FW device driver manual

Introduction

This document describes the software stack for the **L9962** battery management IC, including device driver, interface, hardware abstraction layers, and utility macros. The stack provides a robust and portable solution for cell monitoring, current sensing, balancing, diagnostics, and configuration via I2C, with STM32 HAL as the reference platform.

1 Main modules description

1.1 Modules and dependencies overview

Figure 1. **Drivers architecture**

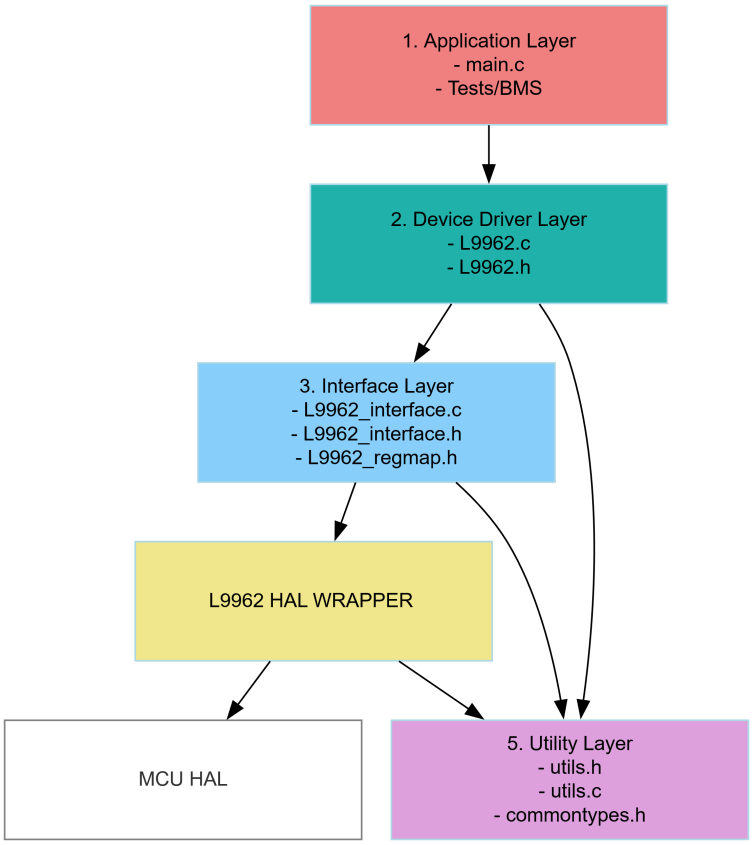
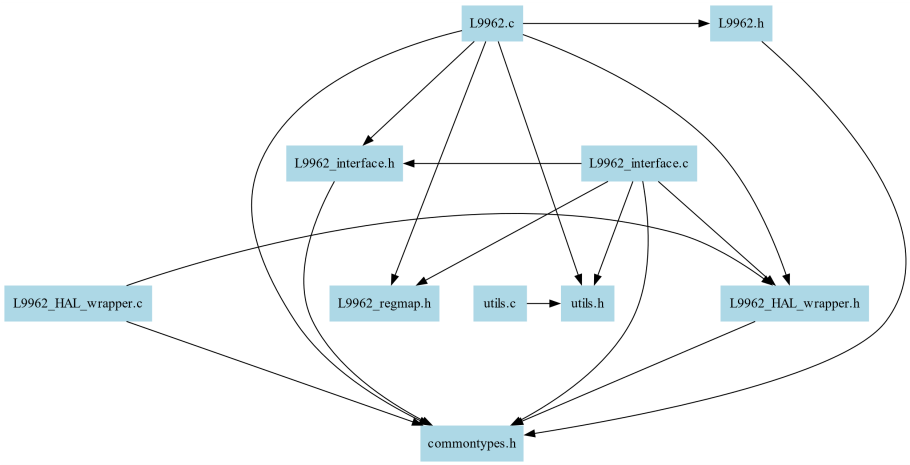


Table 1. **Dependencies table**

Module	Depends On
L9962 Driver	L9962_interface, L9962_regmap, utils, commontypes, L9962_HAL_wrapper
L9962 Interface	L9962_HAL_wrapper, commontypes, utils, L9962_regmap
L9962 HAL Wrapper	STM32 HAL, main.h, commontypes
utils	-
commontypes	-

Figure 2. **Dependencies diagram**



Files:

- L9962.c, L9962.h, L9962_regmap.h

Description:

Implements all device-specific logic for the L9962, including cell monitoring, current measurement, balancing, threshold configuration, diagnostics, and NVM operations.

Key data structures & types:

- CELL_FILTER_T, CURR_FILTER_T, SC_FILTER_T, HS_LS_T, etc.
- Mask enums for VB, die temp, current, balancing, NTC, and cell events.

Important functions:

Initialization, enable/disable, configuration, measurement, diagnostics, NVM, FSM, and mask management.

1.2 L9962 interface module

Files:

- L9962_interface.c, L9962_interface.h

Description:

Handles I²C protocol, CRC calculation, register read/write, and device address management. Abstracts low-level communication for the driver.

Key functions:

- L9962_write_reg, L9962_read_reg, L9962_crc_use, L9962_i2c_crc_check_set/get, etc.

1.3 L9962 HAL wrapper module

Files:

- L9962_HAL_wrapper.c, L9962_HAL_wrapper.h

Description:

Microcontroller (MCU) abstraction layer for I²C, GPIO, and delay. Only this file needs to be ported for a new MCU.

Key Functions:

- HALWRP_I2C_write/read, HALWRAP_read_pin, HALWRAP_write_pin, HALWRP_delay_ms

Hardware Resources:

- I²C interface (e.g., hi2c2)
- GPIOs: WKP, NSHIP, FLTN, RDY

1.4 Utility macros module

Files:

- utils.h, utils.c (currently empty)

Description:

Provides essential macros for register and bitfield manipulation, used throughout the driver and interface layers.

Key Macros:

- UTILS_mask_data(width, offset): Create a bitmask for a field.
- UTILS_set_field_in_reg(reg, field, field_width, field_offset): Set a field in a register.
- UTILS_get_field_from_reg(reg, field_width, field_offset): Extract a field from a register.
- UTILS_SignExtendINT(data, dataSize): Sign-extend an integer value.

Usage Example:

```
uint16_t reg;
UTILS_set_field_in_reg(reg,value, 3, 5); // Set a 3-bit field at offset 5
uint8_t field = UTILS_get_field_from_reg(reg,3, 5);  // Extract the same field
```

Note:

- utils.c is currently a placeholder for future utility function implementations.

2 L9962 driver function descriptions

2.1 Summary table

Table 2. **Summary table**

Function name	Purpose
L9962_CELL_enable_set	Enable monitoring for selected cells via bitmask
L9962_CELL_enable_get	Get current enabled cell bitmask
L9962_CELL_filter_set	Set cell voltage filter time constant
L9962_CELL_filter_get	Get cell voltage filter time constant
L9962_CELL_UV_set	Set cell undervoltage thresholds and counter
L9962_CELL_UV_get	Get cell undervoltage thresholds and counter
L9962_CELL_OV_set	Set cell overvoltage thresholds and counter
L9962_CELL_OV_get	Get cell overvoltage thresholds and counter
L9962_VB_enable_set	Enable/disable VB monitoring and UVLO
L9962_VB_enable_get	Get VB monitoring and UVLO enable status
L9962_NTC_enable_set	Enable/disable NTC monitoring
L9962_NTC_enable_get	Get NTC monitoring enable status
L9962_tmeas_set	Set measurement period in ms
L9962_tmeas_get	Get measurement period in ms
L9962_CURR_enable_set	Enable/disable current measurement and Coulomb counter
L9962_CURR_enable_get	Get current measurement and Coulomb counter enable status
L9962_CURR_filter_set	Set current filter time constant
L9962_CURR_filter_get	Get current filter time constant
L9962_BAL_en_set	Enable balancing for selected cells
L9962_BAL_en_get	Get balancing enable bitmask
L9962_CB_enable_set	Enable/disable charge/discharge circuit breaker
L9962_CB_enable_get	Get charge/discharge circuit breaker enable status
L9962_VB_UV_set	Set VB undervoltage threshold and counter
L9962_VB_OV_set	Set VB overvoltage threshold and counter
L9962_VB_UV_get	Get VB undervoltage threshold and counter
L9962_VB_OV_get	Get VB overvoltage threshold and counter
L9962_NTC_OT_set	Set NTC overtemperature thresholds and counter
L9962_NTC_OT_get	Get NTC overtemperature thresholds and counter
L9962_NTC_UT_set	Set NTC undertemperature threshold and counter
L9962_NTC_UT_get	Get NTC undertemperature threshold and counter
L9962_VB_vs_sum_set	Set VB vs sum threshold
L9962_VB_vs_sum_get	Get VB vs sum threshold
L9962_NTC_MEAS_get	Get NTC measurement value
L9962_VB_diag_get	Get VB diagnostic flags
L9962_NTC_diag_get	Get NTC diagnostic flags

Function name	Purpose
L9962_VB_meas_get	Get VB voltage measurement
L9962_Vcell_meas_get	Get cell voltage measurement
L9962_DIE_TEMP_meas_C_get	Get die temperature in Celsius
L9962_chipid_get	Get silicon and metal ID
L9962_devaddr_get	Get device address
L9962_CELL_sum_meas_get	Get sum of all cell voltages
L9962_CELL_diag_get	Get cell diagnostic flags
L9962_CURR_inst_meas_get	Get instantaneous current measurement
L9962_CURR_overcurr_set	Set overcurrent thresholds and enable
L9962_CURR_overcurr_get	Get overcurrent thresholds and enable
L9962_BAL_threshold_set	Set balancing undervoltage threshold and counter
L9962_BAL_threshold_get	Get balancing undervoltage threshold and counter
L9962_OVC_diag_get	Get overcurrent diagnostic flags
L9962_VB_mask_set	Set VB event/fault mask
L9962_VB_mask_get	Get VB event/fault mask
L9962_DIE_T_mask_set	Set die temperature event/fault mask
L9962_DIE_T_mask_get	Get die temperature event/fault mask
L9962_CURR_mask_set	Set current event/fault mask
L9962_CURR_mask_get	Get current event/fault mask
L9962_DIE_TEMP_diag_get	Get die temperature diagnostic flags
L9962_BAL_mask_set	Set balancing event/fault mask
L9962_BAL_mask_get	Get balancing event/fault mask
L9962_BAL_diag_get	Get balancing diagnostic flags
L9962_CB_mode_set	Set charge/discharge circuit breaker mode (HS/LS)
L9962_CB_mode_get	Get charge/discharge circuit breaker mode (HS/LS)
L9962_NTC_mask_set	Set NTC event/fault mask
L9962_NTC_mask_get	Get NTC event/fault mask
L9962_FUSE_ext_get	Get fuse deployment status
L9962_CELLS_mask_set	Set cell event/fault mask
L9962_CELLS_mask_get	Get cell event/fault mask
L9962_CURR_short_set	Set short-circuit detection parameters
L9962_CURR_short_get	Get short-circuit detection parameters
L9962_CURR_short_diag_get	Get short-circuit diagnostic flags
L9962_FUSE_deploy	Deploy fuse
L9962_global_diag_get	Get global diagnostic status
L9962_NVM_operation	Perform NVM upload/download operation
L9962_NVM_upload_cnt_get	Get NVM upload counter
L9962_CURR_ADC_correction_get	Get current ADC gain correction factor
L9962_CURR_ADC_correction_set	Set current ADC gain correction factor
L9962_i2c_crc_check_set	Enable/disable I2C CRC check
L9962_i2c_crc_check_get	Get I2C CRC check enable status

Function name	Purpose
L9962_CURR_cc_meas_get	Get Coulomb counter measurement
L9962_WAKEUP_set	Enable/disable wake-up function
L9962_NSHIP_get	Get nSHIP pin status
L9962_RDY_get	Get RDY pin status
L9962_FLTN_get	Get FLTN pin status
L9962_FSM_go_to	Set device to FSM state

2.2 Detailed descriptions

L9962_CELL_enable_set

- **Purpose:** Enable monitoring for selected cells using a bitmask.
- **Parameters:** *bit_msk* (bitmask of cells to enable)
- **Returns:** Result code indicating success or error.

L9962_CELL_enable_get

- **Purpose:** Get the current enabled cell bitmask.
- **Parameters:** *bit_msk* (pointer to store bitmask)
- **Returns:** Result code indicating success or error.

L9962_CELL_filter_set

- **Purpose:** Set the filter time constant for cell voltage measurements.
- **Parameters:** *Cell_filter* (filter time constant)
- **Returns:** Result code indicating success or error.

L9962_CELL_filter_get

- **Purpose:** Get the current filter time constant for cell voltage measurements.
- **Parameters:** *Cell_filter* (pointer to store filter time constant)
- **Returns:** Result code indicating success or error.

L9962_CELL_UV_set

- **Purpose:** Set undervoltage thresholds and counter for cell monitoring.
- **Parameters:** *th_uV* (threshold in uV), *severe_th_uV* (severe threshold in uV), *cnt_th* (counter threshold)
- **Returns:** Result code indicating success or error.

L9962_CELL_UV_get

- **Purpose:** Get undervoltage thresholds and counter for cell monitoring.
- **Parameters:** *th_uV*, *severe_th_uV*, *cnt_th* (pointers to store values)
- **Returns:** Result code indicating success or error.

L9962_CELL_OV_set

- **Purpose:** Set overvoltage thresholds and counter for cell monitoring.
- **Parameters:***th_uV*, *severe_th_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_CELL_OV_get

- **Purpose:** Get overvoltage thresholds and counter for cell monitoring.
- **Parameters:***th_uV*, *severe_th_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_VB_enable_set

- **Purpose:** Enable or disable VB (battery voltage) monitoring and UVLO.
- **Parameters:***en* (enable), *uvlo_en* (UVLO enable)
- **Returns:** Result code indicating success or error.

L9962_VB_enable_get

- **Purpose:** Get VB monitoring and UVLO enable status.
- **Parameters:***en*, *uvlo_en* (pointers to store status)
- **Returns:** Result code indicating success or error.

L9962_NTC_enable_set

- **Purpose:** Enable or disable NTC (temperature sensor) monitoring.
- **Parameters:***en* (enable)
- **Returns:** Result code indicating success or error.

L9962_NTC_enable_get

- **Purpose:** Get NTC monitoring enable status.
- **Parameters:***en* (pointer to store status)
- **Returns:** Result code indicating success or error.

L9962_tmeas_set

- **Purpose:** Set the measurement period in milliseconds.
- **Parameters:***time_ms* (measurement period)
- **Returns:** Result code indicating success or error.

L9962_tmeas_get

- **Purpose:** Get the measurement period in milliseconds.
- **Parameters:***time_ms* (pointer to store period)
- **Returns:** Result code indicating success or error.

L9962_CURR_enable_set

- **Purpose:** Enable or disable current measurement and Coulomb counter.
- **Parameters:***inst_en* (instantaneous current enable), *coulomb_cnt* (Coulomb counter enable)
- **Returns:** Result code indicating success or error.

L9962_CURR_enable_get

- **Purpose:** Get current measurement and Coulomb counter enable status.
- **Parameters:** *inst_en*, *coulomb_cnt* (pointers to store status)
- **Returns:** Result code indicating success or error.

L9962_CURR_filter_set

- **Purpose:** Set the filter time constant for current measurements.
- **Parameters:***Curr_filter* (filter time constant)
- **Returns:** Result code indicating success or error.

L9962_CURR_filter_get

- **Purpose:** Get the filter time constant for current measurements.
- **Parameters:***Curr_filter* (pointer to store filter time constant)
- **Returns:** Result code indicating success or error.

L9962_BAL_en_set

- **Purpose:** Enable balancing for selected cells using a bitmask.
- **Parameters:***bit_mask* (bitmask of cells)
- **Returns:** Result code indicating success or error.

L9962_BAL_en_get

- **Purpose:** Get balancing enable bitmask.
- **Parameters:***bit_mask* (pointer to store bitmask)
- **Returns:** Result code indicating success or error.

L9962_CB_enable_set

- **Purpose:** Enable or disable charge/discharge circuit breaker.
- **Parameters:***chg_en* (charge enable), *dchg_en* (discharge enable)
- **Returns:** Result code indicating success or error.

L9962_CB_enable_get

- **Purpose:** Get charge/discharge circuit breaker enable status.
- **Parameters:***chg_en*, *dchg_en* (pointers to store status)
- **Returns:** Result code indicating success or error.

L9962_VB_UV_set

- **Purpose:** Set VB undervoltage threshold and counter.
- **Parameters:***th_uV* (threshold in uV), *cnt_th* (counter threshold)
- **Returns:** Result code indicating success or error.

L9962_VB_OV_set

- **Purpose:** Set VB overvoltage threshold and counter.
- **Parameters:** *th_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_VB_UV_get

- **Purpose:** Get VB undervoltage threshold and counter.
- **Parameters:***th_uV*, *cnt_th* (pointers to store values)
- **Returns:** Result code indicating success or error.

L9962_VB_OV_get

- **Purpose:** Get VB overvoltage threshold and counter.
- **Parameters:***th_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_NTC_OT_set

- **Purpose:** Set NTC overtemperature thresholds and counter.
- **Parameters:***th_uV*, *th_severe_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_NTC_OT_get

- **Purpose:** Get NTC overtemperature thresholds and counter.
- **Parameters:***th_uV*, *th_severe_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_NTC_UT_set

- **Purpose:** Set NTC undertemperature threshold and counter.
- **Parameters:***th_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_NTC_UT_get

- **Purpose:** Get NTC undertemperature threshold and counter.
- **Parameters:***th_uV*, *th_severe_uV*, *cnt_th*
- **Returns:** Result code indicating success or error.

L9962_VB_vs_sum_set

- **Purpose:** Set VB vs sum threshold.
- **Parameters:***th_uV* (threshold in uV)
- **Returns:** Result code indicating success or error.

L9962_VB_vs_sum_get

- **Purpose:** Get VB vs sum threshold.
- **Parameters:** *th_uV* (pointer to store threshold)
- **Returns:** Result code indicating success or error.

L9962_NTC_MEAS_get

- **Purpose:** Get NTC measurement value.
- **Parameters:***NTC_uV* (pointer to store value)
- **Returns:** Result code indicating success or error.

L9962_VB_diag_get

- **Purpose:** Get VB diagnostic flags (UV, OV, UVLO, vs_sum).
- **Parameters:***UV*, *OV*, *UVLO*, *vs_sum* (pointers to store flags)
- **Returns:** Result code indicating success or error.

L9962_NTC_diag_get

- **Purpose:** Get NTC diagnostic flags (UT, OT, OT_severe).
- **Parameters:***UT* (pointer to undertemperature flag), *OT* (pointer to overtemperature flag), *OT_severe* (pointer to severe overtemperature flag)
- **Returns:** Result code indicating success or error.

L9962_VB_meas_get

- **Purpose:** Get VB (battery voltage) measurement.
- **Parameters:***voltage_uV* (pointer to store measured voltage in microvolts)
- **Returns:** Result code indicating success or error.

L9962_Vcell_meas_get

- **Purpose:** Get voltage measurement for a specific cell.
- **Parameters:***cell_idx* (cell index), *voltage_uV* (pointer to store measured voltage)
- **Returns:** Result code indicating success or error.

L9962_DIE_TEMP_meas_C_get

- **Purpose:** Get die temperature in Celsius.
- **Parameters:***Tj_X* (pointer to store temperature in Celsius)
- **Returns:** Result code indicating success or error.

L9962_chipid_get

- **Purpose:** Get silicon and metal ID of the device.
- **Parameters:***silicon* (pointer to silicon ID), *metal* (pointer to metal ID)
- **Returns:** Result code indicating success or error.

L9962_devaddr_get

- **Purpose:** Get device address.
- **Parameters:** *devaddr* (pointer to store device address)
- **Returns:** Result code indicating success or error.

L9962_CELL_sum_meas_get

- **Purpose:** Get sum of all cell voltages.
- **Parameters:***voltage_uV* (pointer to store sum voltage)
- **Returns:** Result code indicating success or error.

L9962_CELL_diag_get

- **Purpose:** Get cell diagnostic flags (UV, OV, OV_severe, UV_severe).
- **Parameters:***UV*, *OV*, *OV_severe*, *UV_severe* (pointers to store flags)
- **Returns:** Result code indicating success or error.

L9962_CURR_inst_meas_get

- **Purpose:** Get instantaneous current measurement.
- **Parameters:***voltage_uV* (pointer to store measured current in microvolts)
- **Returns:** Result code indicating success or error.

L9962_CURR_overcurr_set

- **Purpose:** Set overcurrent detection thresholds and enable.
- **Parameters:***en* (enable), *chg_th_nV* (charge threshold in nV), *dchg_th_nV* (discharge threshold in nV), *th_persist_nV* (persistence threshold in nV)
- **Returns:** Result code indicating success or error.

L9962_CURR_overcurr_get

- **Purpose:** Get overcurrent detection thresholds and enable.
- **Parameters:***en*, *chg_th_nV*, *dchg_th_nV*, *th_persist_nV* (pointers to store values)
- **Returns:** Result code indicating success or error.

L9962_BAL_threshold_set

- **Purpose:** Set balancing undervoltage threshold and counter.
- **Parameters:***UV_uV* (undervoltage threshold in uV), *cnt* (counter threshold)
- **Returns:** Result code indicating success or error.

L9962_BAL_threshold_get

- **Purpose:** Get balancing undervoltage threshold and counter.
- **Parameters:** *UV_uV*, *cnt* (pointers to store values)
- **Returns:** Result code indicating success or error.

L9962_OVC_diag_get

- **Purpose:** Get overcurrent diagnostic flags.
- **Parameters:***ovc_chg*, *ovc_dchg*, *persistent_ovc_chg*, *persistent_ovc_dchg* (pointers to store flags)
- **Returns:** Result code indicating success or error.

L9962_VB_mask_set

- **Purpose:** Set VB event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_VB_mask_get

- **Purpose:** Get VB event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_DIE_T_mask_set

- **Purpose:** Set die temperature event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_DIE_T_mask_get

- **Purpose:** Get die temperature event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_CURR_mask_set

- **Purpose:** Set current event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_CURR_mask_get

- **Purpose:** Get current event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_DIE_TEMP_diag_get

- **Purpose:** Get die temperature diagnostic flags.
- **Parameters:** *ot* (pointer to overtemperature flag)
- **Returns:** Result code indicating success or error.

L9962_BAL_mask_set

- **Purpose:** Set balancing event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_BAL_mask_get

- **Purpose:** Get balancing event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_BAL_diag_get

- **Purpose:** Get balancing diagnostic flags.
- **Parameters:***cell_UV* (pointer to store cell undervoltage flags)
- **Returns:** Result code indicating success or error.

L9962_CB_mode_set

- **Purpose:** Set charge/discharge circuit breaker mode (High-Side/Low-Side).
- **Parameters:***chg_mode* (charge mode), *dchg_mode* (discharge mode)
- **Returns:** Result code indicating success or error.

L9962_CB_mode_get

- **Purpose:** Get charge/discharge circuit breaker mode (High-Side/Low-Side).
- **Parameters:***chg_mode*, *dchg_mode* (pointers to store modes)
- **Returns:** Result code indicating success or error.

L9962_NTC_mask_set

- **Purpose:** Set NTC event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_NTC_mask_get

- **Purpose:** Get NTC event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_FUSE_ext_get

- **Purpose:** Get fuse deployment status.
- **Parameters:** *deploy* (pointer to store deployment status)
- **Returns:** Result code indicating success or error.

L9962_CELLS_mask_set

- **Purpose:** Set cell event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (enable)
- **Returns:** Result code indicating success or error.

L9962_CELLS_mask_get

- **Purpose:** Get cell event/fault mask.
- **Parameters:***mask_sel* (mask selection), *en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_CURR_short_set

- **Purpose:** Set short-circuit detection parameters.
- **Parameters:***en* (enable), *th_uV* (threshold in uV), *sc_filter* (filter selection), *th_persist_uV* (persistence threshold)
- **Returns:** Result code indicating success or error.

L9962_CURR_short_get

- **Purpose:** Get short-circuit detection parameters.
- **Parameters:***en*, *th_uV*, *sc_filter*, *th_persist_uV* (pointers to store values)
- **Returns:** Result code indicating success or error.

L9962_CURR_short_diag_get

- **Purpose:** Get short-circuit diagnostic flags.
- **Parameters:***sc_dchg*, *persist_sc_dchg* (pointers to store flags)
- **Returns:** Result code indicating success or error.

L9962_FUSE_deploy

- **Purpose:** Deploy the fuse.
- **Parameters:** None
- **Returns:** Result code indicating success or error.

L9962_global_diag_get

- **Purpose:** Get global diagnostic status.
- **Parameters:** *sel* (diagnostic selection), *value* (pointer to store value)
- **Returns:** Result code indicating success or error.

L9962_NVM_operation

- **Purpose:** Perform NVM upload or download operation.
- **Parameters:***OP* (operation: upload or download)
- **Returns:** Result code indicating success or error.

L9962_NVM_upload_cnt_get

- **Purpose:** Get NVM upload counter.
- **Parameters:***value* (pointer to store counter)
- **Returns:** Result code indicating success or error.

L9962_CURR_ADC_correction_get

- **Purpose:** Get current ADC gain correction factor.
- **Parameters:***gain_factor* (pointer to store gain factor)
- **Returns:** Result code indicating success or error.

L9962_CURR_ADC_correction_set

- **Purpose:** Set current ADC gain correction factor.
- **Parameters:***gain_factor* (gain correction value)
- **Returns:** Result code indicating success or error.

L9962_i2c_crc_check_set

- **Purpose:** Enable or disable I2C CRC check.
- **Parameters:***en* (enable)
- **Returns:** Result code indicating success or error.

L9962_i2c_crc_check_get

- **Purpose:** Get I2C CRC check enable status.
- **Parameters:***en* (pointer to store enable status)
- **Returns:** Result code indicating success or error.

L9962_CURR_cc_meas_get

- **Purpose:** Get Coulomb counter measurement.
- **Parameters:***voltage_nV* (pointer to store measured value in nV), *smp_cnt* (pointer to sample count), *cc_sat* (pointer to saturation flag)
- **Returns:** Result code indicating success or error.

L9962_WAKEUP_set

- **Purpose:** Enable or disable wake-up function.
- **Parameters:** *wkup_en* (enable)
- **Returns:** Result code indicating success or error.

L9962_NSHIP_get

- **Purpose:** Get nSHIP pin status.
- **Parameters:***nship_en* (pointer to store status)
- **Returns:** Result code indicating success or error.

L9962_RDY_get

- **Purpose:** Get RDY pin status.
- **Parameters:***rdy* (pointer to store status)
- **Returns:** Result code indicating success or error.

L9962_FLTN_get

- **Purpose:** Get FLTN pin status.
- **Parameters:***fltn* (pointer to store status)
- **Returns:** Result code indicating success or error.

L9962_FSM_go_to

- **Purpose:** Set device finite state machine (FSM) state.
- **Parameters:***sts* (desired state)
- **Returns:** Result code indicating success or error.

Additional notes

- All functions return a RESULT_T type, which indicates the success or failure of the operation. Typical values are implementation-specific but usually include codes for success, invalid parameters, or hardware communication errors.
- For functions that use pointers as parameters, the caller must ensure that the pointers are valid and point to sufficient memory to store the results.
- Many functions use bitmasks or enumerated types to select channels, features, or configuration options. Refer to the corresponding enum definitions in the header for valid values.
- Measurement functions typically return values in microvolts (uV), nanovolts (nV), or degrees Celsius (°C), depending on the context.
- Mask and diagnostic functions allow fine-grained control and monitoring of fault and event reporting for safety-critical applications.
- NVM (Non-Volatile Memory) operations are used to save or restore configuration settings to or from the internal device memory.

Example usage

```c uint16_t cell_mask = 0x0F; // Enable first 4 cells
RESULT_T res = L9962_CELL_enable_set(cell_mask);
if (res == RESULT_OK) { // Cells enabled successfully }
uint32_t cell_voltage;
res = L9962_Vcell_meas_get(2, &cell_voltage); // Get voltage for cell 2
if (res == RESULT_OK) { // Use cell_voltage }
```

3 L9962 driver functions description

3.1 Summary table

Table 3. **Summary table**

Function Name	Purpose/Description
L9962_CELL_enable_set/get	Enable/disable cells (bitmask)
L9962_CELL_filter_set/get	Set/get cell filter
L9962_CELL_UV_set/get	Set/get cell undervoltage threshold
L9962_CELL_OV_set/get	Set/get cell overvoltage threshold
L9962_VB_enable_set/get	Enable/disable VB and UVLO
L9962_NTC_enable_set/get	Enable/disable NTC
L9962_tmeas_set/get	Set/get measurement time
L9962_CURR_enable_set/get	Enable/disable current measurement, coulomb counter
L9962_CURR_filter_set/get	Set/get current filter
L9962_BAL_en_set/get	Enable/disable cell balancing (bitmask)
L9962_CB_enable_set/get	Enable/disable charge/discharge
L9962_VB_UV_set/get	Set/get VB undervoltage threshold
L9962_VB_OV_set/get	Set/get VB overvoltage threshold
L9962_NTC_OT_set/get	Set/get NTC overtemperature threshold
L9962_NTC_UT_set/get	Set/get NTC undertemperature threshold
L9962_VB_vs_sum_set/get	Set/get VB vs sum threshold
L9962_NTC_MEAS_get	Get NTC voltage measurement
L9962_VB_diag_get	Get VB diagnostic flags
L9962_NTC_diag_get	Get NTC diagnostic flags
L9962_VB_meas_get	Get VB voltage measurement
L9962_Vcell_meas_get	Get cell voltage measurement
L9962_DIE_TEMP_meas_C_get	Get die temperature (Celsius)
L9962_chipid_get	Get chip silicon and metal ID
L9962_devaddr_get	Get device address
L9962_CELL_sum_meas_get	Get sum of all cell voltages
L9962_CURR_inst_meas_get	Get instantaneous current measurement
L9962_CURR_overcurr_set/get	Set/get overcurrent protection
L9962_BAL_threshold_set/get	Set/get balancing undervoltage threshold
L9962_OVC_diag_get	Get overcurrent diagnostic flags
L9962_CURR_cc_meas_get	Get current accumulator measurement
L9962_CURR_short_set/get	Set/get short-circuit protection
L9962_CURR_short_diag_get	Get short-circuit diagnostic flags
L9962_FUSE_deploy	Deploy fuse
L9962_FUSE_ext_get	Get fuse deployment status
L9962_global_diag_get	Get global diagnostic (e.g., CRC config fail)

Function Name	Purpose/Description
L9962_NVM_operation	Perform NVM upload/download
L9962_NVM_upload_cnt_get	Get NVM upload count
L9962_CURR_ADC_correction_set/get	Set/get current ADC correction gain
L9962_i2c_crc_check_set/get	Enable/disable I2C CRC check
L9962_WAKEUP_set	Set WKP pin
L9962_NSHIP_get	Get NSHIP pin status
L9962_RDY_get	Get RDY pin status
L9962_FLTN_get	Get FLTN pin status
L9962_FSM_go_to	Set device FSM state
Mask management functions	Set/get event masks (VB, Die T, Current, etc.)

3.2 Detailed descriptions

See previous detailed section in Chapter 2.2 for function-by-function descriptions, parameters, and return values.

4 General notes

- **Portability:** Only L9962_HAL_wrapper needs to be ported for a new MCU.
- **Communication:** All register access is via I2C, with optional CRC checking.
- **Device Abstraction:** All device logic is in the driver, with clear separation from hardware.
- **Error Handling:** All functions return a RESULT_T code for robust error detection.
- **Masking:** Fine-grained event masking for all major protection/diagnostic events.
- **NVM:** Upload/download and upload count supported.
- **FSM:** Shipment, Standby, and Normal states supported.

5 Error codes

```
typedef enum {
RES_OK,
RES_FAIL,
RES_HAL_ERR,
RES_CRC_ERR,
RES_INVALID_PARAM,
} RESULT_T;
```

6 Bitfield macros and register map

All register addresses and bitfield offsets/widths are defined in `L9962_regmap.h`. Use utility macros for bitfield manipulation (e.g., `UTILS_set_field_in_reg`, `UTILS_get_field_from_reg`).

Revision history

Table 4. **Document revision history**

Date	Version	Changes
08-Jan-2026	1	Initial release.

Contents

1 **Main modules description** 2
1.1 Modules and dependencies overview 2
1.2 L9962 interface module 3
1.3 L9962 HAL wrapper module 3
1.4 Utility macros module 4
2 **L9962 driver function descriptions** 5
2.1 Summary table 5
2.2 Detailed descriptions 7
3 **L9962 driver functions description** 16
3.1 Summary table 16
3.2 Detailed descriptions 17
4 **General notes** 18
5 **Error codes** 19
6 **Bitfield macros and register map** 20
Revision history 21
List of tables 23
List of figures 24

List of tables

Table 1. Dependencies table 2
Table 2. Summary table 5
Table 3. Summary table 16
Table 4. Document revision history 21

List of figures

Figure 1. Drivers architecture 2
Figure 2. Dependencies diagram 3

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.