

INTECRIOを用いた モデルベースの プロトタイピング

ケタリング大学
Dr. Mark Thompson

ミシガン州Kettering大学の学生によるセーフティ機構付き パワーウィンドウ開発プロジェクト

ドライバーの利便性と使いやすさを高める目的で作られたパワーウィンドウは、今では何百万台もの量産車に標準装備されています。今日の自動車のウィンドウは電子制御式になっているので、ボタンを押すだけで自動開閉できます。

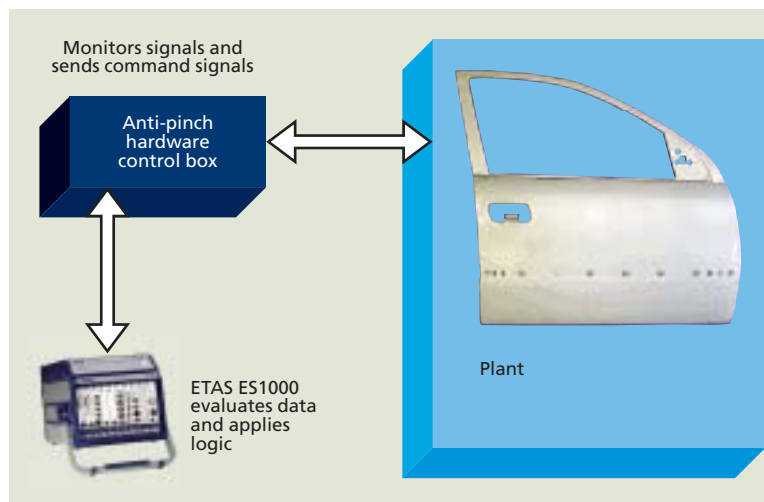
図1：
アンチピンチ制御システムの
ハードウェア構成

1941年に米国でオートウィンドウが利用され始めて以来、自動車メーカーは子どもやペットがはさまったり窒息したりしないようにウィンドウを安全に自動開閉する方法を懸命に模索してきました。現実の技術的問題を解決するという精神は、まさに米国ミシガン州フリントにあるケタリング大学の学生たちが学んでいることでした。この精神の下、ケタリング大学電子工学生チームはピンチ制御（図1）を用いてウィンドウを安全に開閉するシステムのためのパワーエレクトロニクスとソフトウェアを開発しました。このプロジェクトは同学生チームのSenior Capstone Projectの一環でした。

Capstone Projectの設計目標

自動車の安全機能やその他の機能は、米国連邦自動車安全基準の本体に集められている基準に適合しなければなりません。このアンチピンチ機能搭載ウィンドウ制御の設計仕様はFVSS 571.118の基準に基づいています。

そのため、このプロジェクトの仕様のいくつかの様相は、自動車業界のボディエレクトロニクス分野の皆様がマーケティング要件を満足するために従うであろう事柄と一致しています。



この仕様には以下の要件が盛り込まれていました。

- 手動と自動のウィンドウ開／閉モードを備えていること
- 自動開閉時、ウィンドウは5秒以内に完全に開く（閉じる）こと
- ウィンドウが完全に開いた（閉じた）らウィンドウモータは停止すること
- モータおよびレギュレータを保護するために、ウィンドウモータは5秒間連続稼働したら自動的に止まること
- ウィンドウアンチピンチ制御は100N以上の力を示す障害物を検知すること

- 検知された障害物に対する反応として、ウィンドウが約10cm開くこと
- 225ドルの予算内で設計と実装を完遂すること



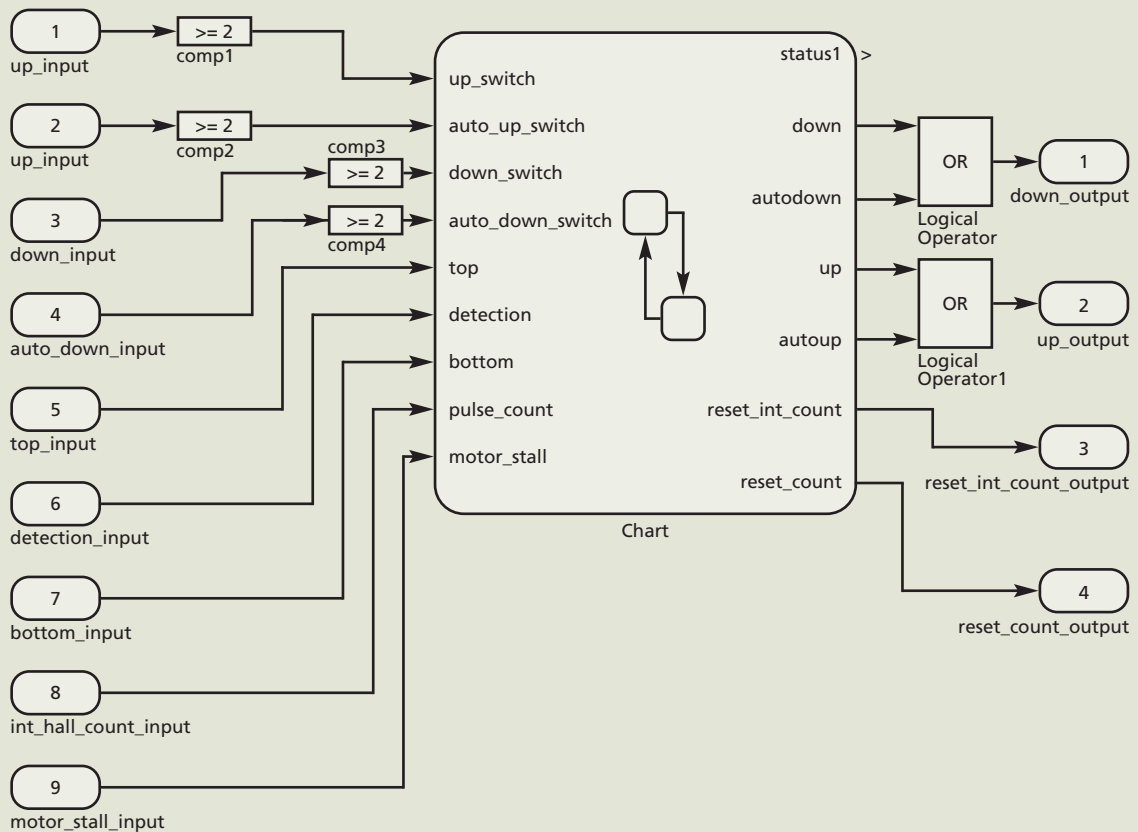


図2：
アンチピンチ
ソフトウェア
モジュールの
ハイレベルビュー

プロジェクトで用いたアンチピンチシステムには、実際に行われている典型的実装と同様に、ドアウィンドウの開閉動力を供給するDCモータの電流引込みを測定する電流センサが組込まれていました。このセンサがモータ電流のスパイクをきれなく検知し、ソフトウェアがそれらのスパイクを解釈するように作られました。モータ電流が所定のしきい値を超えると、ソフトウェアはウィンドウに障害物がはさまった（ピンチ）と解釈してウィンドウを10cm下げます。ウィンドウの現在位置の算出はウィンドウのモータからのホール効果パルスを使用して行われます。ウィンドウが完全に開いている時の位置を表すパルスカウント“0”が“Window Fully Open”として設定され、ウィンドウが完全に閉じている時の位置を表すパルスカウント“Window Fully Closed”はウィンドウが完全に上がりきるために必要なパルス数と等しくなります。ウィンドウ開閉動作中にモータの電流スパイクが発生して障害物の存在を知らせた場合には、10cm相当のパルス数だけウィンドウの動作方向が逆になるようにしました。

ハードウェアサブシステムの設計

このプロジェクトのアンチピンチソフトウェアはETAS ES1000 VME筐体内のES1130シミュレーション/システムコントローラボードにダウンロードされました。このボードはプロダクションコントローラのプロトタイプになりました。ES1000にはA/DボードとD/Aボードが搭載されているので、ソフトウェアはデータを収集し出力することができます。ウィンドウのスイッチ、モータリレー、センサ用/パワーエレクトロニクス用の信号処理回路を収容しているアンチピンチハードウェア制御ボックスが、自動車のドア（＝プラント）から信号を取り込んでES1000に送ったり、ウィンドウモータの制御リレーを駆動する適切なスイッチング回路にES1000からの信号を中継したりします。

設計仕様

このウィンドウ制御システムは手動と自動の両方の開閉モードで使用できるようになっています。2つのモーメンタリスイッチが入力装置（一方は手動開閉用、他方は自動開閉用）として使用されます。これらのスイッチからの入力をコントローラソフトウェアやFETドライブHブリッジとともに使用することにより、ウィンドウを動かすモータの方向を制御します。各スイッチには+5Vの電圧が供給されていて、各スイッチ用の出力は抵抗器によりグラウンドへプルダウンされてスイッチ押下時のノイズが除去されます。スイッチからの信号はES1000のA/Dボードに送られ、スイッチ位置特定用ソフトウェアへのアナログ入力として使用されます。ES1130上の制御ソフトウェアは適切なモータコマンド（手動モード時はinstantaneous、自動モード時はsustained）を判定します。ウィンドウモータの電流引込みに基づいて、ウィンドウが失速したが、あるいは完全に開いた/閉じたと判断された場合には、ソフトウェアはモータの電源を切ります。ソフトウェアからの2つのデジタル出力信号がFETドライブ回路によりHブリッジリレー内のコイルに送られ、ウィンドウが閉じたり開いたりします。

ソフトウェアサブシステム

アンチピンチソフトウェアは、MATLAB®バージョンR2006aと、それに対応するバージョンのSimulink®およびStateflow™を使用して作成されました。完成したモデルはReal-Time Workshopにより実行形式のCコードに変換されました。ETASの統合プラットフォームINTECRIOを使用して、実行形式のソフトウェアコンポーネントとハードウェアが統合されました。アンチピンチソフトウェアモジュールは、スイッチからの入力や現在位置の情報など必要なすべての信号を取り込み、それらにロジックを適用して、ウィンドウモータのリレーや他のカウンタを動かすコマンド信号を出力します。図2はアンチピンチソフトウェアモジュールのハイレベルビューを示しています。

ウィンドウの動きは状態遷移図により決定されます。ウィンドウの主な静的状態はtop（一番上）、middle（中間）およびbottom（一番下）の3つです。最初、ウィンドウはinit1状態になって一番下まで下がり、ホールパルスカウンタがリセットされます。スイッチ値およびパルスカウンタが入力されると、ウィンドウは他のdown、auto_down、up、auto_upなどの状態に遷移する可能性があります。auto_up動作中にウィンドウピンチ状態が検知されると、ウィンドウはdetection1状態に遷移し、ホールパルスをカウントする中間カウンタがリセッ

トされ、ウィンドウが停止します。それから、パルス数が10cm相当（42パルス）になるかウィンドウが一番下まで下がるまでの間、ウィンドウはdetection2状態になります。このプロセス全体が無限ループで続くので、ユーザーはプラント、つまり自動車のドア（ウィンドウ）を連続的に操作できます。

ウィンドウが完全に閉じているか開いているか、あるいは障害物にぶつかったため逆方向に動かす必要があるかどうかを判断するために、ウィンドウ位置特定ソフトウェアモジュールが作成されました。ウィンドウの位置は、ホールセンサカウントおよび電流センサから得られる電圧（どちらもソフトウェアモジュールへの入力として供給されます）に基づいて決められます。ホールカウントがウィンドウの一番上に近い値（>200）で、しかも電圧がストールしきい値より大きくなっている（>1.8V）場合、モジュールは“window at top”信号を出力します。また、ホールカウントがウィンドウの一番下に近い値（<10）で、しかも電圧がストールしきい値より大きくなっている（>1.8V）場合、モジュールは“window at bottom”信号を出力します。最後に、電圧が2つのしきい値の間（1.8V < 電圧 < 2.05V）でウィンドウ位置が一番下でも一番上でもない（10 < ホールカウント < 200）場合、モジュールは“jammed”

信号を出力します。

ホールパルスカウントソフトウェアモジュールでは、ドア内のウィンドウ位置を判定する必要があります。ウィンドウが上がるとパルスカウントが加算され、下がるとパルスカウントが減算されます。図3は、このアンチピンチ制御システムのブロックダイアグラムを示しています。

ケタリング大学の学生は前述したアンチピンチウィンドウ設計要件の最初の6つを満足したうえ、残りの1つの要件だった225ドルという予算を23%も下回る費用でプロジェクトを完遂しました。費用効率の高いハードウェアと現実的なソフトウェア設計を採用するためにあらゆる努力を払い、自動開閉式アンチピンチウィンドウをあらゆる自動車に標準装備できるようにしました。

ケタリングチームのメンバー：Davida Clark、Ben Daniels、Rajesh Gadde、Feba Pothan、Jon Kowalski (ETAS)

図3：
アンチピンチ制御
システムのINTECRIO
ブロックダイアグラム

