Technical Specification - Secondary Microcontroller (m450-sec-1)

Scope

This technical specification associates signal names that are used in the platform software interfaces with hardware functionality. It is not meant to be a complete description of the functionality and capabilities of the BMU.

Connectors

Low Voltage Connectors

Pin	Description	Type
J101-2	AC Charge Positive Contactor	Low side outputs
J101-3	AC Charge Positive Contactor	High side outputs
J101-4	AC Charge Negative Contactor	High side outputs
J101-6	Main Positive Contactor	High side outputs
J101-7	12V Power (VBAT)	Power
J101-8	12V Ground (VBAT_NEG)	Power
J101-9	AC Charge Negative Contactor	Low side outputs
J101-10	Main Positive Contactor	Low side outputs
J102-1	Main Negative Contactor	High side outputs

Pin	Description	Type
J102-2	DC Charge Positive Contactor	High side outputs
J102-3	Pre-Charge	High side outputs
J102-4	Switched Battery	High side outputs
J102-5	Redundant pack current	Analog Inputs
J102-6	Battery Coolant	Analog Inputs
J102-7	5V Sensor supply (VREF)	Supply
J102-8	DC Charge Negative Contactor	High side outputs
J102-9	Main Negative Contactor	Low side outputs
J102-10	DC Charge Positive Contactor	Low side outputs
J102-11	Pre-Charge	Low side outputs
J102-12	Switched Battery Ground	Supply
J102-16	DC Charge Negative Contactor	Low side outputs
J103-5	FEPS	FEPS
J103-6	Spare (DIGINPUTF)	Digital Inputs
J103-9	PT-CAN Low	CAN
J103-10	PT-CAN High	CAN
J103-14	Secondary Flash Code	Reserved
J103-16	Spare (DIGINPUTE)	Digital Inputs
J103-17	Crash Signal	Digital Inputs
J103-18	Redundant contactor command	Digital Inputs
J103-19	CHVIL Return	Digital Inputs
J103-20	MHVIL Return	Digital Inputs

High side outputs

Note: SBAT must be enabled in order for high side outputs to work.

Function	Gate Monitor	Voltage Monitor	Current Monitor
Main Positive	HSGATE0	MAIN_P_PWR_SENSE	HSDCSNS0
Main Negative	HSGATE1	MAIN_N_PWR_SENSE	HSDCSNS1
DC Charge Positive	HSGATE2	DC_CHG_P_PWR_SENSE	HSDCSNS1
DC Charge Negative	HSGATE3	DC_CHG_N_PWR_SENSE	HSDCSNS2
AC Charge Positive	HSGATE4	AC_CHG_P_PWR_SENSE	HSDCSNS0
AC Charge Negative	HSGATE5	AC_CHG_N_PWR_SENSE	HSDCSNS0
Precharge	HSGATE6	PRECHG_PWR_SENSE	HSDCSNS1
Switched Battery	HSGATE7	SWB_PWR_SENSE	HSDCSNS1
Charge HVIL	HSGATE8	none	HSDCSNS2
Main HVIL	HSGATE9	none	HSDCSNS2
Spare	HSGATE10	none	HSDCSNS2
Spare	HSGATE11	none	HSDCSNS0

Voltage monitors:

To convert measured voltage (Vm) to actual voltage (Va) use the equation, Va=Vm*3.938

The high side outputs are driven by ASICs that control four output channels each. The ASICs are capable of reporting the current monitor signal for one channel concurrently per device. The HSDSEL signals determines which outputs are monitored.

The current monitor signals serve a dual purpose. The output must be configured to produce a steady state output rather than a PWM to serve either purpose.

1. If there are not any hard faults (shorts, over-temperature, over-current, open load off), then the signals will report current according to the transfer function.
2. If a hard fault is detected, then the signal will indicate a fault condition by reporting Vm = 5V.

Since the fault value overlaps the valid current range, and The secondary microcontroller can only read these signals as digital inputs, the signals are most suitable for detecting open load in the off state.

HSDSEL1 State	HSDSEL0 State	Current Sense signal	Output current monitored
0	0	HSDCSNS0	Main Positive
0	1	HSDCSNS0	MHSDOUT11
1	0	HSDCSNS0	AC Charge Negative
1	1	HSDCSNS0	AC Charge Positive
0	0	HSDCSNS1	Switched Battery
0	1	HSDCSNS1	Precharge
1	0	HSDCSNS1	DC Charge Positive
1	1	HSDCSNS1	Main Negative
0	0	HSDCSNS2	Main HVIL
0	1	HSDCSNS2	Charge HVIL
1	0	HSDCSNS2	MHSDOUT10
1	1	HSDCSNS2	DC Charge Negative

Low side outputs

Function	Gate Monitor	Voltage Monitor	Current Monitor
Main Positive	LSDGATE0	MAIN_P_V_SENSE	MAIN_P_CURRENT
Main Negative	LSDGATE1	MAIN_N_V_SENSE	MAIN_N_CURRENT
DC Charge Positive	LSDGATE2	DC_CHG_P_SENSE	DC_CHG_P_CURRENT
DC Charge Negative	LSDGATE3	DC_CHG_N_SENSE	DC_CHG_N_CURRENT
AC Charge Positive	LSDGATE4	AC_CHG_P_SENSE	AC_CHG_P_CURRENT
AC Charge Negative	LSDGATE5	AC_CHG_N_SENSE	AC_CHG_N_CURRENT
Precharge	LSDGATE6	PRECHG_SENSE	PRECHG_CURRENT
Redundant MHVIL	LSDGATE7	none	none
Redundant CHVIL	LSDGATE8	none	none
Spare	LSDGATE9	none	none
Spare	LSDGATE10	none	none

Voltage monitors:

To convert measured voltage (Vm) to actual voltage (Va) use the equation, Va=Vm*3.938

Current monitors:

To convert measured voltage (Vm) to current (I) use the equation, I= Vm*2.174

Digital Inputs

Signal Name	Description
CTR_CMD	Redundant contactor command
CRASH	Crash Signal
MHVIL_RTN_IN	MHVIL Return
CHVIL_RTN_IN	CHVIL Return
DIGINPUTE	Spare Digital Input
DIGINPUTF	Spare Digital Input
HSDSEL1	Determine which outputs are current monitored
HSDSEL0	Determine which outputs are current monitored
PWRGOOD5V	5V power supply status good signal
NPWRGOOD3V3	3V3 power supply status **not** good signal
RIP_STATUS	Interprocessor status signal

Analog Inputs

Signal Name	Description	Transfer function
COOL_AI	Battery Coolant	Va = Vm * 4.000
LEM_AI	Redundant pack current	Va = Vm * 4.000
LV_SUPPLY	Internal Low voltage supply monitor	Va = Vm * 8.000
SENS_SUPPLY	Internal Sensor supply voltage monitor	Va = Vm * 1.162

Note: the transfer function describes how to convert measured voltage (Vm) to actual voltage (Va)

Internal Control Signals

HSDOUTEN:

Secondary microcontroller enable signal for high side outputs

- 1 enabled
- 0 disabled

SWBOUTEN:

Secondary microcontroller enable signal for switched battery output

- 1 enabled
- 0 disabled

AHSDOUTEN::

Secondary microcontroller enable signal for alternate high side outputs

- 1 enabled
- 0 disabled

LSDOUTEN:

Secondary microcontroller enable signal for low side outputs

- 1 enabled
- 0 disabled

ALSDOUTEN:

Secondary microcontroller enable signal for alternate low side outputs

- 1 enabled
- 0 disabled

SHOLDON:

Secondary microcontroller power hold signal for keeping both microcontrollers awake in the absence of another wake source

- 1 enabled
- 0 disabled

SECBATENAB:

Secondary microcontroller switched battery control signal

- 1 enabled
- 0 disabled

NRESET1:

Primary microcontroller enable signal

- 1 Enable the primary microcontroller
- 0 Reset the primary microcontroller

IsoSPI Communication

Two Linear Technologies isoSPI interfaces are implemented.

Name	Description
External IsoSPI	External IsoSPI interface
Internal IsoSPI	Internal LTC2949 IsoSPI interface

A decoder selection signal for external IsoSPI direction is available. The external IsoSPI may be routed to either of two LTC6820 devices. If the external daisy chain consists of LTC6810s, then one LTC6820 may be connected to each end of the daisy chain.

ISOSPI_DIR	Active Connection	6820CS0_N state	6820CS1_N state
0	MOD_ISO_SPI	0	1
1	ISOSPI2E	1	0

6820CS0_N, 6820CS1_N are chip select monitor signals that may be used to confirm the correct operation of ISOSPI_DIR. They are only read by the secondary microcontroller.

FEPS

The ECU can run in one of two system modes: reprogramming mode and application mode. In reprogramming mode, both processors of the ECU can be reprogrammed with application software from a calibration tool. In application mode, the ECU runs the programmed application software.

The ECU enters reprogramming mode either by measuring a dedicated external pin called FEPS at power up, or when attempting to reflash over CCP when the application is not inhibiting reprogramming.

Voltage	System mode
< -16V	Enter reprogramming mode. Use the default CCP settings: **Bus:** PT-CAN **Baud rate:** 500kbps **CRO:** 0x6FB **DTO:** 0x6FA **Station:** 1 Use the default UDS settings: **Bus:** PT-CAN **Baud rate:** 500kbps **Functional ID:** 0x7DF **Physical ID:** 0x7E3 **Response ID:** 0x7EB
> -16V	Enter application mode if valid application software has previously been programmed, otherwise enter reprogramming mode.

CAN

Two CAN interfaces are implemented:

Name	Description
PT-CAN	Vehicle power-train CAN
SEC-CAN	BMU internal CAN to primary microcontroller

Note: Reprogramming over CAN is only supported when the Vehicle Wake (Ignition) signal is asserted.

Secondary Supplemental API Guidelines

Type Guidelines

The secondary microcontroller does not support floating point types. The primitive types used by the application should be restricted to those defined in soe_types.h.

Init Routine Guidelines

To avoid unexpected behavior, the application should only call the secondary digital output API, CAN transmit API, and software watchdog API functions from app_timer_routine() or app_background_routine().

Loop Period Measurement Guidelines

sos_timer_loop_usage_pct() and sos_max_timer_loop_usage_pct() are provided to indicate the time taken to execute the timer loop as a percentage of the timer loop period. If a timer loop overrun has occurred, the values reported by these functions may no longer be accurate.

Similarly, the values reported by sos_bkgnd_loop_usage_pct() and sos_max_bkgnd_loop_usage_pct() may no longer be accurate if a background loop overrun has occurred.

Watchdog Guidelines

The watchdog functionality is not enabled by platform software during initialization. It must be enabled by calling the API from the application software.

ISO diagnostic Guidelines

The ISO-15765 diagnostic configuration API only supports configuration with standard 11-bit CAN identifiers.

Diagnostic Service Information

Supported Diagnostic Services

UDS Service 0x10

DiagnosticSessionControl

Enables different diagnostic sessions

Justification

Required for access to reprogramming mode and some other services

Negative Response Codes

- 0x12 sub-functionNotSupported
- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x22 conditionsNotCorrect
- 0x78 requestCorrectlyReceived-ResponsePending

UDS Service 0x11

ECUReset

Request the ECU to reset

Justification

Required for transition from firmware to application after reprogramming

Limitations

- Limited subfunction support:
    - 0x01 (Hard reset)

Negative Response Codes

- 0x12 sub-functionNotSupported
- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x33 securityAccessDenied

UDS Service 0x22

ReadDataByIdentifier

Read data record values identified by one or more dataIdentifiers

Justification:

Required for reporting values by 16-bit PID

Negative Response Codes

- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x14 responseTooLong
- 0x31 requestOutOfRange
- 0x33 securityAccessDenied

UDS Service 0x23

ReadMemoryByAddress

Read data in memory by address and size

Justification:

Required for reading memory

Limitations

- Physical ID only
- Not available in default session

Negative Response Codes

- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x31 requestOutOfRange
- 0x33 securityAccessDenied
- 0x7F serviceNotSupportedInActiveSession

UDS Service 0x28

CommunicationControl

Switch on/off the transmission/reception of certain messages

Justification:

Required to manage bus load

Limitations

- Not available in default session
- nodeIdentificationNumber not supported
- Limited subfunction support
    - 0x00 enableRxAndTx
    - 0x01 enableRxAndDisableTx
    - 0x02 disableRxAndEnableTx
    - 0x03 disableRxAndTx

Negative Response Codes

- 0x12 sub-functionNotSupported
- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x31 requestOutOfRange
- 0x7F serviceNotSupportedInActiveSession

UDS Service 0x31

RoutineControl

Execute a defined sequence of steps and obtain any relevant results

Justification:

Required for reprogramming support

Limitations

Physical addressing only

Negative Response Codes

- 0x12 sub-functionNotSupported
- 0x13 incorrectMessageLengthOrInvalidFormat
- 0x31 requestOutOfRange
- 0x33 securityAccessDenied
- 0x72 GeneralProgrammingFailure
- 0x78 requestCorrectlyReceived-ResponsePending

UDS Service 0x34

RequestDownload

Initiate a data transfer to the ECU

Justification

Required for transferring data during reprogramming

Limitations

- Physical addressing only
- Supported in reprogramming mode only
- compressionMethod not supported
- encryptionMethod not supported

Negative response codes

- 0x31 requestOutOfRange
- 0x33 securityAccessDenied
- 0x7F serviceNotSupportedInActiveSession

UDS Service 0x36

TransferData

Transfer data to or from the ECU

Justification

Required for transferring data during reprogramming

Limitations

- Physical addressing only
- Supported in reprogramming mode only

Negative response codes

- 0x22 conditionsNotCorrect
- 0x31 requestOutOfRange
- 0x7F serviceNotSupportedInActiveSession

UDS Service 0x37

RequestTransferExit

Terminate a data transfer

Justification

Required to end a data transfer during reprogramming

Limitations

- Physical addressing only
- Supported in reprogramming mode only

Negative response codes

- 0x7F serviceNotSupportedInActiveSession

UDS Service 0x3E

TesterPresent

Indicate that the connection is still present and certain services that have been previously active are to remain active

Justification:

Mandatory for support of other services

Negative Response Codes

- 0x12 sub-functionNotSupported
- 0x13 incorrectMessageLengthOrInvalidFormat