Model-Based Prototyping with INTECRIO

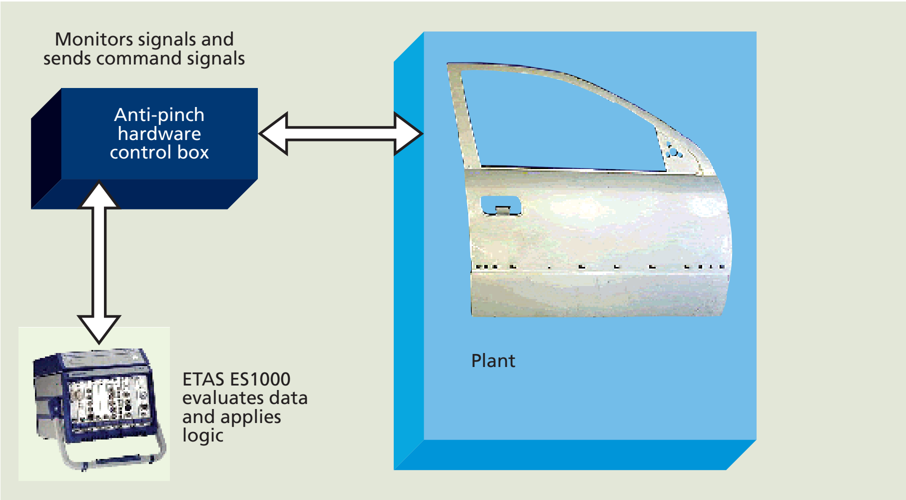
By Dr. Mark Thompson,
Kettering University

Kettering University students take the pinch out of automatic up windows

Electronic windows, designed to increase driver convenience and ease-of-use, are now a standard feature on millions of production vehicles. Today's automobiles have electronic window control, and feature automatic up and down operation with a single button press.

While automatic windows have been enjoyed by drivers in the US since 1941, automakers have struggled with ways to make automatic up/down windows safe so that small children and pets don't inadvertently get pinched or suffocated. Solving real world engineering problems is what students learn at Kettering University, located in Flint, Michigan, USA (see also our report on the clean snowmobile project at Kettering, pp. 34-36 of this issue). In this spirit, a team of Kettering electrical engineering students developed the power electronics and software for a system that safely operates a window with pinch-control (Figure 1). This project was part of the students' Senior Capstone Project.

Figure 1: Hardware setup of the anti-pinch control system.



Capstone Project design goals

Automobile safety and other features have to meet standards collected in the body of Federal Vehicle Safety Standards and Regulations. The design specifications for window control with anti-pinch protection were based on the standard FVSS 571.118.
Consequently, several aspects of the project's specs closely matched those that their body electronics colleagues in the automotive industry would follow to solve a marketing request.

These specifications included the following requirements:

- Window must have manual and automatic open/close modes of operation.
- Window must automatically fully open or close within 5 seconds.
- Window motor must stop when the window is fully opened or fully closed.
- Window motor must shut off after 5 seconds of continuous operation as a safety feature to protect the motor and regulator.
- Window anti-pinch control must detect an obstacle which exerts a force of 100 N or greater.
- Window must respond to a detected obstacle by lowering approximately 10 cm.
- Design and implementation must be completed within a $225 budget.

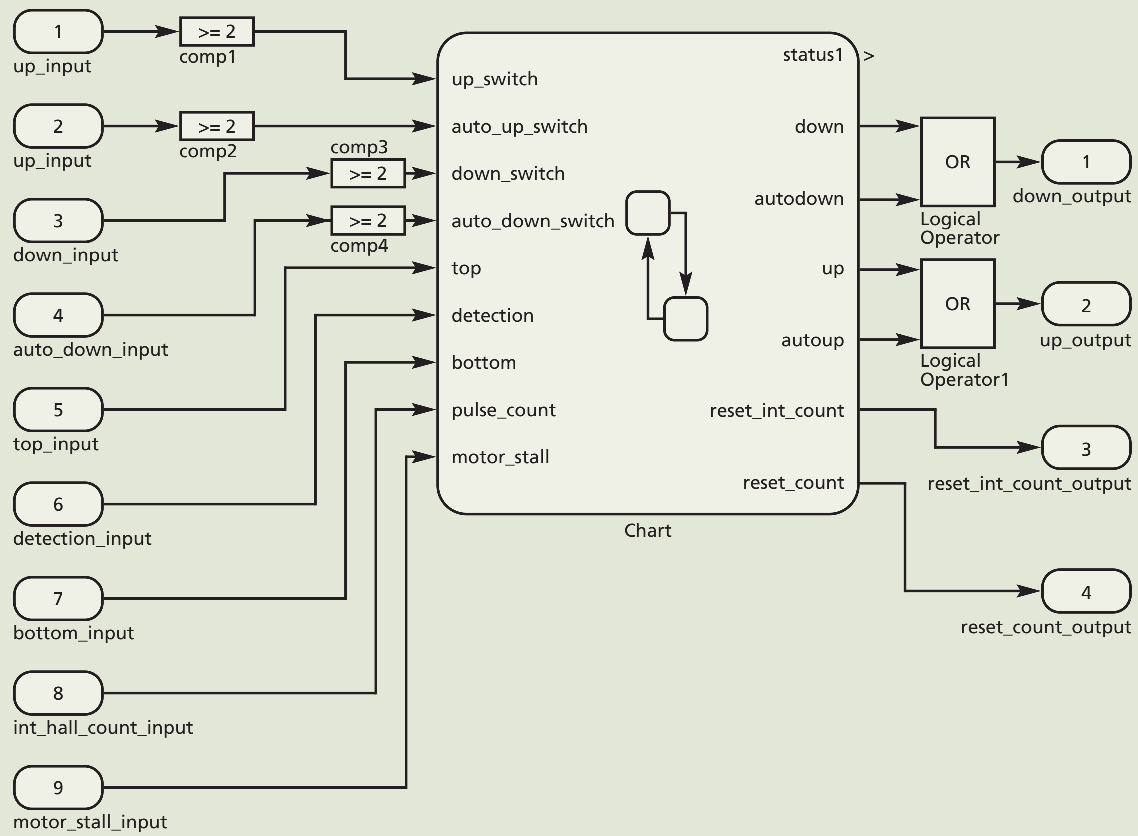


Figure 2: High-level view of the anti-pinch software module.

Similar to a typical implementation in the real world, the anti-pinch system used in the project included a current sensor that measured the current draw of the DC motor that powered the up/down motion of the door's window. This sensor detects any spikes in the motor current, and the software was designed to interpret these spikes. If the motor's current exceeds a specified threshold, the software interpreted the spike as a window obstruction, or pinch, and the window lowered 10 cm. To determine window travel position, the system used hall-effect pulses from the window's motor. Pulses were counted and a bottom position value of "0" was set for "Window Fully Open". The value for the up position "Window Fully Closed" equaled the number of pulses required for the window to be completely raised. If a motor current spike occurred during the window's travel, indicating an obstruction, the travel direction was reversed by pulses equivalent to 10 cm of travel.

Hardware subsystem design

Anti-pinch software for the project was downloaded to an ES1130 simulation/system controller board inside an ETAS ES1000 VME chassis. The board served as a prototype of a production controller. A/D and D/A boards in the ES1000 allowed the software to capture and output data. An anti-pinch hardware control box – which housed the window switches, motor relay, signal conditioning for sensors and power electronics – took in signals from the car door (the physical plant) and sent them to the ES1000, as well as relaying the signals from the ES1000 to the appropriate switching circuits driving the window motor's control relay.

Design specifications

The window control system was designed to have both manual and automatic up and down modes. Two momentary switches were used as input devices – one for manual up/down and the other for automatic up/down. The inputs from these switches were used in conjunction with controller software, as well as a FET drive H-bridge, to control the direction of the motor driving the window. Each switch was supplied with +5V, and the outputs for each switch were pulled down with a resistor to ground – eliminating noise when the switch was pushed. The signals from the switches were sent to the ES1000's A/D board and used as analog inputs to the software for switch position location. The control software on the ES1130 determined the proper motor command, instantaneous if in manual mode, or sustained if in automatic mode. Based on window motor current draw, the software shuts off the motor if the window stalled or if the window had reached the fully closed or fully open position. Two digital software output signals were sent to the coils in the H-bridge relay by the FET drive circuits to raise or lower the window.

Software subsystems

Anti-pinch software was created using MATLAB® version R2006a, and the corresponding versions of Simulink® and Stateflow™. When the model was finished it was turned into executable C code using Real-Time Workshop. ETAS' integration platform INTECRIO was used to integrate the executable software components and the hardware. The anti-pinch software module takes all of the necessary signals, including input from the switches and current location information, applies logic to them, and sends out command signals to run the window motor's relay and other counters. Figure 2 shows a high-level view of the anti-pinch software module.

A state-transition diagram determined the window's operation. There were three main stationary window states: top, middle, and bottom. Initially, the window was placed into the init1 state, causing the window to lower until it reached the bottom, at which point the hall pulse counter resets. With inputs from switches and pulse counts, the window state can transition to other states, such as down, auto_down, up, and auto_up. If a window pinch state is detected during the auto_up operation, the window will transition to the detection1 state, reset the intermediate counter, which counts hall pulses, and stop the window. The window goes to the detection2 state and lowers until the number of pulses reaches 10 cm (42 pulses), or the window is completely down. The whole process continues in an endless loop, allowing the user to operate the plant, i.e., the car door, continuously.

A window location software module was created to determine whether the window was completely up or down, or if the window met an obstacle and should reverse. Window positions were based on hall sensor counts and the voltage from a current sensor, both supplied as inputs to the software module. If the hall counts were near the top of the window (>200), and current had increased above the stall threshold (>1.8V), the module sent a "window at top" signal as an output. If the hall counts were near the bottom of the window (<10), and current had increased above the stall threshold (>1.8V), the module sent a "window at bottom" signal as an output. Finally, if the current was between two thresholds (1.8V<current<2.05V) and the window was between the bottom and the top (10<count<200), the module sent a "jammed" signal as an output.

A hall-pulse-count software module was required to determine window position inside the door. When the window was raised, the pulse counts were added, and when it was lowered, the pulse counts were subtracted. Figure 3 shows a block diagram of the anti-pinch control system.

The students at Kettering met all six of the anti-pinch window design criteria and completed the project 23 % under their original $225 budget. Every effort was made to use cost-effective hardware and a practical software design so that automatic up, anti-pinch windows could be feasible as standard equipment for any vehicle.

The Kettering team included: Davida Clark, Ben Daniels, Rajesh Gadde, Feba Pothen, and Jon Kowalski (ETAS).

Figure 3: INTECRIO block diagram of the anti-pinch control system.