Chapter x

# EPS SUBSYSTEM

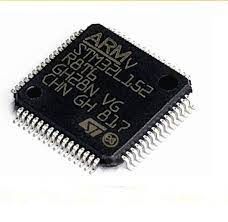
## Introduction

The primary EPS function is capturing solar energy from the sun and albedo with solar cells during the daylight and then to transmit it to the subsystems. So, a part of the captured solar energy must be kept in a battery in order to return it to the subsystems during the eclipse (during this time the solar cells are of course quasi inefficient. When sunlight reaches the Earth’s surface, some of it is absorbed and some is reflected. In the study of the CubeSat EPS we see that we must know the types of solar panels and how we design these to make higher current or higher voltage and the types of batteries we need. Also, we need to know subsystems that we will feed with electricity and calculate our power budget to ensure that our EPS can cover all requirements of our subsystems: On-Board Computer (OBC), Communications Subsystem (COMMS) and Attitude Determination and Control System (ADCS).

## Functional Description

The Electrical Power Subsystem (EPS) is responsible for charging the batteries from the solar panels using the MPPT technique, subsystems power management and batteries temperature control. It is also responsible for the post launch sequence that keeps the subsystems turned off for 30 minutes after the launch from the ISS and after the 30 minutes have passed, it deploys the antennas and the SU m-NLP probes by using a resistor to burn a thread that keeps the mechanism closed.

### Hardware main Components:

* STM32L152(Ultra Low Power Consumption) microcontroller with an ARM cortex M3 cpu core that runs the MPTT algorithm for charging the batteries.
* 3x 2600mAh 18650 Li-ion batteries.



* MOSFET switches for controlling the subsystems power
* 4x 1.5W (6Vx250mA)

### Software main functions:

EPS Subsystem software is concerned with main functions or tasks like:

1. Communication with OBC.
2. Power management and distribution including:

MPPT battery charging system, load control (ON/OFF).

1. Measuring and Sensing:

for voltages, currents, and temperatures for the subsystems, batteries, and solar panels.

1. Performing Safety Checks:

Checking if the voltages, currents, and temperatures are within the safety limits.

1. Status and Error Reporting:

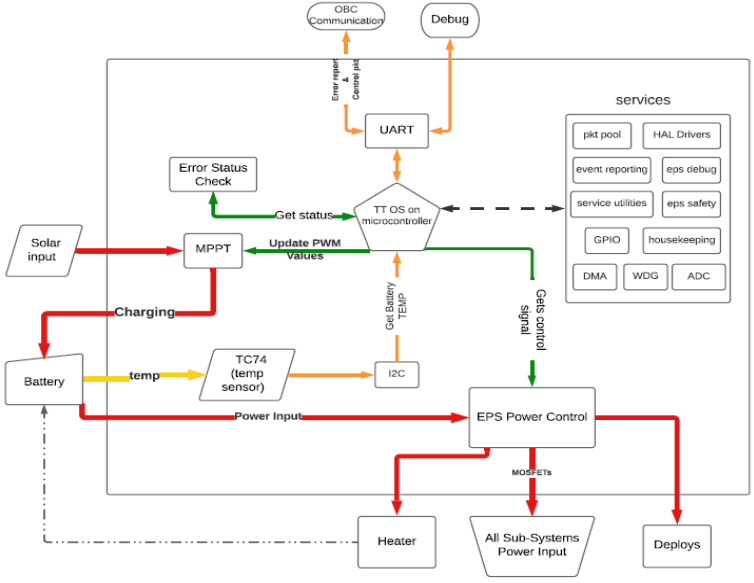
Sends error status via the UART debug port.

Diagram

Description automatically generated

Functional description diagram fig-xx

## Static Software Design



System Block Diagram fig-xx

## Time Triggered OS on microcontroller:

Here we mainly focus on the ttrd19a and its scheduler in which we add tasks in order with its best-case execution time (BCET) and worst-case execution time (WCET).

More details will be discussed in the Dynamic software design.

The EPS has a major tasks and functions that we have focused on while developing the software on the TTRD19A.

## Tasks:

OBC-EPS Communication:

which is done by UART in both directions.

This task includes two modes:

1. EPS to OBC Communication: EPS reports the voltages, currents of all subsystems and the battery status as well as the solar panels voltage, PWM duty cycle of MPPT phase.
2. OBC to EPS Communications: OBC gives packet to EPS which contains what subsystems to be ON or OFF because; each subsystem must report a heartbeat message to OBC and if not the OBC orders the EPS to turn off the power of this subsystem and restart it again

(Power ON RESET), Also if the subsystem consumes a higher current than the threshold will also lead to the same order.

## Power Control & MPPT:

1. This Task is responsible for getting the status of the subsystems, batteries, and solar panels.
2. Switching ON/OFF the power to the subsystems according to OBC packet or the startup sequence.
3. Updating the PWM duty cycle of the MPPT battery charging system.

## Get Reading:

Here the EPS gets all the readings voltages, currents as well as the battery temperature and the ambient temperature.

These readings are collected by ADC Pins except the ambient temperature which is taken by the TC74 sensor which is using I2C Protocol.

## Error Check & Error Reporting:

This task contains subtasks:

1. Load safety limits from the memory
2. Perform Safety limits check
3. UART debug service

## Dynamic Software Design

Diagram

Description automatically generatedIn run time, we have put our main tasks to the scheduler in order and with their BCETs and WCETs.

According to the given data the scheduler will construct a time line of execution