|  |  |
| --- | --- |
| **BMM**  **Detailed Design Document** | |
| **Summary** | This is the Software Detailed Design Document for BMM component of DAIMLER MMA Project |

|  |  |  |
| --- | --- | --- |
| **Author** | **Review** | **Approval** |
| SW Dev: Septimiu-Darex C. Vintila | See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people |
|  |  |  |
| **Distribution** | | |
| See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people | See Project Master Document for the roles and Project Members List for the name of people |

# Table of content

1.1. Revision history \* 5

1.2. Purpose and Scope 5

1.3. Referenced documents 5

1.3.1. External documents 5

1.3.2. Internal Documents 5

1.4. Terminology and definitions 5

2.1. Overview 6

2.2. Traceability 6

3.1. Services 6

3.1.1. BMM\_Autotest\_CheckHallEffectSensor 6

3.1.2. bmm\_ReadSpeed 7

3.1.3. bmm\_ReadSpeedSingleEdge 7

3.1.4. BMM\_runDisableHBSMonitoring 8

3.1.5. BMM\_runEnableHBSMonitoring 9

3.1.6. BMM\_runGetPositionFromT0\_deg 10

3.1.7. BMM\_runGetPositionFromT0\_mm 10

3.1.8. BMM\_runGetPyroActivationStatus 11

3.1.9. BMM\_runGetSpeed\_deg\_s 12

3.1.10. BMM\_runGetSpeed\_mm\_s 13

3.1.11. BMM\_runGetSpeed\_SingleEdge\_deg\_s 14

3.1.12. BMM\_runInit 15

3.1.13. BMM\_runMainFunction 15

3.1.14. bmm\_UpdatePosition 16

3.1.15. Isr\_Bmm\_Hes\_It 17

3.1.16. Isr\_BMM\_HES\_TOUT 18

3.2. Variabiles 19

3.2.1. as16Positions[KU8\_NUMBER\_OF\_PERIOD\_TYPES] 19

3.2.2. au16Captures[KU8\_NUMBER\_OF\_PERIOD\_TYPES] 19

3.2.3. au8Overflows[KU8\_NUMBER\_OF\_PERIOD\_TYPES] 19

3.2.4. b8IsPyroActivated 19

3.2.5. bCriticalNok 20

3.2.6. bmm\_s16BeltPosition\_max\_deg 20

3.2.7. bmm\_s16BeltPosition\_max\_mm 20

3.2.8. bmm\_s16BeltPosition\_min\_deg 20

3.2.9. bmm\_s16BeltPosition\_min\_mm 20

3.2.10. bmm\_s16BeltPositionFromMinPos\_deg 21

3.2.11. bmm\_s16BeltPositionFromMinPos\_mm 21

3.2.12. bmm\_s16BeltPositionFromT0\_deg 21

3.2.13. bmm\_s16BeltPositionFromT0\_mm 21

3.2.14. bmm\_s32BeltSpeed\_deg\_s 22

3.2.15. bmm\_s32BeltSpeed\_mm\_s 22

3.2.16. bmm\_s32BeltSpeed\_SE\_deg\_s 22

3.2.17. bmm\_s32BeltSpeed\_SE\_mm\_s 22

3.2.18. bmm\_State 22

3.2.19. bmm\_tSETimerConfig[((uint8) 2u)] 23

3.2.20. bmm\_tTimerConfig[((uint8) 2u)] 23

3.2.21. bmm\_u32LowSpeedHwCounterTotalOffset 23

3.2.22. bmm\_u32LowSpeedSEHwCounterTotalOffset 23

3.2.23. bmm\_u32StaticNextPeriodLength 23

3.2.24. bmm\_u32StaticSENextPeriodLength 24

3.2.25. bmm\_u8SETimerConfigI 24

3.2.26. bmm\_u8TimerConfigI 24

3.2.27. bNewEvent 24

3.2.28. bPyroDetected 24

3.2.29. s16CurrentPosition 25

3.2.30. s16HighestPosition 25

3.2.31. s16LastPeriodDisplacement 25

3.2.32. s16LowestPosition 25

3.2.33. stSingleEdgeData 26

3.2.34. u16CplTimerCntThrs 26

3.2.35. u32LastPeriodLength 26

3.2.36. u8BMMDowngradedState 26

3.2.37. u8NextPeriodType 26

3.2.38. u8PyroDetectCnt 27

3.3. Macros 27

3.3.1. BMM\_AC\_BeltMovementMonitoring\_START\_SEC\_CODE 27

3.3.2. BMM\_AC\_BeltMovementMonitoring\_STOP\_SEC\_CODE 27

3.3.3. BMM\_BOTH\_SENSORS\_ERROR 27

3.3.4. BMM\_EXPECTED\_TMR\_PRESCALER 27

3.3.5. BMM\_HALL\_A\_ERROR 28

3.3.6. BMM\_HALL\_B\_ERROR 28

3.3.7. BMM\_SENSORS\_OK 28

3.3.8. DIO\_HALL\_A 28

3.3.9. DIO\_HALL\_B 28

3.3.10. EXPECTED\_PRESCALER 28

3.3.11. HARD\_TIMER\_RANGE 29

3.3.12. KS16\_NB\_DEGREES\_PER\_TURN 29

3.3.13. KS16\_NB\_PAIRS\_OF\_POLES\_CFG 29

3.3.14. KS16\_NB\_PULSE\_PER\_PERIOD 29

3.3.15. KS16\_NULL\_DT\_POSITION\_IN\_PERIOD 29

3.3.16. KS16\_RELEASE\_DT\_POSITION\_IN\_PERIOD 29

3.3.17. KS16\_TENSION\_DT\_POSITION\_IN\_PERIOD 30

3.3.18. KS32\_NB\_LSB\_IN\_S 30

3.3.19. KS32\_NB\_LSB\_IN\_S\_SINGLE\_EDGE 30

3.3.20. KU16\_HARD\_TIMER\_HALF\_RANGE 30

3.3.21. KU16\_RESET\_COUNTER 30

3.3.22. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_1 30

3.3.23. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_2 31

3.3.24. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_1 31

3.3.25. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_2 31

3.3.26. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_1 31

3.3.27. KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_2 31

3.3.28. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_1 31

3.3.29. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_2 32

3.3.30. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_1 32

3.3.31. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_2 32

3.3.32. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_1 32

3.3.33. KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_2 32

3.3.34. KU32\_NB\_LSB\_IN\_MS 32

3.3.35. KU32\_NB\_LSB\_IN\_S 33

3.3.36. KU32\_NB\_MS\_IN\_S 33

3.3.37. KU32\_NB\_US\_IN\_MS 33

3.3.38. KU8\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS 33

3.3.39. KU8\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS 33

3.3.40. KU8\_DEGREES\_PER\_EDGE 33

3.3.41. NB\_CLK\_IN\_PERIOD 34

3.3.42. NB\_MS\_IN\_S 34

3.3.43. NB\_US\_IN\_MS 34

3.3.44. NB\_US\_IN\_S 34

3.3.45. SOFT\_TIMER\_RANGE 34

3.3.46. U16\_COUNTER\_VALUE 35

3.3.47. U16\_COUNTER\_VALUE\_STATIC 35

3.3.48. BMM\_CFG\_TIMER\_INPUT\_CLK 35

3.3.49. BMM\_ENABLE\_MAX\_POS\_MONITORING 35

3.3.50. BMM\_ENABLE\_PYRO\_DETECTION 35

3.3.51. BMM\_NB\_PAIRS\_OF\_POLES\_CFG 35

3.3.52. BMM\_PERIOD\_MAX\_MS 36

3.3.53. BMM\_S16\_BOBBIN\_PERIMETER\_IN\_MM\_CFG 36

# General Information

## Revision history \*

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author(s)** | **Description/comment** |
| 1.1.4.1 | 31.08.2022 | Septimiu Vintila | Duplicate to template |
| 1.1.4.2 | 31.08.2022 | Septimiu Vintila | First revision. |
| 1.1.4.3 | 05.09.2022 | Septimiu Vintila | Findings fixes after review. |
| 1.1.4.4 | 07.09.2022 | Septimiu Vintila | Functions diagrams added to table of figures. |
| 1.1.4.5 | 07.03.2023 | Stefan Dominte | Update for R5.1 |
| 1.1.4.6 | 21.06.2023 | Mirela Obada | Update for R6.1 |
| 1.1.4.7 | 17.11.2023 | Razvan Badiu | Fix review findings |

*\* Template history is found in the CM tool used for templates*

## Purpose and Scope

The review of this document is done thanks to …

The purpose of this document is…

## Referenced documents

### External documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  |  |  |
|  |  |  |
|  |  |  |

### Internal Documents

|  |  |  |
| --- | --- | --- |
| **Id** | **Title** | **Reference** |
|  | SBE\_4G\_NVP\_layout.xls |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Terminology and definitions

The generic acronyms are available in the [AEM process and method wiki](https://alvteams.alv.autoliv.int/sites/aeuaeequalityassurance/AEM%20Process%20wiki/acronyms.aspx)

|  |  |
| --- | --- |
| **Terminology** | **Meaning** |
| AAU | Atomic architectural unit |
| SW | software |
| BMM | Belt Movement Monitoring module |

# SW Module Detailed Design

## Overview

The aim of the “BMM” component is to provide information related to the bobbin such as rotation speed and pyro device status.

### BMM system presentation

The BMM module is a low device driver which drives the microcontroller peripheral (itself interfaced with hall effect sensors) to monitor the belt bobbin rotation, in order to provide two types of services:

* To get the relative bobbin position since the software initialization.
* To get the instantaneous bobbin rotation speed.

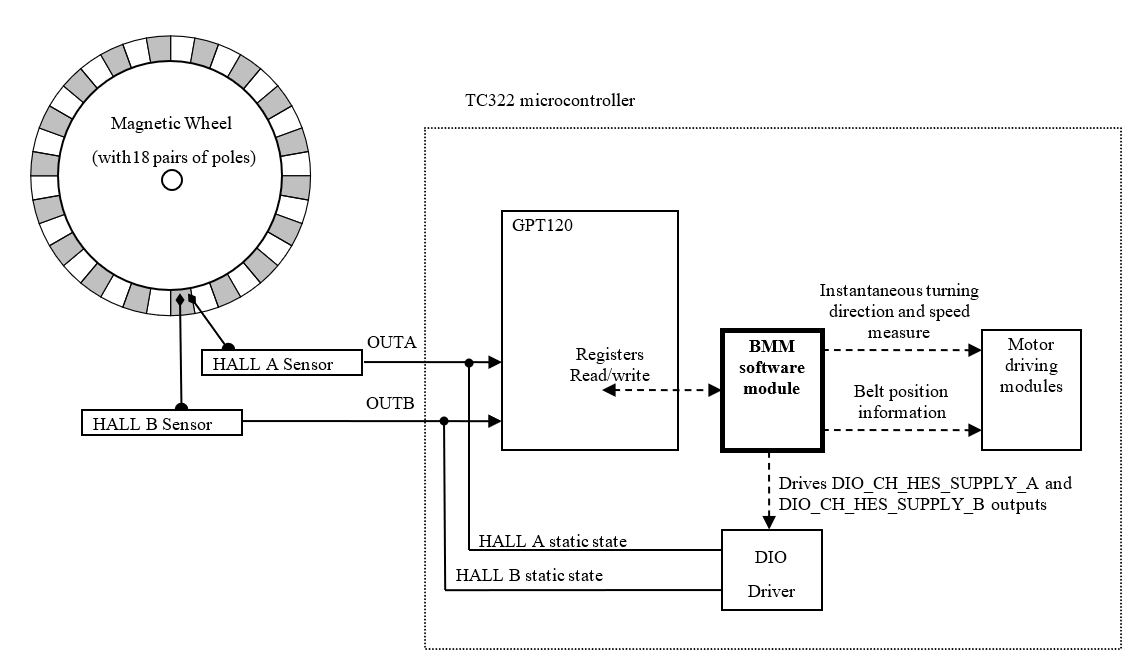


Figure : BMM integration block diagram

* Timer T3 operating in “Incremental Interface Mode” automatically provides information on the sensor’s current position.
* Dynamic information (speed, acceleration, deceleration) may be obtained by measuring the incoming signal periods (Timer T5 in “Combined Capture Mode”).

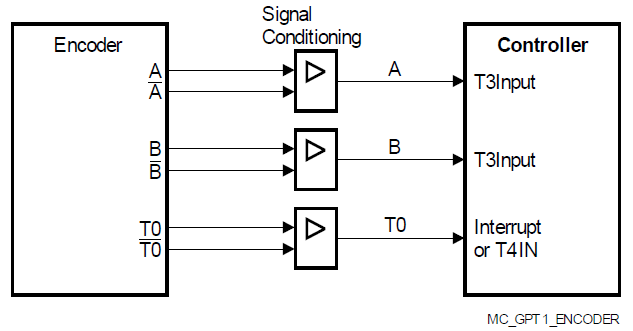


Figure : General principle based on quadrature signals

### General principle:

Each edge (rising or falling) on each channel is detected by the GPT module and taken into account by the BMM module to update the bobbin position in real time.

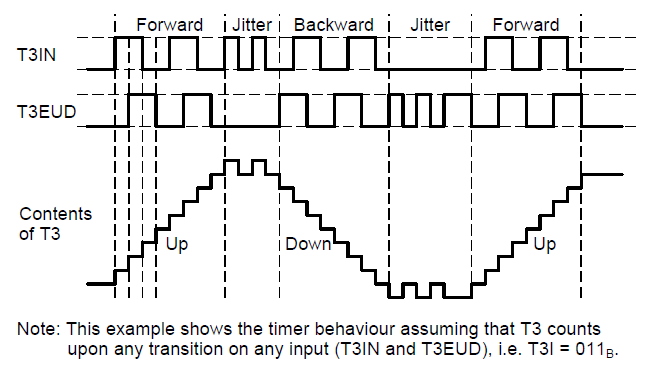


Figure : Evaluation of Incremental Encoder Signals, 2 Count Inputs

The time information to determine the dynamic parameters is generated by capturing the contents of the free running timer T5 into register CAPREL.

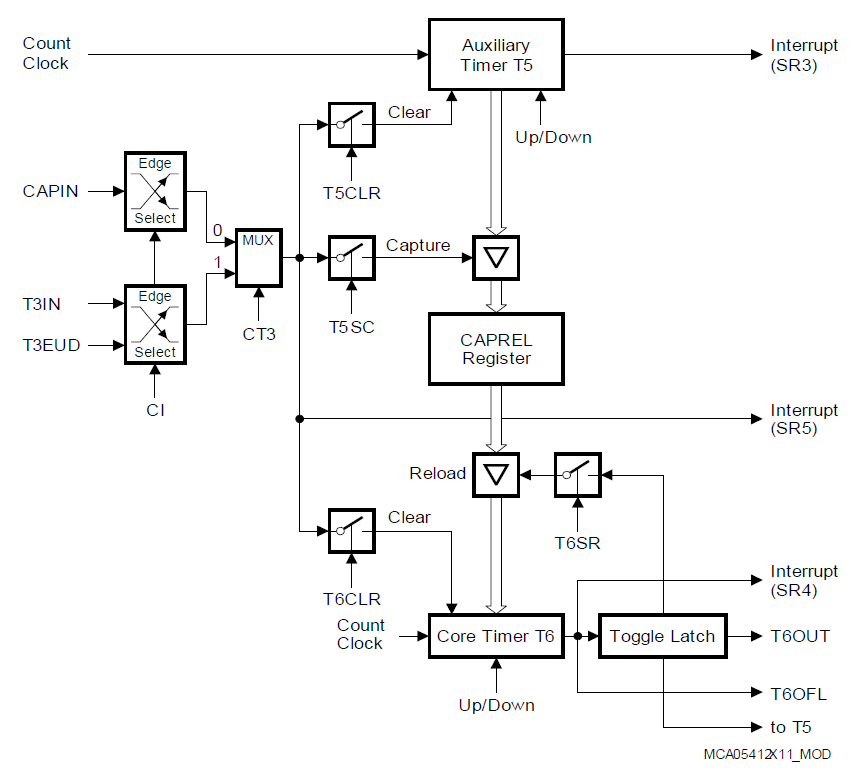


Figure : Capture/Reload Register CAPREL in Capture-And-Reload Mode

If the magnetic wheel has 18 pairs of poles:

* 1 turn = **360°** rotation = 18 periods = 18 x 2 x 2 = **72** total edges,  
  considering both hall A and hall B edges and 2 edges per period and signal

Consequently, each occurring edge means: **360° / 72 = 5°** rotation since previous edge.

To reach a good reliability of speed evaluation, it shall be based on complete periods (see justification in 2.1.3):

* (Speed in °/s) = 20° / (signal’s period in s)

### Support of measures disparities

#### Presentation of sensor assembly errors

While the rotation angle between North Pole and South Pole is 10°, the angle between hall A and hall B sensor places shall be ideally 5 degrees.

Possible minor assembly error can generate unexpected phase shifting of signals regarding each other, even at constant rotation speed.

As a result, a quarter period measures using both signals can’t be taken into account to evaluate the belt speed.

Hall A

Hall B

t1 > t2

Figure : Example with Hall B signal ahead of the Hall A signal quadrature

#### Presentation of the magnetic asymmetry constraint

The magnetic wheel – hall sensor couple is not as perfect as in theory. Indeed, North Pole detection sensitivity and South Pole detection sensitivity are potentially not perfectly symmetric.

As a result, the duty cycle of hall sensors output signals can differ form 50% even at constant rotation speed.   
While low pulses can last more or less than high pulse, the bobbin speed evaluation can’t be based on half period measures.

Hall A

Hall B

t1 > t2

Figure : Asymmetry example: both duty cycles bellow 50%

#### Requirements to handle disparities

The bobbin speed evaluation requires to measure only full periods coming from a same hall sensor.

On the other side, a full period can last several seconds when trying to evaluate very low speed of bobbin rotation (as reference, considering the average bobbin perimeter of 168mm, 1 second full period correspond to belt displacement about 168/18 = 9 mm/s)

To obtain the best software reactivity as possible, the BMM measures simultaneously the 4 existing kinds of full periods:

* Between rising edges on Hall A signal
* Between falling edges on Hall A signal
* Between rising edges on Hall B signal
* Between falling edges on Hall B signal

Using CAPREL interrupt, updates of full period measures are then 4 times more frequent than if only one kind of period was monitored.

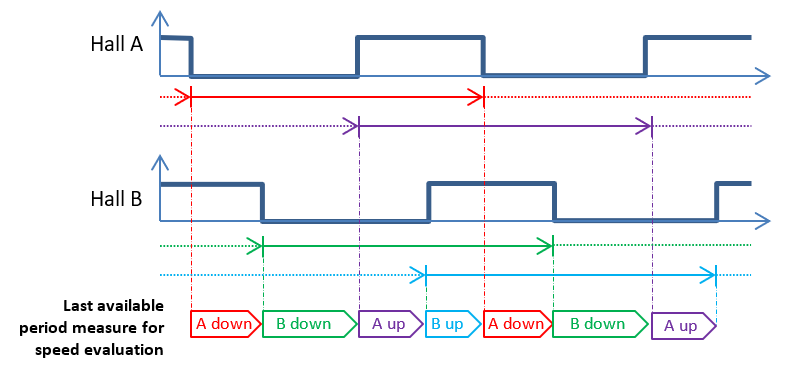


Figure : Four kind of periods simultaneous measuring

Note: Even if both sensor assembly error and magnetic asymmetries are handled by the full period measures, all minor error can’t be definitively canceled. They can potentially come from:

* Translation displacement of the magnetic wheel.
  + Note: this can be only evaluated at mechanical design level.
  + However, the possible impact should be negligible compared to sensor assembly errors impact or magnetic asymmetry impact (without considering the full period).
* Period measure rounding, due to the timer’s granularity:
  + This is visible only at high speed rotation that is to say with very short periods to measure
  + As example:
    - if considering that the timer lsb last 1.192µs
    - if considering a 20000°s rotation speed, then a period of 1ms (360 / (18 x 20000) = 0.001s)
    - 🡺 The resulting error remains below 0.12% which is negligible.
* Speed computation rounding:
  + This is visible only at extreme low speed.
  + As example, the step between 10°s and 11°s represents 10% of the value.

### Single edge measurement

The BMM provides a bobbin velocity computed on a single edge instead of a full period. This data is used by the “Velocity Loop Control”. It’s useful for low speed and add a better reaction time of the whole system.

Every 5° of displacement, the bobbin velocity is computed based on the current timer value.

Using this feature, a no movement belt state can be decided based on the timer T5 overflow interrupt.

### Downgraded states

When one or both sensors are in error, the BMM enters the downgraded state. It is composed of two levels:

* Level 1: One hall is faulty. Its monitoring is disable but we keep the second hall sensor working in order to detect pyro activation. The faulty sensor is disabled to avoid unexpected input signals which could lead to a wrong position/speed computation and pyro detection.
* Level 2: Both halls sensors are faulty. The BMM timer is fully disabled. It’s no longer possible to evaluate belt data and detect the pyro activation.

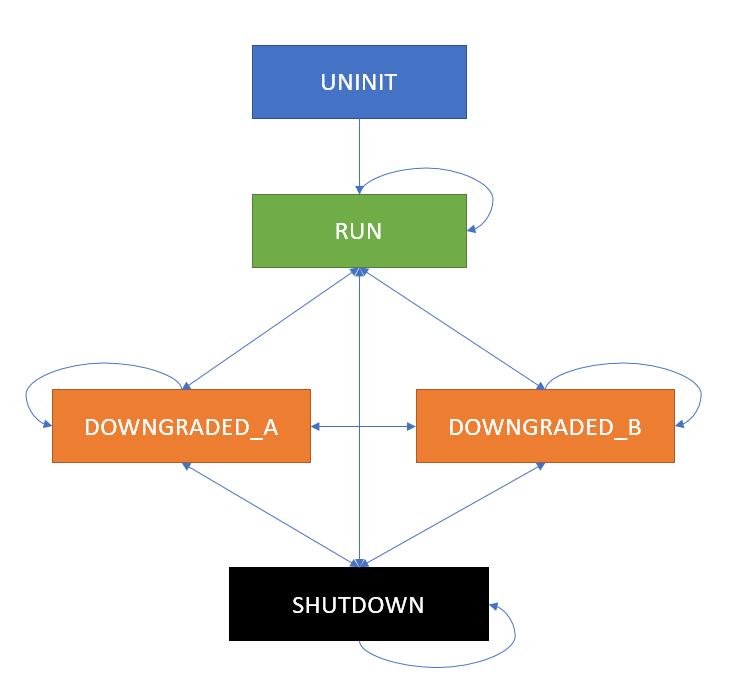


Figure : BMM internal states

#### Impact on belt position measurement

When both all sensors are working, the internal belt position counter is increased or decreased (based on belt rotation) by 1 on each sensor edge.

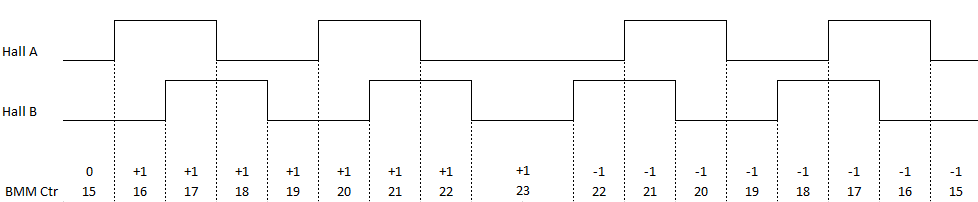


Figure : Standard case

Let’s assume we keep this implementation when one hall sensor is out of order. Will not detect the change in the bobbin rotation. In downgraded state, we are not able to provide the correct belt position.

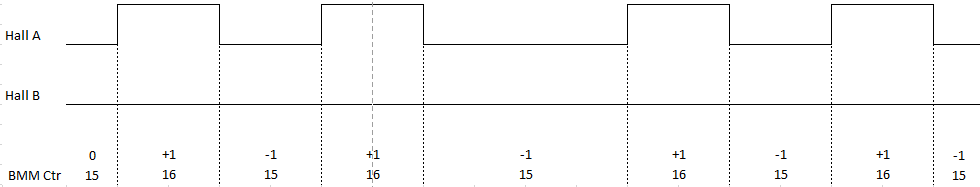


Figure : Hall B malfunction

We need both all sensors to determine the belt movement.

Speed computation can still be made using the single edge measurement feature.

### Detection of pyro activation

The pyro activation (in case of crash) has to be detected very soon to apply appropriate motor driving.  
Very high but short bobbin accelerations are characteristic of the pyro activation. Neither such accelerations can’t be obtained thru the motor tensioning, nor in the releasing direction thru the driver or passenger action.

Considering the required reactivity and the shortness of the high speed peak, the detection tests have to be performed for all measured periods of end of hall effect sensors (external periodic speed testing are prohibited).

Moreover, the pyro detection shall be robust and works even if one hall sensor doesn’t works. Consequently, the belt displacement during the period is intentionally ignored. Detection criteria are:

* 2 consecutive periods bellow a duration threshold customizable in NVP.
* The 2 consecutive periods can occur on 2 different hall signals (nominal case, hall A then hall B, or hall B then hall A) or on the same hall signal (case where one sensor is no longer working).

## Traceability

|  |  |  |  |
| --- | --- | --- | --- |
| Requirements | Criteria | Linked Runnable | Source |
| DSG\_BMM\_0001 | The driver shall monitor at every 2 ms both the filtered physical value of voltages measured on the Hall sensors. | BMM\_runMainFunction | ARCH\_SW\_BMM\_0005, ARCH\_SW\_BMM\_0006, ARCH\_SW\_BMM\_0007 |
| DSG\_BMM\_0002 | In case that both voltages measured on Hall Effect Sensors are out of range, the driver state shall be BMM\_BOTH\_SENSORS\_ERROR. | BMM\_runMainFunction | ARCH\_SW\_BMM\_0005, ARCH\_SW\_BMM\_0006, ARCH\_SW\_BMM\_0007 |
| DSG\_BMM\_0003 | In case only the voltage measured on Hall Effect Sensor A is out of range, the driver state shall be BMM\_HALL\_A\_ERROR. | BMM\_runMainFunction | ARCH\_SW\_BMM\_0005, ARCH\_SW\_BMM\_0006, ARCH\_SW\_BMM\_0007 |
| DSG\_BMM\_0004 | In case only the voltage measured on Hall Effect Sensor B is out of range, the driver state shall be BMM\_HALL\_B\_ERROR. | BMM\_runMainFunction | ARCH\_SW\_BMM\_0005, ARCH\_SW\_BMM\_0006, ARCH\_SW\_BMM\_0007 |
| DSG\_BMM\_0005 | In case that both voltages measured on Hall Effect Sensors are in range, the driver state shall be BMM\_SENSORS\_OK. | BMM\_runMainFunction | ARCH\_SW\_BMM\_0005, ARCH\_SW\_BMM\_0006, ARCH\_SW\_BMM\_0007 |
| DSG\_BMM\_0006 | The driver shall properly set the pins and timers used during the initialization. | BMM\_runInit | ARCH\_SW\_BMM\_0010, ARCH\_SW\_BMM\_0011 |
| DSG\_BMM\_0007 | All related variables shall be set to default during initialization. | BMM\_runInit | ARCH\_SW\_BMM\_0010, ARCH\_SW\_BMM\_0011 |
| DSG\_BMM\_0008 | The position of the belt related to the position recorded at start-up shall be calculated in degrees and made available to the rest of the application. | BMM\_runGetPositionFromT0\_deg | ARCH\_SW\_BMM\_0015 |
| DSG\_BMM\_0009 | The position of the belt related to the position recorded at start-up shall be calculated in millimeters and made available to the rest of the application. | BMM\_runGetPositionFromT0\_mm | ARCH\_SW\_BMM\_0020 |
| DSG\_BMM\_0010 | The position of the belt related to the minimum position measured from start-up shall be calculated in degrees and made available to the rest of the application. | BMM\_runGetPositionFromMinPos\_deg | ARCH\_SW\_BMM\_0025 |
| DSG\_BMM\_0011 | The position of the belt related to the minimum position measured from start-up shall be calculated in millimeters and made available to the rest of the application. | BMM\_runGetPositionFromMinPos\_mm | ARCH\_SW\_BMM\_0030 |
| DSG\_BMM\_0012 | The minimum position of the belt measured since system start-up shall be updated in real time and made available to the application in degrees. | BMM\_runGetPositionRange\_deg | ARCH\_SW\_BMM\_0035 |
| DSG\_BMM\_0013 | The maximum position of the belt measured since system start-up shall be updated in real time and made available to the application in degrees. | BMM\_runGetPositionRange\_deg | ARCH\_SW\_BMM\_0035 |
| DSG\_BMM\_0014 | The minimum position of the belt measured since system start-up shall be updated in real time and made available to the application in millimeters. | BMM\_runGetPositionRange\_mm | ARCH\_SW\_BMM\_0040 |
| DSG\_BMM\_0015 | The maximum position of the belt measured since system start-up shall be updated in real time and made available to the application in millimeters. | BMM\_runGetPositionRange\_mm | ARCH\_SW\_BMM\_0040 |
| DSG\_BMM\_0016 | The speed of the belt shall be calculated in degrees per second and provided real time to the rest of the application. | BMM\_runGetSpeed\_deg\_s | ARCH\_SW\_BMM\_0045 |
| DSG\_BMM\_0017 | The speed of the belt shall be calculated in millimeters per second and provided real time to the rest of the application. | BMM\_runGetSpeed\_mm\_s | ARCH\_SW\_BMM\_0048 |
| DSG\_BMM\_0018 | The voltages on the Hall Effect Sensors outputs shall be monitored and the unexpected values reported to the application. | BMM\_Autotest\_CheckHallEffectSensor | ARCH\_SW\_BMM\_0340, ARCH\_SW\_BMM\_0345, ARCH\_SW\_BMM\_0346, ARCH\_SW\_BMM\_0347, ARCH\_SW\_BMM\_0348 |
| DSG\_BMM\_0019 | The driver shall be able to be deactivated on request. | BMM\_runDisableHBSMonitoring | ARCH\_SW\_BMM\_1060 |
| DSG\_BMM\_0020 | The driver shall be able to be activated on request. | BMM\_runEnableHBSMonitoring | ARCH\_SW\_BMM\_1080 |
| DSG\_BMM\_0021 | When activated, the driver shall monitor and compute for speed, direction and position calculation the signals received from the sensors, taking into consideration the HW state of the devices. | OS\_ISR\_BMM\_HES\_IT | ARCH\_SW\_BMM\_1100 |
| DSG\_BMM\_0022 | When activated, the driver shall be able to detect a “no movement” state of the belt and update the speed and direction accordingly. | OS\_ISR\_BMM\_HES\_TOUT | ARCH\_SW\_BMM\_1200 |
| DSG\_BMM\_0023 | Local variable BMM\_u16MeasuredPeriod will be set as current period by reading the detected HES A and HES B phases. | OS\_ISR\_BMM\_HES\_IT | ARCH\_SW\_BMM\_0053; ARCH\_SW\_BMM\_0054 |
| DSG\_BMM\_0024 | Local variable BMM\_u16MeasuredPeriod will be computed in order to check if the calculated phase is lower than NVP period threshold for setting the pyro activation status as active. | OS\_ISR\_BMM\_HES\_IT | ARCH\_SW\_BMM\_0055 |
| DSG\_BMM\_0025 | Local variable b8IsPyroActivated will be set as pyro device not activated by default. | BMM\_runGetPyroActivationStatus | ARCH\_SW\_BMM\_0057 |
| DSG\_BMM\_0026 | Local variable b8IsPyroActivated will be set with the value pyro device activated if pyro activation is detected. | BMM\_runGetPyroActivationStatus | ARCH\_SW\_BMM\_0058 |
| DSG\_BMM\_0027 | Local variable u16CplTimerCntThrs will hold the computation of Ecpl Bobbin Speed Threshold from microseconds to the timer scale by using NVP\_u16CplBobbinSpeedThrs parameter. | BMM\_runInit | ARCH\_SW\_BMM\_0008 |
| DSG\_BMM\_0028 | In order to detect the a HES signal phase the HES level will be read. | OS\_ISR\_BMM\_HES\_IT | ARCH\_SW\_BMM\_0349; ARCH\_SW\_BMM\_0055 |
| DSG\_BMM\_0029 | The speed of the belt shall be calculated for a single edge in degrees per second and provided real time to the rest of the application. | BMM\_runGetSpeed\_SingleEdge\_deg\_s | ARCH\_SW\_BMM\_0049 |

# Features

## Services

### BMM\_Autotest\_CheckHallEffectSensor

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| This auto-test aims at checking whether both Hall-effect sensors are working properly. This autotest retrieves each HES input voltage value from de ASM module and checks them if they are in correct range. If at least one of both values is out of range, this autotest returns 'failed' result, otherwise it returns 'passed' result. | | | |
| **Prototype** | | | |
| void BMM\_Autotest\_CheckHallEffectSensor (u8TestResultType \* pu8TestResult) | | | |
| **Exceptions** | | | |
| None. | | | |
| **Precondition** | | | |
| None. | | | |
| **Postcondition** | | | |
| None. | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| pu8TestResult | u8TestResultType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | NA | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| \* | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### bmm\_ReadSpeed

|  |  |  |
| --- | --- | --- |
| Object | | |
| Computes the belt displacement speed in degrees per seconds, or in millimetres per seconds. | | |
| **Prototype** | | |
| LOCAL void bmm\_ReadSpeed (const sint32 s32SpeedFactor, sint32 \*const ps32Speed) | | |
| **Exceptions** | | |
| None | | |
| **Precondition** | | |
| None | | |
| **Postcondition** | | |
| None | | |
| **Input parameters** | | |
| Name | Type | Description |
| s32SpeedFactor | const sint32 | multiplication factor used to compute the rotation speed in degrees per seconds, or in millimeters per seconds. |
| **Output parameters** | | |
| Name | Type | Description |
| ps32Speed | sint32 \*const | Pointer to write the bobbin rotation speed |
| **Return value** | | |
| Type | Description | |
| NA | None | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| BMM | \* | |
| **Static aspect** | | |
| \* | | |
| **Constrains** | | |
|  | | |



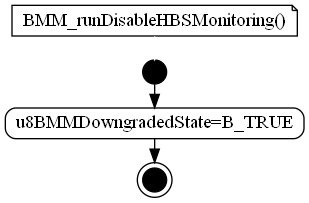
### bmm\_ReadSpeedSingleEdge

|  |  |  |
| --- | --- | --- |
| Object | | |
| Function will return the speed for a single edge. | | |
| **Prototype** | | |
| LOCAL void bmm\_ReadSpeedSingleEdge (const sint32 s32SpeedFactor, sint32 \*const ps32Speed) | | |
| **Exceptions** | | |
| None | | |
| **Precondition** | | |
| None | | |
| **Postcondition** | | |
| None | | |
| **Input parameters** | | |
| Name | Type | Description |
| s32SpeedFactor | const sint32 | multiplication factor used to compute the rotation speed in degrees per seconds, or in millimetres per seconds |
| **Output parameters** | | |
| Name | Type | Description |
| ps32Speed | sint32 \*const | Pointer to write the bobbin rotation speed |
| **Return value** | | |
| Type | Description | |
| NA | None | |
| **Dynamic aspect** | | |
| Who(callers) | Description | |
| BMM | \* | |
| **Static aspect** | | |
| \* | | |
| **Constrains** | | |
|  | | |



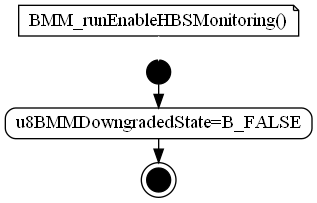
### BMM\_runDisableHBSMonitoring

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will disable the HEs monitoring. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runDisableHBSMonitoring (void) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Runned on request by the application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### BMM\_runEnableHBSMonitoring

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will enable the HES monitoring. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runEnableHBSMonitoring (void) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Runned on request by the application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### BMM\_runGetPositionFromT0\_deg

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will get the position from T0 in degrees. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetPositionFromT0\_deg (s16BeltPosition\_degType \* ps16BeltPosition\_deg) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| ps16BeltPosition\_deg | s16BeltPosition\_degType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



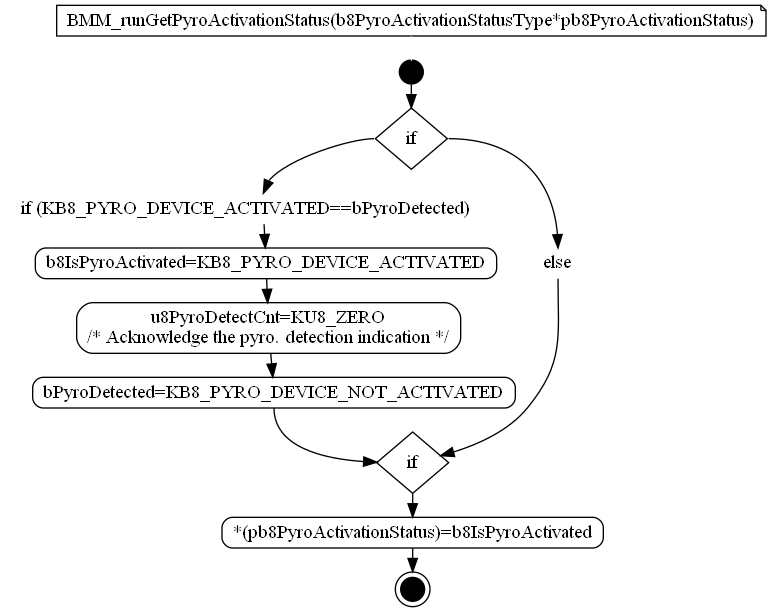
### BMM\_runGetPositionFromT0\_mm

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will get the position from T0 in mm. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetPositionFromT0\_mm (s16BeltPosition\_mmType \* ps16BeltPosition\_mm) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| ps16BeltPosition\_mm | s16BeltPosition\_mmType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



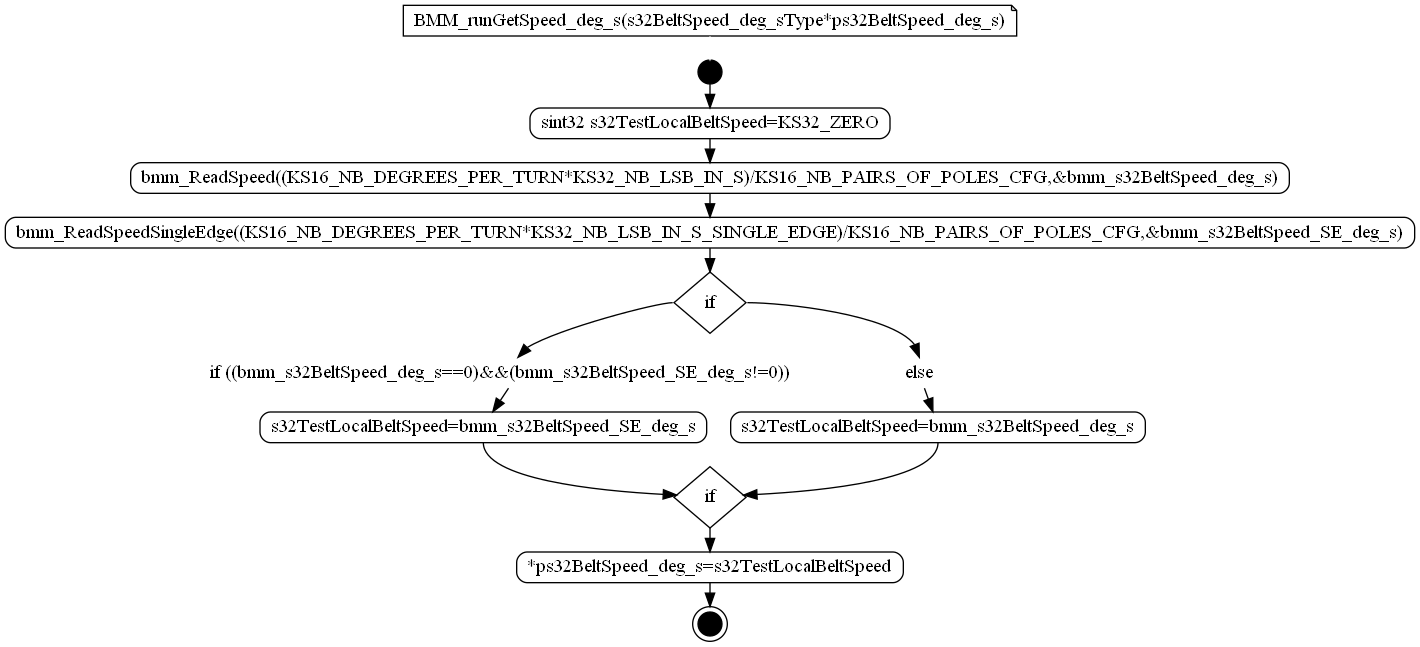
### BMM\_runGetPyroActivationStatus

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will get the pyro activation status. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetPyroActivationStatus (b8PyroActivationStatusType \* pb8PyroActivationStatus) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| pb8PyroActivationStatus | b8PyroActivationStatusType \* | Pointer used to store the status | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### BMM\_runGetSpeed\_deg\_s

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will compute current belt speed in deg. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetSpeed\_deg\_s (s32BeltSpeed\_deg\_sType \* ps32BeltSpeed\_deg\_s) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| ps32BeltSpeed\_deg\_s | s32BeltSpeed\_deg\_sType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



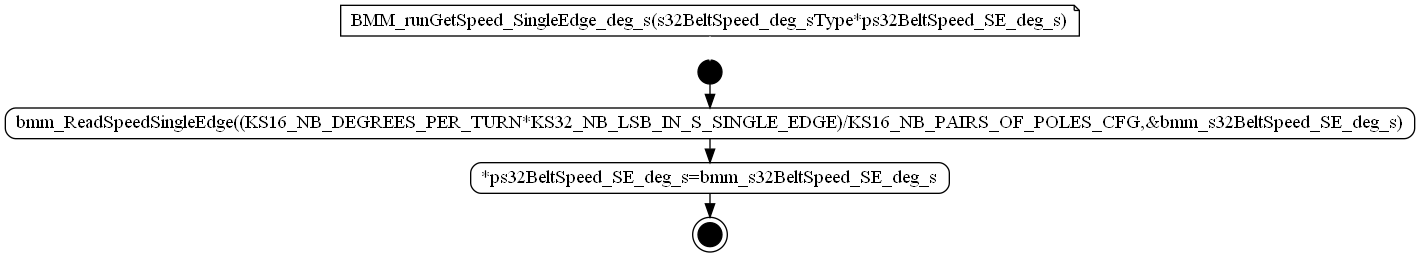
### BMM\_runGetSpeed\_mm\_s

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will compute current belt speed in mm. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetSpeed\_mm\_s (s32BeltSpeed\_mm\_sType \* ps32BeltSpeed\_mm\_s) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| ps32BeltSpeed\_mm\_s | s32BeltSpeed\_mm\_sType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### BMM\_runGetSpeed\_SingleEdge\_deg\_s

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will compute current belt speed from single edge in deg. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runGetSpeed\_SingleEdge\_deg\_s (s32BeltSpeed\_deg\_sType \* ps32BeltSpeed\_SE\_deg\_s) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | |
| ps32BeltSpeed\_SE\_deg\_s | s32BeltSpeed\_deg\_sType \* | Pointer to store the requested result. | |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Asynchronous called from application. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### BMM\_runInit

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Function will initialize the module variables and used HW. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runInit (void) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Scheduler (init phase). | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



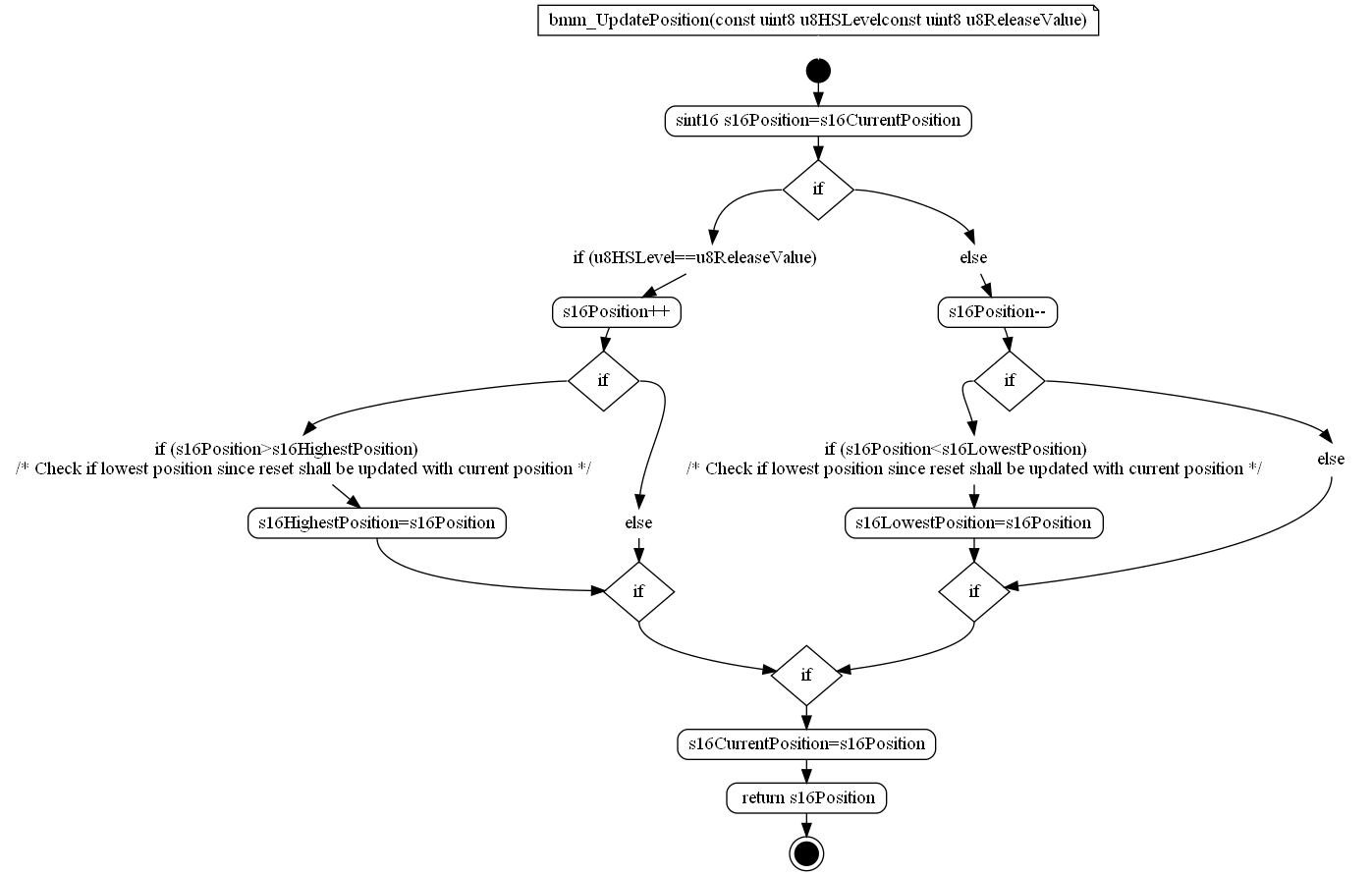
### BMM\_runMainFunction

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Cyclic function of the module, its scope is to verify the electronic integrity of the HAL sensors. | | | |
| **Prototype** | | | |
| EXPORTED void BMM\_runMainFunction (void) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Scheduler (every 2ms) | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### bmm\_UpdatePosition

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Sub function to increment or decrement current position on hall sensor edge. Detect Highest and lowest position exceeding and update if necessary. | | | |
| **Prototype** | | | |
| LOCAL sint16 bmm\_UpdatePosition (const uint8 u8HSLevel, const uint8 u8ReleaseValue) | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | |
| u8HSLevel | const uint8 | The hall output level to be considered (STD\_LOW / STD\_HIGH). | |
| u8ReleaseValue | const uint8 | The expected output level for release direction (STD\_LOW / STD\_HIGH). | |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| sint16 | The just updated position. | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| BMM | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### OS\_ISR\_BMM\_HES\_IT

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Interrupt triggered at each edge transition. | | | |
| **Prototype** | | | |
| void Isr\_Bmm\_Hes\_It (void) | | | |
| **Parameters** | | | |
| None | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Triggered by external physical event. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



### OS\_ISR\_BMM\_HES\_TOUT

|  |  |  |  |
| --- | --- | --- | --- |
| Object | | | |
| Interrupt triggered at Timer 5 overflow. | | | |
| **Prototype** | | | |
| void Isr\_BMM\_HES\_TOUT (void) | | | |
| **Parameters** | | | |
| None | | | |
| **Exceptions** | | | |
| None | | | |
| **Precondition** | | | |
| None | | | |
| **Postcondition** | | | |
| None | | | |
| **Input parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Output parameters** | | | |
| Name | Type | Description | Range |
| NA | NA | NA | NA |
| **Return value** | | | |
| Type | Description | | |
| NA | None | | |
| **Dynamic aspect** | | | |
| Who(callers) | Description | | |
| Triggered by external physical event. | \* | | |
| **Static aspect** | | | |
| \* | | | |
| **Constrains** | | | |
|  | | | |



## Variabiles

### as16Positions[KU8\_NUMBER\_OF\_PERIOD\_TYPES]

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s16PositionType | NA | |
| **Description** | | |
| Remember the bobbin position at last occurrence of each kind of edge marking a period begin. | | |
| **Definition** | | |
| LOCAL s16PositionType as16Positions[KU8\_NUMBER\_OF\_PERIOD\_TYPES] | | |
| **Remarks** | | |
| None. | | |

### au16Captures[KU8\_NUMBER\_OF\_PERIOD\_TYPES]

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint16 | NA | |
| **Description** | | |
| Remember the timer's values which have been captured on last occurrence of each kind of edge marking a period begin. | | |
| **Definition** | | |
| LOCAL volatile uint16 au16Captures[KU8\_NUMBER\_OF\_PERIOD\_TYPES] | | |
| **Remarks** | | |
| None. | | |

### au8Overflows[KU8\_NUMBER\_OF\_PERIOD\_TYPES]

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint8 | NA | |
| **Description** | | |
| Count independently for each kind of input capture edge (regarding polarity and sensor source) the number of times the timer has overflowed during the period. | | |
| **Definition** | | |
| LOCAL volatile uint8 au8Overflows[KU8\_NUMBER\_OF\_PERIOD\_TYPES] | | |
| **Remarks** | | |
| None. | | |

### b8IsPyroActivated

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| b8PyroActivationStatusType | NA | |
| **Description** | | |
| Local pyro device status. | | |
| **Definition** | | |
| LOCAL b8PyroActivationStatusType b8IsPyroActivated | | |
| **Remarks** | | |
| None. | | |

### bCriticalNok

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile boolean | B\_FALSE | |
| **Description** | | |
| Check if all critical auto tests are done and passed. | | |
| **Definition** | | |
| LOCAL volatile boolean bCriticalNok = B\_FALSE | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPosition\_max\_deg

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_degType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_degType bmm\_s16BeltPosition\_max\_deg | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPosition\_max\_mm

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_mmType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_mmType bmm\_s16BeltPosition\_max\_mm | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPosition\_min\_deg

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_degType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_degType bmm\_s16BeltPosition\_min\_deg | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPosition\_min\_mm

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_mmType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_mmType bmm\_s16BeltPosition\_min\_mm | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPositionFromMinPos\_deg

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_degType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_degType bmm\_s16BeltPositionFromMinPos\_deg | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPositionFromMinPos\_mm

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_mmType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_mmType bmm\_s16BeltPositionFromMinPos\_mm | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPositionFromT0\_deg

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_degType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_degType bmm\_s16BeltPositionFromT0\_deg | | |
| **Remarks** | | |
| None. | | |

### bmm\_s16BeltPositionFromT0\_mm

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16BeltPosition\_mmType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL volatile s16BeltPosition\_mmType bmm\_s16BeltPositionFromT0\_mm | | |
| **Remarks** | | |
| None. | | |

### bmm\_s32BeltSpeed\_deg\_s

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s32BeltSpeed\_deg\_sType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL s32BeltSpeed\_deg\_sType bmm\_s32BeltSpeed\_deg\_s | | |
| **Remarks** | | |
| None. | | |

### bmm\_s32BeltSpeed\_mm\_s

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s32BeltSpeed\_mm\_sType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL s32BeltSpeed\_mm\_sType bmm\_s32BeltSpeed\_mm\_s | | |
| **Remarks** | | |
| None. | | |

### bmm\_s32BeltSpeed\_SE\_deg\_s

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s32BeltSpeed\_deg\_sType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL s32BeltSpeed\_deg\_sType bmm\_s32BeltSpeed\_SE\_deg\_s | | |
| **Remarks** | | |
| None. | | |

### bmm\_s32BeltSpeed\_SE\_mm\_s

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s32BeltSpeed\_deg\_sType | NA | |
| **Description** | | |
| Variable used to store values passed by module services. | | |
| **Definition** | | |
| LOCAL s32BeltSpeed\_deg\_sType bmm\_s32BeltSpeed\_SE\_mm\_s | | |
| **Remarks** | | |
| None. | | |

### bmm\_State

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| bmm\_stInternalStateType | {Uninit, Uninit} | |
| **Description** | | |
| BMM internal state. | | |
| **Definition** | | |
| LOCAL bmm\_stInternalStateType bmm\_State = {Uninit, Uninit} | | |
| **Remarks** | | |
| None. | | |

### bmm\_tSETimerConfig[((uint8) 2u)]

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile bmm\_LowSpeedIncHwCounterType | NA | |
| **Description** | | |
| Low speed config. | | |
| **Definition** | | |
| LOCAL volatile bmm\_LowSpeedIncHwCounterType bmm\_tSETimerConfig[((uint8) 2u)] | | |
| **= {** | | |
| { ((uint32)1015808uL) , ((uint32)1245184uL) , ((uint32)0x400ul) }, { ((uint32)1245199uL) , ((uint32)16773120uL) , ((uint32)0xffffuL) }, | | |
| **Remarks** | | |
| None. | | |

### bmm\_tTimerConfig[((uint8) 2u)]

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile bmm\_LowSpeedIncHwCounterType | NA | |
| **Description** | | |
| Low speed config. | | |
| **Definition** | | |
| LOCAL volatile bmm\_LowSpeedIncHwCounterType bmm\_tTimerConfig[((uint8) 2u)] | | |
| **= {** | | |
| { ((uint32)1015808uL) , ((uint32)1245184uL) , ((uint32)0x400uL) }, { ((uint32)1245199uL) , ((uint32)16773120uL) , ((uint32)0xffffuL) }, | | |
| **Remarks** | | |
| None. | | |

### bmm\_u32LowSpeedHwCounterTotalOffset

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint32 | KU32\_ZERO | |
| **Description** | | |
| Variable used in order to increase the timer value exponentially in case of belt deceleration. | | |
| **Definition** | | |
| LOCAL volatile uint32 bmm\_u32LowSpeedHwCounterTotalOffset = KU32\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bmm\_u32LowSpeedSEHwCounterTotalOffset

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint32 | KU32\_ZERO | |
| **Description** | | |
| Variable used in order to increase the timer value exponentially in case of belt deceleration. | | |
| **Definition** | | |
| LOCAL volatile uint32 bmm\_u32LowSpeedSEHwCounterTotalOffset = KU32\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bmm\_u32StaticNextPeriodLength

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint32 | KU32\_ZERO | |
| **Description** | | |
| Variable used to cap the timer value in case of belt deceleration. | | |
| **Definition** | | |
| LOCAL volatile uint32 bmm\_u32StaticNextPeriodLength = KU32\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bmm\_u32StaticSENextPeriodLength

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint32 | KU32\_ZERO | |
| **Description** | | |
| Variable used to cap the timer value in case of belt deceleration. | | |
| **Definition** | | |
| LOCAL volatile uint32 bmm\_u32StaticSENextPeriodLength = KU32\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bmm\_u8SETimerConfigI

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint8 | KU8\_ZERO | |
| **Description** | | |
| Used to check intervals on which to increase the timer value exponentially. | | |
| **Definition** | | |
| LOCAL volatile uint8 bmm\_u8SETimerConfigI = KU8\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bmm\_u8TimerConfigI

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile uint8 | KU8\_ZERO | |
| **Description** | | |
| Used to check intervals on which to increase the timer value exponentially. | | |
| **Definition** | | |
| LOCAL volatile uint8 bmm\_u8TimerConfigI = KU8\_ZERO | | |
| **Remarks** | | |
| None. | | |

### bNewEvent

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile boolean | NA | |
| **Description** | | |
| Semaphore to ensure the consistency of variables modified under interrupts and read for speed evaluation. | | |
| **Definition** | | |
| LOCAL volatile boolean bNewEvent | | |
| **Remarks** | | |
| None. | | |

### bPyroDetected

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile b8PyroActivationStatusType | NA | |
| **Description** | | |
| The pyro firing detection flag. | | |
| **Definition** | | |
| LOCAL volatile b8PyroActivationStatusType bPyroDetected | | |
| **Remarks** | | |
| None. | | |

### s16CurrentPosition

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16PositionType | NA | |
| **Description** | | |
| Belt displacement in number of input edge count (combining hall A and hall B channels for rising and falling edges). = 0 at initialization (the belt has never moved yet = force null speed) > 0 after a releasing belt move < 0 after a tensioning belt move. | | |
| **Definition** | | |
| LOCAL volatile s16PositionType s16CurrentPosition | | |
| **Remarks** | | |
| None. | | |

### s16HighestPosition

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s16PositionType | NA | |
| **Description** | | |
| The lowest and highest value of s16CurrentPosition since reset. | | |
| **Definition** | | |
| LOCAL s16PositionType s16HighestPosition | | |
| **Remarks** | | |
| None. | | |

### s16LastPeriodDisplacement

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile s16PositionType | NA | |
| **Description** | | |
| Delta between begin and end of the last completed period. | | |
| **Definition** | | |
| LOCAL volatile s16PositionType s16LastPeriodDisplacement | | |
| **Remarks** | | |
| None. | | |

### s16LowestPosition

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| s16PositionType | NA | |
| **Description** | | |
| The lowest and highest value of s16CurrentPosition since reset. | | |
| **Definition** | | |
| LOCAL s16PositionType s16LowestPosition | | |
| **Remarks** | | |
| None. | | |

### stSingleEdgeData

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile bmm\_stSingleEdgeData | {KU32\_ZERO, KU16\_ZERO, KU8\_ZERO} | |
| **Description** | | |
| Single edge algorithm data structure. | | |
| **Definition** | | |
| LOCAL volatile bmm\_stSingleEdgeData stSingleEdgeData = {KU32\_ZERO, KU16\_ZERO, KU8\_ZERO} | | |
| **Remarks** | | |
| None. | | |

### u16CplTimerCntThrs

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint16 | NA | |
| **Description** | | |
| Ecpl Bobbin Speed Threshold NVP parameter converted into timer value. | | |
| **Definition** | | |
| LOCAL uint16 u16CplTimerCntThrs | | |
| **Remarks** | | |
| None. | | |

### u32LastPeriodLength

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile u32PeriodType | NA | |
| **Description** | | |
| Last completed period duration. This 'Last' period is supposed to be either between rising edges or between falling edges, occurred either on the hall A or on the hall B channel. The duration is given in timer's counter lsb and depends from hardware clock and timer prescaler configuration. | | |
| **Definition** | | |
| LOCAL volatile u32PeriodType u32LastPeriodLength | | |
| **Remarks** | | |
| None. | | |

### u8BMMDowngradedState

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | NA | |
| **Description** | | |
| BMM internal state base on hall effect sensor AEC B\_TRUE: HES pulse counting disabled => speed = 0 and belt position = the last value measured B\_FALSE: HES pulse counting enabled => nominal case. | | |
| **Definition** | | |
| LOCAL uint8 u8BMMDowngradedState | | |
| **Remarks** | | |
| None. | | |

### u8NextPeriodType

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| volatile bmm\_PeriodTypesType | NA | |
| **Description** | | |
| Expected type of period to be computed on next capture interrupt if the displacement direction does not change. | | |
| **Definition** | | |
| LOCAL volatile bmm\_PeriodTypesType u8NextPeriodType | | |
| **Remarks** | | |
| None. | | |

### u8PyroDetectCnt

|  |  |  |
| --- | --- | --- |
| Type | Value |  |
| uint8 | NA | |
| **Description** | | |
| Count the number of consecutive times a period length has been below the pyro speed detection threshold. | | |
| **Definition** | | |
| LOCAL uint8 u8PyroDetectCnt | | |
| **Remarks** | | |
| None. | | |

## Macros

### BMM\_BOTH\_SENSORS\_ERROR

|  |  |
| --- | --- |
| Name | Value |
| BMM\_BOTH\_SENSORS\_ERROR | ((uint8) 3) |
| **Definition** | |
| #define BMM\_BOTH\_SENSORS\_ERROR ((uint8) 3) | |
| **Description** | |
| HES Fault detection. | |

### BMM\_EXPECTED\_TMR\_PRESCALER

|  |  |
| --- | --- |
| Name | Value |
| BMM\_EXPECTED\_TMR\_PRESCALER | (NB\_CLK\_IN\_PERIOD/SOFT\_TIMER\_RANGE) |
| **Definition** | |
| #define BMM\_EXPECTED\_TMR\_PRESCALER (NB\_CLK\_IN\_PERIOD/SOFT\_TIMER\_RANGE) | |
| **Description** | |
| Macro used for timer prescaler value. | |

### BMM\_HALL\_A\_ERROR

|  |  |
| --- | --- |
| Name | Value |
| BMM\_HALL\_A\_ERROR | ((uint8) 1) |
| **Definition** | |
| #define BMM\_HALL\_A\_ERROR ((uint8) 1) | |
| **Description** | |
| HES Fault detection. | |

### BMM\_HALL\_B\_ERROR

|  |  |
| --- | --- |
| Name | Value |
| BMM\_HALL\_B\_ERROR | ((uint8) 2) |
| **Definition** | |
| #define BMM\_HALL\_B\_ERROR ((uint8) 2) | |
| **Description** | |
| HES Fault detection. | |

### BMM\_SENSORS\_OK

|  |  |
| --- | --- |
| Name | Value |
| BMM\_SENSORS\_OK | ((uint8) 0) |
| **Definition** | |
| #define BMM\_SENSORS\_OK ((uint8) 0) | |
| **Description** | |
| HES Fault detection. | |

### DIO\_HALL\_A

|  |  |
| --- | --- |
| Name | Value |
| DIO\_HALL\_A | (Dio\_ReadChannel(DioConf\_DioChannel\_HES\_A)) |
| **Definition** | |
| #define DIO\_HALL\_A (Dio\_ReadChannel(DioConf\_DioChannel\_HES\_A)) | |
| **Description** | |
| Macro to get the hall sensor 'A' output level, among STD\_LOW and STD\_HIGH. | |

### DIO\_HALL\_B

|  |  |
| --- | --- |
| Name | Value |
| DIO\_HALL\_B | (Dio\_ReadChannel(DioConf\_DioChannel\_HES\_B)) |
| **Definition** | |
| #define DIO\_HALL\_B (Dio\_ReadChannel(DioConf\_DioChannel\_HES\_B)) | |
| **Description** | |
| Macro to get the hall sensor 'B' output level, among STD\_LOW and STD\_HIGH. | |

### EXPECTED\_PRESCALER

|  |  |
| --- | --- |
| Name | Value |
| EXPECTED\_PRESCALER | ((uint32) 128) |
| **Definition** | |
| #define EXPECTED\_PRESCALER ((uint32) 128) | |
| **Description** | |
| Macro used for timer prescaler value. | |

### HARD\_TIMER\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| HARD\_TIMER\_RANGE | (0x10000uL) |
| **Definition** | |
| #define HARD\_TIMER\_RANGE (0x10000uL) | |
| **Description** | |
| The 16 bits hardware timer range. | |

### KS16\_NB\_DEGREES\_PER\_TURN

|  |  |
| --- | --- |
| Name | Value |
| KS16\_NB\_DEGREES\_PER\_TURN | ((sint16)360) |
| **Definition** | |
| #define KS16\_NB\_DEGREES\_PER\_TURN ((sint16)360) | |
| **Description** | |
| Number of degrees rotation in a complete turn, for unit conversion. | |

### KS16\_NB\_PAIRS\_OF\_POLES\_CFG

|  |  |
| --- | --- |
| Name | Value |
| KS16\_NB\_PAIRS\_OF\_POLES\_CFG | ((sint16)BMM\_NB\_PAIRS\_OF\_POLES\_CFG) |
| **Definition** | |
| #define KS16\_NB\_PAIRS\_OF\_POLES\_CFG ((sint16)BMM\_NB\_PAIRS\_OF\_POLES\_CFG) | |
| **Description** | |
| Number of north+south poles couples. | |

### KS16\_NB\_PULSE\_PER\_PERIOD

|  |  |
| --- | --- |
| Name | Value |
| KS16\_NB\_PULSE\_PER\_PERIOD | ((s16PositionType)4) |
| **Definition** | |
| #define KS16\_NB\_PULSE\_PER\_PERIOD ((s16PositionType)4) | |
| **Description** | |
| Number of high pulses and low pulses (that is to say falling edge and rising edge) occurring on one hall effect sensor channel (A or B) during a hall A or hall B signal's period if the direction do not change. | |

### KS16\_NULL\_DT\_POSITION\_IN\_PERIOD

|  |  |
| --- | --- |
| Name | Value |
| KS16\_NULL\_DT\_POSITION\_IN\_PERIOD | ((s16PositionType)0) |
| **Definition** | |
| #define KS16\_NULL\_DT\_POSITION\_IN\_PERIOD ((s16PositionType)0) | |
| **Description** | |
| Position difference corresponding to a hall signal's period caused by both releasing and tensioning displacements that cancel each other. | |

### KS16\_RELEASE\_DT\_POSITION\_IN\_PERIOD

|  |  |
| --- | --- |
| Name | Value |
| KS16\_RELEASE\_DT\_POSITION\_IN\_PERIOD | ((s16PositionType)4) |
| **Definition** | |
| #define KS16\_RELEASE\_DT\_POSITION\_IN\_PERIOD ((s16PositionType)4) | |
| **Description** | |
| Position difference corresponding to a hall signal's period caused by a constant belt releasing displacement. | |

### KS16\_TENSION\_DT\_POSITION\_IN\_PERIOD

|  |  |
| --- | --- |
| Name | Value |
| KS16\_TENSION\_DT\_POSITION\_IN\_PERIOD | ((s16PositionType)-4) |
| **Definition** | |
| #define KS16\_TENSION\_DT\_POSITION\_IN\_PERIOD ((s16PositionType)-4) | |
| **Description** | |
| Position difference corresponding to a hall signal's period caused by a constant belt tensioning displacement. | |

### KS32\_NB\_LSB\_IN\_S

|  |  |
| --- | --- |
| Name | Value |
| KS32\_NB\_LSB\_IN\_S | ((sint32)(BMM\_CFG\_TIMER\_INPUT\_CLK/EXPECTED\_PRESCALER)) |
| **Definition** | |
| #define KS32\_NB\_LSB\_IN\_S ((sint32)(BMM\_CFG\_TIMER\_INPUT\_CLK/EXPECTED\_PRESCALER)) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### KS32\_NB\_LSB\_IN\_S\_SINGLE\_EDGE

|  |  |
| --- | --- |
| Name | Value |
| KS32\_NB\_LSB\_IN\_S\_SINGLE\_EDGE | ((sint32) (KS32\_NB\_LSB\_IN\_S/KU8\_FOUR)) |
| **Definition** | |
| #define KS32\_NB\_LSB\_IN\_S\_SINGLE\_EDGE ((sint32) (KS32\_NB\_LSB\_IN\_S/KU8\_FOUR)) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### KU16\_HARD\_TIMER\_HALF\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| KU16\_HARD\_TIMER\_HALF\_RANGE | ((uint16)0x8000) |
| **Definition** | |
| #define KU16\_HARD\_TIMER\_HALF\_RANGE ((uint16)0x8000) | |
| **Description** | |
| The hardware timer half range. Timer counter value at timer overflow period's middle. | |

### KU16\_RESET\_COUNTER

|  |  |
| --- | --- |
| Name | Value |
| KU16\_RESET\_COUNTER | ((uint16)0) |
| **Definition** | |
| #define KU16\_RESET\_COUNTER ((uint16)0) | |
| **Description** | |
| Macro used to retrieve the reset timer counter value. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_1 | ((uint32)0x400uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_1 ((uint32)0x400uL) | |
| **Description** | |
| Low speed factor 1. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_2 | ((uint32)0xffffuL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_FACTOR\_2 ((uint32)0xffffuL) | |
| **Description** | |
| Low speed factor 1. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_1 | ((uint32)1015808uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_1 ((uint32)1015808uL) | |
| **Description** | |
| Low speed start thres. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_2 | ((uint32)1245199uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_START\_THRS\_2 ((uint32)1245199uL) | |
| **Description** | |
| Low speed start thres. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_1 | ((uint32)1245184uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_1 ((uint32)1245184uL) | |
| **Description** | |
| Low speed stop thres. | |

### KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_2 | ((uint32)16773120uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_STOP\_THRS\_2 ((uint32)16773120uL) | |
| **Description** | |
| Low speed stop thres. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_1 | ((uint32)0x400ul) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_1 ((uint32)0x400ul) | |
| **Description** | |
| Low speed factor 1. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_2 | ((uint32)0xffffuL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_FACTOR\_2 ((uint32)0xffffuL) | |
| **Description** | |
| Low speed factor 1. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_1 | ((uint32)1015808uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_1 ((uint32)1015808uL) | |
| **Description** | |
| Low speed start thres. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_2 | ((uint32)1245199uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_START\_THRS\_2 ((uint32)1245199uL) | |
| **Description** | |
| Low speed start thres. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_1

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_1 | ((uint32)1245184uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_1 ((uint32)1245184uL) | |
| **Description** | |
| Low speed stop thres. | |

### KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_2

|  |  |
| --- | --- |
| Name | Value |
| KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_2 | ((uint32)16773120uL) |
| **Definition** | |
| #define KU32\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_STOP\_THRS\_2 ((uint32)16773120uL) | |
| **Description** | |
| Low speed stop thres. | |

### KU32\_NB\_LSB\_IN\_MS

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NB\_LSB\_IN\_MS | (KU32\_NB\_LSB\_IN\_S / KU32\_NB\_MS\_IN\_S) |
| **Definition** | |
| #define KU32\_NB\_LSB\_IN\_MS (KU32\_NB\_LSB\_IN\_S / KU32\_NB\_MS\_IN\_S) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### KU32\_NB\_LSB\_IN\_S

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NB\_LSB\_IN\_S | ((uint32)(BMM\_CFG\_TIMER\_INPUT\_CLK/EXPECTED\_PRESCALER)) |
| **Definition** | |
| #define KU32\_NB\_LSB\_IN\_S ((uint32)(BMM\_CFG\_TIMER\_INPUT\_CLK/EXPECTED\_PRESCALER)) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### KU32\_NB\_MS\_IN\_S

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NB\_MS\_IN\_S | ((uint32) NB\_MS\_IN\_S) |
| **Definition** | |
| #define KU32\_NB\_MS\_IN\_S ((uint32) NB\_MS\_IN\_S) | |
| **Description** | |
| NA | |

### KU32\_NB\_US\_IN\_MS

|  |  |
| --- | --- |
| Name | Value |
| KU32\_NB\_US\_IN\_MS | ((uint32)NB\_US\_IN\_MS) |
| **Definition** | |
| #define KU32\_NB\_US\_IN\_MS ((uint32)NB\_US\_IN\_MS) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### KU8\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS

|  |  |
| --- | --- |
| Name | Value |
| KU8\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS | ((uint8)2u) |
| **Definition** | |
| #define KU8\_BMM\_LOW\_SPEED\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS ((uint8)2u) | |
| **Description** | |
| Number of low speed configurations. | |

### KU8\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS

|  |  |
| --- | --- |
| Name | Value |
| KU8\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS | ((uint8)2u) |
| **Definition** | |
| #define KU8\_BMM\_LOW\_SPEED\_SE\_INC\_HW\_COUNTER\_NB\_OF\_INTERVALS ((uint8)2u) | |
| **Description** | |
| Number of low speed configurations. | |

### KU8\_DEGREES\_PER\_EDGE

|  |  |
| --- | --- |
| Name | Value |
| KU8\_DEGREES\_PER\_EDGE | ((uint8)5) |
| **Definition** | |
| #define KU8\_DEGREES\_PER\_EDGE ((uint8)5) | |
| **Description** | |
| Number of degrees rotation per same edge. | |

### NB\_CLK\_IN\_PERIOD

|  |  |
| --- | --- |
| Name | Value |
| NB\_CLK\_IN\_PERIOD | ((BMM\_CFG\_TIMER\_INPUT\_CLK/NB\_MS\_IN\_S)\*BMM\_PERIOD\_MAX\_MS) |
| **Definition** | |
| #define NB\_CLK\_IN\_PERIOD ((BMM\_CFG\_TIMER\_INPUT\_CLK/NB\_MS\_IN\_S)\*BMM\_PERIOD\_MAX\_MS) | |
| **Description** | |
| The period max configuration parameter converted into core clock lsb, knowing that the core clocks directly the Timer1. | |

### NB\_MS\_IN\_S

|  |  |
| --- | --- |
| Name | Value |
| NB\_MS\_IN\_S | (1000uL) |
| **Definition** | |
| #define NB\_MS\_IN\_S (1000uL) | |
| **Description** | |
| Number of milliseconds in a seconds, for unit conversion. | |

### NB\_US\_IN\_MS

|  |  |
| --- | --- |
| Name | Value |
| NB\_US\_IN\_MS | (1000uL) |
| **Definition** | |
| #define NB\_US\_IN\_MS (1000uL) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### NB\_US\_IN\_S

|  |  |
| --- | --- |
| Name | Value |
| NB\_US\_IN\_S | (1000000uL) |
| **Definition** | |
| #define NB\_US\_IN\_S (1000000uL) | |
| **Description** | |
| Compute the counting frequency in Hz and kHz. | |

### SOFT\_TIMER\_RANGE

|  |  |
| --- | --- |
| Name | Value |
| SOFT\_TIMER\_RANGE | (0x100u \* HARD\_TIMER\_RANGE) |
| **Definition** | |
| #define SOFT\_TIMER\_RANGE (0x100u \* HARD\_TIMER\_RANGE) | |
| **Description** | |
| The software timer full range: Correspond to a 24 bits timer, made with a combination of the hardware 16 bits timer and the overflow 8 bits counter. | |

### U16\_COUNTER\_VALUE

|  |  |
| --- | --- |
| Name | Value |
| U16\_COUNTER\_VALUE | ((uint16)GPT120\_T5.B.T5) |
| **Definition** | |
| #define U16\_COUNTER\_VALUE ((uint16)GPT120\_T5.B.T5) | |
| **Description** | |
| Macro used to retrieve the current timer counter value. | |

### U16\_COUNTER\_VALUE\_STATIC

|  |  |
| --- | --- |
| Name | Value |
| U16\_COUNTER\_VALUE\_STATIC | ((uint16)GPT120\_CAPREL.B.CAPREL) |
| **Definition** | |
| #define U16\_COUNTER\_VALUE\_STATIC ((uint16)GPT120\_CAPREL.B.CAPREL) | |
| **Description** | |
| Macro used to retrieve the stored timer counter value. | |

### BMM\_CFG\_TIMER\_INPUT\_CLK

|  |  |
| --- | --- |
| Name | Value |
| BMM\_CFG\_TIMER\_INPUT\_CLK | (80000000ul) |
| **Definition** | |
| #define BMM\_CFG\_TIMER\_INPUT\_CLK (80000000ul) | |
| **Description** | |
| The clock of timer used for BRS function in Hz. | |

### BMM\_ENABLE\_MAX\_POS\_MONITORING

|  |  |
| --- | --- |
| Name | Value |
| BMM\_ENABLE\_MAX\_POS\_MONITORING | 1 |
| **Definition** | |
| #define BMM\_ENABLE\_MAX\_POS\_MONITORING 1 | |
| **Description** | |
| Macro to enable lowest and highest positions detection feature. | |

### BMM\_ENABLE\_PYRO\_DETECTION

|  |  |
| --- | --- |
| Name | Value |
| BMM\_ENABLE\_PYRO\_DETECTION | 1 |
| **Definition** | |
| #define BMM\_ENABLE\_PYRO\_DETECTION 1 | |
| **Description** | |
| Macro to enable pyro detection feature. | |

### BMM\_NB\_PAIRS\_OF\_POLES\_CFG

|  |  |
| --- | --- |
| Name | Value |
| BMM\_NB\_PAIRS\_OF\_POLES\_CFG | (18uL) |
| **Definition** | |
| #define BMM\_NB\_PAIRS\_OF\_POLES\_CFG (18uL) | |
| **Description** | |
| Configuration parameter which defines the number of pairs of poles present on the magnetic wheel. | |

### BMM\_PERIOD\_MAX\_MS

|  |  |
| --- | --- |
| Name | Value |
| BMM\_PERIOD\_MAX\_MS | (1000uL\*(360uL/BMM\_NB\_PAIRS\_OF\_POLES\_CFG)) |
| **Definition** | |
| #define BMM\_PERIOD\_MAX\_MS (1000uL\*(360uL/BMM\_NB\_PAIRS\_OF\_POLES\_CFG)) | |
| **Description** | |
| NA | |

### BMM\_S16\_BOBBIN\_PERIMETER\_IN\_MM\_CFG

|  |  |
| --- | --- |
| Name | Value |
| BMM\_S16\_BOBBIN\_PERIMETER\_IN\_MM\_CFG | ((sint32)168) |
| **Definition** | |
| #define BMM\_S16\_BOBBIN\_PERIMETER\_IN\_MM\_CFG ((sint32)168) | |
| **Description** | |
| Defines the average belt bobbin perimeter in mm. | |

# EEPROM

The EEPROM parameters are all specified in [Doc1 = SBE\_4G\_NVP\_layout.xls].

Refer to this document for more details.

# Configuration

No special configuration for BMM software component.

# Compilation Options

No compilation options for BMM software component.