eMD ICM426xx Driver API

2.0.1

Generated by Doxygen 1.8.11

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 4

Module Documentation

4.1 Icm426xx driver high level functions

High-level function to setup an Icm426xx device.

Files

- file Icm426xxDriver_HL.h

  High-level function to setup an Icm426xx device.

Classes

- struct inv_icm426xx_sensor_event_t

  Sensor event structure definition.
- struct inv_icm426xx

  Icm426xx driver states definition.
- struct inv_icm426xx_interrupt_parameter_t

  Icm426xx set of interrupt enable flag.

Macros

- #define INV_ICM426XX_LIGHTWEIGHT_DRIVER 0

  Ligthen driver logic by stripping out procedures on transitions.
- #define PLL_SCALE_FACTOR_Q24 (1UL<<24)

  Scale factor and max ODR Dependant of chip.
- #define ACCEL_CONFIG0_FS_SEL_MAX ICM426XX_ACCEL_CONFIG0_FS_SEL_16g

  Max FSR values for accel and gyro Dependant of chip.
- #define RTC_SUPPORTED 0

  RTC Support flag Define whether the RTC mode is supported Dependant of chip.
- #define ICM426XX_FIFO_MIRRORING_SIZE 16 * 129

  Icm426xx maximum buffer size mirrored from FIFO at polling time.
- #define ICM426XX_DEFAULT_WOM_THS_MG 52>>2 /* = 52mg/4 */

  Default value for the WOM threshold Resolution of the threshold is ∼= 4mg.
- #define ICM426XX_ACC_STARTUP_TIME_US 20000U

  Icm426xx Accelerometer start-up time before having correct data.
- #define ICM426XX_GYR_STARTUP_TIME_US 60000U

  Icm426xx Gyroscope start-up time before having correct data.

Enumerations

Functions

- int inv_icm426xx_set_reg_bank (struct inv_icm426xx ∗s, uint8_t bank)

  Set register bank index.
- int inv_icm426xx_init (struct inv_icm426xx ∗s, struct inv_icm426xx_serif ∗serif, void(∗sensor_event_cb)(inv←_icm426xx_sensor_event_t ∗event))

  Configure the serial interface used to access the device and execute hardware initialization.
- int inv_icm426xx_device_reset (struct inv_icm426xx ∗s)

  Perform a soft reset of the device.
- int inv_icm426xx_get_who_am_i (struct inv_icm426xx ∗s, uint8_t ∗who_am_i)

  return WHOAMI value
- int inv_icm426xx_force_clock_source (struct inv_icm426xx ∗s, ICM426XX_INTF_CONFIG1_ACCEL_LP_←CLK_t clk_src)

  Configure Accel clock source.
- int inv_icm426xx_enable_accel_low_power_mode (struct inv_icm426xx ∗s)

  Enable/put accel in low power mode.
- int inv_icm426xx_enable_accel_low_noise_mode (struct inv_icm426xx ∗s)

  Enable/put accel in low noise mode.
- int inv_icm426xx_disable_accel (struct inv_icm426xx ∗s)

  Disable all 3 axes of accel.
- int inv_icm426xx_enable_gyro_low_noise_mode (struct inv_icm426xx ∗s)

  Enable/put gyro in low noise mode.
- int inv_icm426xx_disable_gyro (struct inv_icm426xx ∗s)

  Disable all 3 axes of gyro.
- int inv_icm426xx_enable_fsync (struct inv_icm426xx ∗s)

  Enable fsync tagging functionnality.
- int inv_icm426xx_disable_fsync (struct inv_icm426xx ∗s)

  Disable fsync tagging functionnality.
- int inv_icm426xx_configure_timestamp_resolution (struct inv_icm426xx ∗s, ICM426XX_TMST_CONFIG_←RESOL_t resol)

  Configure timestamp resolution from FIFO.
- int inv_icm426xx_set_config_ibi (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt_←to_configure)

  Configure which interrupt source can trigger ibi interruptions.
- int inv_icm426xx_get_config_ibi (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt_←to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_set_config_int1 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt←_to_configure)

  Configure which interrupt source can trigger INT1.
- int inv_icm426xx_get_config_int1 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt←_to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_set_config_int2 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt←_to_configure)

  Configure which interrupt source can trigger INT2.
- int inv_icm426xx_get_config_int2 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt←_to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_get_data_from_registers (struct inv_icm426xx ∗s)

Read all registers containing data (tempereature, accelerometer and gyroscope).

- int inv_icm426xx_get_data_from_fifo (struct inv_icm426xx ∗s)

  Read all available packets from the FIFO.

- uint32_t inv_icm426xx_convert_odr_bitfield_to_us (uint32_t odr_bitfield)

  Converts ICM426XX_ACCEL_CONFIG0_ODR_t or ICM426XX_GYRO_CONFIG0_ODR_t enums to period expressed in us.

- int inv_icm426xx_set_accel_frequency (struct inv_icm426xx ∗s, const ICM426XX_ACCEL_CONFIG0_OD←R_t frequency)

  Configure accel Output Data Rate.

- int inv_icm426xx_set_gyro_frequency (struct inv_icm426xx ∗s, const ICM426XX_GYRO_CONFIG0_ODR←_t frequency)

  Configure gyro Output Data Rate.

- int inv_icm426xx_set_accel_fsr (struct inv_icm426xx ∗s, ICM426XX_ACCEL_CONFIG0_FS_SEL_t accel←_fsr_g)

  Set accel full scale range.

- int inv_icm426xx_get_accel_fsr (struct inv_icm426xx ∗s, ICM426XX_ACCEL_CONFIG0_FS_SEL_t ∗accel←_fsr_g)

  Access accel full scale range.

- int inv_icm426xx_set_gyro_fsr (struct inv_icm426xx ∗s, ICM426XX_GYRO_CONFIG0_FS_SEL_t gyro_fsr←_dps)

  Set gyro full scale range.

- int inv_icm426xx_get_gyro_fsr (struct inv_icm426xx ∗s, ICM426XX_GYRO_CONFIG0_FS_SEL_t ∗gyro_←fsr_dps)

  Access gyro full scale range.

- int inv_icm426xx_set_accel_lp_avg (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_ACC←EL_FILT_AVG_t acc_avg)

  Set accel Low-Power averaging value.

- int inv_icm426xx_set_accel_ln_bw (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_ACC←EL_FILT_BW_t acc_bw)

  Set accel Low-Noise bandwidth value.

- int inv_icm426xx_set_gyro_ln_bw (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_GYRO←_FILT_BW_t gyr_bw)

  Set gyro Low-Noise bandwidth value.

- int inv_icm426xx_reset_fifo (struct inv_icm426xx ∗s)

  reset ICM426XX fifo

- int inv_icm426xx_enable_timestamp_to_register (struct inv_icm426xx ∗s)

  Enable the 20bits-timestamp register access to read in a reliable way the strobed timestamp.

- int inv_icm426xx_disable_timestamp_to_register (struct inv_icm426xx ∗s)

  Disable the 20bits-timestamp register access.

- int inv_icm426xx_get_current_timestamp (struct inv_icm426xx ∗s, uint32_t ∗icm_time)

  Get the timestamp value of icm426xx from register.

- int inv_icm426xx_enable_clkin_rtc (struct inv_icm426xx ∗s, uint8_t enable)

  Enable or disable CLKIN/RTC capability.

- int inv_icm426xx_get_clkin_rtc_status (struct inv_icm426xx ∗s)

  Get CLKIN/RTC feature status.

- int inv_icm426xx_enable_high_resolution_fifo (struct inv_icm426xx ∗s)

  Enable 20 bits raw acc and raw gyr data in fifo.

- int inv_icm426xx_disable_high_resolution_fifo (struct inv_icm426xx ∗s)

  Disable 20 bits raw acc and raw gyr data in fifo.

- int inv_icm426xx_configure_fifo (struct inv_icm426xx ∗s, INV_ICM426XX_FIFO_CONFIG_t fifo_config)

  Configure Fifo to select the way data are gathered.

- int inv_icm426xx_configure_fifo_wm (struct inv_icm426xx ∗s, uint16_t wm)

Configure Fifo watermark (also refered to as fifo threshold)

- uint32_t inv_icm426xx_get_fifo_timestamp_resolution_us_q24 (struct inv_icm426xx ∗s)

  Get FIFO timestamp resolution.
- uint32_t inv_icm426xx_get_reg_timestamp_resolution_us_q24 (struct inv_icm426xx ∗s)

  Get register timestamp resolution.
- const char ∗ inv_icm426xx_get_version (void)

  Return driver version x.y.z-suffix as a char array.

4.1.1 Detailed Description

High-level function to setup an Icm426xx device.

4.1.2 Macro Definition Documentation

4.1.2.1 #define ICM426XX_FIFO_MIRRORING_SIZE 16 ∗ 129

Icm426xx maximum buffer size mirrored from FIFO at polling time.

Warning

fifo_idx type variable must be large enough to parse the FIFO_MIRRORING_SIZE

4.1.2.2 #define INV_ICM426XX_LIGHTWEIGHT_DRIVER 0

Ligthen driver logic by stripping out procedures on transitions.

In the nominal case, ie. when sensors are enabled and their output has settled, ICM-426XX will not need the logic to handle each transition. They are part of each API function so this code will be linked in regardless. This might not be desirable for the most size-constrained platforms and it can be avoided by setting this define to 1.

4.1.3 Enumeration Type Documentation

4.1.3.1 enum INV_ICM426XX_FIFO_CONFIG_t

Configure Fifo usage.

Enumerator

INV_ICM426XX_FIFO_DISABLED	Fifo is disabled and data source is sensors registers.
INV_ICM426XX_FIFO_ENABLED	Fifo is used as data source.

4.1.3.2 enum inv_icm426xx_sensor

Sensor identifier for UI control and OIS function.

Enumerator

INV_ICM426XX_SENSOR_ACCEL	Accelerometer (UI control path)
INV_ICM426XX_SENSOR_GYRO	Gyroscope (UI control path)
INV_ICM426XX_SENSOR_FSYNC_EVENT	Used by OIS and UI control layers.
INV_ICM426XX_SENSOR_OIS	Only used by OIS layer.
INV_ICM426XX_SENSOR_TEMPERATURE	Chip temperature, enabled by default. However, it will be reported only if Accel and/or Gyro are also enabled. The Temperature's ODR (Output Data Rate) will match the ODR of Accel or Gyro, or the fastest if both are enabled
INV_ICM426XX_SENSOR_TAP	Tap and Double tap.
INV_ICM426XX_SENSOR_DMP_PEDOMETER_EVENT	Pedometer: step is detected.
INV_ICM426XX_SENSOR_DMP_PEDOMETER_COUNT	Pedometer: step counter.
INV_ICM426XX_SENSOR_DMP_TILT	Tilt.
INV_ICM426XX_SENSOR_DMP_R2W	Raise to wake.

4.1.4 Function Documentation

4.1.4.1 int inv_icm426xx_configure_fifo (struct inv_icm426xx ∗ *s*, INV_ICM426XX_FIFO_CONFIG_t *fifo_config*)

Configure Fifo to select the way data are gathered.

Parameters

`in`	*fifo_config*	Fifo configuration method : if enabled data are comming from fifo and Interrupt is configured on Fifo Watermark if disabled data are comming from sensor registers and Interrupt is configured on Data ready

See also

INV_ICM426XX_FIFO_CONFIG_t

4.1.4.2 int inv_icm426xx_configure_fifo_wm (struct inv_icm426xx ∗ *s*, uint16_t *wm*)

Configure Fifo watermark (also refered to as fifo threshold)

Parameters

`in`	*wm*	Watermark value

4.1.4.3 int inv_icm426xx_configure_timestamp_resolution (struct inv_icm426xx ∗ *s*, ICM426XX_TMST_CONFIG_RESOL_t *resol*)

Configure timestamp resolution from FIFO.

Parameters

in	*resol*	The expected resolution of the timestamp. See enum ICM426XX_TMST_CONFIG_RESOL_t.

Returns

0 on success, negative value on error.

Warning

The resolution will have no effect if RTC is enabled

4.1.4.4 uint32_t inv_icm426xx_convert_odr_bitfield_to_us (uint32_t *odr_bitfield*)

Converts ICM426XX_ACCEL_CONFIG0_ODR_t or ICM426XX_GYRO_CONFIG0_ODR_t enums to period expressed in us.

Parameters

in	*odr_bitfield*	An ICM426XX_ACCEL_CONFIG0_ODR_t or ICM426XX_GYRO_CONFIG0_ODR_t enum

Returns

The corresponding period expressed in us

4.1.4.5 int inv_icm426xx_device_reset (struct inv_icm426xx ∗ *s*)

Perform a soft reset of the device.

Returns

0 on success, negative value on error.

4.1.4.6 int inv_icm426xx_disable_accel (struct inv_icm426xx ∗ *s*)

Disable all 3 axes of accel.

Returns

0 on success, negative value on error.

If both accel and gyro are turned off as a result of this function, they will also be removed from the FIFO and a FIFO reset will be performed (to guarantee no side effects until the next enable sensor call)

Warning

inv_icm426xx::register_cache::pwr_mngt_0_reg is modified by this function

4.1.4.7 int inv_icm426xx_disable_fsync (struct inv_icm426xx ∗ *s*)

Disable fsync tagging functionnality.

In details it:

- disables fsync
- disables timestamp to registers. Once fysnc is disabled timestamp is pushed to fifo instead of fsync counter. So in order to decrease power consumption, timestamp is no more available in registers.
- disables fsync related interrupt

  Returns

  0 on success, negative value on error.

4.1.4.8 int inv_icm426xx_disable_gyro (struct inv_icm426xx ∗ *s*)

Disable all 3 axes of gyro.

Returns

0 on success, negative value on error.

If both accel and gyro are turned off as a result of this function, they will also be removed from the FIFO and a FIFO reset will be performed (to guarantee no side effects until the next enable sensor call)

Warning

inv_icm426xx::register_cache::pwr_mngt_0_reg is modified by this function

4.1.4.9 int inv_icm426xx_disable_high_resolution_fifo (struct inv_icm426xx ∗ *s*)

Disable 20 bits raw acc and raw gyr data in fifo.

Returns

0 on success, negative return code otherwise

4.1.4.10 int inv_icm426xx_disable_timestamp_to_register (struct inv_icm426xx ∗ *s*)

Disable the 20bits-timestamp register access.

Register read always return 0's.

Returns

0 on success, negative value on error.

4.1.4.11 int inv_icm426xx_enable_accel_low_noise_mode (struct **inv_icm426xx** ∗ *s*)

Enable/put accel in low noise mode.

Returns

0 on success, negative value on error.

It enables accel and gyro data in the FIFO (so the packet format is 16 bytes). If called first, the configuration will be applied, otherwise it will be ignored if the FIFO is not empty (but since the new configuration is identical it is not a issue).

Warning

inv_icm426xx::register_cache::pwr_mngt_0_reg is modified by this function

4.1.4.12 int inv_icm426xx_enable_accel_low_power_mode (struct **inv_icm426xx** ∗ *s*)

Enable/put accel in low power mode.

Returns

0 on success, negative value on error.

It enables accel and gyro data in the FIFO (so the packet format is 16 bytes). If called first, the configuration will be applied, otherwise it will be ignored if the FIFO is not empty (but since the new configuration is identical it is not a issue).

Warning

inv_icm426xx::register_cache::pwr_mngt_0_reg is modified by this function

4.1.4.13 int inv_icm426xx_enable_clkin_rtc (struct **inv_icm426xx** ∗ *s*, uint8_t *enable*)

Enable or disable CLKIN/RTC capability.

Parameters

`in`	*enable*	1 if external 32kHz is provided to ICM, 0 otherwise

Returns

0 on success, negative value on error.

Warning

In case CLKIN is disabled, it is recommended to call inv_icm426xx_configure_timestamp_resolution() just afterwards so that timestamp resolution is in line with system request
inv_icm426xx::register_cache::intf_cfg_1_reg is modified by this function

4.1.4.14 int inv_icm426xx_enable_fsync (struct inv_icm426xx ∗ *s*)

Enable fsync tagging functionnality.

In details it:

- enables fsync
- enables timestamp to registers. Once fysnc is enabled fsync counter is pushed to fifo instead of timestamp. So timestamp is made available in registers. Note that this increase power consumption.
- enables fsync related interrupt

  Returns

  0 on success, negative value on error.

4.1.4.15 int inv_icm426xx_enable_gyro_low_noise_mode (struct inv_icm426xx ∗ *s*)

Enable/put gyro in low noise mode.

Returns

0 on success, negative value on error.

It enables gyro and accel data in the FIFO (so the packet format is 16 bytes). If called first, the configuration will be applied, otherwise it will be ignored if the FIFO is not empty (but since the new configuration is identical it is not a issue).

Warning

inv_icm426xx::register_cache::pwr_mngt_0_reg is modified by this function

4.1.4.16 int inv_icm426xx_enable_high_resolution_fifo (struct inv_icm426xx ∗ *s*)

Enable 20 bits raw acc and raw gyr data in fifo.

Returns

0 on success, negative return code otherwise

4.1.4.17 int inv_icm426xx_enable_timestamp_to_register (struct inv_icm426xx ∗ *s*)

Enable the 20bits-timestamp register access to read in a reliable way the strobed timestamp.

To do that, the fine clock is forced enabled at some power cost.

Returns

0 on success, negative value on error.

4.1.4.18 int inv_icm426xx_force_clock_source (struct inv_icm426xx ∗ *s*, ICM426XX_INTF_CONFIG1_ACCEL_LP_CLK_t *clk_src*)

Configure Accel clock source.

Parameters

in	*new*	clock source to use

Returns

0 on success, negative value on error

Note

Transitions when enabling/disabling sensors are already handled by the driver. This function is here only to force a specific clock source and shall be used with care.

4.1.4.19 int inv_icm426xx_get_accel_fsr (struct inv_icm426xx ∗ *s*, ICM426XX_ACCEL_CONFIG0_FS_SEL_t ∗ *accel_fsr_g*)

Access accel full scale range.

Parameters

out	*accel_fsr←_g*	current full scale range.

See also

ICM426XX_ACCEL_CONFIG0_FS_SEL_t.

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::accel_cfg_0_reg is relied upon by this function

4.1.4.20 int inv_icm426xx_get_clkin_rtc_status (struct inv_icm426xx ∗ *s*)

Get CLKIN/RTC feature status.

Returns

0 if CLKIN is disabled, 1 if enabled.

Warning

In case CLKIN is disabled, it is recommended to call inv_icm426xx_configure_timestamp_resolution() just afterwards so that timestamp resolution is in line with system request
inv_icm426xx::register_cache::intf_cfg_1_reg is relied upon by this function

4.1.4.21 int inv_icm426xx_get_config_ibi (struct inv_icm426xx ∗ *s*, inv_icm426xx_interrupt_parameter_t ∗ *interrupt_to_configure*)

Retrieve interrupts configuration.

Parameters

`in`	*interrupt_to_configure*	structure with the corresponding state to manage IBI interruptions.

Returns

0 on success, negative value on error.

4.1.4.22 int inv_icm426xx_get_config_int1 (struct inv_icm426xx ∗ *s*, inv_icm426xx_interrupt_parameter_t ∗ *interrupt_to_configure*)

Retrieve interrupts configuration.

Parameters

`in`	*interrupt_to_configure*	structure with the corresponding state to manage INT1.

Returns

0 on success, negative value on error.

4.1.4.23 int inv_icm426xx_get_config_int2 (struct inv_icm426xx ∗ *s*, inv_icm426xx_interrupt_parameter_t ∗ *interrupt_to_configure*)

Retrieve interrupts configuration.

Parameters

`in`	*interrupt_to_configure*	structure with the corresponding state to manage INT2.

Returns

0 on success, negative value on error.

4.1.4.24 int inv_icm426xx_get_current_timestamp (struct inv_icm426xx ∗ *s*, uint32_t ∗ *icm_time*)

Get the timestamp value of icm426xx from register.

Parameters

`in`	*icm_time*	timestamp read from register

Returns

0 on success, negative value on error.

Warning

Prior to call this API, the read access to timestamp register must be enabled (see inv_icm426xx_enable_↩timestamp_to_register() function)

4.1.4.25 int inv_icm426xx_get_data_from_fifo (struct inv_icm426xx ∗ *s*)

Read all available packets from the FIFO.

For each packet function builds a sensor event containing packet data and validity information. Then it calls sensor↩_event_cb funtion passed in parameter of inv_icm426xx_init function for each packet.

Returns

0 on success, negative value on error.

4.1.4.26 int inv_icm426xx_get_data_from_registers (struct inv_icm426xx ∗ *s*)

Read all registers containing data (tempereature, accelerometer and gyroscope).

Then it calls sensor_event_cb funtion passed in parameter of inv_icm426xx_init function for each packet

Returns

0 on success, negative value on error.

4.1.4.27 uint32_t inv_icm426xx_get_fifo_timestamp_resolution_us_q24 (struct inv_icm426xx ∗ *s*)

Get FIFO timestamp resolution.

Returns

the timestamp resolution in us as a q24 or 0 in case of error

4.1.4.28 int inv_icm426xx_get_gyro_fsr (struct inv_icm426xx ∗ *s*, ICM426XX_GYRO_CONFIG0_FS_SEL_t ∗ *gyro_fsr_dps*)

Access gyro full scale range.

Parameters

`out`	*gyro_fsr_dps*	current full scale range.

See also

ICM426XX_GYRO_CONFIG0_FS_SEL_t.

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::gyro_cfg_0_reg is relied upon by this function

4.1.4.29 uint32_t inv_icm426xx_get_reg_timestamp_resolution_us_q24 (struct inv_icm426xx * *s*)

Get register timestamp resolution.

Returns

the timestamp resolution in us as a q24 or 0 in case of error

4.1.4.30 const char* inv_icm426xx_get_version (void)

Return driver version x.y.z-suffix as a char array.

Return values

driver	version a char array "x.y.z-suffix"

4.1.4.31 int inv_icm426xx_get_who_am_i (struct inv_icm426xx * *s*, uint8_t * *who_am_i*)

return WHOAMI value

Parameters

out	*who_↩ am_i*	WHOAMI for device

Returns

0 on success, negative value on error

4.1.4.32 int inv_icm426xx_init (struct inv_icm426xx * *s*, struct inv_icm426xx_serif * *serif*, void(*)(inv_icm426xx_sensor_event_t *event) *sensor_event_cb*)

Configure the serial interface used to access the device and execute hardware initialization.

This functions first configures serial interface passed in parameter to make sure device is accessible both in read and write. Thus no serial access should be done before succesfully executing the present function.

Then if requested serial interface is a primary interface (aka UI interface or AP interface), this function initializes the device using the following hardware settings:

- gyroscope fsr = 2000dps
- accelerometer fsr = 4g
- set timestamp resolution to 16us
- enable FIFO mechanism with the following configuration:
  - FIFO record mode i.e FIFO count unit is packet
  - FIFO snapshot mode i.e drop the data when the FIFO overflows
  - Timestamp is logged in FIFO
  - Little Endian fifo_count and fifo_data
  - generate FIFO threshold interrupt when packet count reaches FIFO watermark
  - set FIFO watermark to 1 packet
  - enable temperature and timestamp data to go to FIFO

In case requested serial interface is an auxliary interface (i.e. AUX1 or AUX2) this function returns an error.

Parameters

`in`	*s*	driver structure. Note that first field of this structure MUST be a struct inv_icm426xx_serif.
`in`	*serif*	pointer on serial interface structure to be used to access icm426xx.
`in`	*sensor_event_cb*	callback executed by inv_icm426xx_get_data_from_fifo function each time it extracts some valid data from fifo. Or inv_icm426xx_get_data_from_registers read data from register. Thus this parameter is optional as long as inv_icm426xx_get_data_from_fifo/inv_icm426xx_get_data_from_registers function is not used.

Returns

0 on success, negative value on error.

4.1.4.33 int inv_icm426xx_reset_fifo (struct **inv_icm426xx** ∗ *s*)

reset ICM426XX fifo

Returns

0 on success, negative value on error.

4.1.4.34 int inv_icm426xx_set_accel_frequency (struct **inv_icm426xx** ∗ *s*, const **ICM426XX_ACCEL_CONFIG0_OD↩R_t** *frequency*)

Configure accel Output Data Rate.

Parameters

in	*frequency*	The requested frequency.

See also

ICM426XX_ACCEL_CONFIG0_ODR_t

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::accel_cfg_0_reg is modified by this function

4.1.4.35 int inv_icm426xx_set_accel_fsr (struct **inv_icm426xx** ∗ *s*, ICM426XX_ACCEL_CONFIG0_FS_SEL_t *accel_fsr_g*)

Set accel full scale range.

Parameters

in	*accel_fsr↩_g*	requested full scale range.

See also

ICM426XX_ACCEL_CONFIG0_FS_SEL_t.

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::accel_cfg_0_reg is modified by this function

4.1.4.36 int inv_icm426xx_set_accel_ln_bw (struct **inv_icm426xx** ∗ *s*, ICM426XX_GYRO_ACCEL_CONFIG0_ACCEL_FILT_↩BW_t *acc_bw*)

Set accel Low-Noise bandwidth value.

Parameters

in	*acc_bw*	requested averaging value

See also

ICM426XX_GYRO_ACCEL_CONFIG0_ACCEL_FILT_BW_t

Returns

0 on success, negative value on error.

4.1.4.37 int inv_icm426xx_set_accel_lp_avg (struct inv_icm426xx * *s*, ICM426XX_GYRO_ACCEL_CONFIG0_ACCEL_FILT↵_AVG_t *acc_avg*)

Set accel Low-Power averaging value.

Parameters

`in`	*acc_avg*	requested averaging value

See also

ICM426XX_GYRO_ACCEL_CONFIG0_ACCEL_FILT_AVG_t

Returns

0 on success, negative value on error.

4.1.4.38 int inv_icm426xx_set_config_ibi (struct inv_icm426xx * *s*, inv_icm426xx_interrupt_parameter_t * *interrupt_to_configure*)

Configure which interrupt source can trigger ibi interruptions.

Parameters

`in`	*interrupt_to_configure*	structure with the corresponding state to manage ibi interruptions.

Returns

0 on success, negative value on error.

4.1.4.39 int inv_icm426xx_set_config_int1 (struct inv_icm426xx * *s*, inv_icm426xx_interrupt_parameter_t * *interrupt_to_configure*)

Configure which interrupt source can trigger INT1.

Parameters

`in`	*interrupt_to_configure*	structure with the corresponding state to manage INT1.

Returns

0 on success, negative value on error.

4.1.4.40 int inv_icm426xx_set_config_int2 (struct inv_icm426xx ∗ *s*, inv_icm426xx_interrupt_parameter_t ∗ *interrupt_to_configure*)

Configure which interrupt source can trigger INT2.

Parameters

in	*interrupt_to_configure*	structure with the corresponding state to INT2.

Returns

0 on success, negative value on error.

4.1.4.41 int inv_icm426xx_set_gyro_frequency (struct inv_icm426xx ∗ *s*, const ICM426XX_GYRO_CONFIG0_ODR_t *frequency*)

Configure gyro Output Data Rate.

Parameters

in	*frequency*	The requested frequency.

See also

ICM426XX_GYRO_CONFIG0_ODR_t

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::gyro_cfg_0_reg is modified by this function

4.1.4.42 int inv_icm426xx_set_gyro_fsr (struct inv_icm426xx ∗ *s*, ICM426XX_GYRO_CONFIG0_FS_SEL_t *gyro_fsr_dps*)

Set gyro full scale range.

Parameters

in	*gyro_fsr_dps*	requested full scale range.

See also

ICM426XX_GYRO_CONFIG0_FS_SEL_t.

Returns

0 on success, negative value on error.

Warning

inv_icm426xx::register_cache::gyro_cfg_0_reg is modified by this function

4.1.4.43 int inv_icm426xx_set_gyro_ln_bw (struct inv_icm426xx * *s*, ICM426XX_GYRO_ACCEL_CONFIG0_GYRO_FILT_B↩W_t *gyr_bw*)

Set gyro Low-Noise bandwidth value.

Parameters

in	*gyr_bw*	requested averaging value

See also

ICM426XX_GYRO_ACCEL_CONFIG0_GYRO_FILT_BW_t

Returns

0 on success, negative value on error.

4.1.4.44 int inv_icm426xx_set_reg_bank (struct inv_icm426xx * *s*, uint8_t *bank*)

Set register bank index.

Parameters

bank	new bank to be set

Returns

0 on success, negative value otherwise

4.2 Icm426xx driver high level functions related to APEX

High-level function to setup an Icm426xx device.

Files

- file Icm426xxDriver_HL_apex.h

  High-level function to setup an Icm426xx device.

Classes

- struct inv_icm426xx_tap_parameters_t

  Icm426xx TAP inputs parameters definition.
- struct inv_icm426xx_apex_parameters

  Icm426xx APEX inputs parameters definition.
- struct inv_icm426xx_apex_step_activity

  APEX pedometer outputs.
- struct inv_icm426xx_tap_data

  TAP outputs.

Typedefs

- typedef struct inv_icm426xx_apex_parameters inv_icm426xx_apex_parameters_t

  Icm426xx APEX inputs parameters definition.
- typedef struct inv_icm426xx_apex_step_activity inv_icm426xx_apex_step_activity_t

  APEX pedometer outputs.
- typedef struct inv_icm426xx_tap_data inv_icm426xx_tap_data_t

  TAP outputs.

Functions

- int inv_icm426xx_configure_smd_wom (struct inv_icm426xx ∗s, const uint8_t x_th, const uint8_t y_th, const uint8_t z_th, ICM426XX_SMD_CONFIG_WOM_INT_MODE_t wom_int, ICM426XX_SMD_CONFIG_WO↩M_MODE_t wom_mode)

  Configure Wake On Motion and SMD thresholds.
- int inv_icm426xx_enable_wom (struct inv_icm426xx ∗s)

  Enable Wake On Motion.
- int inv_icm426xx_disable_wom (struct inv_icm426xx ∗s)

  Disable Wake On Motion.
- int inv_icm426xx_enable_smd (struct inv_icm426xx ∗s)

  Enable Significant Motion Detection.
- int inv_icm426xx_disable_smd (struct inv_icm426xx ∗s)

  Disable Significant Motion Detection.
- int inv_icm426xx_init_tap_parameters_struct (struct inv_icm426xx ∗s, inv_icm426xx_tap_parameters_↩t ∗tap_inputs)

  Fill the TAP parameters structure with all the default parameters for TAP algorithm.
- int inv_icm426xx_configure_tap_parameters (struct inv_icm426xx ∗s, const inv_icm426xx_tap_parameters↩_t ∗tap_inputs)

Configure TAP.
- int inv_icm426xx_get_tap_parameters (struct inv_icm426xx ∗s, inv_icm426xx_tap_parameters_t ∗tap_↩ params)

    Returns current TAP parameters.
- int inv_icm426xx_enable_tap (struct inv_icm426xx ∗s)

    Enable TAP.
- int inv_icm426xx_disable_tap (struct inv_icm426xx ∗s)

    Disable TAP.
- int inv_icm426xx_init_apex_parameters_struct (struct inv_icm426xx ∗s, inv_icm426xx_apex_parameters_↩ t ∗apex_inputs)

    Fill the APEX parameters structure with all the default parameters for APEX algorithms (pedometer, tilt)
- int inv_icm426xx_configure_apex_parameters (struct inv_icm426xx ∗s, const inv_icm426xx_apex_↩ parameters_t ∗apex_inputs)

    Configures DMP parameters for APEX algorithms (pedometer, tilt).
- int inv_icm426xx_get_apex_parameters (struct inv_icm426xx ∗s, inv_icm426xx_apex_parameters_t ∗apex↩ _params)

    Returns current DMP parameters for APEX algorithms (pedometer, tilt).
- int inv_icm426xx_set_apex_frequency (struct inv_icm426xx ∗s, const ICM426XX_APEX_CONFIG0_DMP↩ _ODR_t frequency)

    Configure DMP Output Data Rate for APEX algorithms (pedometer, tilt)
- int inv_icm426xx_start_dmp (struct inv_icm426xx ∗s)

    Start DMP for APEX algorithms.
- int inv_icm426xx_enable_apex_pedometer (struct inv_icm426xx ∗s)

    Enable APEX algorithm Pedometer.
- int inv_icm426xx_disable_apex_pedometer (struct inv_icm426xx ∗s)

    Disable APEX algorithm Pedometer.
- int inv_icm426xx_enable_apex_r2w (struct inv_icm426xx ∗s)

    Enable APEX algorithm Raise to wake.
- int inv_icm426xx_disable_apex_r2w (struct inv_icm426xx ∗s)

    Disable APEX algorithm Raise to wake.
- int inv_icm426xx_enable_apex_tilt (struct inv_icm426xx ∗s)

    Enable APEX algorithm Tilt.
- int inv_icm426xx_disable_apex_tilt (struct inv_icm426xx ∗s)

    Disable APEX algorithm Tilt.
- int inv_icm426xx_get_apex_data_activity (struct inv_icm426xx ∗s, inv_icm426xx_apex_step_activity_↩ t ∗apex_activity)

    Retrieve APEX pedometer outputs and format them.
- int inv_icm426xx_get_tap_data (struct inv_icm426xx ∗s, inv_icm426xx_tap_data_t ∗tap_data)

    Retrieve tap outputs.

4.2.1 Detailed Description

High-level function to setup an Icm426xx device.

4.2.2 Function Documentation

4.2.2.1 int inv_icm426xx_configure_apex_parameters (struct inv_icm426xx ∗ *s*, const inv_icm426xx_apex_parameters_t ∗ *apex_inputs*)

Configures DMP parameters for APEX algorithms (pedometer, tilt).

This programmable parameters will be decoded and propagate to the SRAM to be executed at DMP start.

Parameters

in	*apex_inputs*	The requested input parameters. See

See also

inv_icm426xx_apex_parameters_t

Warning

APEX inputs can't change on the fly, this API should be called before enabling any APEX features.
APEX configuration can't be done too frequently, but only once every 10ms. Otherwise it can create unknown behavior.

Returns

0 on success, negative value on error.

4.2.2.2 int inv_icm426xx_configure_smd_wom (struct inv_icm426xx ∗ *s*, const uint8_t *x_th*, const uint8_t *y_th*, const uint8_t *z_th*, ICM426XX_SMD_CONFIG_WOM_INT_MODE_t *wom_int*, ICM426XX_SMD_CONFIG_WOM_MODE_t *wom_mode*)

Configure Wake On Motion and SMD thresholds.

Parameters

in	*x_th*	threshold value for the Wake on Motion Interrupt for X-axis accelerometer.
in	*y_th*	threshold value for the Wake on Motion Interrupt for Y-axis accelerometer.
in	*z_th*	threshold value for the Wake on Motion Interrupt for Z-axis accelerometer.
in	*wom_int*	select which mode between AND/OR is used to generate interrupt.
in	*wom_mode*	select which comparison mode is used for WoM detection.

Returns

0 on success, negative value on error.

4.2.2.3 int inv_icm426xx_configure_tap_parameters (struct inv_icm426xx ∗ *s*, const inv_icm426xx_tap_parameters_t ∗ *tap_inputs*)

Configure TAP.

Parameters

in	*tap_inputs*	The requested input parameters. See

See also

inv_icm426xx_tap_parameters_t

Returns

0 on success, negative value on error.

4.2.2.4 int inv_icm426xx_disable_apex_pedometer (struct inv_icm426xx ∗ *s*)

Disable APEX algorithm Pedometer.

Returns

0 on success, negative value on error.

4.2.2.5 int inv_icm426xx_disable_apex_r2w (struct inv_icm426xx ∗ *s*)

Disable APEX algorithm Raise to wake.

Returns

0 on success, negative value on error.

4.2.2.6 int inv_icm426xx_disable_apex_tilt (struct inv_icm426xx ∗ *s*)

Disable APEX algorithm Tilt.

Returns

0 on success, negative value on error.

4.2.2.7 int inv_icm426xx_disable_smd (struct inv_icm426xx ∗ *s*)

Disable Significant Motion Detection.

Disables SMD event generation and disables SMD interrupt.

Returns

0 on success, negative value on error.

4.2.2.8 int inv_icm426xx_disable_tap (struct inv_icm426xx ∗ *s*)

Disable TAP.

Returns

0 on success, negative value on error.

4.2.2.9 int inv_icm426xx_disable_wom (struct inv_icm426xx ∗ *s*)

Disable Wake On Motion.

Disables WoM event generation and reconfigures interrupt to fire on Fifo water-mark.

Returns

0 on success, negative value on error.

4.2.2.10 int inv_icm426xx_enable_apex_pedometer (struct inv_icm426xx ∗ *s*)

Enable APEX algorithm Pedometer.

note : Pedometer request to have the accelerometer enabled to works with accelerometer frequency less than dmp frequency.

Returns

0 on success, negative value on error.

Warning

Pedometer must be turned OFF to reconfigure it

4.2.2.11 int inv_icm426xx_enable_apex_r2w (struct inv_icm426xx ∗ *s*)

Enable APEX algorithm Raise to wake.

note : Raise to wake request to have the accelerometer enabled to works with accelerometer frequency less than dmp frequency.

Returns

0 on success, negative value on error.

4.2.2.12 int inv_icm426xx_enable_apex_tilt (struct inv_icm426xx ∗ *s*)

Enable APEX algorithm Tilt.

note : Tilt request to have the accelerometer enabled to works with accelerometer frequency less than dmp frequency.

Returns

0 on success, negative value on error.

4.2.2.13 int inv_icm426xx_enable_smd (struct inv_icm426xx ∗ *s*)

Enable Significant Motion Detection.

note : SMD requests to have the accelerometer enabled to work. Enables SMD event generation and configures interrupt to fire on SMD event. WoM event will also be generated. To have good performance, it's recommended to set accelerometer ODR (Output Data Rate) to 20ms and the accelerometer in Low Power Mode.

Returns

0 on success, negative value on error.

4.2.2.14 int inv_icm426xx_enable_tap (struct inv_icm426xx ∗ *s*)

Enable TAP.

note : TAP requests to have the accelerometer enabled to work. To have good performance, it's recommended to set accelerometer ODR (Output Data Rate) to 1ms and the accelerometer in Low Noise Mode.

Returns

0 on success, negative value on error.

4.2.2.15 int inv_icm426xx_enable_wom (struct inv_icm426xx ∗ *s*)

Enable Wake On Motion.

note : WoM requests to have the accelerometer enabled to work. Enables WoM event generation and configures interrupt to fire on WoM event. As a consequence Fifo water-mark interrupt is disabled. To have good performance, it's recommended to set accelerometer ODR (Output Data Rate) to 20ms and the accelerometer in Low Power Mode.

Returns

0 on success, negative value on error.

4.2.2.16 int inv_icm426xx_get_apex_data_activity (struct inv_icm426xx ∗ *s*, inv_icm426xx_apex_step_activity_t ∗ *apex_activity*)

Retrieve APEX pedometer outputs and format them.

Parameters

out	*apex_activity*	Apex step and activity data value.

Returns

0 in case of success, negative value on error. See enum inv_error

4.2.2.17 int inv_icm426xx_get_apex_parameters (struct inv_icm426xx * *s*, inv_icm426xx_apex_parameters_t * *apex_params*)

Returns current DMP parameters for APEX algorithms (pedometer, tilt).

Parameters

out	*apex_params*	The current parameter, fetched from registers. See

See also

inv_icm426xx_apex_parameters_t

Returns

0 on success, negative value on error.

4.2.2.18 int inv_icm426xx_get_tap_data (struct inv_icm426xx * *s*, inv_icm426xx_tap_data_t * *tap_data*)

Retrieve tap outputs.

Parameters

out	*tap_data*	Tap axis, direction and type

Returns

0 in case of success, negative value on error. See enum inv_error

4.2.2.19 int inv_icm426xx_get_tap_parameters (struct inv_icm426xx * *s*, inv_icm426xx_tap_parameters_t * *tap_params*)

Returns current TAP parameters.

Parameters

out	*tap_params*	The current parameter, fetched from registers. See

See also

inv_icm426xx_tap_parameters_t

Returns

0 on success, negative value on error.

4.2.2.20 int inv_icm426xx_init_apex_parameters_struct (struct **inv_icm426xx** ∗ *s*, **inv_icm426xx_apex_parameters_t** ∗ *apex_inputs*)

Fill the APEX parameters structure with all the default parameters for APEX algorithms (pedometer, tilt)

Parameters

out	*apex_inputs*	Default input parameters. See

See also

inv_icm426xx_apex_parameters_t

Returns

0 on success, negative value on error.

4.2.2.21 int inv_icm426xx_init_tap_parameters_struct (struct **inv_icm426xx** ∗ *s*, **inv_icm426xx_tap_parameters_t** ∗ *tap_inputs*)

Fill the TAP parameters structure with all the default parameters for TAP algorithm.

Parameters

out	*tap_inputs*	Default input parameters. See

See also

inv_icm426xx_tap_parameters_t

Returns

0 on success, negative value on error.

4.2.2.22 int inv_icm426xx_set_apex_frequency (struct **inv_icm426xx** ∗ *s*, const **ICM426XX_APEX_CONFIG0_DMP↵_ODR_t** *frequency*)

Configure DMP Output Data Rate for APEX algorithms (pedometer, tilt)

Parameters

`in`	*frequency*	The requested frequency.

See also

ICM426XX_APEX_CONFIG0_DMP_ODR_t

Warning

DMP_ODR can change on the fly, and the DMP code will accommodate necessary modifications
The user needs to take care to set Accel frequency >= DMP frequency. This is a hard constraint since HW will not handle incorrect setting.

Returns

0 on success, negative value on error.

4.2.2.23 int inv_icm426xx_start_dmp (struct inv_icm426xx * *s*)

Start DMP for APEX algorithms.

Returns

0 on success, negative value on error.

4.3 Icm426xx driver extern functions

Extern functions for Icm426xx devices.

Files

- file Icm426xxExtFunc.h

  Extern functions for Icm426xx devices.

Functions

- void inv_icm426xx_sleep_us (uint32_t us)

  Hook for low-level high res system sleep() function to be implemented by upper layer ∼100us resolution is sufficient.
- uint64_t inv_icm426xx_get_time_us (void)

  Hook for low-level high res system get_time() function to be implemented by upper layer Timer should be on 64bit with a 1 us resolution.

4.3.1 Detailed Description

Extern functions for Icm426xx devices.

4.3.2 Function Documentation

4.3.2.1 uint64_t inv_icm426xx_get_time_us (void)

Hook for low-level high res system get_time() function to be implemented by upper layer Timer should be on 64bit with a 1 us resolution.

Parameters

`out`	*The*	current time in us

4.3.2.2 void inv_icm426xx_sleep_us (uint32_t *us*)

Hook for low-level high res system sleep() function to be implemented by upper layer ∼100us resolution is sufficient.

Parameters

`in`	*us*	number of us the calling thread should sleep

4.4 Icm426xx selftest

Low-level function to run selftest on a Icm426xx device.

Files

- file Icm426xxSelfTest.h

    Low-level function to run selftest on a Icm426xx device.

Functions

- int inv_icm426xx_run_selftest (struct inv_icm426xx ∗s, int ∗result)

    Perform hardware self-test for Accel and Gyro.
- int inv_icm426xx_get_st_bias (struct inv_icm426xx ∗s, int st_bias[6])

    Retrieve bias collected by self-test.
- int inv_icm426xx_set_st_bias (struct inv_icm426xx ∗s, const int st_bias[6])

    Apply bias.

4.4.1 Detailed Description

Low-level function to run selftest on a Icm426xx device.

4.4.2 Function Documentation

4.4.2.1 int inv_icm426xx_get_st_bias (struct **inv_icm426xx** ∗ *s*, int *st_bias[6]*)

Retrieve bias collected by self-test.

Parameters

`out`	*st_bias*	bias scaled by 2^16, accel is gee and gyro is dps. The buffer will be filled as below. Gyro LN mode X,Y,Z Accel LN mode X,Y,Z

Returns

0 if success, error code if failure

4.4.2.2 int inv_icm426xx_run_selftest (struct **inv_icm426xx** ∗ *s*, int ∗ *result*)

Perform hardware self-test for Accel and Gyro.

Parameters

`in`	*result*	containing ACCEL_SUCCESS<<1 | GYRO_SUCCESS so 3

Returns

0 if success, error code if failure

4.4.2.3 int inv_icm426xx_set_st_bias (struct inv_icm426xx * *s*, const int *st_bias[6]*)

Apply bias.

Parameters

`in`	*st_bias*	bias scaled by 2^16, accel is gee and gyro is dps. The buffer must be filled as below. Gyro LN mode X,Y,Z Accel LN mode X,Y,Z

Returns

0 if success, error code if failure

4.5 Icm426xx driver transport

Low-level Icm426xx register access.

Files

- file Icm426xxTransport.h

  Low-level Icm426xx register access.

Classes

- struct inv_icm426xx_serif

  basesensor serial interface
- struct inv_icm426xx_transport

  transport interface

Enumerations

Functions

- int inv_icm426xx_init_transport (struct inv_icm426xx ∗s)

  Init cache variable.
- int inv_icm426xx_read_reg (struct inv_icm426xx ∗s, uint8_t reg, uint32_t len, uint8_t ∗buf)

  Reads data from a register on Icm426xx.
- int inv_icm426xx_write_reg (struct inv_icm426xx ∗s, uint8_t reg, uint32_t len, const uint8_t ∗buf)

  Writes data to a register on Icm426xx.

4.5.1 Detailed Description

Low-level Icm426xx register access.

4.5.2 Function Documentation

4.5.2.1 int inv_icm426xx_init_transport (struct **inv_icm426xx** ∗ *s*)

Init cache variable.

Returns

0 in case of success, -1 for any error

4.5.2.2 int inv_icm426xx_read_reg (struct **inv_icm426xx** ∗ *s,* uint8_t *reg,* uint32_t *len,* uint8_t ∗ *buf*)

Reads data from a register on Icm426xx.

Parameters

`in`	*reg*	register address to be read
`in`	*len*	number of byte to be read
`out`	*buf*	output data from the register

Returns

0 in case of success, -1 for any error

4.5.2.3 int inv_icm426xx_write_reg (struct inv_icm426xx ∗ *s*, uint8_t *reg*, uint32_t *len*, const uint8_t ∗ *buf*)

Writes data to a register on Icm426xx.

Parameters

`in`	*reg*	register address to be written
`in`	*len*	number of byte to be written
`in`	*buf*	input data to write

Returns

0 in case of success, -1 for any error

Chapter 5

Class Documentation

5.1 fifo_header_t Union Reference

Describe the content of the FIFO header.

```
#include <Icm426xxDefs.h>
```

5.1.1 Detailed Description

Describe the content of the FIFO header.

The documentation for this union was generated from the following file:

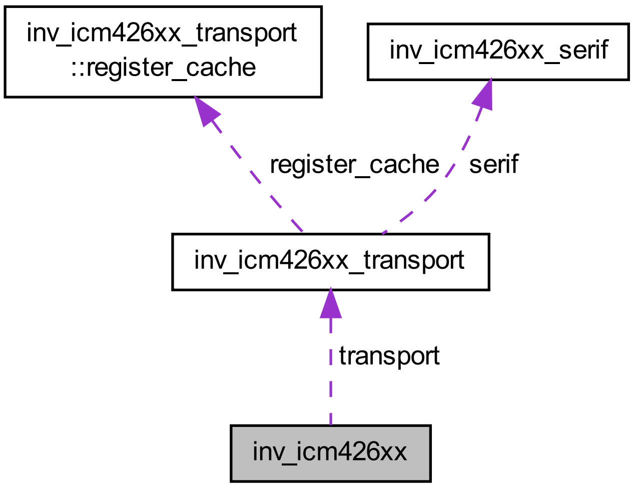
- Icm426xxDefs.h

5.2 inv_icm426xx Struct Reference

Icm426xx driver states definition.

```
#include <Icm426xxDriver_HL.h>
```

Collaboration diagram for inv_icm426xx:



Public Attributes

- struct inv_icm426xx_transport transport

  Warning : this field MUST be the first one of struct icm426xx.
- void(∗ sensor_event_cb)(inv_icm426xx_sensor_event_t ∗event)

  callback executed by inv_icm426xx_get_data_from_fifo function for each data packet extracted from fifo or inv_↩ icm426xx_get_data_from_registers read data from register This field may be NULL if inv_icm426xx_get_data_↩ from_fifo/inv_icm426xx_get_data_from_registers is not used by application.
- int gyro_st_bias [3]

  collected bias values (lsb) during self test
- int st_result

  Flag to keep track if self-test has been already run by storing acc and gyr results.
- uint8_t fifo_data [ICM426XX_FIFO_MIRRORING_SIZE]

  FIFO mirroring memory area.
- uint8_t tmst_to_reg_en_cnt

  internal counter to keep track of the timestamp to register access availability
- uint8_t dmp_is_on

  DMP started status.
- uint64_t gyro_start_time_us

  internal state needed to discard first gyro samples
- uint64_t accel_start_time_us

  internal state needed to discard first accel samples
- uint8_t endianess_data

  internal status of data endianess mode to report correctly data
- uint8_t fifo_highres_enabled

  FIFO packets are 20 bytes long.
- INV_ICM426XX_FIFO_CONFIG_t fifo_is_used

  Data are get from FIFO or from sensor registers.

- uint8_t wom_smd_mask

  This variable keeps track if wom or smd is enabled.
- uint8_t wom_enable

  This variable keeps track if wom is enabled.
- uint64_t gyro_power_off_tmst

  This variable keeps track of timestamp when gyro is power off.
- uint8_t acc_lp_avg

  Low-power averaging setting for accelerometer.
- uint8_t reserved

  reserved field
- uint8_t acc_ln_bw

  Low-noise filter bandwidth setting for accelerometer.
- uint8_t gyr_ln_bw

  Low-noise filter bandwidth setting for gyroscope.

5.2.1 Detailed Description

Icm426xx driver states definition.

5.2.2 Member Data Documentation

5.2.2.1 INV_ICM426XX_FIFO_CONFIG_t inv_icm426xx::fifo_is_used

Data are get from FIFO or from sensor registers.

By default Fifo is used

5.2.2.2 void(∗ inv_icm426xx::sensor_event_cb) (inv_icm426xx_sensor_event_t ∗event)

callback executed by inv_icm426xx_get_data_from_fifo function for each data packet extracted from fifo or inv_↩
icm426xx_get_data_from_registers read data from register This field may be NULL if inv_icm426xx_get_data_↩
from_fifo/inv_icm426xx_get_data_from_registers is not used by application.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL.h

5.3 inv_icm426xx_apex_parameters Struct Reference

Icm426xx APEX inputs parameters definition.

```
#include <Icm426xxDriver_HL_apex.h>
```

Public Attributes

- ICM426XX_APEX_CONFIG2_PEDO_AMP_TH_t pedo_amp_th

  Peak threshold value to be considered as a valid step (mg)
- uint8_t pedo_step_cnt_th

  Minimum number of steps that must be detected before the pedometer step count begins incrementing.
- uint8_t pedo_step_det_th

  Minimum number of low latency steps that must be detected before the pedometer step count begins incrementing.
- ICM426XX_APEX_CONFIG3_PEDO_SB_TIMER_TH_t pedo_sb_timer_th

  Duration of non-walk to exit the current walk mode, pedo_step_cnt_th number of steps must again be detected before step count starts to increase.
- ICM426XX_APEX_CONFIG3_PEDO_HI_ENRGY_TH_t pedo_hi_enrgy_th

  Threshold to improve run detection if not steps are counted while running.
- ICM426XX_APEX_CONFIG4_TILT_WAIT_TIME_t tilt_wait_time

  Number of accelerometer samples to wait before triggering tilt event.
- ICM426XX_APEX_CONFIG1_DMP_POWER_SAVE_TIME_t power_save_time

  The time after which DMP goes in power save mode according to the DMP ODR configured.
- ICM426XX_APEX_CONFIG0_DMP_POWER_SAVE_t power_save

  Power save mode for APEX algorithms.
- ICM426XX_APEX_CONFIG9_SENSITIVITY_MODE_t sensitivity_mode

  Sensitivity mode Normal(0) or Slow walk(1).
- ICM426XX_APEX_CONFIG4_R2W_SLEEP_TIME_OUT_t r2w_sleep_time_out

  Time out for a sleep detection.
- ICM426XX_APEX_CONFIG5_R2W_MOUNTING_MATRIX_t r2w_mounting_matrix

  Mounting matrix, chip to device frame.
- ICM426XX_APEX_CONFIG6_R2W_SLEEP_GEST_DELAY_t r2w_gest_delay

  Detection window for a sleep gesture detection.
- ICM426XX_APEX_CONFIG1_LOW_ENERGY_AMP_TH_t low_energy_amp_th

  Peak threshold value to be considered as a valid step (mg) in Slow walk mode.

5.3.1 Detailed Description

Icm426xx APEX inputs parameters definition.

5.3.2 Member Data Documentation

5.3.2.1 ICM426XX_APEX_CONFIG0_DMP_POWER_SAVE_t inv_icm426xx_apex_parameters::power_save

Power save mode for APEX algorithms.

This mode will put APEX features into sleep mode, leaving only the WOM running to wake-up the DMP

5.3.2.2 ICM426XX_APEX_CONFIG6_R2W_SLEEP_GEST_DELAY_t inv_icm426xx_apex_parameters::r2w_gest_delay

Detection window for a sleep gesture detection.

5.3.2.3 ICM426XX_APEX_CONFIG5_R2W_MOUNTING_MATRIX_t inv_icm426xx_apex_parameters::r2w_mounting_matrix

Mounting matrix, chip to device frame.

5.3.2.4 ICM426XX_APEX_CONFIG4_R2W_SLEEP_TIME_OUT_t inv_icm426xx_apex_parameters::r2w_sleep_time_out

Time out for a sleep detection.

5.3.2.5 ICM426XX_APEX_CONFIG9_SENSITIVITY_MODE_t inv_icm426xx_apex_parameters::sensitivity_mode

Sensitivity mode Normal(0) or Slow walk(1).

The Slow walk mode improve the slow walk detection (<1Hz) but in return the number of false detection might be increase.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL_apex.h

5.4 inv_icm426xx_apex_step_activity Struct Reference

APEX pedometer outputs.

```
#include <Icm426xxDriver_HL_apex.h>
```

Public Attributes

- uint16_t step_cnt

  Number of steps taken.
- uint8_t step_cadence

  Walk/run cadency in number of samples.
- uint8_t activity_class

  Detected activity unknown (0), walk (1) or run (2)

5.4.1 Detailed Description

APEX pedometer outputs.

5.4.2 Member Data Documentation

5.4.2.1 uint8_t inv_icm426xx_apex_step_activity::step_cadence

Walk/run cadency in number of samples.

Format is u6.2. E.g, At 50Hz and 2Hz walk frequency, if the cadency is 25 samples. The register will output 100.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL_apex.h

5.5 inv_icm426xx_interrupt_parameter_t Struct Reference

Icm426xx set of interrupt enable flag.

```
#include <Icm426xxDriver_HL.h>
```

5.5.1 Detailed Description

Icm426xx set of interrupt enable flag.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL.h

5.6 inv_icm426xx_sensor_event_t Struct Reference

Sensor event structure definition.

```
#include <Icm426xxDriver_HL.h>
```

5.6.1 Detailed Description

Sensor event structure definition.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL.h

5.7 inv_icm426xx_serif Struct Reference

basesensor serial interface

```
#include <Icm426xxTransport.h>
```

5.7.1 Detailed Description

basesensor serial interface

The documentation for this struct was generated from the following file:

- Icm426xxTransport.h

5.8 inv_icm426xx_tap_data Struct Reference

TAP outputs.

```
#include <Icm426xxDriver_HL_apex.h>
```

Public Attributes

- ICM426XX_APEX_DATA4_TAP_NUM_t tap_num

  Detects single (1) or double (2) tap.
- ICM426XX_APEX_DATA4_TAP_AXIS_t tap_axis

  Axis along which tap has been detected.
- ICM426XX_APEX_DATA4_TAP_DIR_t tap_dir

  Direction of the tap, either +axis (1) or -axis (0)
- uint8_t double_tap_timing

  Timing between both taps of a double tap expressed in 1/16th of odr in ms (e.g At 500Hz, 2 means 64ms between each tap)

5.8.1 Detailed Description

TAP outputs.

The documentation for this struct was generated from the following file:

- Icm426xxDriver_HL_apex.h

5.9 inv_icm426xx_tap_parameters_t Struct Reference

Icm426xx TAP inputs parameters definition.

```
#include <Icm426xxDriver_HL_apex.h>
```

Public Attributes

- uint8_t min_jerk_thr

  Minimum Jerk Threshold.
- ICM426XX_APEX_CONFIG7_TAP_MAX_PEAK_TOL_t max_peak_tol

  Maximum peak tolerance.
- ICM426XX_APEX_CONFIG8_TAP_TMAX_t tmax

  Tap measurement window.
- ICM426XX_APEX_CONFIG8_TAP_TAVG_t tavg

  Energy measuremerent window.
- ICM426XX_APEX_CONFIG8_TAP_TMIN_t tmin

  Single tap window.

5.9.1 Detailed Description

Icm426xx TAP inputs parameters definition.

The documentation for this struct was generated from the following file:

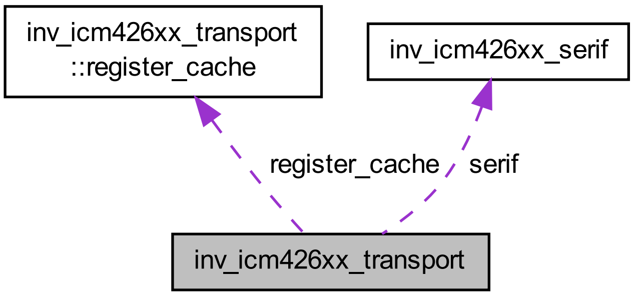
- Icm426xxDriver_HL_apex.h

5.10 inv_icm426xx_transport Struct Reference

transport interface

```
#include <Icm426xxTransport.h>
```

Collaboration diagram for inv_icm426xx_transport:



Classes

- struct register_cache

    Contains mirrored values of some IP registers.

Public Attributes

- struct inv_icm426xx_serif serif

    Warning : this field MUST be the first one of struct inv_icm426xx_transport.
- struct inv_icm426xx_transport::register_cache register_cache

    Store mostly used register values on SRAM.

5.10.1 Detailed Description

transport interface

5.10.2 Member Data Documentation

5.10.2.1 struct inv_icm426xx_transport::register_cache inv_icm426xx_transport::register_cache

Store mostly used register values on SRAM.

MPUREG_OTP_SEC_STATUS_B1 and MPUREG_INT_STATUS registers are read before the cache has a chance to be initialized. Therefore, these registers shall never be added to the cache Registers from bank 1,2,3 or 4 shall never be added to the cache

The documentation for this struct was generated from the following file:

- Icm426xxTransport.h

5.11 recover_regs Struct Reference

Contains the current register values.

5.11.1 Detailed Description

Contains the current register values.

Used to reapply values after the ST procedure

The documentation for this struct was generated from the following file:

- Icm426xxSelfTest.c

5.12 inv_icm426xx_transport::register_cache Struct Reference

Contains mirrored values of some IP registers.

```
#include <Icm426xxTransport.h>
```

Public Attributes

- uint8_t intf_cfg_1_reg

  INTF_CONFIG1, Bank: 0, Address: 0x4D.
- uint8_t pwr_mngt_0_reg

  PWR_MGMT_0, Bank: 0, Address: 0x4E.
- uint8_t gyro_cfg_0_reg

  GYRO_CONFIG0, Bank: 0, Address: 0x4F.
- uint8_t accel_cfg_0_reg

  ACCEL_CONFIG0, Bank: 0, Address: 0x50.
- uint8_t tmst_cfg_reg

  TMST_CONFIG, Bank: 0, Address: 0x54.
- uint8_t bank_sel_reg

  MPUREG_REG_BANK_SEL, All banks, Address 0x76.

5.12.1 Detailed Description

Contains mirrored values of some IP registers.

The documentation for this struct was generated from the following file:

- Icm426xxTransport.h

Chapter 6

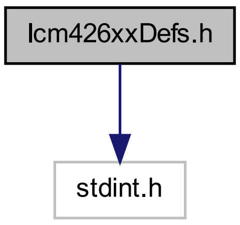
File Documentation

6.1 Icm426xxDefs.h File Reference

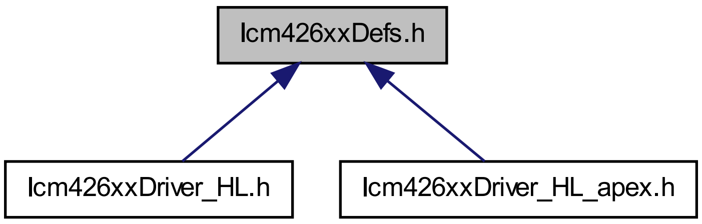
File exposing the device register map.

```
#include <stdint.h>
```

Include dependency graph for Icm426xxDefs.h:



This graph shows which files directly or indirectly include this file:



Classes

- union fifo_header_t

  Describe the content of the FIFO header.

Macros

- #define BIT_APEX_DATA4_TAP_NUM_POS 3

  TAP status flags: non-zero value - tap detected bit0 - positive or negative edge bit1 and 2 - axis detected : 0-X ; 1-Y ; 2-Z bit3 and 4 - tap type : 1-single ; 2 -double.

Enumerations

6.1.1 Detailed Description

File exposing the device register map.

6.1.2 Enumeration Type Documentation

6.1.2.1 enum ICM426XX_ACCEL_CONFIG0_FS_SEL_t

Accelerometer FSR selection.

Enumerator

ICM426XX_ACCEL_CONFIG0_FS_SEL_2g	2g
ICM426XX_ACCEL_CONFIG0_FS_SEL_4g	4g
ICM426XX_ACCEL_CONFIG0_FS_SEL_8g	8g
ICM426XX_ACCEL_CONFIG0_FS_SEL_16g	16g

6.1.2.2 enum ICM426XX_ACCEL_CONFIG0_ODR_t

Accelerometer ODR selection.

Enumerator

ICM426XX_ACCEL_CONFIG0_ODR_500_HZ	500 Hz (2 ms)
ICM426XX_ACCEL_CONFIG0_ODR_1_5625_HZ	1.5625 Hz (640 ms)
ICM426XX_ACCEL_CONFIG0_ODR_3_125_HZ	3.125 Hz (320 ms)
ICM426XX_ACCEL_CONFIG0_ODR_6_25_HZ	6.25 Hz (160 ms)
ICM426XX_ACCEL_CONFIG0_ODR_12_5_HZ	12.5 Hz (80 ms)
ICM426XX_ACCEL_CONFIG0_ODR_25_HZ	25 Hz (40 ms)
ICM426XX_ACCEL_CONFIG0_ODR_50_HZ	50 Hz (20 ms)
ICM426XX_ACCEL_CONFIG0_ODR_100_HZ	100 Hz (10 ms)
ICM426XX_ACCEL_CONFIG0_ODR_200_HZ	200 Hz (5 ms)
ICM426XX_ACCEL_CONFIG0_ODR_1_KHZ	1 KHz (1 ms)
ICM426XX_ACCEL_CONFIG0_ODR_2_KHZ	2 KHz (500 us)
ICM426XX_ACCEL_CONFIG0_ODR_4_KHZ	4 KHz (250 us)
ICM426XX_ACCEL_CONFIG0_ODR_8_KHZ	8 KHz (125 us)
ICM426XX_ACCEL_CONFIG0_ODR_16_KHZ	16 KHz (62.5 us)
ICM426XX_ACCEL_CONFIG0_ODR_32_KHZ	32 KHz (31.25 us)

6.1.2.3 enum ICM426XX_APEX_CONFIG0_DMP_ODR_t

DMP ODR selection.

Enumerator

ICM426XX_APEX_CONFIG0_DMP_ODR_25Hz	25Hz (40ms)
ICM426XX_APEX_CONFIG0_DMP_ODR_50Hz	50Hz (20ms)
ICM426XX_APEX_CONFIG0_DMP_ODR_RESERVED	Reserved.

6.1.2.4 enum ICM426XX_GYRO_CONFIG0_FS_SEL_t

Gyroscope FSR selection.

Enumerator

ICM426XX_GYRO_CONFIG0_FS_SEL_16dps	16dps
ICM426XX_GYRO_CONFIG0_FS_SEL_31dps	31dps
ICM426XX_GYRO_CONFIG0_FS_SEL_62dps	62dps
ICM426XX_GYRO_CONFIG0_FS_SEL_125dps	125dps
ICM426XX_GYRO_CONFIG0_FS_SEL_250dps	250dps
ICM426XX_GYRO_CONFIG0_FS_SEL_500dps	500dps
ICM426XX_GYRO_CONFIG0_FS_SEL_1000dps	1000dps
ICM426XX_GYRO_CONFIG0_FS_SEL_2000dps	2000dps

6.1.2.5 enum ICM426XX_GYRO_CONFIG0_ODR_t

Gyroscope ODR selection.

Enumerator

ICM426XX_GYRO_CONFIG0_ODR_500_HZ	500 Hz (2 ms)
ICM426XX_GYRO_CONFIG0_ODR_12_5_HZ	12.5 Hz (80 ms)
ICM426XX_GYRO_CONFIG0_ODR_25_HZ	25 Hz (40 ms)
ICM426XX_GYRO_CONFIG0_ODR_50_HZ	50 Hz (20 ms)
ICM426XX_GYRO_CONFIG0_ODR_100_HZ	100 Hz (10 ms)
ICM426XX_GYRO_CONFIG0_ODR_200_HZ	200 Hz (5 ms)
ICM426XX_GYRO_CONFIG0_ODR_1_KHZ	1 KHz (1 ms)
ICM426XX_GYRO_CONFIG0_ODR_2_KHZ	2 KHz (500 us)
ICM426XX_GYRO_CONFIG0_ODR_4_KHZ	4 KHz (250 us)
ICM426XX_GYRO_CONFIG0_ODR_8_KHZ	8 KHz (125 us)
ICM426XX_GYRO_CONFIG0_ODR_16_KHZ	16 KHz (62.5 us)
ICM426XX_GYRO_CONFIG0_ODR_32_KHZ	32 KHz (31.25 us)

6.1.2.6 enum ICM426XX_OIS1_CONFIG1_DEC_t

OIS1 rate selection (base clock fixed by OTP divided by decimator value)

Enumerator

ICM426XX_OIS1_CONFIG1_DEC_1	OTP_OIS_clock / 1.
ICM426XX_OIS1_CONFIG1_DEC_2	OTP_OIS_clock / 2.
ICM426XX_OIS1_CONFIG1_DEC_4	OTP_OIS_clock / 4.
ICM426XX_OIS1_CONFIG1_DEC_8	OTP_OIS_clock / 8.
ICM426XX_OIS1_CONFIG1_DEC_16	OTP_OIS_clock / 16.
ICM426XX_OIS1_CONFIG1_DEC_32	OTP_OIS_clock / 32.

6.1.2.7 enum ICM426XX_OIS1_CONFIG2_ACCEL_FS_SEL_t

OIS1 accelerometer FSR selection.

Enumerator

ICM426XX_OIS1_CONFIG2_ACCEL_FS_SEL_2g	2g
ICM426XX_OIS1_CONFIG2_ACCEL_FS_SEL_4g	4g
ICM426XX_OIS1_CONFIG2_ACCEL_FS_SEL_8g	8g
ICM426XX_OIS1_CONFIG2_ACCEL_FS_SEL_16g	16g

6.1.2.8 enum ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_t

OIS1 gyroscope FSR selection.

Enumerator

ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_16dps	15.625 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_31dps	31.25 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_62dps	62.5 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_125dps	125 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_250dps	250 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_500dps	500 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_1000dps	1000 dps
ICM426XX_OIS1_CONFIG2_GYRO_FS_SEL_2000dps	2000 dps

6.1.2.9 enum ICM426XX_OIS2_CONFIG1_DEC_t

OIS2 rate selection (base clock fixed by OTP divided by decimator value)

Enumerator

ICM426XX_OIS2_CONFIG1_DEC_1	OTP_OIS_clock / 1.
ICM426XX_OIS2_CONFIG1_DEC_2	OTP_OIS_clock / 2.
ICM426XX_OIS2_CONFIG1_DEC_4	OTP_OIS_clock / 4.
ICM426XX_OIS2_CONFIG1_DEC_8	OTP_OIS_clock / 8.
ICM426XX_OIS2_CONFIG1_DEC_16	OTP_OIS_clock / 16.
ICM426XX_OIS2_CONFIG1_DEC_32	OTP_OIS_clock / 32.

6.1.2.10 enum ICM426XX_OIS2_CONFIG2_ACCEL_FS_SEL_t

OIS2 accelerometer FSR selection.

Enumerator

ICM426XX_OIS2_CONFIG2_ACCEL_FS_SEL_2g	2g
ICM426XX_OIS2_CONFIG2_ACCEL_FS_SEL_4g	4g
ICM426XX_OIS2_CONFIG2_ACCEL_FS_SEL_8g	8g
ICM426XX_OIS2_CONFIG2_ACCEL_FS_SEL_16g	16g

6.1.2.11 enum ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_t

OIS2 gyroscope FSR selection.

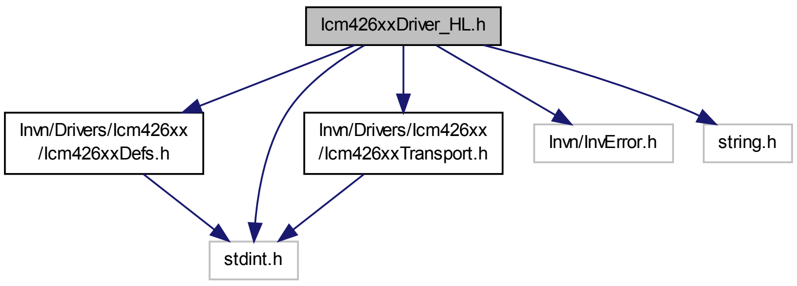
Enumerator

ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_16dps	15.625 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_31dps	31.25 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_62dps	62.5 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_125dps	125 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_250dps	250 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_500dps	500 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_1000dps	1000 dps
ICM426XX_OIS2_CONFIG2_GYRO_FS_SEL_2000dps	2000 dps

6.2 Icm426xxDriver_HL.h File Reference

High-level function to setup an Icm426xx device.

```
#include "Invn/Drivers/Icm426xx/Icm426xxDefs.h"
#include "Invn/Drivers/Icm426xx/Icm426xxTransport.h"
#include "Invn/InvError.h"
#include <stdint.h>
#include <string.h>
```
Include dependency graph for Icm426xxDriver_HL.h:


Icm426xxDriver_HL.h
Invn/Drivers/Icm426xx/Icm426xxDefs.h
Invn/Drivers/Icm426xx/Icm426xxTransport.h
Invn/InvError.h
string.h
stdint.h

Classes

- struct inv_icm426xx_sensor_event_t

  Sensor event structure definition.
- struct inv_icm426xx

  Icm426xx driver states definition.
- struct inv_icm426xx_interrupt_parameter_t

  Icm426xx set of interrupt enable flag.

Macros

- #define INV_ICM426XX_LIGHTWEIGHT_DRIVER 0

  Ligthen driver logic by stripping out procedures on transitions.
- #define PLL_SCALE_FACTOR_Q24 (1UL<<24)

  Scale factor and max ODR Dependant of chip.
- #define ACCEL_CONFIG0_FS_SEL_MAX ICM426XX_ACCEL_CONFIG0_FS_SEL_16g

  Max FSR values for accel and gyro Dependant of chip.
- #define RTC_SUPPORTED 0

  RTC Support flag Define whether the RTC mode is supported Dependant of chip.
- #define ICM426XX_FIFO_MIRRORING_SIZE 16 ∗ 129

  Icm426xx maximum buffer size mirrored from FIFO at polling time.
- #define ICM426XX_DEFAULT_WOM_THS_MG 52>>2 /∗ = 52mg/4 ∗/

  Default value for the WOM threshold Resolution of the threshold is ∼= 4mg.
- #define ICM426XX_ACC_STARTUP_TIME_US 20000U

  Icm426xx Accelerometer start-up time before having correct data.
- #define ICM426XX_GYR_STARTUP_TIME_US 60000U

  Icm426xx Gyroscope start-up time before having correct data.

Enumerations

Functions

- int inv_icm426xx_set_reg_bank (struct inv_icm426xx ∗s, uint8_t bank)

  Set register bank index.
- int inv_icm426xx_init (struct inv_icm426xx ∗s, struct inv_icm426xx_serif ∗serif, void(∗sensor_event_cb)(inv↩_icm426xx_sensor_event_t ∗event))

  Configure the serial interface used to access the device and execute hardware initialization.
- int inv_icm426xx_device_reset (struct inv_icm426xx ∗s)

  Perform a soft reset of the device.
- int inv_icm426xx_get_who_am_i (struct inv_icm426xx ∗s, uint8_t ∗who_am_i)

  return WHOAMI value
- int inv_icm426xx_force_clock_source (struct inv_icm426xx ∗s, ICM426XX_INTF_CONFIG1_ACCEL_LP_↩CLK_t clk_src)

  Configure Accel clock source.
- int inv_icm426xx_enable_accel_low_power_mode (struct inv_icm426xx ∗s)

  Enable/put accel in low power mode.
- int inv_icm426xx_enable_accel_low_noise_mode (struct inv_icm426xx ∗s)

  Enable/put accel in low noise mode.
- int inv_icm426xx_disable_accel (struct inv_icm426xx ∗s)

Disable all 3 axes of accel.

- int inv_icm426xx_enable_gyro_low_noise_mode (struct inv_icm426xx ∗s)

  Enable/put gyro in low noise mode.
- int inv_icm426xx_disable_gyro (struct inv_icm426xx ∗s)

  Disable all 3 axes of gyro.
- int inv_icm426xx_enable_fsync (struct inv_icm426xx ∗s)

  Enable fsync tagging functionnality.
- int inv_icm426xx_disable_fsync (struct inv_icm426xx ∗s)

  Disable fsync tagging functionnality.
- int inv_icm426xx_configure_timestamp_resolution (struct inv_icm426xx ∗s, ICM426XX_TMST_CONFIG_↩ RESOL_t resol)

  Configure timestamp resolution from FIFO.
- int inv_icm426xx_set_config_ibi (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt_↩ to_configure)

  Configure which interrupt source can trigger ibi interruptions.
- int inv_icm426xx_get_config_ibi (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt_↩ to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_set_config_int1 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt↩ _to_configure)

  Configure which interrupt source can trigger INT1.
- int inv_icm426xx_get_config_int1 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt↩ _to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_set_config_int2 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt↩ _to_configure)

  Configure which interrupt source can trigger INT2.
- int inv_icm426xx_get_config_int2 (struct inv_icm426xx ∗s, inv_icm426xx_interrupt_parameter_t ∗interrupt↩ _to_configure)

  Retrieve interrupts configuration.
- int inv_icm426xx_get_data_from_registers (struct inv_icm426xx ∗s)

  Read all registers containing data (tempereature, accelerometer and gyroscope).
- int inv_icm426xx_get_data_from_fifo (struct inv_icm426xx ∗s)

  Read all available packets from the FIFO.
- uint32_t inv_icm426xx_convert_odr_bitfield_to_us (uint32_t odr_bitfield)

  Converts ICM426XX_ACCEL_CONFIG0_ODR_t or ICM426XX_GYRO_CONFIG0_ODR_t enums to period expressed in us.
- int inv_icm426xx_set_accel_frequency (struct inv_icm426xx ∗s, const ICM426XX_ACCEL_CONFIG0_OD↩ R_t frequency)

  Configure accel Output Data Rate.
- int inv_icm426xx_set_gyro_frequency (struct inv_icm426xx ∗s, const ICM426XX_GYRO_CONFIG0_ODR↩ _t frequency)

  Configure gyro Output Data Rate.
- int inv_icm426xx_set_accel_fsr (struct inv_icm426xx ∗s, ICM426XX_ACCEL_CONFIG0_FS_SEL_t accel↩ _fsr_g)

  Set accel full scale range.
- int inv_icm426xx_get_accel_fsr (struct inv_icm426xx ∗s, ICM426XX_ACCEL_CONFIG0_FS_SEL_t ∗accel↩ _fsr_g)

  Access accel full scale range.
- int inv_icm426xx_set_gyro_fsr (struct inv_icm426xx ∗s, ICM426XX_GYRO_CONFIG0_FS_SEL_t gyro_fsr↩ _dps)

  Set gyro full scale range.

- int inv_icm426xx_get_gyro_fsr (struct inv_icm426xx ∗s, ICM426XX_GYRO_CONFIG0_FS_SEL_t ∗gyro_↩ fsr_dps)

  Access gyro full scale range.
- int inv_icm426xx_set_accel_lp_avg (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_ACC↩ EL_FILT_AVG_t acc_avg)

  Set accel Low-Power averaging value.
- int inv_icm426xx_set_accel_ln_bw (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_ACC↩ EL_FILT_BW_t acc_bw)

  Set accel Low-Noise bandwidth value.
- int inv_icm426xx_set_gyro_ln_bw (struct inv_icm426xx ∗s, ICM426XX_GYRO_ACCEL_CONFIG0_GYRO↩ _FILT_BW_t gyr_bw)

  Set gyro Low-Noise bandwidth value.
- int inv_icm426xx_reset_fifo (struct inv_icm426xx ∗s)

  reset ICM426XX fifo
- int inv_icm426xx_enable_timestamp_to_register (struct inv_icm426xx ∗s)

  Enable the 20bits-timestamp register access to read in a reliable way the strobed timestamp.
- int inv_icm426xx_disable_timestamp_to_register (struct inv_icm426xx ∗s)

  Disable the 20bits-timestamp register access.
- int inv_icm426xx_get_current_timestamp (struct inv_icm426xx ∗s, uint32_t ∗icm_time)

  Get the timestamp value of icm426xx from register.
- int inv_icm426xx_enable_clkin_rtc (struct inv_icm426xx ∗s, uint8_t enable)

  Enable or disable CLKIN/RTC capability.
- int inv_icm426xx_get_clkin_rtc_status (struct inv_icm426xx ∗s)

  Get CLKIN/RTC feature status.
- int inv_icm426xx_enable_high_resolution_fifo (struct inv_icm426xx ∗s)

  Enable 20 bits raw acc and raw gyr data in fifo.
- int inv_icm426xx_disable_high_resolution_fifo (struct inv_icm426xx ∗s)

  Disable 20 bits raw acc and raw gyr data in fifo.
- int inv_icm426xx_configure_fifo (struct inv_icm426xx ∗s, INV_ICM426XX_FIFO_CONFIG_t fifo_config)

  Configure Fifo to select the way data are gathered.
- int inv_icm426xx_configure_fifo_wm (struct inv_icm426xx ∗s, uint16_t wm)

  Configure Fifo watermark (also refered to as fifo threshold)
- uint32_t inv_icm426xx_get_fifo_timestamp_resolution_us_q24 (struct inv_icm426xx ∗s)

  Get FIFO timestamp resolution.
- uint32_t inv_icm426xx_get_reg_timestamp_resolution_us_q24 (struct inv_icm426xx ∗s)

  Get register timestamp resolution.
- const char ∗ inv_icm426xx_get_version (void)

  Return driver version x.y.z-suffix as a char array.

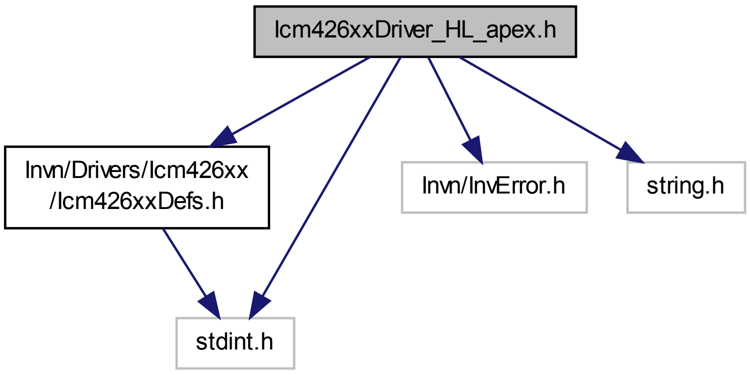
6.2.1 Detailed Description

High-level function to setup an Icm426xx device.

6.3 Icm426xxDriver_HL_apex.h File Reference

High-level function to setup an Icm426xx device.

```
#include "Invn/Drivers/Icm426xx/Icm426xxDefs.h"
#include "Invn/InvError.h"
#include <stdint.h>
#include <string.h>
```
Include dependency graph for Icm426xxDriver_HL_apex.h:



Classes

- struct inv_icm426xx_tap_parameters_t

    Icm426xx TAP inputs parameters definition.
- struct inv_icm426xx_apex_parameters

    Icm426xx APEX inputs parameters definition.
- struct inv_icm426xx_apex_step_activity

    APEX pedometer outputs.
- struct inv_icm426xx_tap_data

    TAP outputs.

Typedefs

- typedef struct inv_icm426xx_apex_parameters inv_icm426xx_apex_parameters_t

    Icm426xx APEX inputs parameters definition.
- typedef struct inv_icm426xx_apex_step_activity inv_icm426xx_apex_step_activity_t

    APEX pedometer outputs.
- typedef struct inv_icm426xx_tap_data inv_icm426xx_tap_data_t

    TAP outputs.

Functions

- int inv_icm426xx_configure_smd_wom (struct inv_icm426xx ∗s, const uint8_t x_th, const uint8_t y_th, const uint8_t z_th, ICM426XX_SMD_CONFIG_WOM_INT_MODE_t wom_int, ICM426XX_SMD_CONFIG_WO↩ M_MODE_t wom_mode)

    Configure Wake On Motion and SMD thresholds.
- int inv_icm426xx_enable_wom (struct inv_icm426xx ∗s)

    Enable Wake On Motion.
- int inv_icm426xx_disable_wom (struct inv_icm426xx ∗s)

    Disable Wake On Motion.
- int inv_icm426xx_enable_smd (struct inv_icm426xx ∗s)

    Enable Significant Motion Detection.
- int inv_icm426xx_disable_smd (struct inv_icm426xx ∗s)

    Disable Significant Motion Detection.
- int inv_icm426xx_init_tap_parameters_struct (struct inv_icm426xx ∗s, inv_icm426xx_tap_parameters_↩ t ∗tap_inputs)

    Fill the TAP parameters structure with all the default parameters for TAP algorithm.
- int inv_icm426xx_configure_tap_parameters (struct inv_icm426xx ∗s, const inv_icm426xx_tap_parameters↩ _t ∗tap_inputs)

    Configure TAP.
- int inv_icm426xx_get_tap_parameters (struct inv_icm426xx ∗s, inv_icm426xx_tap_parameters_t ∗tap_↩ params)

    Returns current TAP parameters.
- int inv_icm426xx_enable_tap (struct inv_icm426xx ∗s)

    Enable TAP.
- int inv_icm426xx_disable_tap (struct inv_icm426xx ∗s)

    Disable TAP.
- int inv_icm426xx_init_apex_parameters_struct (struct inv_icm426xx ∗s, inv_icm426xx_apex_parameters_↩ t ∗apex_inputs)

    Fill the APEX parameters structure with all the default parameters for APEX algorithms (pedometer, tilt)
- int inv_icm426xx_configure_apex_parameters (struct inv_icm426xx ∗s, const inv_icm426xx_apex_↩ parameters_t ∗apex_inputs)

    Configures DMP parameters for APEX algorithms (pedometer, tilt).
- int inv_icm426xx_get_apex_parameters (struct inv_icm426xx ∗s, inv_icm426xx_apex_parameters_t ∗apex↩ _params)

    Returns current DMP parameters for APEX algorithms (pedometer, tilt).
- int inv_icm426xx_set_apex_frequency (struct inv_icm426xx ∗s, const ICM426XX_APEX_CONFIG0_DMP↩ _ODR_t frequency)

    Configure DMP Output Data Rate for APEX algorithms (pedometer, tilt)
- int inv_icm426xx_start_dmp (struct inv_icm426xx ∗s)

    Start DMP for APEX algorithms.
- int inv_icm426xx_enable_apex_pedometer (struct inv_icm426xx ∗s)

    Enable APEX algorithm Pedometer.
- int inv_icm426xx_disable_apex_pedometer (struct inv_icm426xx ∗s)

    Disable APEX algorithm Pedometer.
- int inv_icm426xx_enable_apex_r2w (struct inv_icm426xx ∗s)

    Enable APEX algorithm Raise to wake.
- int inv_icm426xx_disable_apex_r2w (struct inv_icm426xx ∗s)

    Disable APEX algorithm Raise to wake.
- int inv_icm426xx_enable_apex_tilt (struct inv_icm426xx ∗s)

    Enable APEX algorithm Tilt.
- int inv_icm426xx_disable_apex_tilt (struct inv_icm426xx ∗s)

Disable APEX algorithm Tilt.

- int inv_icm426xx_get_apex_data_activity (struct inv_icm426xx *s, inv_icm426xx_apex_step_activity_↵
t *apex_activity)

  Retrieve APEX pedometer outputs and format them.

- int inv_icm426xx_get_tap_data (struct inv_icm426xx *s, inv_icm426xx_tap_data_t *tap_data)

  Retrieve tap outputs.

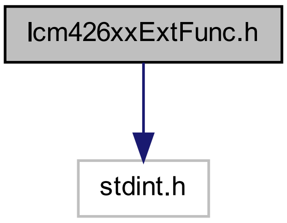
6.3.1 Detailed Description

High-level function to setup an Icm426xx device.

6.4 Icm426xxExtFunc.h File Reference

Extern functions for Icm426xx devices.

`#include <stdint.h>`
Include dependency graph for Icm426xxExtFunc.h:



Functions

- void inv_icm426xx_sleep_us (uint32_t us)

  Hook for low-level high res system sleep() function to be implemented by upper layer ∼100us resolution is sufficient.

- uint64_t inv_icm426xx_get_time_us (void)

  Hook for low-level high res system get_time() function to be implemented by upper layer Timer should be on 64bit with a 1 us resolution.

6.4.1 Detailed Description

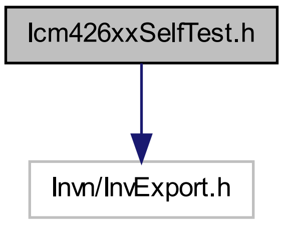
Extern functions for Icm426xx devices.

6.5 Icm426xxSelfTest.h File Reference

Low-level function to run selftest on a Icm426xx device.

```
#include "Invn/InvExport.h"
```

Include dependency graph for Icm426xxSelfTest.h:



Functions

- int inv_icm426xx_run_selftest (struct inv_icm426xx ∗s, int ∗result)

  Perform hardware self-test for Accel and Gyro.
- int inv_icm426xx_get_st_bias (struct inv_icm426xx ∗s, int st_bias[6])

  Retrieve bias collected by self-test.
- int inv_icm426xx_set_st_bias (struct inv_icm426xx ∗s, const int st_bias[6])

  Apply bias.

6.5.1 Detailed Description

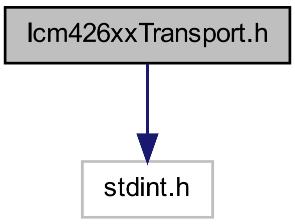
Low-level function to run selftest on a Icm426xx device.

6.6 Icm426xxTransport.h File Reference

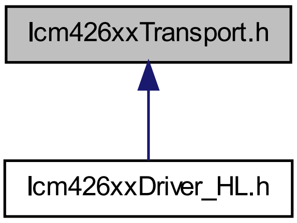
Low-level Icm426xx register access.

```
#include <stdint.h>
```

Include dependency graph for Icm426xxTransport.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct inv_icm426xx_serif

    basesensor serial interface
- struct inv_icm426xx_transport

    transport interface
- struct inv_icm426xx_transport::register_cache

    Contains mirrored values of some IP registers.

Enumerations

Functions

- int inv_icm426xx_init_transport (struct inv_icm426xx ∗s)

    Init cache variable.
- int inv_icm426xx_read_reg (struct inv_icm426xx ∗s, uint8_t reg, uint32_t len, uint8_t ∗buf)

    Reads data from a register on Icm426xx.
- int inv_icm426xx_write_reg (struct inv_icm426xx ∗s, uint8_t reg, uint32_t len, const uint8_t ∗buf)

    Writes data to a register on Icm426xx.

6.6.1 Detailed Description

Low-level Icm426xx register access.

Index