Project „Safe Temperature Monitoring “

Software system und module specification

Modification history:

Version	Date	Modification	Creator	Auditor
1.0	2020-05-15	Creation systemspecification und module monitorTemp();	SaS	DS
1.1	2020-05-26	specification of checkTemp(); calcF2C(); and calcC2F()	SaS	DS
1.2	2020-05-27	specification of displayTemp();	SaS	DS
1.3	2021-04-06	Formal corrections	SaS	DS

Content

1. Software system specification

1.1.1 Summary of the design tools [SW-SMS1]

Microsoft Word been chosen as the SW module specification designing tool.

1.1.2 Description of the function

The function to be developed is "Safe Temperature Monitoring system". The objective of this safe function is to monitor the temperature data from the 1oo2 system and checks whether the current temperature exceeds the user specified limit or not.

1.1.3 Definition of the temperature limits for the safety function [SW-SMS2]

Absolute maximum possible temperature	1000°C or 1832°F
Absolute minimum possible temperature	-273.15°C or -459°F

Functional limit: Minimum temperature must not be higher than the maximum limit

1.1.4 Dividing the function into modules [SW-SMS3]

Overview of all modules	
No.	**module name**
1	monitorTemp()
2	checkTemp()
3	calcC2F()
4	calcF2C()
5	displayTemp()

1.1.5 Additional program-modules and their application [SW-SMS10]

It is assumed that the integrity of the input temperature sensor values has already been verified the checksum concept implemented in the main function.

1.1.6 Representation of the relationship between the software modules [SW-SMS4]

monitorTemp() function is the main safe function which has 4 sub functions namely "checkTemp(), calcC2F(), calcF2C() and displayTemp()".

1.1.7 Libraries [SW-SMS9]

This safe function shall be implemented as a hardware independent library (with exception to displayTemp() function which is a hardware specific function) in monitorTemp.c and monitorTemp.h files.

1.1.8 Coding guidelines [SW-SMS5]

MISRA C:2012 coding guidelines shall be used.

1.0 Software module specification [SW-SMS6]

2.1 Module „monitorTemp () “ [SW-SMS6.1]

2.1.1 Module description

This module acts as an interface between users and the underlying safe function implementation. The objective of this function is to get the temperature data of two sensors (1oo2 system) with the temperature data format from the user.

Then this module shall utilize other sub modules to check whether the current temperature exceeds the user defined limits and outputs the system state to the user

2.1.2 Function parameters

- Are parameters passed to the software module? - Yes

No.	name	Data type	Description	Plausibility check/ fault tolerance
1	TempFormat	char	Specifies the temperature format of the user data	Plausibility check
2	min_Temp	float	Minimum temperature limit specified by user	Plausibility check
3	max_Temp	float	Maximum temperature limit specified by user	Plausibility check
4	max_DisCr	float	Allowed discrepancies between two temperature sensor values	Fault tolerance
5	TempS1	float	Temperature data from Sensor1	Plausibility check & Fault tolerance
6	TempS2	float	Temperature data from Sensor2	Plausibility check & Fault tolerance

Fault tolerance:	max_DisCr, TempS1 & TempS2
After plausibility checks are done, these parameters shall be used to check the system state.	

Plausibility check:	TempFormat	true	false
Check the character passed to this parameter equals ‘C’ or ‘F’ or ‘c’ or ‘f’		Proceed with temperature value plausibility checks	Update error code as ‘5 – function error’

Plausibility check:	minTemp	True	false
Check the minTemp is below physically possible absolute minimum temperature value		Proceed with maximum temperature value plausibility checks	Update error code as ‘5 – function error’

Plausibility check:	maxTemp	True	false
Check the maxTemp is below physically possible absolute minimum temperature value		Proceed with temperature discrepancy checks	Update error code as ‘5 – function error’

Plausibility check:	TempS1	True	false
Check the temperature value of from Sensor1 is within the physically possible absolute temperature limits		Proceed with temperature discrepancy checks	Update error code as ‘5 – function error’

Plausibility check:	TempS2	True	false
Check the temperature value of from Sensor1 is within the physically possible absolute temperature limits		Proceed with temperature discrepancy checks	Update error code as ‘5 – function error’

2.1.3 Return value

- Is a parameter returned by the function? - Yes

No.	Name	Data type	Description	Plausibility Check
1	TempOK	uint8	Returns the system state after checking the input sensor data	Yes

Plausibility check:	TempOK	True	False
Check whether the return value has ‘7 – Temperature range OK’, ‘5 – Function error’, or ‘3 – Temperature range limits exceeded - alarm’		Action if true	Action if false

2.1.4 Module variables

- Enter the (local) module variables including the range check for the values and check of physical quantities.

No.	Name	Data type	Description	Plausibility check
1	TempOK	uint8	Returns the system state after checking the input sensor data	Yes
2	converted_temp	float	Used to convert the mean value of TempS1 and TempS2 for displaying the current temperature	No

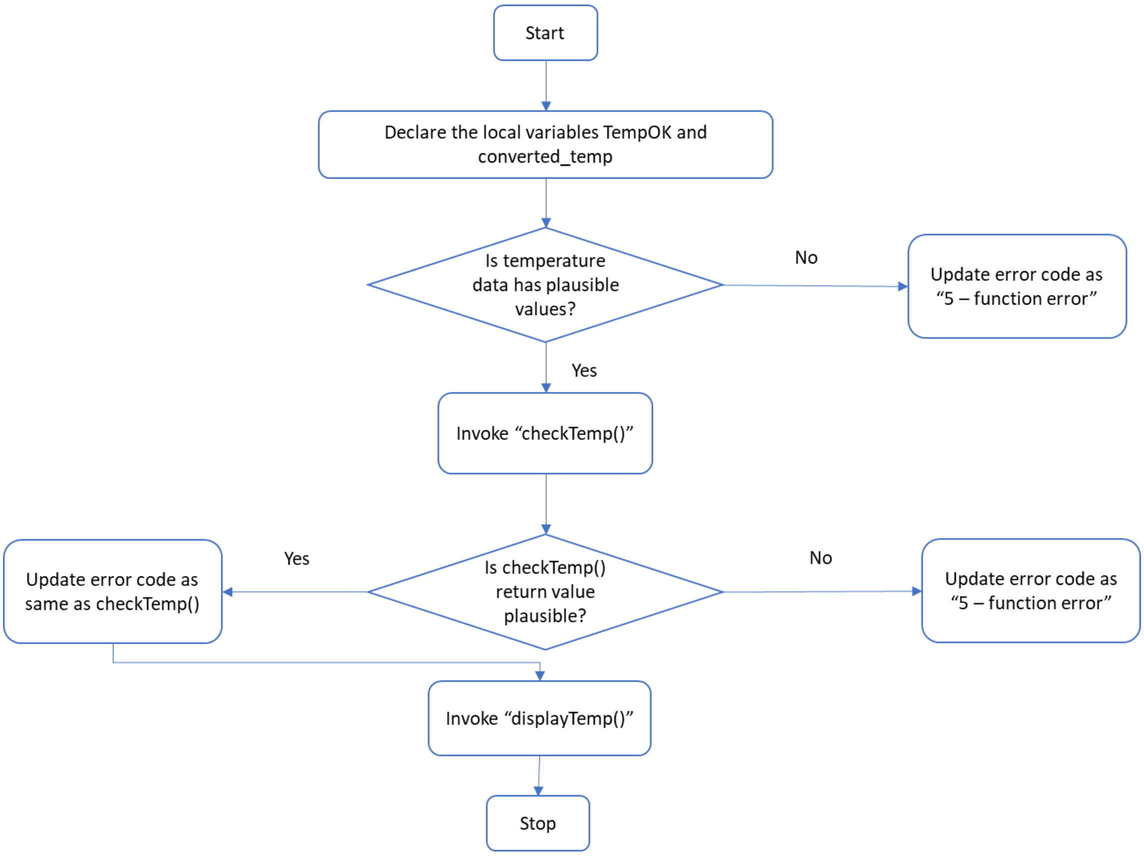
2.1.5 Assertion programming

If assertion programming was selected within the software architecture specification, a precondition and postcondition must also be specified for the module, if applicable.

- precondition of the safety function:
  n.a

- postcondition of the safety function:
  n.a.

2.1.6 Diagram of module implementation


Start
Declare the local variables TempOK and converted_temp
Is temperature data has plausible values?
No
Update error code as "5 – function error"
Yes
Invoke "checkTemp()"
Yes
Update error code as same as checkTemp()
Is checkTemp() return value plausible?
No
Update error code as "5 – function error"
Invoke "displayTemp()"
Stop

2.2 Module „checkTemp() “ [SW-SMS6.2]

2.2.1 Module description

The objective of this function is to check the temperature data of two sensors (1oo2 system) with the temperature data format from the user.

2.2.2 Parameters

- Are parameters passed to the module?

No.	Name	Data type	description	Plausibility check / fault tolerance
1	min_Temp	float	Minimum temperature limit specified by user	Plausibility check
2	max_Temp	float	Maximum temperature limit specified by user	Plausibility check
3	max_DisCr	float	Allowed discrepancies between two temperature sensor values	Fault tolerance
4	TempS1	float	Temperature data from Sensor1	Plausibility check & Fault tolerance
5	TempS2	float	Temperature data from Sensor2	Plausibility check & Fault tolerance

2.2.3 Return value

- Is a parameter returned by the function? - Yes

No.	Name	Data type	Description	Plausibility check
1	TempOK	uint8	Returns the system state after checking the input sensor data	No

2.2.4 Module variables

- Enter the (local) module variables including the range check for the values and check of physical quantities.

No.	Name	Data type	Description	Plausibility check
1	TempOK	uint8	Returns the system state after checking the input sensor data	Yes

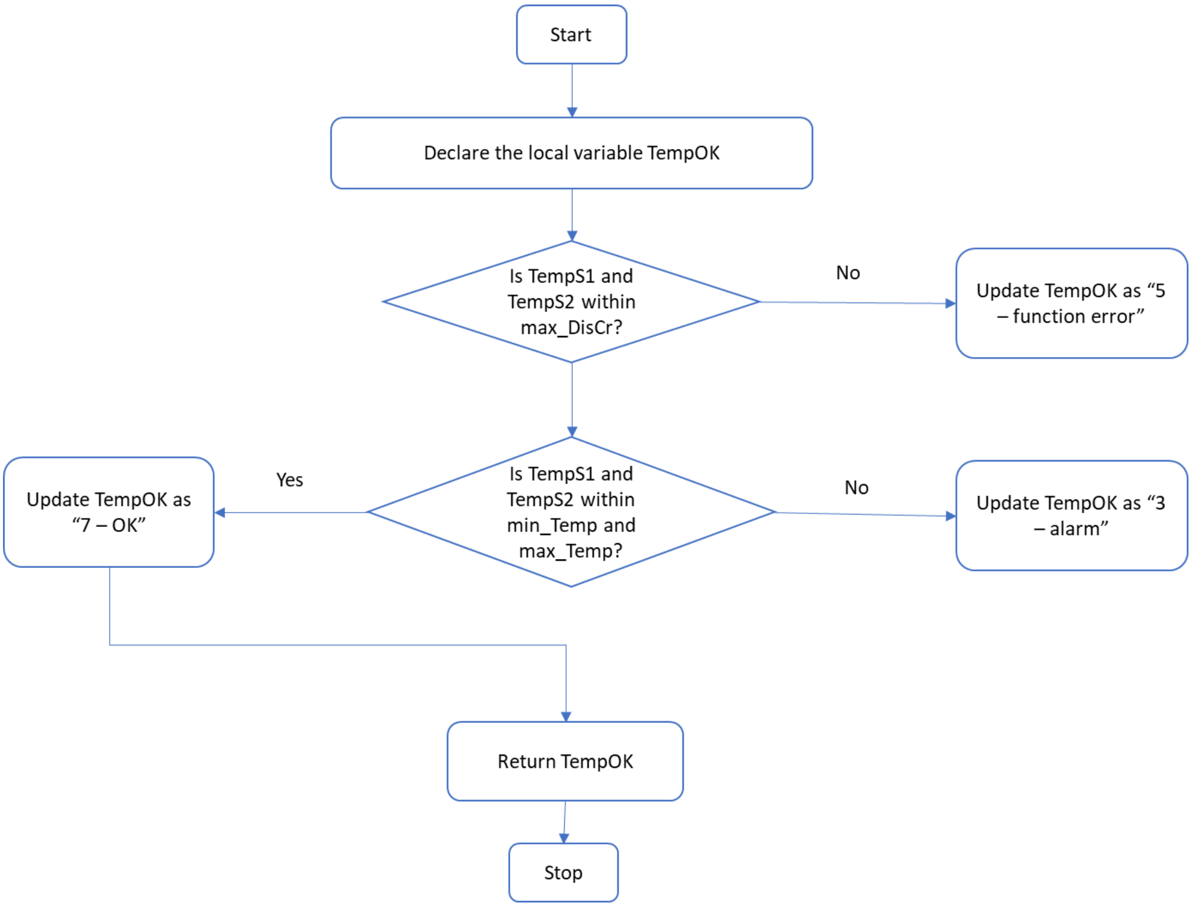
2.2.4 Assertion programming

If assertion programming was selected within the software architecture specification, a precondition and postcondition must also be specified for the module if applicable.

- precondition of the safety function:
  n.a.

- postcondition of the safety function:
  n.a.

2.2.5 Diagram of module implementation


Start
Declare the local variable TempOK
Is TempS1 and TempS2 within max_DisCr?
No
Update TempOK as "5 – function error"
Is TempS1 and TempS2 within min_Temp and max_Temp?
Yes
Update TempOK as "7 – OK"
No
Update TempOK as "3 – alarm"
Return TempOK
Stop

2.3 Module „calcC2F()“ [SW-SMS6.3]

2.3.1 Module description

This module is used as a sub routine in displayTemp() function where the mean value of TempS1 and Temp S2 (in °C) is converted to °F

Formula: Fahrenheit = (9/5) * (Celcius) +32

2.3.2 Parameters

Temp (float) - the mean value of TempS1 and Temp S2 (in °C)

2.3.3 Return values

Converted temperature (float) in °F

2.4 Module „calcF2C()“ [SW-SMS6.4]

2.4.1 Module description

This module is used as a sub routine in displayTemp() function where the mean value of TempS1 and Temp S2 (in °F) is converted to °C

Formula: Celcius = (Fahrenheit – 32) * (5/9)

2.4.2 Parameters

Temp (float) - the mean value of TempS1 and Temp S2 (in °F)

2.4.3 Return values

Converted temperature (float) in °C

2.5 Module „displayTemp()“ [SW-SMS6.5]

2.5.1 Module description

This module outputs the current system state to the user.

Output format:

Line 1:
The current temperature in the selected format (e.g. Celsius for Europe).
Line 2:
The temperature is to be indicated in the respective other format.
Line 3:
The temperature range is OK or if there is an error or if an alarm has been triggered.

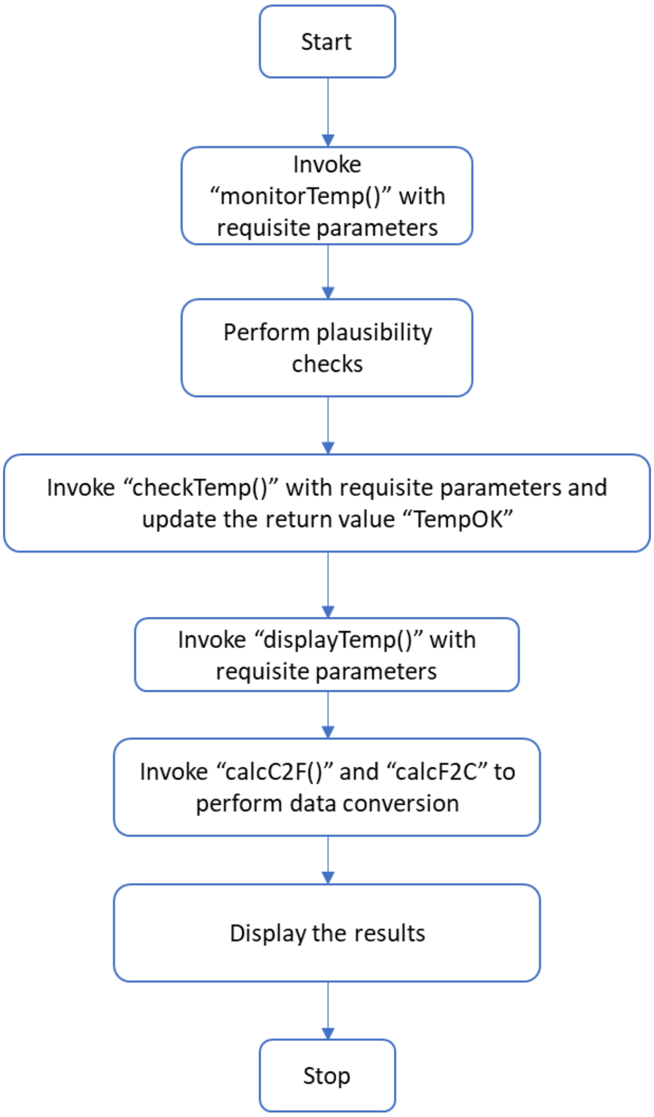
2.5.2 Parameters

No.	Name	Data type	description
1	TempFormat	char	Temperature limit specified by user
2	TempS1	float	Temperature data from Sensor1
3	TempS2	float	Temperature data from Sensor2
4	TempOK	uint8	Calculated system state

2.5.3 Return value

This function has no return value.

3.0 Flow chart of the whole function


Start
Invoke "monitorTemp()" with requisite parameters
Perform plausibility checks
Invoke "checkTemp()" with requisite parameters and update the return value "TempOK"
Invoke "displayTemp()" with requisite parameters
Invoke "calcC2F()" and "calcF2C" to perform data conversion
Display the results
Stop