Military
EMBEDDED SYSTEMS

Articles, Technology Feature

All discretes are not created equal

No longer just the basic On/Off "workhorse," Discrete I/O interfaces are standing up to the rigors of modern control systems.

By

Amir Shafy
NORTH ATLANTIC INDUSTRIES

In modern military, aerospace, and industrial control systems, the Discrete I/O interface is so common that it is often overlooked in the early design integration phase. This can lead to missed operational envelope considerations and other interfacing issues that might yield further complexity, increased cost, and missed operational Size, Weight, and Power (SWaP) savings. Meanwhile, Discrete I/O interfaces have evolved from basic On/Off “workhorses” into more sophisticated, intelligent devices that meet the challenges of designing control systems with Discrete I/O applications. Input/output considerations and the benefits of these new Discrete I/O control systems are examined.

Discrete I/O, in its simplest definition, is the sensing or driving of two-state voltage/current sources. The two states (On/Off or High/Low) are commonly defined by the system power and GND buses for simplicity and robustness. Perhaps because it is the most common interface, it is frequently overlooked and considered relatively minor in the overall design of complex control systems; this oversight, however, can lead to last-minute design scrambles and unforeseen variables such as unusual rail voltages, unstable power supplies, and noisy actuators.

Many complex military systems, such as engine and power control, data distribution, navigation, positioning, motion control, and weapons and targeting have discrete interfaces dispersed throughout. Battery system interlocks, servo brakes, hatch closure mechanisms, and arming indicators are just a few examples of simple but effective areas for discrete interfaces. But today’s Discrete I/O is much more than a simple On/Off “workhorse”; it has evolved into a sophisticated, intelligent, programmable, and wide-operating envelope control device. Thus, a closer look at the input/output considerations and benefits of these more modern, COTS-based I/O control devices is presented.

Managing system design with programmable I/O

Discrete I/O, usually identified on a per-channel basis, is normally preset as an *input*- or *output*-type interface. As an input, the Discrete I/O is generally used to detect the state (On/Off) of a particular device. In a typical input configuration, Discrete I/O interfaces directly with common actuators, switches, push buttons, state sensors, relays, indicators, and many other similar control- or contact-type devices. As an output, the discrete is frequently used to control the state of a particular device. Configured as an output, I/O may directly drive relay coils, solenoids, incandescent lamps, LEDs, and many other devices. Cumulatively, discretes used as inputs are decisively managed to control outputs and vice versa. In fact, some complete control systems can be managed by Discrete I/O only. This flexibility allows last-minute additions or changes in the overall system to be maintained seamlessly.

Input considerations

Configured as an input, Discrete I/O interfaces are referred to as either *voltage* or *contact sensing.* In voltage sensing, the device source will drive the input to either the Low or High state. On the surface, this might seem like a very simple interface to manage. However, using multiple power supply rails for different sensor inputs can cause chaos for the system designer because different channels might require different threshold levels. Programmable thresholds on a per-channel basis enable the system engineer to address this issue much later in the design process. Knowing that the Discrete I/O can handle a wide range of voltage levels (typically up to 60 V) affords engineers more time for other design considerations. In contact-sensing applications – such as GND-OPEN or OPEN-VCC levels (see Figure 1 for detailed circuit implementation) – it is the “open” sense or “floating” voltage level (which is in an undetermined state) that usually requires more effort from the system designer to implement. In the past, an external pull-up or pull-down resistor would be added to provide the hard-level state change required. Today, fully programmable pull-up or pull-down current sources can be set to provide the level that transitions require.

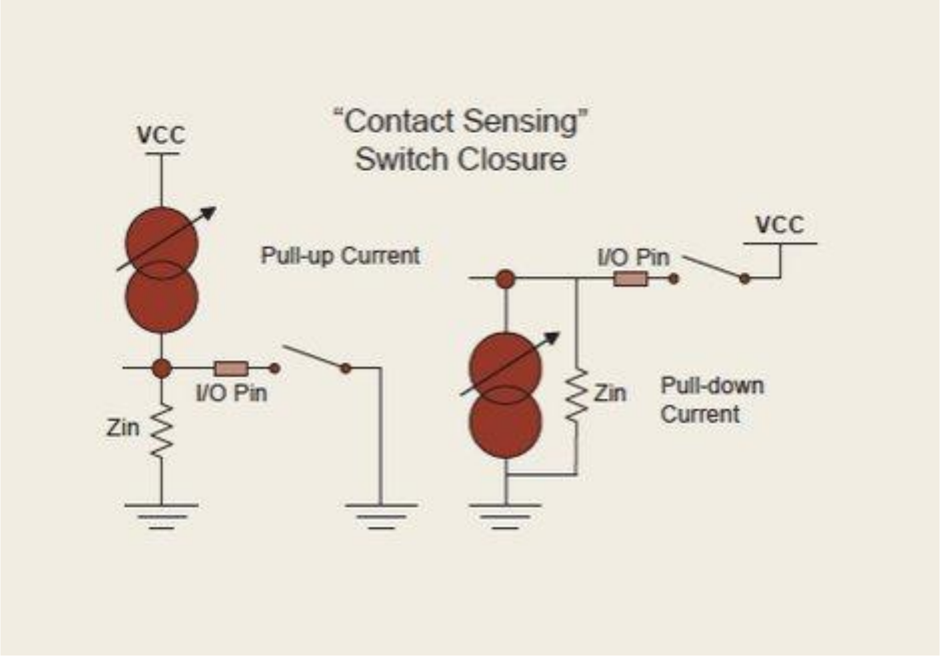

Figure 1: Input contact sensing circuit example with integrated pull-up/pull-down current sources

Dealing with “relay chatter” poses another challenge. Relay chatter is the oscillation that occurs when mechanical contacts are initially closed and there is an inherent “voltage bounce” that could be interpreted as multiple-level changes. For example, if a dry contact switch is used to “arm” a sensor, multiple voltage transients over several milliseconds could cause multiple “re-arming” commands that might yield unwanted results such as a misfire. With the ability to program a selected time-span debounce filter for the discrete input, this issue is managed by effectively ignoring any spurious transients that might be interpreted as a real signal.

Programmable features such as input or output selection, adjustable current source with “pull-up” or “pull-down” capability, and variable programmable filtering save integration time because no external components (such as external sensor modifications, cabling additions, or added resistors) are required.

Output considerations

Configured as an output, Discrete I/O interfaces are often referred to as *current sources* (high-side drive), *current sinks* (low-side drive), or *push-pull drives* (sourcing or sinking, depending on state). For example, discretes can serve as the output drive for a multi-indicator flight status panel. The panel might have many different types of illumination devices including LEDs and incandescent bulbs. Each illuminating device might have its own requirements – some might be configured to turn on by applying a voltage and some by supplying the GND (or return). Incandescent bulbs also have inherent issues with initial current surge, whereby the filament, prior to heating up, might draw more current than the rated average steady state. For example, the wide-use legacy GE382 miniature indicator bulb is typically rated at 14 V/80 mA. The initial cold turn-on might actually yield a current draw 10x greater for 20 msec.

Since the discrete can handle initial current surges greater than the rated average current draw, the size of the component selections required for driving the indicator is reduced. This subsequently provides smaller size and overall higher channel density. Without specifically knowing which device is connected to which channel, the systems engineer would be required to research and customize the hardware design to interface with each device independently. The ability to program the type of output drive simplifies the engineering integration required. Additionally, a COTS-based Discrete I/O technology can drive any device "out-of-the-box," saving time, effort, and engineering dollars. Push/Pull configurations (see Figure 2 for detailed circuit implementation) can also save space and weight in interconnecting cabling where two devices can effectively be controlled with one channel in a complementary On-Off-On configuration.

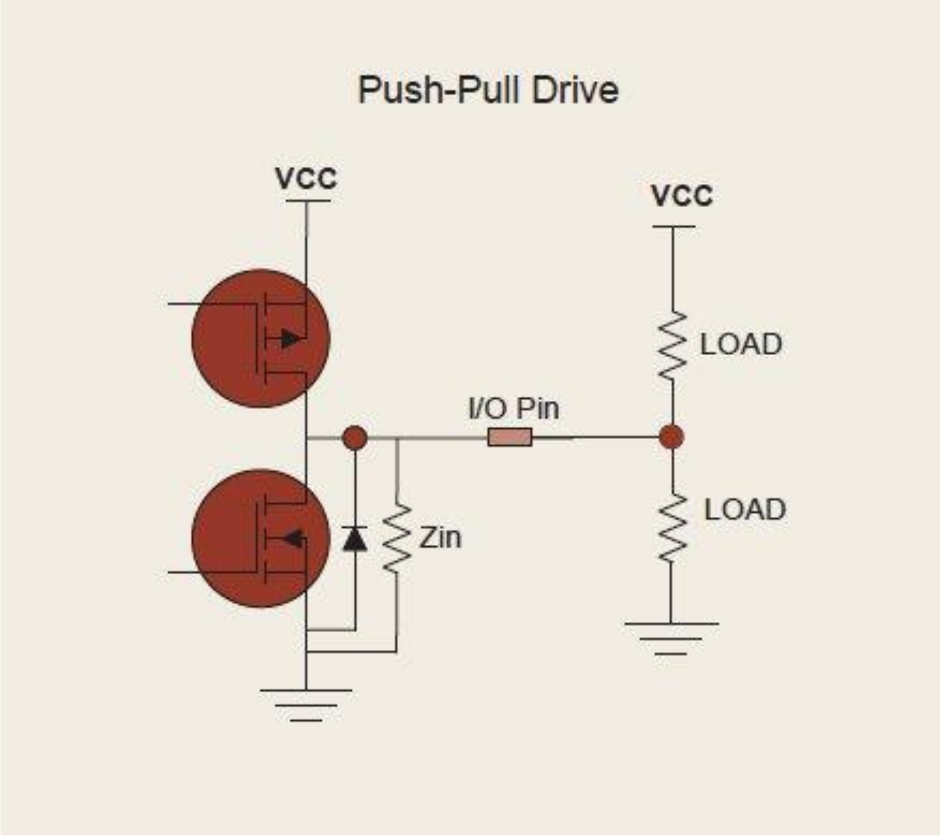

Figure 2: Output programmable push-pull single channel/complementary load circuit example

Incorporating fast response times

The simple discrete interface has evolved into providing more than just an On/Off interface and offers greater benefits. With the advent of DSP and FPGA technology, the once-basic Discrete I/O – originally limited to only slow sensing, slow driving applications – has emerged into new interfacing applications. For example, a simple parallel discrete transceiver can be realized and implemented for subsystem intercommunications, as individual channels are typically managed simultaneously in 16-channel “module banks.” Also, combined with a fast FET front end, integrating requirements such as an LED dimming control in addition to On/Off levels can be realized. Previously, performing dimming control with a discrete would involve additional circuitry to pulse the discrete output or VCC source, thereby complicating the integration effort.

Additionally, simple preconfigured routines can be implemented in the DSP and FPGA control circuitry so that pulsed outputs or inputs can be managed simply by programming a period and duty cycle. The update rate of discretes today is at 50 KHz as opposed to perhaps 100 Hz in the past, depending on the Discrete I/O design – quite an accomplishment realizing that voltage levels up to 60 V are switched. Having basic programmable control at the channel level may relieve system processors from maintaining a continuous pulsed train effort.

On-line background status and health monitoring with Built-In-Self-Test

Typically, embedded systems are located in strategic access areas and are difficult to maintain. In military systems where event failures cannot be tolerated, a Built-In-Self-Test (BIT or BIST) is paramount. The modern Discrete I/O design includes a background BIT for monitoring the health and status of each channel. The status can be polled or interrupted on specific events, such as “overcurrent detect.” Combined with the ability to read the actual voltage or current being sourced or the input from a sensor/load during actual operation (online, without shutting down the application), the operational status of all channels can be monitored without any redundant hardware or additional software overhead.

Supporting common platforms with high channel density

As the Discrete I/O is arguably the most common interface used in most embedded control platforms, there is often a need for many channels in a control system. For example, one (of many) flight control panels might have 40 indicators and 20 switches – 60 channels total. Another example might be a distributed control system interface used on modern amphibious assault ships. Controlling multiple subsystems such as engine control and monitoring, power distribution, weapons stores and status, navigation, and others yields literally thousands of Discrete I/O monitoring channels that must all be accessed, monitored, and controlled on a real-time basis. Networked subsystems via GbE (at the card level) can be interconnected, yielding a true distributed networked I/O system targeting individual areas.

With component size reductions, modern, state-of-the-art Discrete I/O interfaces can deliver the maximum channel density count versus programmability versus operational characteristics envelope (voltage and current). These designs typically employ a modular concept and – combined with common platforms such as VME, VPX, CompactPCI, and others – this concept can be combined to provide in excess of 130 channels per card. Combining the common card platform with processing capability (Figure 3), the Discrete I/O can be maximized for channel count and/or integrated with other I/O functions such as A/D, D/A, communications interfacing, and so on, for a complete distributed I/O package capable and programmable for any I/O application.

Figure 3: Common platform/distributed Discrete I/O

The evolution of the "workhorse"

The Discrete I/O, no longer the overlooked basic On/Off "workhorse," has evolved into an intelligent, programmable, and wide-operating envelope control device. With continued dedicated engineering and close customer interfacing, common COTS products are available to simplify the system integrator’s daunting task. Companies such as North Atlantic Industries (NAI) have introduced COTS-based I/O interfacing products that meet wide operating-envelope demands, mitigating the costs of design and field risks commonly overlooked early in the control system design process.

Amir Shafy is a Product/Applications Engineering Manager for North Atlantic Industries (NAI). Amir has a Bachelor’s degree in Electrical Engineering and Technology (BSEET) and has held various engineering design and integration positions in the rugged MIL peripherals, embedded, and portable computing industries. He can be contacted at ashafy@naii.com.

North Atlantic Industries (NAI)

www.naii.com