Auto FW Fan Control with Alert Example Project

1.0

Features

- Full featured fan control application
- Control completely in firmware
- Fan stall and speed regulation failure alerts

General Description

This example project demonstrates the unique benefit of the Fan Controller component whereby the fan control algorithm is implemented in firmware inside PSoC. This example project also demonstrates the component's ability to detect fan stall and other fault conditions. It is intended for users who need to get up and running quickly on a full-featured fan control application.

Development Kit Configuration

The following configuration instructions provide a guideline to test this design. For simplicity, the instructions describe the stepwise process to be followed when testing this design with the PSoC Development Kit (CY8CKIT-001) board, but can be generalized for the PSoC 3 Development Kit (CY8CKIT-030) and PSoC 5 LP Development Kit (CY8CKIT-050) as well.

1. Set LCD power jumper J12 to ON position and leave the rest of the board at default configuration.
2. Connect two 4-wire fans to the appropriate pins as shown in Figure 1. Note that if using the PSoC Thermal Management EBK (CY8CKIT-036), this can be connected to Port A on the CY8CKIT-001 or Port E on CY8CKIT-030/050.
3. Connect P1[6] and P1[7] to SW1 and SW2 on the board. For the CY8CKIT-030/050, reassign these pins to P6[1] and P15[5].
4. Ensure that the Character LCD is connected to LCD header on the development board.

Project Configuration

The TopDesign schematic looks as shown in Figure 1 below. The FanController is set to Automatic Firmware (CPU) Control. Stall and Speed Alerts are enabled in the 'Basic' tab of the configuration window. In the 'Fans' tab, 2 10-bit PWM outputs are defined. It is crucial to enter two data-points from the duty-cycle-to-RPM curve corresponding to each fan being controlled. These values are typically provided by the fan manufacturer and documented in the fan datasheet. The Fan Controller component configuration windows are shown in Figure 2,

Figure 3 and Figure 4 below. SW1 and SW2 are chosen as digital input pins, and control the RPM of the fans. The tach_1 and tach_2 digital input pins serve as indicators of the Fan RPM; while digital output pins, pwm_1 and pwm_2, drive the two fans. The 'alert' output of the component is connected to an ISR and a digital output pin which can be observed with an oscilloscope. The Character LCD is configured in its default mode. Finally, the 1-bit Status Register is configured to be sticky so that the EOC pulse is not missed.

Figure 1. TopDesign schematic

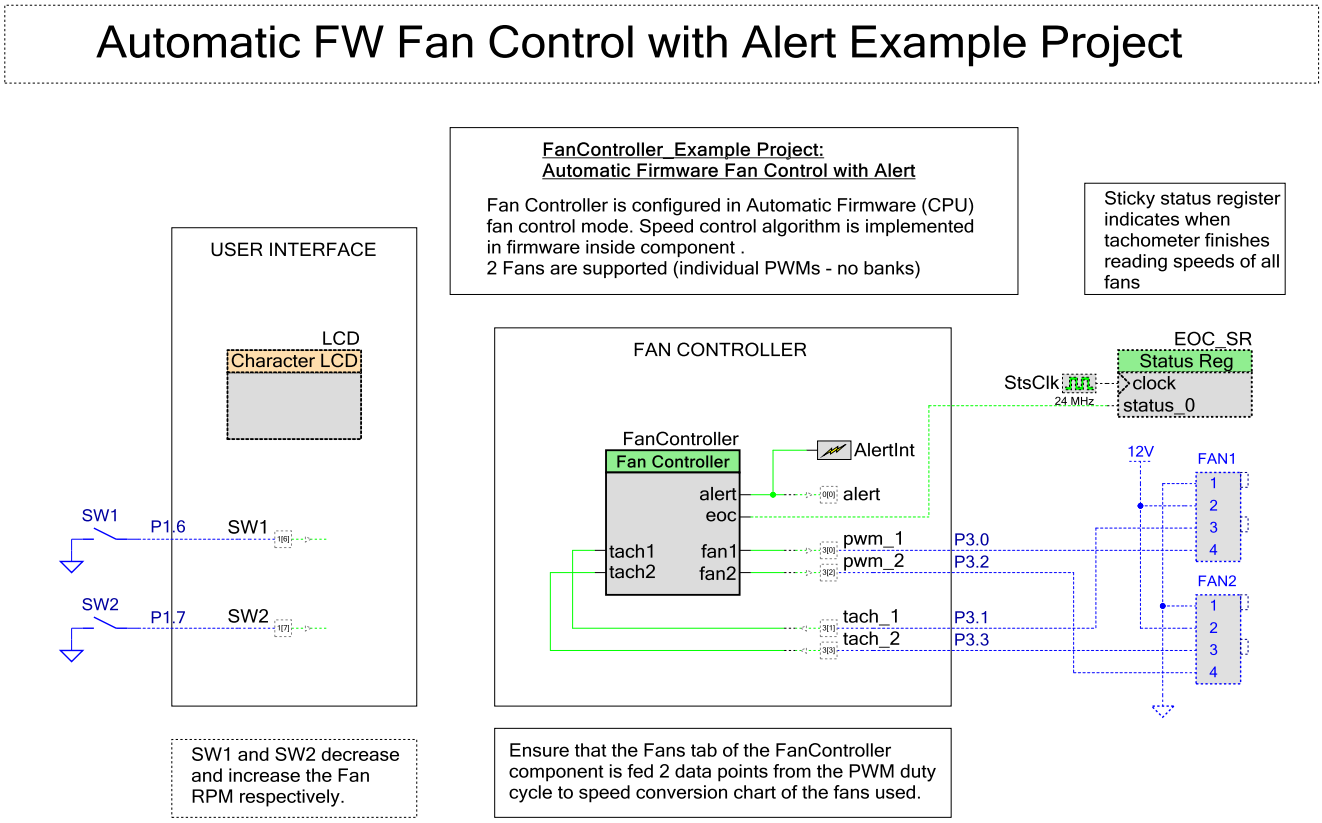


Figure 2. Fan Controller Component Configuration - Basic Tab

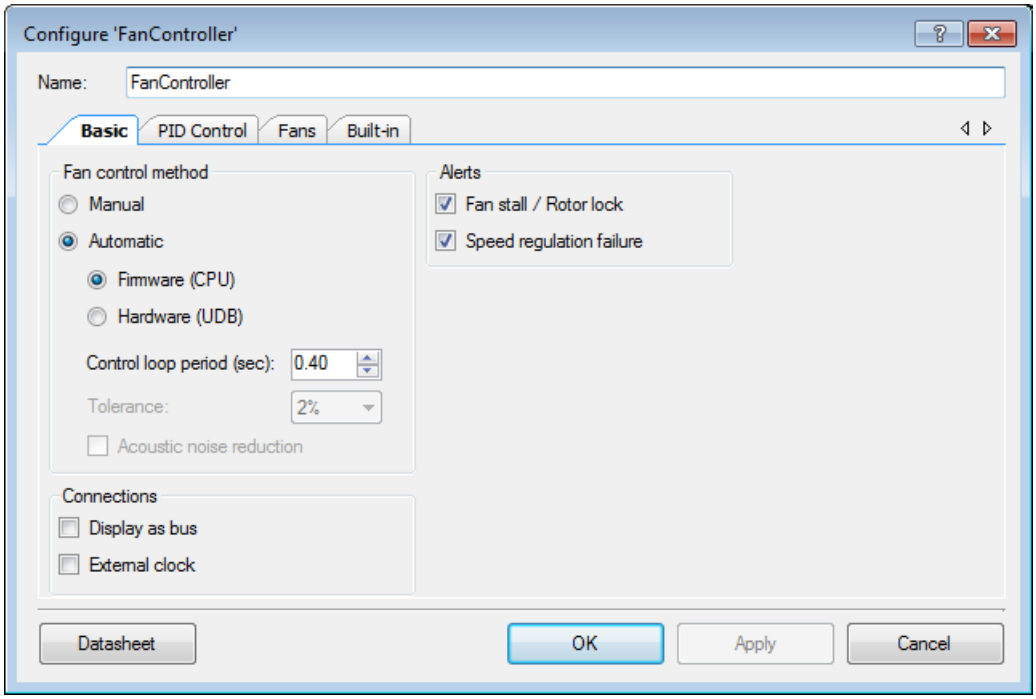


Figure 3. Fan Controller Component Configuration - PID Control

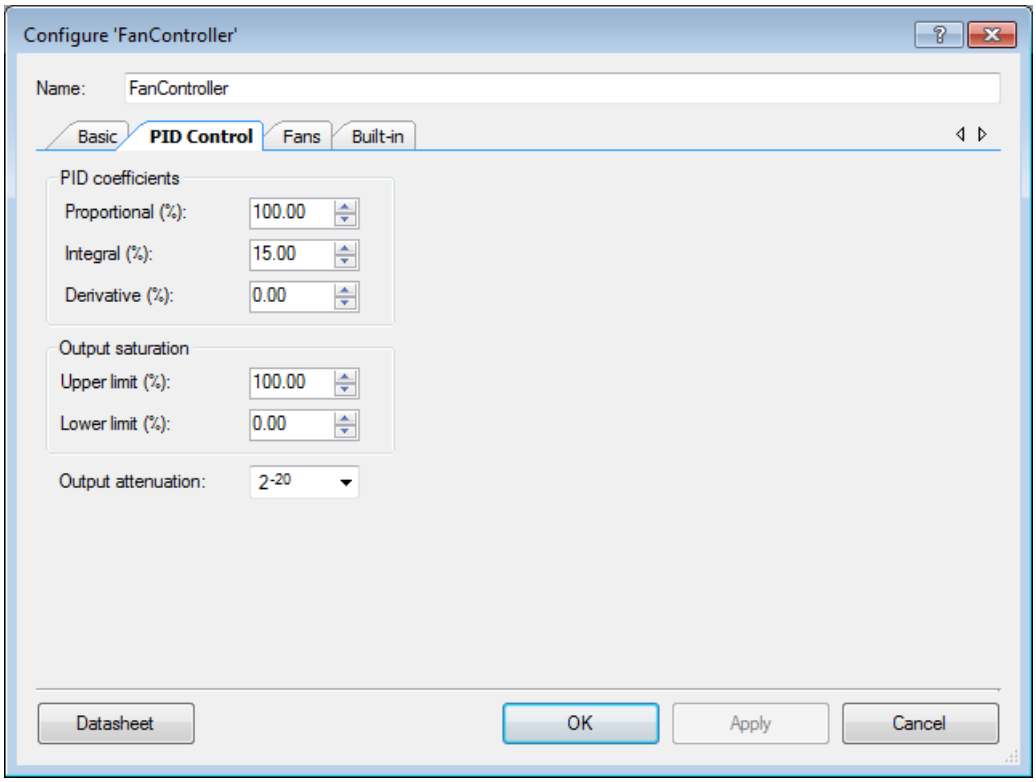
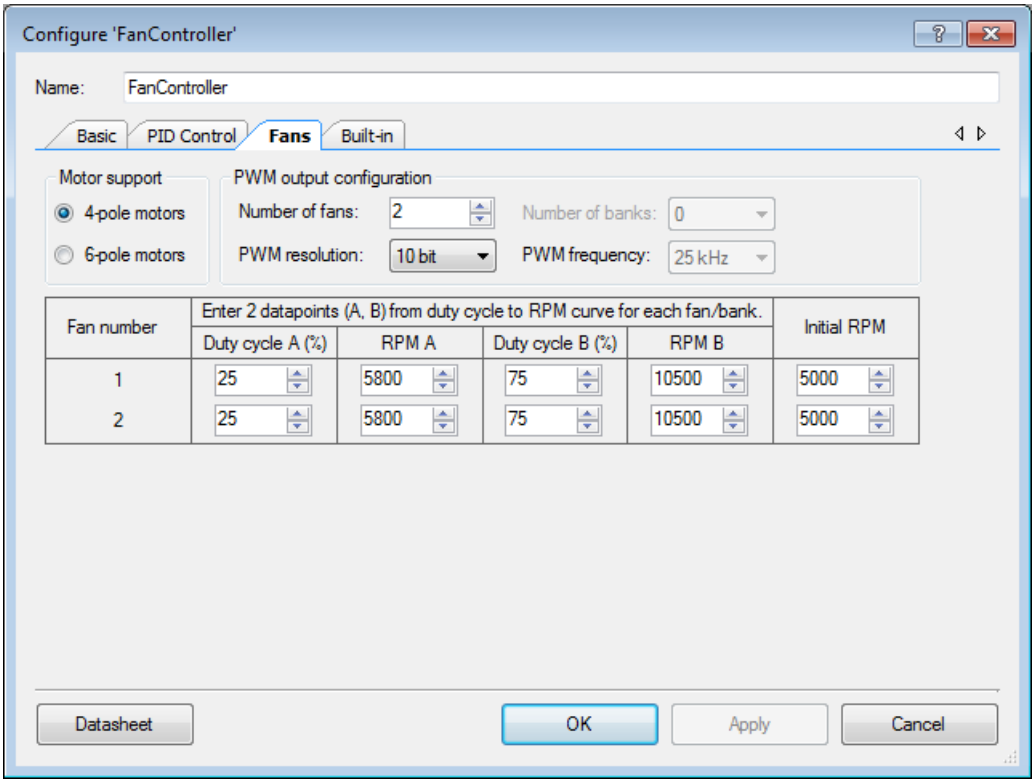


Figure 4. Fan Controller Component Configuration - Fans Tab



Project Description

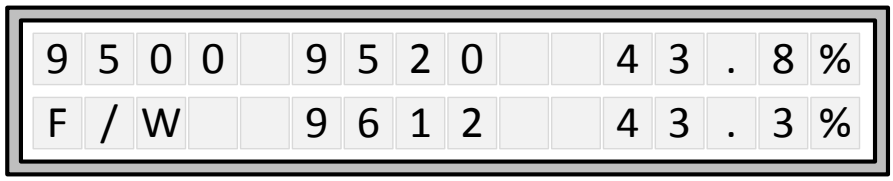
In the main.c file, the FanController and LCD components are started and the desired initial RPM is set for both fans. Global interrupts, fan alerts, and alert interrupt are enabled. If a stall/speed regulation failure is detected, a flag is set, and this flag is polled in the forever loop. Further, the inputs from SW1 and SW2 are read and the RPM of the fans is set accordingly.

Expected Results

The Fan Controller component will automatically maintain speed regulation. Fault conditions such as fan stall, broken or disconnected wiring, or attempted use of the fan outside its electromechanical limits are automatically detected by the component and flagged as alerts.

SW1 decreases fan speed, SW2 increases fan speed. Fault conditions are displayed on the LCD display and are indicated via a pulse on the 'alert' output pin P0[0].

Figure 5. Expected LCD output



Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

Phone : 408-943-2600
Fax : 408-943-4730
Website : www.cypress.com