시키며 분석하는 "동적 프로파일링" 방법이 있다. 특히, 동적 프로파일링은 꼭 필요한 과정인데, 같은 코드라 하더라도 실제 운영되는 타겟의 환경과 컴파일러를 통해 생성된 기계어에 따라 다양한 성능 차이가 발생하기 때문이다.

컴퓨터에서 심장은 CPU라고 할수 있다.

심장이 제대로 뛰고 있어야 온 몸에 혈류 공급이 되고 신진대사가 이루어 질수 있듯이, CPU에서 정확한 시스템 클럭과 타이밍에 맞춰 주변장치에 명령어를 쓰고 상태를 확인한다.

실제 BMS에서 CPU Clock은 정상인지, 정확한 타이밍에 맞춰 동작하는건지, 메모리 사용량은 이상이 없는지, Stack 사용량은 이상이 없는지 직·간접적으로 측정/파악하는 알고리즘을 구현 적용하면 화재발생 예방에 효과가 있을 것으로 기대가 된다.

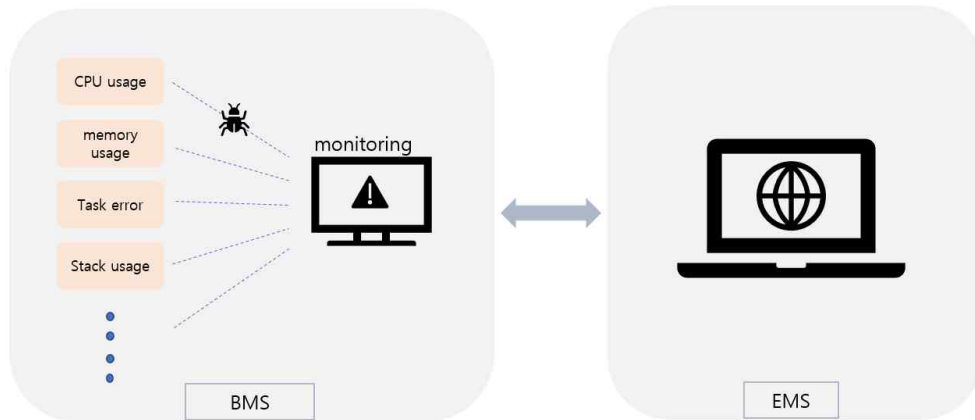


그림 3 BMS 리소스 모니터링

### ○ 프로파일링

프로파일링은 실행 중인 임베디드 시스템이나 시뮬레이션에서 타이밍 파라미터에 접근하는 과정을 기술한다. 한가지 접근 방법은 실행시간 측정 수행과 동시에 그 자리에서 원하는 타이밍 파라미터를 직접 도출하는 것이다. 다른 접근 방법은 트레이싱을 통해 간접으로 수행하는 것이다. 트레이싱을 할 때는 일단 트레이스 데이터 수집만 하고, 이후에 이 수집된 데이터를 통해 타이밍 파라미터를 추출할 수 있다.

### ○ 임베디드시스템 자원관리 :

임베디드 시스템 응용프로그램의 비기능적 요구사항 - 응용프로그램은 제한된 하드웨어 자원 위에서 Real-time 제약 조건을 수행한다.

### ○ BMS System CPU 사용량 모니터링 실시간 성능 측정

목표 : 시스템 동작중 CPU 부하 및 사용량 모니터링

- CPU 부하 및 사용량 : 모든 시스템 태스크, 인터럽트, 지연, 타이밍 파라미터, 타이밍 요구사항 반영 등을 반영하여 스케줄링의 상태를 모니터링
- 모든 TASK/Interrupt의 Worstcase를 포함하여 모니터링 한다.

A. CPU 사용량  $U = \frac{t_e}{t_o}$

( $t_o$ :관측 기간,  $t_e$ 유휴 코드 이외의 다른 코드를 처리하기 위해 CPU가 소비한 시간)

B. CPU 사용량 :  $U = \frac{CET}{PER_0}$

(CET : 실제실행시간,  $PER_0$ :Task시간)

C. 관측기간동안의 실행시간의 합 :  $t_e = \sum_{n=1}^N CET_n(t_e\text{:관측시간})$



D. 관측기간을 고려한 CPU 부하 :  $U = \frac{\sum_{n=1}^N CET_n}{t_0}$

예를 들어 4,8,16,1000ms의 주기적인 태스크에서 각각 실행시간이 1,2,3,1ms일 경우  $CET_{4ms} = 1$ ,  $CET_{8ms} = 2$ ,  $CET_{16ms} = 3$ ,  $CET_{1000ms} = 1$  이라 할수 있으며 위 공식에 적용시 아래와 같은 결과를 얻을 수 있다.

$$U = \frac{1ms}{4ms} + \frac{2ms}{8ms} + \frac{3ms}{16ms} + \frac{1ms}{1000ms} = 0.6885 = 68.85\%$$

$$MCU\ load\ [\%] = \frac{(1 - count\ at\ load)}{count\ at\ no\ load} \times 100$$

기대효과 : CPU 부하가 어느정도 여유가 있음을 실시간 모니터링할 수 있다.

- 명시적으로 idle 상태를 정의
- Timer tick을 이용하여 Count를 증가시킨다.
- Idle 상태에 이르지 못한다면 Count는 되지 않으며, MCU load는 증가
- 실시간 CPU 부하를 가시적으로 표현하며 사용량 모니터링을 통해 위험도를 도출할수 있다.

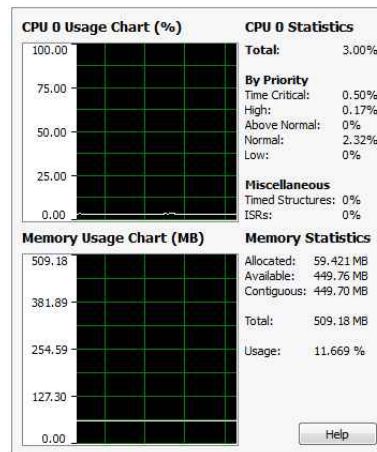


그림 4 CPU 사용량 예시