Ethernet to CAN Converter Health Status

Document version: 3
Date: April 5, 2021
Written by: Oleksandr Bogush, Axiomatic Technologies Corporation.

Contents

Introduction 2
Heath Status Definition 2
Operational Statuses 2
Controller Configuration Parameters Loading 3
CAN Configuration Parameters Loading 4
Ethernet Configuration Parameters Loading 4
WiFi Configuration Parameters Loading 4
CAN Driver Initialization 4
Ethernet Driver Initialization 5
WiFi Driver Initialization 5
Flash Memory Driver Initialization 5
Logging Memory Driver Initialization 5
LwIp Stack Initialization 5
CAN Operation 6
Ethernet Operation 6
WiFi Operation 6
CAN Logging Operation 6
Power Supply Voltage 7
Output CAN Voltage 7
Internal BAT Voltage 7
uC Temperature 8
Health Data Field 8
References 9

Introduction

The following document describes the *Health Status* run-time parameter used in Axiomatic Ethernet to CAN and WiFi to CAN Converters.

The document contains terminology and acronyms from CAN protocol. Their meaning is explained in the appropriate CAN documentation.

Heath Status Definition

The device *Health Status* is an aggregated system run-time parameter calculated on the base of operational statuses of the major device hardware and software components. It presents the overall operational status of the Ethernet to CAN Converter.

The *Health Status*, together with individual operational statuses, can take one of the four states:

HS ϵ { *Undefined*, *Normal*, *Warning*, *Error* }

The *Health Status* is calculated combining all individual operational statuses the following way:

Table 1. Heath Status aggregation rules

State	Condition
Error	*Error* is reported when at least one operational status is in the *Error* state
Warning	*Warning* is reported when at least one operational status is in the *Warning* state and there are no operational statuses in the *Error* state
Undefined	*Undefined* is reported when at least one operational status is in the *Undefined* state and there are no operational statuses in the *Error* or *Warning* state
Normal	*Normal* is reported when all operational statuses are in the *Normal* state

The health status states are encoded in a two-bit field the following way.

Table 2. Health and Operational Status States

Value	Bit 1	Bit 0	Health and Operational Status States
0	0	0	Undefined
1	0	1	Normal
2	1	0	Warning
3	1	1	Error

Operational Statuses

The following operational statuses are currently defined and used for the *Health Status* calculation.

Table 3. Operational Statuses

Operational Status	Converter Type	Operational Status Type	Description
Controller Configuration Parameters Loading	All	Software	Result of the loading operation for the common configuration parameters

Operational Status	Converter Type	Operational Status Type	Description
CAN Configuration Parameters Loading	All		Result of the loading operation for CAN configuration parameters
Ethernet Configuration Parameters Loading	0		Result of the loading operation for Ethernet configuration parameters
WiFi Configuration Parameter Loading	1,2		Result of the loading operation for WiFi configuration parameters
CAN Driver Initialization	All		Result of the CAN driver initialization
Ethernet Driver Initialization	0		Result of the Ethernet driver initialization
WiFi Driver Initialization	1,2		Result of the WiFi driver initialization
Flash Memory Driver Initialization	All		Result of the flash memory driver initialization
Logging Memory Driver Initialization	2		Result of the logging memory driver initialization
LwIp Stack Initialization	All		Result of the LwIp stack initialization
CAN Operation	All		Run-time CAN status
Ethernet Operation	0		Run-time Ethernet status
WiFi Operation	1,2		Run-time WiFi status
CAN Logging Operation	2		Run-time CAN logging status
Power Supply Voltage	All	Hardware	Power Supply Voltage status
Output CAN Voltage	0		Output CAN Voltage status
Internal BAT Voltage	All		Internal microcontroller voltage status
uC Temperature	All		Microcontroller temperature status

The Converter Types are defined in [1] the following way:

Table 4. Converter Types

Converter Type	Name
0[1]	Ethernet to CAN converter with CAN Voltage Output
1	WiFi to CAN converter
2	WiFi to CAN converter with CAN datalogging capability

Each operational status is described below in an individual sub-section.

Controller Configuration Parameters Loading

The *Controller Configuration Parameters Loading* is a software operational status, which defines the result of the loading operation for the common configuration parameters of the Ethernet to CAN Converter.

There are only two states defined for this operational status, see Table 5:

Table 5. Controller Configuration Parameters Loading Status

Controller Configuration Parameters Loading Status	Description
Error	Common configuration parameters of the device have not been loaded during initialization
Normal	Normal loading

CAN Configuration Parameters Loading

This software operational status defines the result of the loading operation for CAN configuration parameters.

Table 6. CAN Configuration Parameters Loading Status

CAN Configuration Parameters Loading Status	Description
Error	CAN configuration parameters of the device have not been loaded during initialization
Normal	Normal loading

Ethernet Configuration Parameters Loading

This software operational status defines the result of the loading operation for Ethernet configuration parameters. It is defined only for Ethernet to CAN converters.

Table 7. Ethernet Configuration Parameters Loading Status

Ethernet Configuration Parameters Loading Status	Description
Error	Ethernet configuration parameters of the device have not been loaded during initialization
Normal	Normal loading

WiFi Configuration Parameters Loading

This software operational status defines the result of the loading operation for WiFi configuration parameters. It is defined only for WiFi to CAN converters.

Table 8. WiFi Configuration Parameters Loading Status

WiFi Configuration Parameters Loading Status	Description
Error	WiFi configuration parameters of the device have not been loaded during initialization
Normal	Normal loading

CAN Driver Initialization

The *CAN Driver Initialization* is a software operational status that defines the result of the CAN driver initialization.

Table 9. CAN Driver Initialization Status

CAN Driver Initialization Status	Description
Error	CAN driver has not been initialized correctly
Normal	Normal initialization

Ethernet Driver Initialization

The *Ethernet Driver Initialization* is a software operational status that defines the result of the Ethernet driver initialization. It is defined only for Ethernet to CAN converters.

Table 10. Ethernet Driver Initialization Status

Ethernet Driver Initialization Status	Description
Error	Ethernet driver has not been initialized correctly
Normal	Normal initialization

WiFi Driver Initialization

The *WiFi Driver Initialization* is a software operational status that defines the result of the WiFi driver initialization. It is defined only for WiFi to CAN converters.

Table 11. WiFi Driver Initialization Status

WiFi Driver Initialization Status	Description
Error	WiFi driver has not been initialized correctly
Normal	Normal initialization

Flash Memory Driver Initialization

The *Flash Memory Driver Initialization* is a software operational status that defines the result of the flash memory driver initialization.

Table 12. Flash Memory Driver Initialization Status

Flash Memory Driver Initialization Status	Description
Error	The flash memory driver initialization has failed
Normal	Normal initialization

Logging Memory Driver Initialization

The *Logging Memory Driver Initialization* is a software operational status that defines the result of the logging memory driver initialization. It is defined only for data-logging converters.

Table 13. Logging Memory Driver Initialization Status

Logging Memory Driver Initialization Status	Description
Error	The logging memory driver initialization has failed
Normal	Normal initialization

LwIp Stack Initialization

The *LwIp Stack Initialization* is a software operational status that defines the result of the LwIp Stack initialization operation.

Table 14. LwIp Stack Initialization Status

LwIp Stack Initialization Status	Description
Error	LwIp Stack has not been initialized correctly
Normal	Normal Stack initialization

CAN Operation

The *CAN Operation* is a software operational status that defines the runtime status of the CAN device peripheral controller.

Table 15. CAN Operation Status

CAN Operation Status	Description
Error	CAN controller is in the *Bus-Off* or *Error Passive* state
Warning	CAN controller is in the *Warning* state
Normal	CAN is operating normally
Undefined	CAN status is undefined due to an error in the CAN driver initialization

Ethernet Operation

The *Ethernet Operation* is a software operational status that defines a runtime status of the Ethernet port. It is defined only for Ethernet to CAN converters.

Table 16. Ethernet Operation Status

Ethernet Operation Status	Description
Error	Ethernet port is not connected to the network
Warning	Error during parsing of the protocol messages. Can indicate software incompatibility between the converter and the paired device.
Normal	Ethernet is operating normally
Undefined	Ethernet status is undefined due to an error in the Ethernet driver initialization

WiFi Operation

The *WiFi Operation* is a software operational status that defines a runtime status of the WiFi port. It is defined only for WiFi to CAN converters.

Table 17. WiFi Operation Status

WiFi Operation Status	Description
Error	WiFi port is not connected to the access point or other stations.
Warning	Error during parsing of the protocol messages. Can indicate software incompatibility between the converter and the paired device.
Normal	WiFi port is operating normally
Undefined	WiFi status is undefined due to an error during WiFi driver initialization

CAN Logging Operation

The *CAN Logging Operation* is a software operational status that defines a runtime status of the CAN logging operation. It is defined only for data-logging converters.

Table 18. CAN Logging Operation Status

CAN Logging Operation Status	Description
Error	Error during logging CAN data. Logging memory can be full.
Warning	More than 90% of the logging memory has been used.
Normal	CAN data logging is operating normally.

Undefined	CAN Logging status is undefined due to an error in the CAN driver initialization

Power Supply Voltage

The *Power Supply Voltage* is a hardware operational status that indicates whether the power supply voltage V_{PS} is within the operational range. The *Error* state is not defined for this status. V_IN_MIN and V_IN_MAX are hardware dependent.

Table 19. Power Supply Voltage Status

Power Supply Voltage Status	Description
Warning	$V_{PS} < V\_IN\_MIN$ or $V_{PS} > V\_IN\_MAX$
Normal	Power supply voltage is within the operational range: $V_{PS} \geq V\_IN\_MIN$ and $V_{PS} \leq V\_IN\_MAX$
Undefined	Power supply voltage has not been measured yet. This state should not be normally observed other than at startup

Output CAN Voltage

The *Output CAN Voltage* is a hardware operational status indicating whether the CAN output voltage V_{CAN}^{OUT} is within the operational range. It is supported by hardware in Ethernet to CAN converters. The status rules depend on the state of the CAN output voltage switch. Constants are hardware dependent.

Table 20. Output CAN Voltage Status

Output CAN Voltage Status	Switch State	Description
Error	ON	OUTPUT_FAULT signal is asserted
	OFF	$V_{CAN}^{OUT} > V\_OUT\_SWITCH\_OFF\_MAX$
Warning	ON	OUTPUT_FAULT signal is deasserted and $V_{CAN}^{OUT} < V\_OUT\_MIN$ or $V_{CAN}^{OUT} > V\_OUT\_MAX$
	OFF	Undefined. Not generated
Normal	ON	OUTPUT_FAULT signal is deasserted and CAN output voltage is within the operational range: $V_{CAN}^{OUT} \geq V\_OUT\_MIN$ and $V_{CAN}^{OUT} \leq V\_OUT\_MAX$
	OFF	$V_{CAN}^{OUT} \leq V\_OUT\_SWITCH\_OFF\_MAX$
Undefined	ON	OUTPUT_FAULT signal is deasserted and CAN output voltage has not been measured yet. This state should not be normally observed other than at startup
	OFF	CAN output voltage has not been measured yet. This state should not be normally observed other than at startup

Internal BAT Voltage

The *Internal BAT Voltage* is a hardware operational status that indicates whether the internal BAT voltage V_{BAT} is within the operational range. The V_{BAT} voltage is measured on the +3.3V rail. The *Error* state is not defined for this status. Constants are hardware dependent.

Table 21. Internal BAT Voltage Status

Internal BAT Voltage Status	Description
Warning	$V_{BAT} < V\_BAT\_MIN$ or $V_{BAT} > V\_BAT\_MAX$

Internal BAT Voltage Status	Description
Normal	Internal BAT voltage is within the operational range: $V_{BAT} \geq V\_BAT\_MIN$ and $V_{BAT} \leq V\_BAT\_MAX$
Undefined	Internal BAT voltage has not been measured yet. This state should not be normally observed other than at startup

uC Temperature

The *uC Temperature* is a hardware operational status that indicates whether the internal microcontroller temperature T_{uC} is within the operational range. The *Error* state is not defined for this status. Constants are hardware dependent.

Table 22. uC Temperature Status

uC Temperature Status	Description
Warning	$T_{uC} < TEMP\_MIN$ or $T_{uC} > TEMP\_MAX$
Normal	uC Temperature is within the operational range: $T_{uC} \geq TEMP\_MIN$ and $T_{uC} \leq TEMP\_MAX$
Undefined	uC Temperature has not been measured yet. This state should not be normally observed other than at startup

Health Data Field

The complete health status information is presented in the 4-byte *Health Data* field, which is a part of the *Status Response and Heartbeat* messages [1]. The *Health Data* field has the following format:

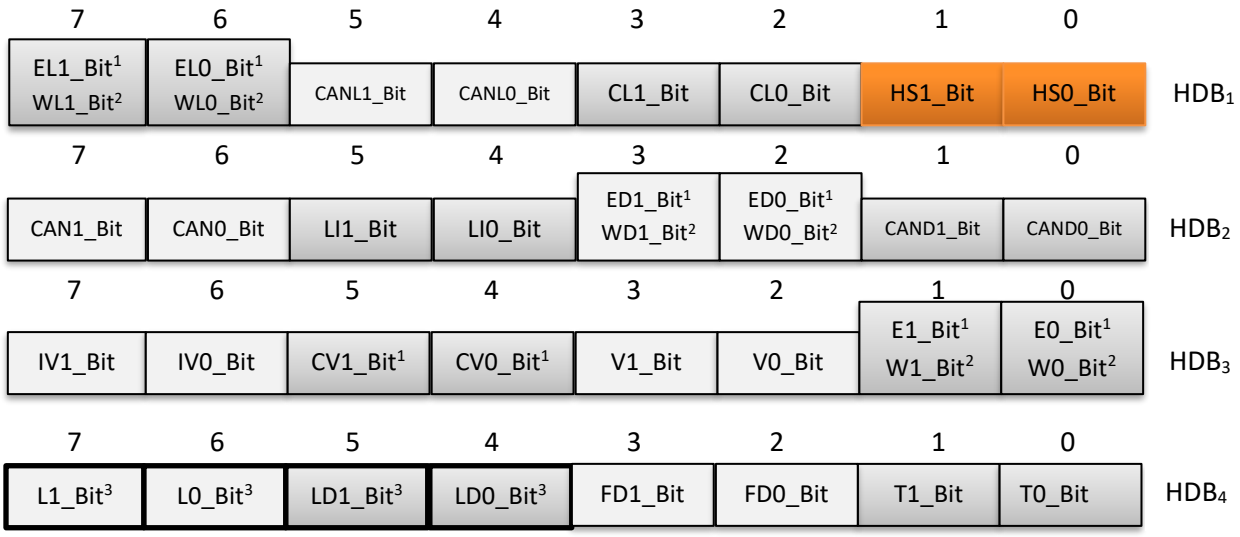

Figure 1. Health Data Field Format.

Bit 0:1 in HDB_1 = HS_Bit[1:0] : *Health Status*
Bit 2:3 in HDB_1 = CL_Bit[1:0] : *Controller Configuration Parameters Loading Status*
Bit 4:5 in HDB_1 = CANL_Bit[1:0] : *CAN Configuration Parameters Loading Status*
Bit 6:7 in HDB_1 = EL_Bit[1:0] : *Ethernet Configuration Parameters Loading Status*[1]
Bit 6:7 in HDB_1 = WL_Bit[1:0] : *WiFi Configuration Parameters Loading Status*[2]

Bit 0:1 in HDB_2 = CAND_Bit[1:0] : *CAN Driver Initialization Status*
Bit 2:3 in HDB_2 = ED_Bit[1:0] : *Ethernet Driver Initialization Status*[1]
Bit 2:3 in HDB_2 = WD_Bit[1:0] : *WiFi Driver Initialization Status*[2]

Bit 4:5 in HDB_2 = LI_Bit[1:0] : *LwIp Stack Initialization Status*
Bit 6:7 in HDB_2 = CAN_Bit[1:0] : *CAN Operation Status*

Bit 0:1 in HDB_3 = E_Bit[1:0] : *Ethernet Operation Status*[1]
Bit 0:1 in HDB_3 = W_Bit[1:0] : *WiFi Operation Status*[2]
Bit 2:3 in HDB_3 = V_Bit[1:0] : *Power Supply Voltage Status*
Bit 4:5 in HDB_3 = CV_Bit[1:0] : *Output CAN Voltage Status*[1]
Bit 6:7 in HDB_3 = IV_Bit[1:0] : *Internal BAT Voltage Status*

Bit 0:1 in HDB_4 = T_Bit[1:0] : *uC Temperature Status*
Bit 2:3 in HDB_4 = FD_Bit[1:0] : *Flash Memory Driver Initialization Status*
Bit 4:5 in HDB_4 = LD_Bit[1:0] : *Logging Memory Driver Initialization Status*[3]
Bit 6:7 in HDB_4 = L_Bit[1:0] : *Logging Operation Status*[3]

[1]Only for Ethernet to CAN converters. [2]Only for WiFi to CAN converters. [3]Only for WiFi to CAN data-logging converters.

References

[1] O. Bogush, "Ethernet to CAN Converter Communication Protocol. Document version: 4," Axiomatic Technologies Corporation, April 5, 2021.

Document Version History

Document Version	Date	Author	Changes
3	April 5, 2021	Olek Bogush	Added health status for Axiomatic WiFi-CAN converters. Added health status format (this information was transferred from “Ethernet to CAN Converter Communication Protocol”). Added *Health Data Field* section. Removed *Constants* section. Added *Converter Type* numbers in Operational Statuses description table. Updated *References* section.
2	June 27, 2016	Olek Bogush	Added the Flash Memory Driver Initialization status. Corrected Table 14. LwIp Stack Initialization Status.
1A	February 5, 2016	Olek Bogush	Made the document generic. Removed references to the project 14010, except for the hardware constants.

			Replaced the *Ethernet to CAN Gateway* term with the term: *Ethernet to CAN Converter*. Corrected Table 23. Heath Status aggregation rules.
1	October 28, 2015	Olek Bogush	Initial version.