Small Satellite Research Laboratory
Franklin College of Arts and Sciences
UNIVERSITY OF GEORGIA

Microcontroller 'EPS' Firmware 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 List of Acronyms

BBS Bulletin Board System
CDH Command and Data Handling
EPS Electrical Power System
GDS Ground Data System
GPIO General Purpose Input/Output
INA-219 Current Monitor
I2C Inter-Integrated Circuit
MCU Microcontroller Unit
OBC On-Board Computer
OSAL Operating System Abstraction Layer
OTA Over-The-Air
OTAU Over-The-Air Updater
RTOS Real-Time Operating System
UART Universal Asynchronous Receiver-Transmitter
UHF Ultra High Frequency

3 Revision History

Changes	**Authors**	**Version**
[31-10-2022] Document Created	Michael Starks	0.0.0
[10-02-2023] Revisions to general structure and additional information regarding UART Packetization, modes, and sensors	Aidan LeCroy	0.0.1
[05-09-2023] Updated wording to be more clear	Aidan LeCroy	0.0.2
[05-18-2023] Removed old or outdated information	Aidan LeCroy	0.0.3
[02-26-2024] Updated to include new changes to sensor service and state machine. Inserted updated pinout	Aidan LeCroy	0.0.4
[03-11-2024] Checked for Feasibility Review	Olivia Beattie	0.1.4
[04-04-2024] Updated the STM Pinout	Jake VanEssendelft	0.1.5
[04-05-2024] Updated References Section. Approved for Feasibility	Isaac Garon	1.0.0
[04-01-2025] Updated to include modified packetization structures, altered state machine control flow and revised state transitions	Samuel Lemus	1.1.0

4 Introduction

5 Purpose

This document describes the EPS MCU (STM32F303RET6) Firmware Architecture for MEMESat-1. The information held here aims to detail the high-level functionality of the Zephyr RTOS-based firmware and its interactions with other satellite subsystems.

6 Interactions

The EPS MCU interacts with the OBC (On-Board Computer) via a UART interface. The OBC utilizes the EpsUart FSW component to send and receive packets to and from the EPS MCU.

7 Design

7.1 EPS Architecture

Zephyr RTOS
The EPS firmware will be built using the Zephyr RTOS framework - a small real-time operating system for resource-constrained systems. Functionally it is split into four software components running in threads: OBC communications service, sensor service, state machine service, and UHF communications service.

Multithreaded Design
The first thread spawned is the state machine thread, which will in-turn spawn the sensor and communications threads following the detumbling sequence. Sensor data is shared throughout all threads via a mutex-protected array.

7.1.1 Concept of Operations

The stm32 shall exist and has several critical functions.
Raw state: The STM is en-boot. The stm should emit a heartbeat signal to the OBC.

1. The STM should tend to an internal state machine which keeps track of the current functions of the STM with regards to its global state.

2. The STM should prepare for sensor reading of 7 INA's which exist over the i2c0 bus.

3. The STM should respond to commands from the OBC.

4. The STM should be able to change internal variables or parameters (not limited to the state machine state) from commands sent from the OBC.

5. The STM should trigger listening the UHF radio on command from the OBC.

Inherent states: Sovereign, Idle, Cruise, Critical Power, Autonomous

State Descriptions

(1) Sovereign

1. This state is the initial nominal state denoting that functionality should be has without regard to responses from the OBC.
2. This state should permit heartbeat dispatching to the obc.
3. This state should monitor sensor readings and have the potential to switch to necessary response states.
4. Potential goal states from the sovereign state: Idle, Critical Power, Autonomous

(2) Idle

1. The preliminary state for this state transition needs to be the sovereign state.
2. This state is triggered when the obc has responded to a heartbeat emission with a heartbeat ack packet.
3. In this state, the stm is knowledgable of the obc being functional and is ready to begin nominal communications with it.
4. At this state, we are waiting for a ground station command declaring that the obc may switch to nominal functions in cruise mode. If this is to occur, the next state should be sent toi the stm from the obc and the stm is to switch to the Cruise state.
5. Sensor communications/logging from the stm to the obc may occur in this state.
6. A timeout may occur from when the last time the obc has communicated with the stm a heartbeat message. This would result in the stm switching back to the Sovereign state.
7. If sensor readings trigger the state of Critical Power, the stm should relay to the obc that it is to switch to the Critical Power state and yield for a moment to allow for the obc to switch its state and log the telemetry accordingly.
8. Potential goal states from the Idle state: Sovereign, Cruise, Critical Power

(3) Cruise

1. The preliminary state for this change needs to be the Idle state.
2. This state occurs as a response from the obc where the obc has received transmission from the ground station declaring permission for the obc to ensure cruise-mode functions.
3. Sensor logging from the stm to the obc is permitted in this state.
4. In this state the stm may receive parameter changes or requests from the obc. The stm should respond accordingly.

5. In this state the stm may receive the parameter change to have it listen to the radio.
6. Potential goal states from the Cruise state: Sovereign, Idle, Critical Power

(4) Critical Power

1. This state is reached when sensor logging has indicated that the power needed to provide minimal functionality from other components in the system is not present.
2. A state change to sovereign should ensue once sensor logging has indicated that power to the overall system has been restored to acceptable values.
3. Potential goal states from the Critical Power state: Sovereign

(5) Autonomous

1. The STM should switch to this state whenever the obc is in a state of non-recoverability. The stm should attempt to cycle states at this point in the explicit purpose of attempting to retain communications with the obc.
2. I would image at this point that the stm may benefit from a gpio pin which can allocate (or deallocate) a voltage line going to the obc. This should be 'pulled high' by default but in an attempt at manually power-cycling the obc for the purpose of functionality retention, this may prove useful.
3. At this state sensor logging and relay to the obc needn't occur.
4. Potential goal states from the Autonomous state: Sovereign, Critical Power

7.2 OBC Communications Service

UartController
The OBC communications thread will gather and package data based on the message type and data in the packets received from the OBC. This communications thread will also relay telemetry data to the OBC as well as periodically relay a heartbeat signal. This thread should be of lesser priority than sensor and state machine updates.

7.2.1 UART Receive

When the UART device receives data from the OBC and a break in transmission (time-out) occurs, the program will receive an interrupt and all other processes will halt until the message is properly stored. (This interrupt is not thread-based, and will interrupt everything happening on all other threads).

7.2.2 UART Parse

UART parsing will occur whenever the OBC communications thread isn't busy storing received information. Parsing am message will often result in a message being sent.

7.2.3 UART Packetization

UART packets will have the following format:

1 byte	1 byte	1 byte	1 byte	0-56 bytes	1 byte	1 byte
Sync High	Sync Low	Msg Type	Msg Length	Msg Data	Checksum msb	Checksum lsb

The packet sizes can range from anywhere from 6 to 64 bytes. Sending more than 64 bytes will result in additional bytes being dropped.

7.2.4 Message Types - From the OBC to the STM

Byte Value	**Type Name**	**Description**
0x00	ack	report packet ack
0x01	err	signal to stm to switch to safe anom state
0x02	rst	signal to stm to restart the fsm
0x03	rsnd	signal to stm to resend last packet
0x04	hb	send a heartbeat signal from the state machine to the stm
0x05	stChng	send a state change signal from the state machine to the stm
0x06	resp	confirm or deny a request (boolean)
0x07	setVal	
0x0A	reqPrm	

7.2.5 Packet Model - From the OBC to the STM

Type	**Length**	**Data**
ack::0x00	0x00	NULL
err::0x01	0x00	NULL
rst::0x02	0x00	NULL
rsnd::0x03	0x01	resends remaining
hb::0x04	0x00	NULL
stChng::0x05	0x01	targetState
valid::0x06	0x01	boolean
setVal::0x07	0x02	targetVal
reqPrm::0x0A	0x01	targetPrm

7.2.6 Message Types - From the STM to the OBC

Byte Value	Type Name	Description
0x00	ack	report packet ack
0x01	hb	hearbeat field
0x02	stUp	state update field
0x03	snsr	sensor specification field
0x04	grp	group field
0x05	prm	parameter field

7.2.7 Packet Model - From the STM to the OBC

Type	Length	Data Type
ack::0x00	0x01	NULL
hb::0x01	0x01	NULL
stUp::0x02	0x01	NULL
snsr::0x03	0x01	NULL
grp::0x04	0x01	0x00 0x00 (enum w 256 max opt.)
prm::0x05	0x01	0x00 (denote param val)

7.3 Sensor Service

Sensor Monitor

The sensor monitor will gather sensor data (thermistors, current monitors, and battery heaters) to monitor the health of the satellite. This thread will analyze the sensor data for anomalies and signal a state change if the OBC does not respond. This thread will have the same priority as the State Machine thread.

I2C Sensors

Most voltage and current monitoring will be handled via readings from INA219 power monitors through the I2C bus. I2C sensors will be polled every second and their data will be stored in a mutex-protected array, sharing memory with the Communications thread.

7.4 State Machine Service

State Machine

The MCU State Machine will be responsible for tracking the operational state of the satellite, it should coordinate with but not be controlled by the state of the OBC. THis thread will have the highest priority to minimize time between error occurrence and state transition.

7.4.1 Startup Mode

Startup Mode

Due to the STM32 not having any conventional writable memory, we need a latch to track whether or not the MCU has undergone startup. This latch will be set when the burn wire is initially enabled, and the output will loop back around to a pin on the MCU.

7.5 STM Pinout

Pin #	Type	Pin Description	Net	Net Notes
1	S	Vbat	3V3_LDO	Low-Dropout Voltage to Power STM
2	I/O	PC13-TAMPER-RTC		
3	I/O	PC 14-O SC 32_IN	PC140SC32_IN	External LF oscillator in
4	I/O	PC 15-O SC 32_OUT	PC140SC32_OUT	External LF oscillator out
5	I	PD0-OSC_IN		
6	O	PD1-OSC_OUT		
7	I/O	NRST	NRST	Inverted RESET
8	I/O	PC0	Solar_TELEM_1	Solar Telemetry Data
9	I/O	PC1	Solar_TELEM_2	Solar Telemetry Data
10	I/O	PC2	Solar_TELEM_3	Solar Telemetry Data
11	I/O	PC3	Solar_TELEM_4	Solar Telemetry Data
12	S	Vssa	GND	Ground
13	S	Vdda	3V3_LDO	Low-Dropout Voltage to Power STM
14	S	PA 0-WK UP		
15	I/O	PA1	BWE0	Burn Wire Enable
16	I/O	PA2	3V3_stmDisableP	Turn off 3V3_Primary Rail (STM)
17	I/O	PA3	3V3_stmDisableS	Turn off 3V3_Secondary Rail
18	S	Vss_4	GND	Ground
19	S	Vdd_4	3V3_LDO	Low-Dropout Voltage to Power STM
20	I/O	PA4	STM_HB	Heart Beat to Watch Dog Circuit
21	I/O	PA5	5V_stmDisableS	Turn off 5V_Secondary Rail

Pin #	Type	Pin Description	Net	Net Notes
22	I/O	PA6	VBatt_stmEnableP	Turn on VBatt_Primary Rail (Battery Heater)
23	I/O	PA7	VBatt_stmDisableS	Turn off VBatt_Secondary Rail
24	I/O	PC4	CEmppt	Controlls if MPTT is charging or not
25	I/O	PC5	5V_stmEnableP1	1/2 control for turning on 5V_Primary Rail (OBC)
26	I/O	PB0	B.5_VD	Single Cell Voltage after Voltage Divider (3/4.2)
27	I/O	PB1	TEMP_TELEM	Telemetry from Temperature sensing on Boba
28	I/O	PB2		
29	I/O	PB10	5V_stmEnableP2	1/2 control for turning on 5V_Primary Rail (OBC)
30	I/O	PB11		
31	S	Vss_1	GND	Ground
32	S	Vdd_1	3V3_LDFO	Low-Dropout Voltage to Power STM
33	I/O	PB12	TEMP_TELEM_2	Telemetry from Temperature sensing on Boba
34	I/O	PB13		
35	I/O	PB14		
36	I/O	PB15		
37	I/O	PC6	OBC_WD	OBC Watchdog circuit output
38	I/O	PC7	OBC_RST	to RST pin on OBC; need to set internal pull-down to turn OBC off

Pin #	**Type**	**Pin Description**	**Net**	**Net Notes**
39	I/O	PC8	StartupWrite	Write to startup memory
40	I/O	PC9	StartupRead	Read from startup memory
41	I/O	PA8	STM_SCL	I2C SCL line
42	I/O	PA9	STM_SDA	I2C SDA line
43	I/O	PA10	I2CFault	HIGH when the I2C buffer detects an error
44	I/O	PA11		
45	I/O	PA12		
46	I/O	PA13	SWDIO	Seria Wire Data for Flashing
47	S	Vss_2	GND	Ground
48	S	Vdd_2	3V3_LDO	Low-Dropout Voltage to Power STM
49	I/O	PA14	SWCLK	Serial Wire Clock for Flashing
50	I/O	PA15		
51	I/O	PC10	UART_RX	UART from OBC to STM
52	I/O	PC11	UART_TX	UART from STM to OBC
53	I/O	PC12		
54	I/O	PD2		
55	I/O	PB3		
56	I/O	PB4	RS485-RX	Output of the receiver on the RS-485 device
57	I/O	PB5	RX_nEnable	Inverted Enable for the receiver on the RS-485 device
58	I/O	PB6	TX_Enable	Enable for the transmitter on the RS-485 device

Pin #	**Type**	**Pin Description**	**Net**	**Net Notes**
59	I/O	PB7	RS485-TX	Input of the transmitter on the RS-485 device
60	I	BOOT0	Boot0	Boot Mode
61	I/O	PB8		
62	I/O	PB9		
63	S	Vss_3	GND	Ground
64	S	Vss_3	3V3_LDO	Low-Dropout Voltage to Power STM