

API 仕様書

Rev0.4

最終更新日：2021/1/20

Rev0.1	2020/05/14	初回作成
Rev0.2	2020/08/06	IC-EG のレビュー指摘対応
Rev0.3	2020/08/07	getHoodDoor 追加
Rev0.4	2021/01/20	clusterInit、clusterTerm 関数追加。及び、誤記修正

目次

API 仕様書.....	1
1 目的.....	6
2 概要.....	6
3 API 詳細.....	8
3-1 Enum Definition.....	8
3-1-1 IC_HMI_ON_OFF.....	8
3-1-2 IC_HMI_GEAR_AT_VAL.....	9
3-1-3 IC_HMI_GEAR_MT_VAL.....	9
3-1-4 IC_HMI_SP_UNIT_VAL.....	10
3-1-5 IC_HMI_TRCOM_UNIT_VAL.....	10
3-1-6 IC_HMI_FUEL_ECONOMY_UNIT_VAL.....	11
3-1-7 IC_HMI_OTEMP_UNIT_VAL.....	11
3-2 Macro Definition.....	12
3-2-1 TellTale.....	12
3-2-1-1 IC_HMI_TT_TURN_R.....	12
3-2-1-2 IC_HMI_TT_TURN_L.....	12
3-2-1-3 IC_HMI_TT_BRAKE.....	13
3-2-1-4 IC_HMI_TT_SEATBELT.....	13
3-2-1-5 IC_HMI_TT_HIGHBEAM.....	13
3-2-1-6 IC_HMI_TT_DOOR.....	14
3-2-1-7 IC_HMI_TT_EPS.....	14
3-2-1-8 IC_HMI_TT_SRS_AIRBAG.....	14
3-2-1-9 IC_HMI_TT_ABS.....	15
3-2-1-10 IC_HMI_TT_LOW_BATTERY.....	15
3-2-1-11 IC_HMI_TT_OIL_PRESS.....	15
3-2-1-12 IC_HMI_TT_ENGINE.....	16
3-2-1-13 IC_HMI_TT_FUEL.....	16
3-2-1-14 IC_HMI_TT_IMMOBI.....	17
3-2-1-15 IC_HMI_TT_TM_FAIL.....	17
3-2-1-16 IC_HMI_TT_ESP_ACT.....	17
3-2-1-17 IC_HMI_TT_ESP_OFF.....	18
3-2-1-18 IC_HMI_TT_ADAPTING_LIGHTING.....	18
3-2-1-19 IC_HMI_TT_AUTO_STOP.....	18
3-2-1-20 IC_HMI_TT_AUTO_STOP_FAIL.....	19

3-2-1-21 IC_HMI_TT_PARKING_LIGHTS	19
3-2-1-22 IC_HMI_TT_FRONT_FOG	19
3-2-1-23 IC_HMI_TT_EXTERIOR_LIGHT_FAULT	20
3-2-1-24 IC_HMI_TT_ACC_FAIL	20
3-2-1-25 IC_HMI_TT_LDW_OFF	21
3-2-1-26 IC_HMI_TT_HILL_DESCENT	21
3-2-1-27 IC_HMI_TT_AUTO_HI_BEAM_GREEN	21
3-2-1-28 IC_HMI_TT_AUTO_HI_BEAM_AMBER	22
3-2-1-29 IC_HMI_TT_LDW_OPERATE	22
3-2-1-30 IC_HMI_TT_GENERAL_WARN	22
3-2-1-31 IC_HMI_TT_SPORTS_MODE	23
3-2-1-32 IC_HMI_TT_DRIVING_POWER_MODE	23
3-2-1-33 IC_HMI_TT_HOT_TEMP	23
3-2-1-34 IC_HMI_TT_LOW_TEMP	24
3-2-1-35 IC_HMI_TT_ALL	24
3-3 Callback Function	25
3-3-1 IC_HMI_FUNC_NOTIFY_IC_HMI	25
3-4 Telltale	25
3-4-1 getTurnR	25
3-4-2 getTurnL	26
3-4-3 getBrake	26
3-4-4 getSeatbelt	27
3-4-5 getFrontRightSeatbelt	28
3-4-6 getFrontCenterSeatbelt	28
3-4-7 getFrontLeftSeatbelt	29
3-4-8 getMid1RightSeatbelt	29
3-4-9 getMid1CenterSeatbelt	30
3-4-10 getMid1LeftSeatbelt	30
3-4-11 getMid2RightSeatbelt	31
3-4-12 getMid2CenterSeatbelt	32
3-4-13 getMid2LeftSeatbelt	32
3-4-14 getRearRightSeatbelt	33
3-4-15 getRearCenterSeatbelt	33
3-4-16 getRearLeftSeatbelt	34
3-4-17 getHighbeam	34
3-4-18 getDoor	35

3-4-19	getFrontRightDoor	35
3-4-20	getFrontLeftDoor.....	36
3-4-21	getRearRightDoor.....	37
3-4-22	getRearLeftDoor	37
3-4-23	getTrunkDoor	38
3-4-24	getHoodDoor.....	38
3-4-25	getEps	39
3-4-26	getSrsAirbag	39
3-4-27	getAbs.....	40
3-4-28	getLowBattery	40
3-4-29	getOilPress	41
3-4-30	getEngine.....	42
3-4-31	getFuel	42
3-4-32	getImmobi.....	43
3-4-33	getTMFail	43
3-4-34	getEspAct.....	44
3-4-35	getEspOff.....	44
3-4-36	getAdaptingLighting.....	45
3-4-37	getAutoStop	45
3-4-38	getAutoStopFail	46
3-4-39	getParkingLights.....	47
3-4-40	getFrontFog.....	47
3-4-41	getExteriorLightFault	48
3-4-42	getAccFail	48
3-4-43	getLdwOff.....	49
3-4-44	getHillDescent.....	49
3-4-45	getAutoHiBeamGreen.....	50
3-4-46	getAutoHiBeamAmber.....	50
3-4-47	getSportsMode.....	51
3-4-48	getLdwOperate	52
3-4-49	getGeneralWarn.....	52
3-4-50	getDrivingPowerMode	53
3-4-51	getHotTemp.....	53
3-4-52	getLowTemp.....	54
3-5	ShiftPosition	54
3-5-1	getGearAtVal	54

3-5-2 getGearMtVal.....	55
3-6 Speed	56
3-6-1 getSpAnalogVal.....	56
3-6-2 getSpAnaDigUnitVal.....	57
3-7 Tacho.....	57
3-7-1 getTaAnalogVal	57
3-8 TripComputer.....	58
3-8-1 getTrcomTripAVal	58
3-8-2 getTrcomTripBVal	59
3-8-3 getTrcomOdoVal	59
3-8-4 getTrcomUnitVal	60
3-8-5 getAvgSpeedAVal	60
3-8-6 getAvgSpeedBVal	61
3-8-7 getHourAVal	62
3-8-8 getHourBVal	62
3-8-9 getMinuteAVal	63
3-8-10 getMinuteBVal.....	64
3-8-11 getSecondAVal.....	64
3-8-12 getSecondBVal.....	65
3-8-13 getOTempVal.....	66
3-8-14 getOTempUnitVal.....	66
3-8-15 getCruRangeVal.....	67
3-8-16 getAvgFuelAVal.....	67
3-8-17 getAvgFuelBVal.....	68
3-8-18 getInsFuelAVal.....	69
3-8-19 getInsFuelBVal.....	69
3-8-20 getFuelEconomyUnitVal	70
3-9 Register/Notify	71
3-9-1 registerIchmi.....	71
3-9-2 notifyIchmi	72
3-10 Initalize.....	73
3-10-1 clusterInit	73
3-10-2 clusterTerm	74
4.改訂履歷.....	75

1 目的

本仕様書は、HMI Layer の Cluster UI が、共有メモリにセットされた IC-Service のシグナル情報を取得するための Interface について記載する。

2 概要

画面描画を行う HMI Layer の Cluster UI と機能処理を行う Service Layer の IC-Service 間における API を定義する。

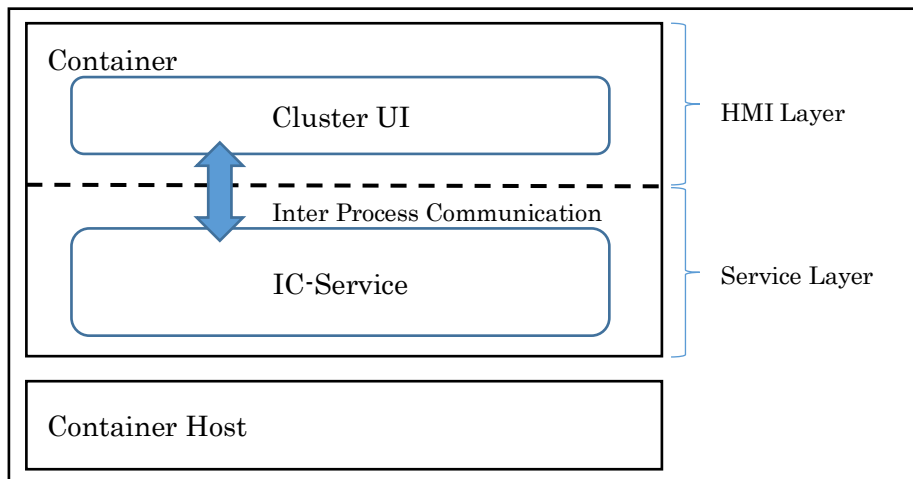


図 2.1 HMI Layer の位置づけ

HMI Layer の Cluster UI と Service Layer の IC-Service 間は、Inter Process Communication の共有メモリを使用して、通信を行う。

共有メモリの使い方は、Polling 方式と Event 方式を用意する。

① Polling 方式

IC-Service は共有メモリの値に変化があった場合、setData 関数などを用いて、値の更新を行う。

Cluster UI は周期的に getData 関数などを実行し、値の変化にかかわらず、共有メモリから値を取得する。

以下に概略図を示す。

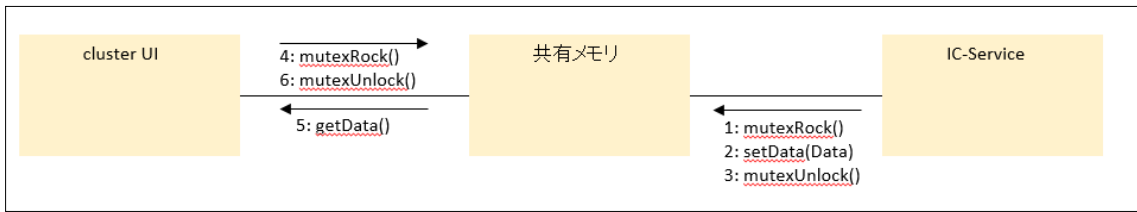


図 2.2 Polling 方式における通信

② Event 方式

Cluster UI は通知を受けたいシグナルを IC-Service に登録する。IC-Service は共有メモリの値を変更する場合は、setData 関数などを用いて、値の更新を行う。その際、Cluster UI から登録された共有メモリの値が変わった場合は、Notify 関数で Cluster UI へ通知を行う。

Notify 関数では変更したシグナルと状態を ClusterUI に通知させ、Cluster UI は値を取得する。

以下に概略図を示す。

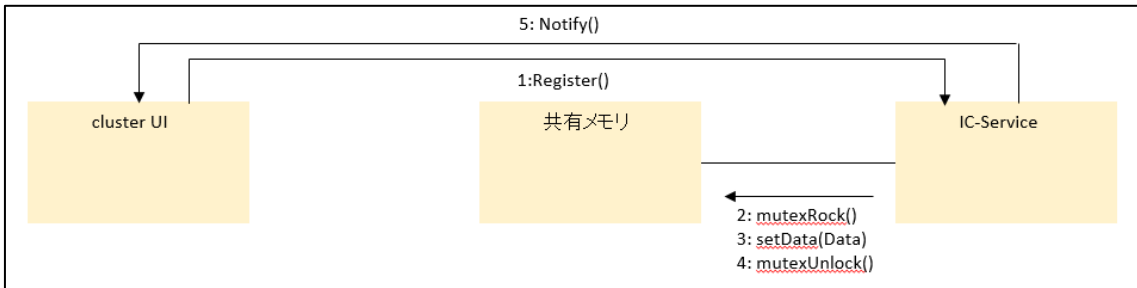


図 2.3 Event 方式における通信

3 API 詳細

本項は、各シグナル取得するために使用する API の詳細を記述する。

Enum 定数定義の詳細については次の 1～4 で形成される。

- 1 : API の名前
- 2 : API の説明
- 3 : 定数名、値及び説明
- 4 : 特記事項

Macro 定数定義の詳細については次の 1～4 で形成される。

- 1 : API の名前
- 2 : API の説明
- 3 : 値及び説明
- 4 : 特記事項

シグナル取得 API の詳細については次の 1～6 で形成される。

- 1 : プロトタイプ宣言
- 2 : API の説明
- 3 : 戻り値の型種別
- 4 : 戻り値の詳細
- 5 : 引数の詳細
- 6 : 特記事項

3-1 Enum Definition

3-1-1 IC_HMI_ON_OFF

1:API の名前

IC_HMI_ON_OFF

2:API の説明

Telltale の ON/OFF 状態

3:定数名及び値

0 : IC_HMI_OFF OFF 状態

1 : IC_HMI_ON ON 状態

4:特記事項

特になし

3-1-2 IC_HMI_GEAR_AT_VAL

1: API の名前

IC_HMI_GEAR_AT_VAL

2:API の説明

ギア状態(AT)

3:定数名及び値

0 : IC_HMI_AT_OFF	ギアが非表示状態
1 : IC_HMI_AT_PARKING	ギアが Parking 状態
2 : IC_HMI_AT_REVERSE	ギアが Reverse 状態
3 : IC_HMI_AT_NEUTRAL	ギアが Neutral 状態
4 : IC_HMI_AT_DRIVE	ギアが Drive 状態
5 : IC_HMI_AT_SECOND	ギアが Second 状態
6 : IC_HMI_AT_LOW_GEAR	ギアが LowGear 状態
7 : IC_HMI_AT_FIRST	ギアが First 状態
8 : IC_HMI_AT_MANUAL	ギアが Manual 状態
9 : IC_HMI_AT_BRAKE	ギアが Brake 状態
10 : IC_HMI_AT_ALL_ON	全点灯状態
11 : IC_HMI_AT_ALL_OFF	全消灯状態
12 : IC_HMI_AT_UNUSED	ギアが未使用状態

4:特記事項

特になし

3-1-3 IC_HMI_GEAR_MT_VAL

1: API の名前

IC_HMI_GEAR_MT_VAL

2:API の説明

ギア状態(MT)

3:定数名及び値

0 : IC_HMI_MT_OFF	ギアが非表示状態
1 : IC_HMI_MT_FIRST	ギアが 1 速状態
2 : IC_HMI_MT_SECOND	ギアが 2 速状態
3 : IC_HMI_MT_THIRD	ギアが 3 速状態
4 : IC_HMI_MT_FOURTH	ギアが 4 速状態
5 : IC_HMI_MT_FIFTH	ギアが 5 速状態
6 : IC_HMI_MT_SIXTH	ギアが 6 速状態
7 : IC_HMI_MT_SEVENTH	ギアが 7 速状態
8 : IC_HMI_MT_EIGHTTH	ギアが 8 速状態

4:特記事項

特になし

3-1-4 IC_HMI_SP_UNIT_VAL

1: API の名前

IC_HMI_SP_UNIT_VAL

2:API の説明

車速単位

3:定数名及び値

0 : IC_HMI_SP_KM_H	車速単位が km/h 状態
1 : IC_HMI_SP_MPH	車速単位が mph 状態

4:特記事項

特になし

3-1-5 IC_HMI_TRCOM_UNIT_VAL

1: API の名前

IC_HMI_TRCOM_UNIT_VAL

2:API の説明

トリップコンピュータ単位

3:定数名及び値

0 : IC_HMI_TRCOM_KM トリップコンピュータ単位が km 状態
1 : IC_HMI_TRCOM_MILE トリップコンピュータ単位が mile 状態

4:特記事項

特になし

3-1-6 IC_HMI_FUEL_ECONOMY_UNIT_VAL

1:API の名前

IC_HMI_FUEL_ECONOMY_UNIT_VAL

2:API の説明

燃費表示単位

- ・瞬間燃費
- ・平均燃費 A/B
- ・瞬間電費
- ・瞬間電費 A/B

3:定数名及び値

0 : IC_HMI_FUEL_KM_L 燃費表示単位が km/L 状態
1 : IC_HMI_FUEL_MPG_US 燃費表示単位が MPG(U.S gallon) 状態
2 : IC_HMI_FUEL_MPG_IG 燃費表示単位が MPG(imperial gallon) 状態
3 : IC_HMI_FUEL_L_100KM 燃費表示単位が L/100km 状態
4 : IC_HMI_FUEL_MILE_KWH 燃費表示単位が mile/kwh 状態
5 : IC_HMI_FUEL_KM_KWH 燃費表示単位が km/kwh 状態
6 : IC_HMI_FUEL_MILE_KG 燃費表示単位が mile/kg 状態
7 : IC_HMI_FUEL_KM_KG 燃費表示単位が km/kg 状態
8 : IC_HMI_FUEL_KWH_100KM 燃費表示単位が kwh/100km 状態

4:特記事項

特になし

3-1-7 IC_HMI_OTEMP_UNIT_VAL

1: API の名前

IC_HMI_OTEMP_UNIT_VAL

2:API の説明

外気温単位

3:定数名及び値

0 : IC_HMI_OTEMP_UNIT_C

外気温単位が摂氏状態

1 : IC_HMI_OTEMP_UNIT_K

外気温単位が華氏状態

4:特記事項

特になし

3-2 Macro Definition

3-2-1 TellTale

3-2-1-1 IC_HMI_TT_TURN_R

1: API の名前

IC_HMI_TT_TURN_R

2:API の説明

右方向指示器状態のビットフラグ

3:値

0x0000000000000001

4:特記事項

特になし

3-2-1-2 IC_HMI_TT_TURN_L

1: API の名前

IC_HMI_TT_TURN_L

2:API の説明

左方向指示器状態のビットフラグ

3:値

0x0000000000000002

4:特記事項

特になし

3-2-1-3 IC_HMI_TT_BRAKE

1: API の名前

IC_HMI_TT_BRAKE

2:API の説明

ブレーキ警告灯状態のビットフラグ

3:値

0x0000000000000004

4:特記事項

特になし

3-2-1-4 IC_HMI_TT_SEATBELT

1: API の名前

IC_HMI_TT_SEATBELT

2:API の説明

シートベルト警告灯状態のビットフラグ

3:値

0x0000000000000008

4:特記事項

特になし

3-2-1-5 IC_HMI_TT_HIGHBEAM

1: API の名前

IC_HMI_TT_HIGHBEAM

2:API の説明

ビームパッシング警告灯状態のビットフラグ

3:値

0x00000000000000010

4:特記事項

特になし

3-2-1-6 IC_HMI_TT_DOOR

1: API の名前

IC_HMI_TT_DOOR

2:API の説明

半ドア警告灯状態のビットフラグ

3:値

0x00000000000000020

4:特記事項

特になし

3-2-1-7 IC_HMI_TT_EPS

1: API の名前

IC_HMI_TT_EPS

2:API の説明

EPS 警告灯状態のビットフラグ

3:値

0x00000000000000040

4:特記事項

特になし

3-2-1-8 IC_HMI_TT_SRS_AIRBAG

1: API の名前

IC_HMI_SRS_TT_AIRBAG

2:API の説明

SRS エアバッグ警告灯状態のビットフラグ

3:値

0x00000000000000080

4:特記事項

特になし

3-2-1-9 IC_HMI_TT_ABS

1: API の名前

IC_HMI_TT_ABS

2:API の説明

ABS 警告灯状態のビットフラグ

3:値

0x00000000000000100

4:特記事項

特になし

3-2-1-10 IC_HMI_TT_LOW_BATTERY

1: API の名前

IC_HMI_TT_LOW_BATTERY

2:API の説明

チャージ警告灯状態のビットフラグ

3:値

0x00000000000000200

4:特記事項

特になし

3-2-1-11 IC_HMI_TT_OIL_PRESS

1: API の名前

IC_HMI_TT_OIL_PRESS

2: API の説明

オイルプレッシャー警告灯状態のビットフラグ

3: 値

0x00000000000000400

4: 特記事項

特になし

3-2-1-12 IC_HMI_TT_ENGINE

1: API の名前

IC_HMI_TT_ENGINE

2: API の説明

エンジン警告灯状態のビットフラグ

3: 値

0x00000000000000800

4: 特記事項

特になし

3-2-1-13 IC_HMI_TT_FUEL

1: API の名前

IC_HMI_TT_FUEL

2: API の説明

燃料残量警告灯状態のビットフラグ

3: 値

0x00000000000001000

4: 特記事項

特になし

3-2-1-14 IC_HMI_TT_IMMOBI

1: API の名前

IC_HMI_TT_IMMOBI

2: API の説明

Immobilizer 警告灯状態のビットフラグ

3: 値

0x00000000000002000

4: 特記事項

特になし

3-2-1-15 IC_HMI_TT_TM_FAIL

1: API の名前

IC_HMI_TT_TM_FAIL

2: API の説明

TM Failure 警告灯状態のビットフラグ

3: 値

0x00000000000004000

4: 特記事項

特になし

3-2-1-16 IC_HMI_TT_ESP_ACT

1: API の名前

IC_HMI_TT_ESP_ACT

2: API の説明

ESP-ACT 警告灯状態のビットフラグ

3: 値

0x00000000000008000

4:特記事項

特になし

3-2-1-17 IC_HMI_TT_ESP_OFF

1: API の名前

IC_HMI_TT_ESP_OFF

2:API の説明

ESP-OFF 警告灯状態のビットフラグ

3:値

0x00000000000010000

4:特記事項

特になし

3-2-1-18 IC_HMI_TT_ADAPTING_LIGHTING

1: API の名前

IC_HMI_TT_ADAPTING_LIGHTING

2:API の説明

Adapting Lightig 警告灯状態のビットフラグ

3:値

0x00000000000020000

4:特記事項

特になし

3-2-1-19 IC_HMI_TT_AUTO_STOP

1: API の名前

IC_HMI_TT_AUTO_STOP

2:API の説明

Auto Stop 警告灯状態のビットフラグ

3:値

0x00000000000040000

4:特記事項

特になし

3-2-1-20 IC_HMI_TT_AUTO_STOP_FAIL

1: API の名前

IC_HMI_TT_AUTO_STOP_FAIL

2:API の説明

Auto Stop Fail 警告灯状態のビットフラグ

3:値

0x00000000000080000

4:特記事項

特になし

3-2-1-21 IC_HMI_TT_PARKING_LIGHTS

1: API の名前

IC_HMI_TT_PARKING_LIGHTS

2:API の説明

Parking Lights 警告灯状態のビットフラグ

3:値

0x0000000000100000

4:特記事項

特になし

3-2-1-22 IC_HMI_TT_FRONT_FOG

1: API の名前

IC_HMI_TT_FRONT_FOG

2:API の説明

FrontFog 警告灯状態のビットフラグ

3:値

0x0000000000200000

4:特記事項

特になし

3-2-1-23 IC_HMI_TT_EXTERIOR_LIGHT_FAULT

1: API の名前

IC_HMI_TT_EXTERIOR_LIGHT_FAULT

2:API の説明

ExteriorLightFault 警告灯状態のビットフラグ

3:値

0x0000000000400000

4:特記事項

特になし

3-2-1-24 IC_HMI_TT_ACC_FAIL

1: API の名前

IC_HMI_TT_ACC_FAIL

2:API の説明

ACC-Failure 警告灯状態のビットフラグ

3:値

0x0000000000800000

4:特記事項

特になし

3-2-1-25 IC_HMI_TT_LDW_OFF

1: API の名前

IC_HMI_TT_LDW_OFF

2: API の説明

Lane Departure Warning OFF 警告灯状態のビットフラグ

3: 値

0x0000000001000000

4: 特記事項

特になし

3-2-1-26 IC_HMI_TT_HILL_DESCENT

1: API の名前

IC_HMI_TT_HILL_DESCENT

2: API の説明

Hill-Descent 警告灯状態のビットフラグ

3: 値

0x0000000002000000

4: 特記事項

特になし

3-2-1-27 IC_HMI_TT_AUTO_HI_BEAM_GREEN

1: API の名前

IC_HMI_TT_AUTO_HI_BEAM_GREEN

2: API の説明

AutoHiBeamGreen 警告灯状態のビットフラグ

3: 値

0x0000000004000000

4:特記事項

特になし

3-2-1-28 IC_HMI_TT_AUTO_HI_BEAM_AMBER

1: API の名前

IC_HMI_TT_AUTO_HI_BEAM_AMBER

2:API の説明

AutoHiBeamAmber 警告灯状態のビットフラグ

3:値

0x0000000008000000

4:特記事項

特になし

3-2-1-29 IC_HMI_TT_LDW_OPERATE

1: API の名前

IC_HMI_TT_LDW_OPERATE

2:API の説明

Lane Departure Warning Operate 警告灯状態のビットフラグ

3:値

0x0000000010000000

4:特記事項

特になし

3-2-1-30 IC_HMI_TT_GENERAL_WARN

1: API の名前

IC_HMI_TT_GENERAL_WARN

2:API の説明

General Warn 警告灯状態のビットフラグ

3:値

0x0000000020000000

4:特記事項

特になし

3-2-1-31 IC_HMI_TT_SPORTS_MODE

1: API の名前

IC_HMI_TT_SPORTS_MODE

2:API の説明

Sports Mode 警告灯状態のビットフラグ

3:値

0x0000000040000000

4:特記事項

特になし

3-2-1-32 IC_HMI_TT_DRIVING_POWER_MODE

1: API の名前

IC_HMI_TT_DRIVING_POWER_MODE

2:API の説明

Driver Power Mode 警告灯状態のビットフラグ

3:値

0x0000000080000000

4:特記事項

特になし

3-2-1-33 IC_HMI_TT_HOT_TEMP

1: API の名前

IC_HMI_TT_HOT_TEMP

2:API の説明

Hot Temp 警告灯状態のビットフラグ

3:値

0x0000000100000000

4:特記事項

特になし

3-2-1-34 IC_HMI_TT_LOW_TEMP

1: API の名前

IC_HMI_TT_LOW_TEMP

2:API の説明

Low Temp 警告灯状態のビットフラグ

3:値

0x0000000200000000

4:特記事項

特になし

3-2-1-35 IC_HMI_TT_ALL

1: API の名前

IC_HMI_TT_ALL

2:API の説明

全警告灯を ON にするビットフラグ

3:値

0xFFFFFFFFFFFFFFFF

4:特記事項

特になし

3-3 Callback Function

3-3-1 IC_HMI_FUNC_NOTIFY_IC_HMI

1:API の名前

```
typedef void(* IC_HMI_FUNC_NOTIFY_IC_HMI)(unsigned long long arg_1,  
                                           IC_HMI_ON_OFF arg_2)
```

2:API の説明

Event 方式で使用する registerIchmi 関数の引数に渡すコールバック関数の型

3:返り値の型種別

void

4:返り値の詳細

None

5:引数の詳細

unsigned long long arg_1 : 取得するシグナルに対応したビットフラグ
各ビットに対応するシグナルは [3-2-1](#) 参照

IC_HMI_ON_OFF arg_2 : 変化検出後のシグナルの状態
各状態については [3-1-1](#) 参照

6:特記事項

特筆事項無し

3-4 Telltale

3-4-1 getTurnR

1:プロトタイプ宣言

```
IC_HMI_ON_OFF getTurnR(void)
```

2:API の説明

右方向指示器状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

点滅制御は IC-Service です。

3-4-2 getTurnL

1: プロトタイプ宣言

IC_HMI_ON_OFF getTurnL(void)

2: API の説明

左方向指示器状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

点滅制御は IC-Service です。

3-4-3 getBrake

1: プロトタイプ宣言

IC_HMI_ON_OFF getBrake(void)

2:API の説明

ブレーキ警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-4 getSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getSeatbelt(void)

2:API の説明

シートベルト警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

3-4-5 から 3-4-16 いずれか 1 つが OFF ならば戻り値 ON になる

3-4-5 getFrontRightSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getFrontRightSeatbelt(void)

2:APIの説明

最前部座席右のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-6 getFrontCenterSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getFrontCenterSeatbelt(void)

2:APIの説明

最前部座席中央のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-7 getFrontLeftSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getFrontLeftSeatbelt(void)

2:APIの説明

最前部座席左のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-8 getMid1RightSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getMid1RightSeatbelt(void)

2:APIの説明

2 列目右のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

3 列シート以上の車両における 2 列目のシートベルト情報を返す。

3 列シート未満の車両では使用しない

3-4-9 getMid1CenterSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getMid1CenterSeatbelt(void)

2:API の説明

2 列目中央のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

3 列シート以上の車両における 2 列目のシートベルト情報を返す。

3 列シート未満の車両では使用しない

3-4-10 getMid1LeftSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getMid1LeftSeatbelt(void)

2:API の説明

2 列目左のシートベルト状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5: 引数の詳細

None

6: 特記事項

3 列シート以上の車両における 2 列目のシートベルト情報を返す。

3 列シート未満の車両では使用しない

3-4-11 getMid2RightSeatbelt

1: プロトタイプ宣言

IC_HMI_ON_OFF getMid2RightSeatbelt(void)

2: API の説明

3 列目右のシートベルト状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5: 引数の詳細

None

6: 特記事項

4 列シート以上の車両における 3 列目のシートベルト情報を返す。

4 列シート未満の車両では使用しない

3-4-12 getMid2CenterSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getMid2CenterSeatbelt(void)

2:APIの説明

3 列目中央のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

4 列シート以上の車両における 3 列目のシートベルト情報を返す。

4 列シート未満の車両では使用しない

3-4-13 getMid2LeftSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getMid2LeftSeatbelt(void)

2:APIの説明

3 列目左のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

4 列シート以上の車両における 3 列目のシートベルト情報を返す。

4 列シート未満の車両では使用しない

3-4-14 getRearRightSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getRearRightSeatbelt(void)

2:API の説明

最後部座席右のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 未装着

IC_HMI_ON : 装着

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-15 getRearCenterSeatbelt

1:プロトタイプ宣言

IC_HMI_ON_OFF getRearCenterSeatbelt(void)

2:API の説明

最後部座席中央のシートベルト状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 未装着
IC_HMI_ON : 装着

5: 引数の詳細

None

6: 特記事項

特筆事項なし

3-4-16 getRearLeftSeatbelt

1: プロトタイプ宣言

IC_HMI_ON_OFF getRearLeftSeatbelt(void)

2: API の説明

最後部座席左のシートベルト状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 未装着
IC_HMI_ON : 装着

5: 引数の詳細

None

6: 特記事項

特筆事項なし

3-4-17 getHighbeam

1: プロトタイプ宣言

IC_HMI_ON_OFF getHighbeam(void)

2: API の説明

ビームパッシング警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項なし

3-4-18 getDoor

1: プロトタイプ宣言

IC_HMI_ON_OFF getDoor(void)

2: API の説明

半ドア警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

3-4-19 から 3-4-24 いずれか 1 つが ON ならば戻り値 ON になる

3-4-19 getFrontRightDoor

1: プロトタイプ宣言

IC_HMI_ON_OFF getFrontRightDoor(void)

2:API の説明

右前ドアの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 閉

IC_HMI_ON : 開

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-20 getFrontLeftDoor

1:プロトタイプ宣言

IC_HMI_ON_OFF getFrontLeftDoor(void)

2:API の説明

左前ドアの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 閉

IC_HMI_ON : 開

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-21 getRearRightDoor

1:プロトタイプ宣言

IC_HMI_ON_OFF getRearRightDoor(void)

2:APIの説明

右後ドアの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 閉

IC_HMI_ON : 開

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-22 getRearLeftDoor

1:プロトタイプ宣言

IC_HMI_ON_OFF getRearLeftDoor(void)

2:APIの説明

左後ドアの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 閉

IC_HMI_ON : 開

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-23 getTrunkDoor

1:プロトタイプ宣言

IC_HMI_ON_OFF getTrunkDoor(void)

2:APIの説明

トランクドアの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 閉

IC_HMI_ON : 開

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-24 getHoodDoor

1:プロトタイプ宣言

IC_HMI_ON_OFF getHoodDoor(void)

2:APIの説明

ボンネットの開閉状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 閉
IC_HMI_ON : 開

5: 引数の詳細

None

6: 特記事項

特筆事項なし

3-4-25 getEps

1: プロトタイプ宣言

IC_HMI_ON_OFF getEps(void)

2: API の説明

EPS 警告灯状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し

3-4-26 getSrsAirbag

1: プロトタイプ宣言

IC_HMI_ON_OFF getSrsAirbag(void)

2: API の説明

SRS エアバッグ警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し

3-4-27 getAbs

1: プロトタイプ宣言

IC_HMI_ON_OFF getAbs(void)

2: API の説明

ABS 警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し

3-4-28 getLowBattery

1: プロトタイプ宣言

IC_HMI_ON_OFF getLowBattery(void)

2:API の説明

チャージ警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し

3-4-29 getOilPress

1:プロトタイプ宣言

IC_HMI_ON_OFF getOilPress(void)

2:API の説明

オイルプレッシャー警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し

3-4-30 getEngine

1:プロトタイプ宣言

IC_HMI_ON_OFF getEngine(void)

2:APIの説明

エンジン警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項なし

3-4-31 getFuel

1:プロトタイプ宣言

IC_HMI_ON_OFF getFuel(void)

2:APIの説明

燃料残量警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-32 getImmobi

1:プロトタイプ宣言

IC_HMI_ON_OFF getImmobi(void)

2:API の説明

Immobilizer 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-33 getTMFail

1:プロトタイプ宣言

IC_HMI_ON_OFF getTMFail(void)

2:API の説明

TM Failure 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-34 getEspAct

1: プロトタイプ宣言

IC_HMI_ON_OFF getEspAct(void)

2: API の説明

ESP-ACT 警告灯状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-35 getEspOff

1: プロトタイプ宣言

IC_HMI_ON_OFF getEspOff(void)

2: API の説明

ESP-OFF 警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-36 getAdaptingLighting

1: プロトタイプ宣言

IC_HMI_ON_OFF getAdaptingLighting(void)

2: API の説明

Adapting Lighting 警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-37 getAutoStop

1: プロトタイプ宣言

IC_HMI_ON_OFF getAutoStop(void)

2:API の説明

Auto Stop 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-38 getAutoStopFail

1:プロトタイプ宣言

IC_HMI_ON_OFF getAutoStopFail(void)

2:API の説明

Auto Stop Fail 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-39 getParkingLights

1:プロトタイプ宣言

```
IC_HMI_ON_OFF getParkingLights(void)
```

2:API の説明

Parking Lights 警告灯状態を取得する。

3:返り値の型種別

```
IC_HMI_ON_OFF
```

4:返り値の詳細

```
IC_HMI_OFF    : 消灯
```

```
IC_HMI_ON     : 点灯
```

5:引数の詳細

```
None
```

6:特記事項

特筆事項無し。

3-4-40 getFrontFog

1:プロトタイプ宣言

```
IC_HMI_ON_OFF getFrontFog(void)
```

2:API の説明

FrontFog 警告灯状態を取得する。

3:返り値の型種別

```
IC_HMI_ON_OFF
```

4:返り値の詳細

```
IC_HMI_OFF    : 消灯
```

```
IC_HMI_ON     : 点灯
```

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-41 getExteriorLightFault

1:プロトタイプ宣言

IC_HMI_ON_OFF getExteriorLightFault(void)

2:APIの説明

ExteriorLightFault 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-42 getAccFail

1:プロトタイプ宣言

IC_HMI_ON_OFF getAccFail(void)

2:APIの説明

ACC-Failure 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-43 getLdwOff

1: プロトタイプ宣言

IC_HMI_ON_OFF getLdwOff(void)

2: API の説明

Lane Departure Warning OFF 警告灯状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-44 getHillDescent

1: プロトタイプ宣言

IC_HMI_ON_OFF getHillDescent(void)

2: API の説明

Hill-Descent 警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-45 getAutoHiBeamGreen

1: プロトタイプ宣言

IC_HMI_ON_OFF getAutoHiBeamGreen(void)

2: API の説明

AutoHiBeamGreen 警告灯状態を取得する。

3: 戻り値の型種別

IC_HMI_ON_OFF

4: 戻り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-46 getAutoHiBeamAmber

1: プロトタイプ宣言

IC_HMI_ON_OFF getAutoHiBeamAmber(void)

2:API の説明

AutoHiBeamAmber 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-47 getSportsMode

1:プロトタイプ宣言

IC_HMI_ON_OFF getSportsMode(void)

2:API の説明

Sports Mode 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-48 getLdwOperate

1:プロトタイプ宣言

IC_HMI_ON_OFF getLdwOperate(void)

2:APIの説明

Lane Departure Warning Operate 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-49 getGeneralWarn

1:プロトタイプ宣言

IC_HMI_ON_OFF getGeneralWarn(void)

2:APIの説明

General Warn 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-50 getDrivingPowerMode

1:プロトタイプ宣言

```
IC_HMI_ON_OFF getDrivingPowerMode(void)
```

2:APIの説明

Driver Power Mode 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4:返り値の詳細

IC_HMI_OFF : 消灯

IC_HMI_ON : 点灯

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-4-51 getHotTemp

1:プロトタイプ宣言

```
IC_HMI_ON_OFF getHotTemp(void)
```

2:APIの説明

Hot Temp 警告灯状態を取得する。

3:返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-4-52 getLowTemp

1: プロトタイプ宣言

IC_HMI_ON_OFF getLowTemp(void)

2: API の説明

Low Temp 警告灯状態を取得する。

3: 返り値の型種別

IC_HMI_ON_OFF

4: 返り値の詳細

IC_HMI_OFF : 消灯
IC_HMI_ON : 点灯

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-5 ShiftPosition

3-5-1 getGearAtVal

1: プロトタイプ宣言

IC_HMI_GEAR_AT_VAL getGearAtVal(void)

2:API の説明

ギア状態値を取得する。

3:返り値の型種別

IC_HMI_GEAR_AT_VAL

4:返り値の詳細

0 : IC_HMI_AT_OFF	ギアが非表示状態を表す。
1 : IC_HMI_AT_PARKING	ギアが Parking 状態を表す。
2 : IC_HMI_AT_REVERSE	ギアが Reverse 状態を表す。
3 : IC_HMI_AT_NEUTRAL	ギアが Neutral 状態を表す。
4 : IC_HMI_AT_DRIVE	ギアが Drive 状態を表す。
5 : IC_HMI_AT_SECOND	ギアが Second 状態を表す。
6 : IC_HMI_AT_LOW_GEAR	ギアが LowGear 状態を表す。
7 : IC_HMI_AT_FIRST	ギアが First 状態を表す。
8 : IC_HMI_AT_MANUAL	ギアが Manual 状態を表す。
9 : IC_HMI_AT_BRAKE	ギアが Brake 状態を表す。
10 : IC_HMI_AT_ALL_ON	故障全点灯状態を表す。
11 : IC_HMI_AT_ALL_OFF	故障全消灯状態を表す。
12 : IC_HMI_AT_UNUSED	ギアが未使用状態を表す。

5:引数の詳細

None

6:特記事項

特筆事項無し。

3-5-2 getGearMtVal

1:プロトタイプ宣言

IC_HMI_GEAR_MT_VAL getGearMtVal(void)

2:API の説明

ギア状態値を取得する。

3:返り値の型種別

IC_HMI_GEAR_MT_VAL

4: 返り値の詳細

0 : IC_HMI_MT_OFF	ギアが非表示状態を表す。
1 : IC_HMI_MT_FIRST	ギアが 1 速状態を表す。
2 : IC_HMI_MT_SECOND	ギアが 2 速状態を表す。
3 : IC_HMI_MT_THIRD	ギアが 3 速状態を表す。
4 : IC_HMI_MT_FOURTH	ギアが 4 速状態を表す。
5 : IC_HMI_MT_FIFTH	ギアが 5 速状態を表す。
6 : IC_HMI_MT_SIXTH	ギアが 6 速状態を表す。
7 : IC_HMI_MT_SEVENTH	ギアが 7 速状態を表す。
8 : IC_HMI_MT_EIGHTH	ギアが 8 速状態を表す。

5: 引数の詳細

None

6: 特記事項

特筆事項無し。

3-6 Speed

3-6-1 getSpAnalogVal

1: プロトタイプ宣言

```
unsigned long getSpAnalogVal(void)
```

2: API の説明

車速アナログ指示値(分解能 : 0.01 平滑化処理後の値)を取得する。

3: 返り値の型種別

unsigned long

4: 返り値の詳細

0x00000000	: Min Speed	(0.00)
0x00007530	: Max Speed	(300.00)
0x000007531~0xFFFFFFFF	: 未使用	(300.01~42949672.95)

5:引数の詳細

None

6:特記事項

単位は仕向依存で対応する

3-6-2 getSpAnaDigUnitVal

1:プロトタイプ宣言

IC_HMI_SP_UNIT_VAL getSpAnaDigUnitVal(void)

2:API の説明

車速単位を取得する。

3:返り値の型種別

IC_HMI_SP_UNIT_VAL

4:返り値の詳細

0 : IC_HMI_SP_KM_H

車速単位を km/h で表示する。

1 : IC_HMI_SP_MPH

車速単位を mph で表示する。

5:引数の詳細

None

6:特記事項

特筆事項無し

3-7 Tacho

3-7-1 getTaAnalogVal

1:プロトタイプ宣言

unsigned long getTaAnalogVal(void)

2:API の説明

Tacho アナログ指示値(分解能 : 1 平滑化処理後の値)を取得する。単位は rpm。

3: 返り値の型種別

unsigned long

4: 返り値の詳細

0x00000000	: Min rpm	(0)
0x00004E20	: Max rpm	(20000)
0x00004E21~0xFFFFFFFF	: 未使用	(20001~4294967295)

5: 引数の詳細

None

6: 特記事項

特筆事項無し

3-8 TripComputer

3-8-1 getTrcomTripAVal

1: プロトタイプ宣言

unsigned long getTrcomTripAVal(void)

2: API の説明

TripA 値(分解能 : 0.1)を取得する。

3: 返り値の型種別

unsigned long

4: 返り値の詳細

0x00000000	: TripA Min	(0.0)
0x0001869F	: TripA Max	(9999.9)
0x00018670 ~0xFFFFFFFFD	: 未使用	(10000.0~429496729.5)
0xFFFFFFFFE	: “—”表示	
0xFFFFFFFFF	: 非表示	

5: 引数の詳細

None

6:特記事項

表示桁数は機種依存にする

3-8-2 getTrcomTripBVal

1:プロトタイプ宣言

```
unsigned long getTrcomTripBVal(void)
```

2:API の説明

TripB 値(分解能 : 0.1)を取得する。

3:返り値の型種別

unsigned long

4:返り値の詳細

0x00000000	: TripB Min	(0.0)
0x0001869F	: TripB Max	(9999.9)
0x00018670 ~0xFFFFFFFFD	: 未使用	(10000.0~429496729.5)
0xFFFFFFFFE	: “-”表示	
0xFFFFFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示桁数は機種依存にする。

3-8-3 getTrcomOdoVal

1:プロトタイプ宣言

```
unsigned long getTrcomOdoVal(void)
```

2:API の説明

ODO 値(分解能 : 1)を取得する。

3:返り値の型種別

unsigned long

4: 返り値の詳細

0x00000000	: ODO Min	(0)
0x000F423F	: ODO Max	(999999)
0x000F4240 ~ 0xFFFFFFFFD	: 未使用	(1000000~4294967295)
0xFFFFFFFFE	: “—”表示	
0xFFFFFFFFF	: 非表示	

5: 引数の詳細

None

6: 特記事項

表示桁数は機種依存にする。

3-8-4 getTrcomUnitVal

1: プロトタイプ宣言

```
IC_HMI_TRCOM_UNIT_VAL getTrcomUnitVal(void)
```

2: API の説明

トリップコンピュータ単位を取得する。

3: 返り値の型種別

IC_HMI_TRCOM_UNIT_VAL

4: 返り値の詳細

0 : IC_HMI_TRCOM_KM	トリップコンピュータ単位を km で表示する。
1 : IC_HMI_TRCOM_MILE	トリップコンピュータ単位を mile で表示する。

5: 引数の詳細

TripA/B および ODO 値の単位取得に使用する。

3-8-5 getAvgSpeedAVal

1: プロトタイプ宣言

```
unsigned short getAvgSpeedAVal(void)
```

2: API の説明

TripAと連動した平均車速値(分解能：1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Average Speed A Min (0)
0x012C	: Average Speed A Max (300)
0x012D~0xFFFFD	: 未使用 (300~65533)
0xFFFFE	: “-”表示
0xFFFFF	: 非表示

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-6 getAvgSpeedBVal

1:プロトタイプ宣言

unsigned short getAvgSpeedBVal(void)

2:APIの説明

TripBと平均車速値(分解能：1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Average Speed B Min (0)
0x012C	: Average Speed B Max (300)
0x012D~0xFFFFD	: 未使用 (300~65533)
0xFFFFE	: “-”表示
0xFFFFF	: 非表示

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-7 getHourAVal

1:プロトタイプ宣言

unsigned short getHourAVal(void)

2:API の説明

TripAと連動した経過時間(時) (分解能 : 1) を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Hour A Min	(0)
0x03E7	: Hour A Max	(999)
0x03E8~0xFFFFD	: 未使用	(1000~65533)
0xFFFFE	: “—”表示	
0xFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-8 getHourBVal

1:プロトタイプ宣言

unsigned short getHourBVal(void)

2:API の説明

TripBと連動した経過時間(時) (分解能 : 1)を取得する。

3:返り値の型種別

unsigned short

4: 返り値の詳細

0x0000	: Hour B Min	(0)
0x03E7	: Hour B Max	(999)
0x03E8~0xFFFFD	: 未使用	(1000~65533)
0xFFFFE	: “—”表示	
0xFFFFF	: 非表示	

5: 引数の詳細

None

6: 特記事項

表示範囲は機種依存にする。

3-8-9 getMinuteAVal

1: プロトタイプ宣言

```
unsigned char getMinuteAVal(void)
```

2: API の説明

TripA と連動した経過時間(分) (分解能 : 1)を取得する。

3: 返り値の型種別

unsigned char

4: 返り値の詳細

0x00	: Minute A Min	(0)
0x3B	: Minute A Max	(59)
0x3C ~ 0xFD	: 未使用	(60~253)
0xFE	: “—”表示	
0xFF	: 非表示	

5: 引数の詳細

None

6: 特記事項

表示範囲は機種依存にする。

3-8-10 getMinuteBVal

1:プロトタイプ宣言

```
unsigned char getMinuteBVal(void)
```

2:API の説明

TripB と連動した経過時間(分) (分解能 : 1)を取得する。

3:返り値の型種別

```
unsigned char
```

4:返り値の詳細

0x00	:	Minute B Min (0)
0x3B	:	Minute B Max (59)
0x3C ~ 0xFD	:	未使用 (60~253)
0xFE	:	“—”表示
0xFF	:	非表示

5:引数の詳細

```
None
```

6:特記事項

表示範囲は機種依存にする。

3-8-11 getSecondAVal

1:プロトタイプ宣言

```
unsigned char getSecondAVal(void)
```

2:API の説明

TripA と連動した経過時間(秒) (分解能 : 1)を取得する。

3:返り値の型種別

```
unsigned char
```

4:返り値の詳細

0x00 : Second A Min (0)
0x3B : Second A Max (59)
0x3C ~ 0xFD : 未使用 (60~253)
0xFE : “—”表示
0xFF : 非表示

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-12 getSecondBVal

1:プロトタイプ宣言

```
unsigned char getSecondBVal(void)
```

2:APIの説明

TripBと連動した経過時間(秒) (分解能：1)を取得する。

3:返り値の型種別

unsigned char

4:返り値の詳細

0x00 : Second B Min (0)
0x3B : Second B Max (59)
0x3C ~ 0xFD : 未使用 (60~253)
0xFE : “—”表示
0xFF : 非表示

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-13 getOTempVal

1:プロトタイプ宣言

signed short getOTempVal(void)

2:API の説明

外気温表示値(分解能 : 1)を取得する。

3:返り値の型種別

signed short

4:返り値の詳細

0x8001~0xFFD7	: 未使用	(-32767 ~ -41)
0xFFD8	: Outside Temp Min	(-40)
0x0083	: Outside Temp Max	(131)
0x0084 ~0x7FFE	: 未使用	(132~32765)
0x7FFE	: “-”表示	(32766)
0x7FFF	: 非表示	(32767)

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-14 getOTempUnitVal

1:プロトタイプ宣言

IC_HMI_OTEMP_UNIT_VAL getOTempUnitVal(void)

2:API の説明

外気温表示値に対する単位を取得する。

3:返り値の型種別

IC_HMI_OTEMP_UNIT_VAL

4:返り値の詳細

0 : IC_HMI_OTEMP_UNIT_C 外気温単位を摂氏状態に表示する。

1 : IC_HMI_OTEMP_UNIT_K

外気温単位を華氏状態に表示する。

5:引数の詳細

None

6:特記事項

特になし

3-8-15 getCruRangeVal

1:プロトタイプ宣言

unsigned short getCruRangeVal(void)

2:API の説明

航続距離可能表示値(分解能 : 1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Cruise Range Min	(0)
0x064A	: Cruise Range Max	(1610)
0x064B ~ 0xFFFFD	: 未使用	(1611~65533)
0xFFFFE	: “—”表示	
0xFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-16 getAvgFuelAVal

1:プロトタイプ宣言

unsigned short getAvgFuelAVal(void)

2:API の説明

TripAと連動した平均燃費値(分解能 : 0.1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Average Fuel TripA Min	(0.0)
0x07CF	: Average Fuel TripA Max	(199.9)
0x07D0 ~ 0xFFFFD	: 未使用	(200.0~6553.3)
0xFFFFE	: “-”表示	
0xFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-17 getAvgFuelBVal

1:プロトタイプ宣言

unsigned short getAvgFuelBVal(void)

2:API の説明

TripBと連動した平均燃費値(分解能 : 0.1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Average Fuel TripB Min	(0.0)
0x07CF	: Average Fuel TripB Max	(199.9)
0x07D0 ~ 0xFFFFD	: 未使用	(200.0~6553.3)
0xFFFFE	: “-”表示	
0xFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-18 getInsFuelAVal

1:プロトタイプ宣言

unsigned short getInsFuelAVal(void)

2:API の説明

TripAと連動した瞬間燃費値(分解能 : 0.1)を取得する。

3:返り値の型種別

unsigned short

4:返り値の詳細

0x0000	: Instant Fuel TripA Min	(0.0)
0x07CF	: Instant Fuel TripA Max	(199.9)
0x07D0 ~ 0xFFFFD	: 未使用	(200.0~6553.3)
0xFFFFE	: “—”表示	
0xFFFFF	: 非表示	

5:引数の詳細

None

6:特記事項

表示範囲は機種依存にする。

3-8-19 getInsFuelBVal

1:プロトタイプ宣言

unsigned short getInsFuelBVal(void)

2:API の説明

TripBと連動した瞬間燃費値(分解能 : 0.1)を取得する。

3:返り値の型種別

unsigned short

4: 返り値の詳細

0x0000	: Instant Fuel TripB Min	(0.0)
0x07CF	: Instant Fuel TripB Max	(199.9)
0x07D0 ~ 0xFFFFD	: 未使用	(200.0~6553.3)
0xFFFFE	: “-”表示	
0xFFFFF	: 非表示	

5: 引数の詳細

None

6: 特記事項

表示範囲は機種依存にする。

3-8-20 getFuelEconomyUnitVal

1: プロトタイプ宣言

```
IC_HMI_FUEL_ECONOMY_UNIT_VAL getFuelEconomyUnitVal(void)
```

2: API の説明

燃費表示単位を取得する。

3: 返り値の型種別

```
IC_HMI_FUEL_ECONOMY_UNIT_VAL
```

4: 返り値の詳細

0 : IC_HMI_FUEL_KM_L	燃費表示単位を km/L で表示する。
1 : IC_HMI_FUEL_MPG_US	燃費表示単位を MPG(U.S gallon)で表示する。
2 : IC_HMI_FUEL_MPG_IG	燃費表示単位を MPG(imperial gallon)で表示する。
3 : IC_HMI_FUEL_L_100KM	燃費表示単位を L/100km で表示する。
4 : IC_HMI_FUEL_MILE_KWH	燃費表示単位を mile/kwh で表示する。
5 : IC_HMI_FUEL_KM_KWH	燃費表示単位を km/kwh で表示する。
6 : IC_HMI_FUEL_MILE_KG	燃費表示単位を mile/kg で表示する。
7 : IC_HMI_FUEL_KM_KG	燃費表示単位を km/kg で表示する。
8 : IC_HMI_FUEL_KWH_100KM	燃費表示単位を kwh/100km で表示する。

5:引数の詳細

平均燃費と瞬間燃費の単位取得に使用する。

6:特記事項

特記事項無し

3-9 Register/Notify

3-9-1 registerIcHmi

1:プロトタイプ宣言

```
bool registerIcHmi(unsigned long long arg_1,  
                  IC_HMI_FUNC_NOTIFY_IC_HMI addr)
```

2:API の説明

Event 方式通信において Cluster UI 側で取得したいシグナルを決定するために使用される。通信方式については 2 章参照。

シグナル決定方法は第 1 引数にマスクビットを指定し、立ったフラグに該当するシグナルを取得する。第 2 引数にはアプリケーション側で作成したコールバック関数を指定する。

3:返り値の型種別

bool

4:返り値の詳細

true : 登録成功
false : 登録失敗

5:引数の詳細

unsigned long long arg_1 : 取得するシグナルに対するビットフラグ。
各ビットに対応するシグナルは [3-2-1](#) 参照。
未使用フラグを設定した場合は無視する。

IC_HMI_FUNC_NOTIFY_IC_HMI addr :
定義されたコールバック関数の先頭アドレス。
コールバック関数の詳細は [3-3-1](#) 参照。

addr に null が設定された場合、登録失敗とする。

6:特記事項

TurnR と TurnL のシグナルを登録する場合の実装例を以下に示す。

```
registerIchHmi((IC_HMI_TT_TURN_R||IC_HMI_TT_TURN_L ),  
              IC_HMI_FUNC_NOTIFY_IC_HMI addr)
```

当該シグナル変化検出時に addr で指定したコールバック関数が呼ばれる。

3-9-2 notifyIchHmi

1:プロトタイプ宣言

```
void notifyIchHmi(unsigned long long arg_1, IC_HMI_ON_OFF arg_2)
```

2:API の説明

登録されたシグナルが変更されたことを IC-Service から Cluster UI に通知する。変化されたタイミングで本関数は呼ばれる。第 1 引数にはどのシグナルかを示すビットフラグ、第 2 引数には変化後の値が通知される。

3:返り値の型種別

void

4:返り値の詳細

None

5:引数の詳細

unsigned long long arg_1 : 取得するシグナルに対応したビットフラグ。
各シグナルのビットフラグは [3-2-1](#) 参照

IC_HMI_ON_OFF arg_2 : 変化検出後のシグナルの状態
各状態については [3-1-1](#) 参照

6:特記事項

当該シグナル変化検出時に 3-9-1 で指定したコールバック関数が呼ばれる

```
void notifyIchHmi(unsigned long long arg_1, IC_HMI_ON_OFF arg_2)
```



```

{
    if(arg_1 == TRUN_R){
        //TurnR 変化検出時の処理
        if(arg_2 == IC_HMI_ON){
            //ON 時の処理
        }else{
            //OFF 時の処理
        }
    }
}

```

3-10 Initalize

3-10-1 clusterInit

1:プロトタイプ宣言

```
bool clusterInit(void)
```

2:API の説明

IC-Service を使用するための Cluster API を初期化する。

3:返り値の型種別

bool

4:返り値の詳細

true : 初期化処理成功
false : 初期化処理失敗

5:引数の詳細

None

6:特記事項

本 API を呼ばずに他の API を呼び出した場合の動作は未定義である。
clusterTerm()を呼ばずに本 API を 2 連続で呼び出すとエラーとなる。
IC-Service 側が起動していない場合、エラーとなる。

3-10-2 clusterTerm

1:プロトタイプ宣言

```
bool clusterTerm(void)
```

2:API の説明

Cluster API を終了する。

Cluster API が使用していたリソースを解放する。

3:返り値の型種別

```
bool
```

4:返り値の詳細

true : 終了処理成功

false : 終了処理失敗

5:引数の詳細

```
None
```

6:特記事項

clusterInit()を呼ばずに本 API を呼び出すとエラーとなる。

本 API を 2 連続で呼び出すとエラーとなる。

4.改訂履歴

Rev0.1	2020/05/14	–	初版
Rev0.2	2020/08/06	全体	IC-EG のレビュー指摘の対応を実施した
Rev0.3	2020/08/07	3-4-24 getHoodDoor	ボンネットの開閉状態を取得為の API が漏れていた為、追加
Rev0.4	2020/10/26	3-2-1-4 IC_HMI_TT_SEATBELT 2.API の説明	“ブレーキ警告灯状態”となっていたので、“シートベルト警告灯”に誤記修正
		3-2-1-5 ~ 3-2-1-33 1. API の名前	API の名前に Talltale を示す “_TT” の漏れが合った為、追記
		3-2-1-6 IC_HMI_TT_DOOR 1. API の名前	“半ドア警告灯状態のビットフラグ”と API の説明が記載されていたので、“IC_HMI_TT_DOOR”に誤記修正
		3-2-1-6 IC_HMI_TT_DOOR 2.API の説明	“左方向指示器”となっていたので、“半ドア警告灯状態のビットフラグ”と誤記修正
		3-2-1-12 IC_HMI_TT_ENGINE 2:API の説明	“燃料残量警告灯状態”となっていたので、“エンジン警告灯状態”と誤記修正
		3-2-1-20 IC_HMI_TT_AUTO_STOP_FAIL 1: API の名前	“Auto Stop Fail 警告灯状態のビットフラグ”と API の説明が記載されていたので、“IC_HMI_TT_AUTO_STOP_FAIL”と誤記修正
		3-2-1-20 IC_HMI_TT_AUTO_STOP_FAIL 2:API の説明	“左方向指示器”となっていたので、“Auto Stop Fail 警告灯状態のビットフラグ”と誤記修正
		3-4-30 getEngine 2:API の説明	“燃料残量警告灯状態”となっていたので、“エンジン警告灯状態”と誤記修正
	2020/11/30	3-4-4 getSeatbelt 6.特記事項	“いずれか 1 つが ON ならば戻り値 ON になる”となっており、シートベルトが装着されると警告灯が表示されてしまうので、“いずれか 1 つが OFF ならば戻り値 ON になる”と修正
		3-8-5 getAvgSpeedAVal 1.プロトタイプ宣言	“unsigned char”となっており、戻り値と一致しない為、“unsigned short”と誤記修正
		3-8-6 getAvgSpeedBVal	“unsigned char”となっており、戻り値と一

		1.プロトタイプ宣言	致しない為、“unsigned short”と誤記修正
2020/12/02	3-4-40 getFrontFog 3:返り値の型種別		4.返り値の詳細の内容が記述されていたので、誤記修正
	3-4-40 getFrontFog 4:返り値の詳細		3.返り値の型種別の内容が記述されていたので、誤記修正
	3-8-9 ~ 3-8-12 4:返り値の詳細		“0xFE”が“0xFÉ”となっていたので誤記修正
	全体		フォントを統一した
2020/12/23	3-4-7 getFrontLeftSeatbelt 1:プロトタイプ宣言		“getFrontRightSeatbelt”となっていたので誤記修正
	3-4-50 getDrivingPowerMode 1:プロトタイプ宣言		“getDriverPowerMode”となっていたので誤記修正
	3-8-5 getAvgSpeedAVal 1:プロトタイプ宣言		“getAveSpeedAVal”となっていたので誤記修正
	3-8-6 getAvgSpeedBVal 1:プロトタイプ宣言		“getAveSpeedBVal”となっていたので誤記修正
	3-8-18 getInsFuelAVal 1:プロトタイプ宣言		“getInsAVal”となっていたので誤記修正
	3-8-19 getInsFuelBVal 1:プロトタイプ宣言		“getInsBVal”となっていたので誤記修正
2021/01/20	3-8-19 getInsFuelBVal 4:返り値の詳細		“Instant Fuel TripA Min”、“Instant Fuel TripA Max”となっていたので誤記修正
	3-10 Initialize		実装上必要な API の記載が漏れていた為、追記