Project

Goal: Implementing a basic control system for an autonomous catamaran.

A microcontroller board is connected to two outboard motors. The outboard motors are composed by a DC motor and a propeller installed at the end of its shaft. Together, the two outboard motors allow the catamaran to move and rotate in the water.

The microcontroller receives desired reference values for the rotation speed of the motors from a control PC, in terms of motor RPMs (rounds per minute). These reference signals are sent through a serial interface. The microcontroller sends some feedback messages back to the control PC with a period between 1 and 5 seconds.

Hardware specifications

- Each motor can run from -10000 to +10000 RPMs
- The RPM are controlled through a PWM signal.
    - The frequency should be at least 1 kHz.
    - 50% duty cycle corresponds to 0 RPM, 0% corresponds to -10000 RPM and 100% corresponds to 10000 RPMs.
- The propeller installed on the shaft of each motor limits the maximum RPMs to -8000 and +8000.

Firmware requirements

- The control system must never generate PWM signals outside of the specifications of the motor and its propeller.
    - If any reference value is given outside the specifications, the system should saturate it to the minimum/maximum allowed value.
- If no references are received for more than 5 seconds, the firmware should enter a timeout mode:
    - Both motors velocity should be set to zero
    - Led D4 should blink to signal timeout
    - When a new reference is read, then the led D4 should stop blinking and commands should be given again to the motor
- The firmware acquires the temperature sensor at 1 Hz frequency
- The control system should blink led D3 at 1 Hz to signal a correct functioning of the main loop.
- The user can set new minimum and maximum values through a dedicated command. These values must be within the allowed range of the propeller.
- If any of the buttons (S5/S6) is pressed, the firmware should enter a safe mode
    - Motors are stopped immediately and reference signals are ignored until the microcontroller receives an enable message.

Messages from the PC

$HLREF,n1,n2* where n1 and n2 are the RPMs for the left and right motors respectively.

$HLSAT,min,max*, where min and max represent the minimum and maximum RPMs allowed.

$HLENA*, enables the firmware to send references to the motors (to exit safe mode)

Messages to the PC

$MCFBK,n1,n2,state*, where n1 and n2 are the *applied* reference signals and state is 2 if the microcontroller is in safe mode, 1 if it is in timeout mode, 0 otherwise.

$MCTEM,temp*, where temp is the temperature