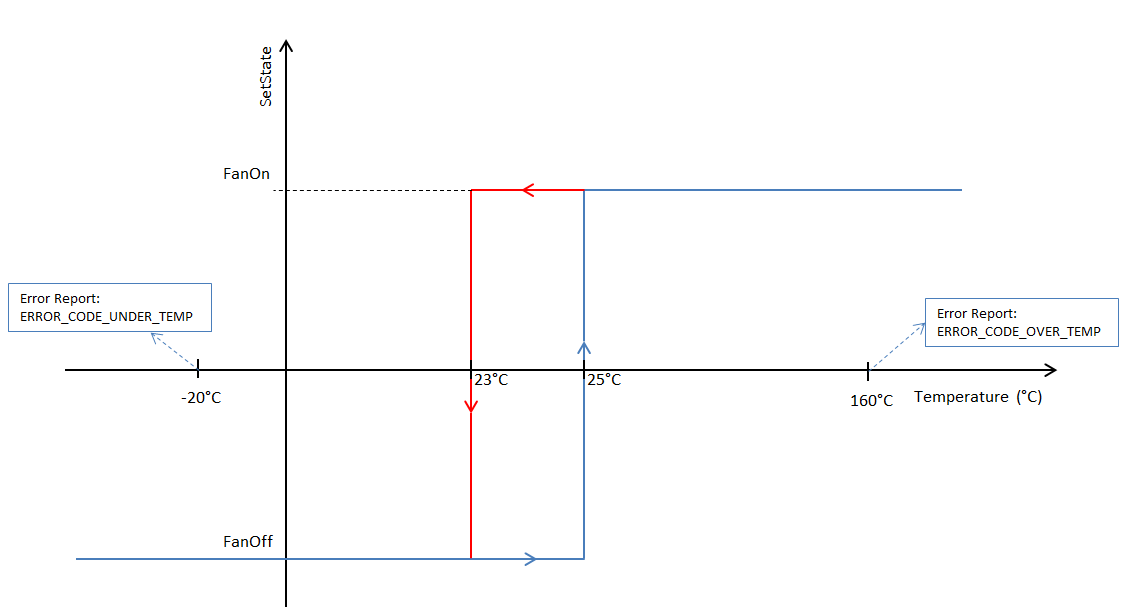
# FanControl

Please write a program in ANSI C99 programming language.

FanControl shall realize a hysteresis switching functionality based on the given requirements.

The program controls a simple fan that shall be turned on/off according to the temperature.

## Requirements:

1. FanControl function shall turn on the fan if the temperature is over the limit and it shall turn off the fan when the temperature is below the limit.
   1. Hysteresis shall be applied on the switching limits symmetrically.
   2. The hysteresis and the default limit shall be configurable on compile time.
      1. The default limit shall be 25°C and the default hysteresis value shall be 2°C.
   3. The temperature shall be acquired by reading the GetTemperature() function.
   4. The fan shall be turned on by calling SetState() function with the argument FAN\_ON and it shall be turned off by calling it with FAN\_OFF argument.
2. In case of the input temperature goes beyond its valid limits (-20°C..+160°C) then the component shall report an error.
   1. The error report shall be sent out by SendErrorReport() function. The reason of the report shall be stored into the errorCode field and the temperature value into the temperature field.
      1. In case of over temperature the errorCode shall be set to ERROR\_CODE\_OVER\_TEMP.
      2. In case of under temperature the errorCode shall be set to ERROR\_CODE\_UNDER\_TEMP.

## Design constraints

* FanControl function can use only the following APIs: GetTemperature, SetState, SendErrorReport.
* The FanControl function will be called periodically by the tasking with fixed periodicity and it shall be optimized for runtime.

Please provide a FanControl.c file which implements the required functionality and send it to me.

The subject of this mail shall be: [FanControl] - Student Team Candidate

## The implementation shall extend the following template:

**#ifndef COMPONENTTEST**

**typedef** **unsigned** **char** boolean;

**typedef** **unsigned** **char** uint8;

**typedef** **unsigned** **short** uint16;

**typedef** **unsigned** **int** uint32;

**typedef** **float** float32;

**typedef** **struct**

{

uint32 errorCode;

float32 temperature;

} dtErrorMessage;

**typedef** **enum**

{

*FAN\_OFF*,

*FAN\_ON*

} dtFanState;

**#define** ERROR\_CODE\_OVER\_TEMP ((uint32)1u)

**#define** ERROR\_CODE\_UNDER\_TEMP ((uint32)2u)

**extern** float32 **GetTemperature**(**void**);

**extern** **void** **SetState**(dtFanState fanState);

**extern** **void** **SendErrorReport**(dtErrorMessage\* errorMsg);

#endif /\* !COMPONENTTEST \*/

/\* Please add your own additions here! \*/

**void** **FanControl**(**void**)

{

}