



Small Satellite Research Laboratory

Franklin College of Arts and Sciences

UNIVERSITY OF GEORGIA

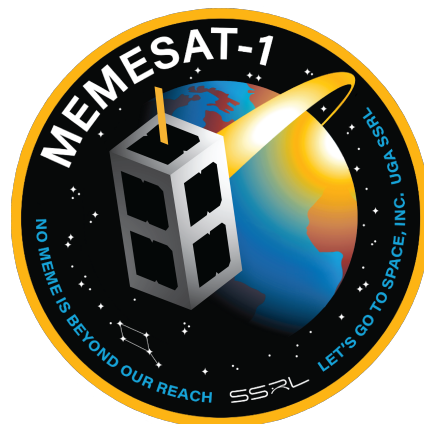
EpsUart Component Documentation

Prepared for
The Small Satellite Research Laboratory
Fall 2024 - Spring 2025

Principal Investigator
Dr. Deepak Mishra

The University of Georgia
Geography-Geology Department
Athens, Georgia
United States of America

Prepared by
The MEMESat-1 CDH Team
University of Georgia
Athens, Georgia
United States of America



Contents

1	List of Tables	2
2	List of Acronyms	2
3	Revision History	2
4	Purpose	3
5	Overview	3
6	Design	3
6.1	Requirements	3
6.2	Conceptual Control Flow	4
6.3	Ports	4
6.4	Custom Ports	5
6.5	Custom Types	5
6.6	Events	7
6.7	Telemetry	7
7	Implementation	8
7.1	Behaviors	8

1 List of Tables

2 List of Acronyms

BBS Bulletin Board System
CDH Command and Data Handling
EPS Electrical Power System
FSM Finite State Machine
GDS Ground Data System
MCU Microcontroller Unit
OBC On-Board Computer
OSAL Operating System Abstraction Layer
OTA Over-The-Air
OTAU Over-The-Air Updater
UART Universal Asynchronous Receiver-Transmitter
UHF Ultra High Frequency

3 Revision History

Changes	Authors	Version
[08-26-2023] Document created.	Preetam Jain	1.0.0
[02-26-2024] Document updated.	Preetam Jain	1.0.1
[04-01-2025] Preparing for IMR-3	Samuel Lemus	1.1.0

4 Purpose

This document is an overview of the EpsUart component, which is intended to communicate with the Electrical Power System (EPS), log telemetry such as voltage and current, and coordinate with the MCU's FSM states.

5 Overview

This component is responsible for processing UART packets send from the STM to the OBC and logging via telemetry and events. More specifically, this component uses the ByteStreamRecv and ByteStreamSend port interfaces provided by the OSAL (through the F' framework) to send and receive data. It received data from the EPS and uses it to set system telemetry values for downlinking in the form of Health Beacons. The StateMachine component also receives data from the EpsUart component to decide which state to switch to - depending on the data received. The UartRead port first checks the serial_status for an error. If there is no error, it verifies the length and content of the packet before passing it into the parseUart function. This function evaluates the packet's 4 byte msb/lsb checksum and asks for a resend if it fails the comparison. On comparison, the third byte field (packet type indication) is read and processed accordingly. If the byte is meant to set a value, it passes it to the logTelem function which writes various values to telemetry such as battery voltage and current. Additionally the component communicates with the EPS using the outsideCommandBuffer port, whcih creates a packet containing two sync bytes, a type byte, a length byte, the message data, and a checksum - and writes it to the UART buffer.

Description of Directory Files:

- EpsUart.fpp - contains fprime interactions for the component.
- EpsUart.hpp - used as a container for function signatures and prototypes.
- EpsUart.cpp - implementation of the fprime interactions.

6 Design

6.1 Requirements

The requirements for the component are as follows:

Requirement	Description	Verification Method
EPS_UART-00	The component shall accept UART packets and validate/parse through them to facilitate communication from the EPS.	Unit Testing
EPS_UART-01	The component shall write EPS-given system values to telemetry for the OBC.	Unit Testing
EPS_UART-02	The component shall write to the EPS through UART communication.	Unit Testing

6.2 Conceptual Control Flow

The stm32 (stm) shall exist and in return, the flight software (fsw) should have the capability to listen to it.

Raw state: The stm is en-boot. The stm should put out a heartbeat to the on-board computer (obc). Cases:

- OBC is also en-boot and there exists some downtime from the system receiving power, os-boot time, and fsw service start-time.
 - stm emits heartbeat signal without regard to a response.
- OBC has just started up and is running the fsw - a heartbeat is then heard from the stm.
 - OBC changes state from start-up to idle.
 - OBC responds to heartbeat packet with 'ack' response.
 - stateMachine reports state change by sending a changeState packet to the stm.
 - stateMachine sends heartbeat packet to stm when cycles are had (every second)
 - stm hears heartbeat packet ack from OBC
 - stm responds to the heartbeat packet ack by starting a timer that waits for 2 seconds and is reset every time a heartbeat signal is heard from the obc. In return, the stm disables the internal heartbeat timer until the 2 second wait-timer is triggered.
 - stm responds to an incoming heartbeat packet with an ack for said packet.
- OBC is running the fsw and a heartbeat ack is not received from the stm.

6.3 Ports

The EpsUart component has six f-prime ports. The UartRead port is an input that receives incoming buffer data from the EPS. The outsideCommandBuffer port is an input port that receives data, incorporates it into packets, and then sends the packets to the EPS. The Uart-Buffers port is an output port for providing the serial driver to receive incoming data from the STM. The UartWrite port is an output port that writes data to the UART buffer to be sent to the EPS.

Port Data Type	Name	Direction	Kind	Usage
ByteStreamRecv	UartRead	Input	Asynchronous	Port for receiving UART communication.
BufferSend	outsideCommandBuffer	Output	Synchronous	Port for receiving packet data and incorporating data into a packet and sending it to the EPS.
BufferSend	UartBuffers	Output	N/A	Port for sending back buffer to the serial driver in case of the fault packet or invalid status
ByteStreamSend	UartWrite	Output	N/A	Port for writing UART out to EPS
ChangeState	ChangeState	Output	N/A	Port for changing FSM states if packet designated.
RequestState	RequestState	Output	N/A	Port for requesting state from FSM

6.4 Custom Ports

There are two custom ports, ChangeState and RequestState. The ChangeState port is an output port used to signal to the FSM to change states if the packet received is designated to do so. The RequestState port is an output that requests the FSM's current state.

Name	Parameter	Type	Description
ChangeState	portNum	NATIVE_INT_TYPE	The port number
	state	State	The state of the component
RequestState	portNum	NATIVE_INT_TYPE	The port number
	request	bool	Whether or not the state has been requested

6.5 Custom Types

There is 1 custom type, MessageType. This type represents the type of packet received by the component, of which there are 8 types. The enumeration name clearly labels the function of

each enumeration.

Name	Type	Parameter	Description
MessageType	enum	NOP	A message type to write to the EPS
		ACK	A message type to write to the EPS
		ERROR	A message type designating to change states to safe anomaly
		RESET	A message type designation to restart the FSM
		GET_DATA	A message type requesting data from the EPS
		RETURN_DATA	(Not implemented)
		SET_VALUE	A message type designating to write telemetry
		RESEND	A message type requesting to resend the packet

Name	Type	Parameter
SensorData	enum	VBATT_VOLTAGE
		VBATT_CURRENT
		CELL_VOLTAGE
		CELL_CURRENT
		HEATER_STATUS
		SOLAR_PANEL_VOLTAGE
		SOLAR_PANEL_CURRENT
		VOLTAGE_5V0
		CURRENT_5V0
		VOLTAGE_3V3
		CURRENT_3V3
		SWITCH_STATE
		SENSOR_DATA
		FSM_STATE

Name	Type	Parameter	Description
States	enum	START_UP CRUISE SAFE_ANOM SAFE_CRIT_PWR RESTART SHUTDOWN	

6.6 Events

The events are used to indicate any changes or errors that occur during flight operations.

Name	Severity	Arguments	Description
EPS_UartMsgIn	activity high	msg : F32 : The first operand	An event that is triggered when uart is received.
BAD_CHECKSUM	warning high		An event that indicates a packet parsed contained a bad checksum
BAD_PACKET	warning high		An event that indicates a packet has invalid sync bytes

6.7 Telemetry

The telemetry values for this component are used to describe health data of the satellite and determine the state of the system.

Name	Data Type	Update	Description
EPS_UartRecvBytes	U32	On receive	Number of bytes received on UART
EPS_LastMsg	String	On receive	Last UART text message
VBATT_VOLTAGE	I64	On send	Telemetry for the battery voltage
VBATT_CURRENT	I64	On send	Telemetry for the battery current
CELL_VOLTAGE	CELL_DATA	On send	Telemetry for the cell voltage
CELL_CURRENT	CELL_DATA	On send	Telemetry for the cell current
BATT_TEMP	I64	On send	Telemetry for the cell temperature
HEATER_STATUS	bool	On send	Telemetry for the heater status
SOLAR_PANEL_VOLTAGE	PANEL_DATA	On send	Telemetry for the solar panel voltage
SOLAR_PANEL_CURRENT	PANEL_DATA	On send	Telemetry for the solar panel current
VOLTAGE_5V0	I64	On send	Telemetry for the voltage 5V0
CURRENT_5V0	I64	On send	Telemetry for the current 5V0
VOLTAGE_3V3	I64	On send	Telemetry for the voltage 3V3
CURRENT_3V3	I64	On send	Telemetry for the current 3V3
SWITCH_STATE	SWITCH_STATE	On send	Telemetry for the state to switch to

7 Implementation

7.1 Behaviors

The obc should have the following messaging capabilities to the stm.

- Send ack message to a received packet.
- request a resending of the last packet sent from the eps.

- variable re-try count (start with 3)
- trigger a state change of the stm
- purvey a 'heartbeat' from the state machine to the stm when running
- confirm or validate a prompt from the stm
- set a specific value on the stm
- request the status of a parameter tracked on the stm

The stm should have the following messaging capabilities to the obc.

- send ack message to a received packet.
- store a recently sent message and resend on request.
 - keep track of the count of resent messages.
- send a single sensor value to the obc.
- send multiple sensor values to the obc in a single packet via a sensor-group definition.
- purvey a 'heartbeat' to the obc.