Maximize infotainment system design reuse with I2C GPIO expanders

Roland Pang, Product Marketing Engineer, High Volume Linear Texas Instruments - January 03, 2017

Consumers are exposed to the latest technologies through advancements in mobile devices, home electronics, and computing. It is now the expectation to have the same user experience in the automobile. Many of today’s infotainment systems are based on an infotainment processor with an established ecosystem platform. While the system engineer often can reuse a core system, some challenges to interface next generation technologies to legacy subsystems are still present. A lack of general purpose inputs/outputs (GPIOs) for controlling peripheral devices or board-routing problems emerge as the system increases in complexity. This article highlights the convergence of infotainment and cluster systems to form an integrated cockpit. I2C GPIO expanders provide a great avenue to reusing the existing design, while integrating new feature sets that meet consumer demands.

System integration

Infotainment is the presentation of information and entertainment that encompasses work and recreation. Today’s infotainment systems revolve around content and connectivity. To meet consumer expectations, the industry is obsessed with keeping the driver and passengers connected through digital technology. While this presents new distractions to the driver, automakers are taking measures to refocus the driver’s attention back on the road.

Infotainment subsystems are increasingly complex, largely because of the need to stay connected to the world while efficiently processing this information. Most automakers do not want to make drastic changes to proven system architectures that have gone through a long, arduous qualification process. However, there must be an avenue to bridge existing designs to keep up with consumer demands.

For decades, the in-dash car stereo or head unit has been at the center of the infotainment system, storing and delivering content and processing data that enhances the user experience. Over time, Bluetooth has been incorporated into these head units to help keep drivers focused on the road. By 2022, Bluetooth will be in over 90 percent of all vehicles worldwide. More recently, with the advent of Apple Carplay and Android Auto, anybody with even an entry-level smartphone can experience a feature-rich, integrated navigation environment. What was once deemed a luxury option is now becoming commonplace.

As these high-end features migrate to mid- and low-end vehicles, automakers are pushing the limits of existing architectures to accommodate next-generation feature sets. Merging the head unit with the instrumentation cluster electronics, along with some body control module (BCM) functions, delivers a seamless integrated cockpit experience to the driver and passengers. Infotainment
companies have invested a lot of engineering and qualification time to conceptualize, design, and implement the latest system hardware and software. Although development time for automotive electronics can extend over the course of several years, infotainment is one of the fastest moving and evolving aspects of the automobile. It is in the best interest for infotainment engineers to reuse both hardware and software while differentiating their end product by seamlessly adding new features and capabilities.

**Migration to next generation processors**

At the heart of the system is the processor, where several design options are streamlined specifically for infotainment. A select few are powerful enough to handle the demands of an integrated cockpit. The experienced engineer often will invest much of their time in these high-end systems. However, these systems need to be then scaled back into mid-to-low end mass market systems. The software is often ported to a smaller, lower grade processor with less GPIOs, while requiring the need to connect to peripheral devices. This is where the GPIO expander offers flexibility to the engineer. I2C is a common interface that is available through a shared bus in most infotainment systems. The engineer can easily recover the GPIOs needed to control any number of peripheral devices.

Much like mobile phones, these infotainment systems are platform-based. This means that from the low-end to the high-end there is a common base platform design to enable basic functionality. The high-end platform will have add-on peripherals to enable exclusive functionality and capabilities. As these systems become more modularized, it is critical to have a common bus that is able to communicate between different subsystems.

One key advantage of I2C is that this interface standard is a shared bus based on a master and slave protocol. This commonality helps in that the I2C bus can be routed throughout the system, noting that the maximum number of nodes is limited by the address space and the total bus capacitance of 400 pF. One of the most common uses of an I2C IO expander is to control inputs that are located remotely on the PCB. Routing to these peripheral devices can be a challenge. The example in Figure 1 highlights a typical infotainment system, where the control inputs of the peripheral devices are being controlled by the main processor.
**Figure 1** Control inputs are directly controlled by the main processor.

An I2C IO expander can reduce board routing complexity significantly by using a pre-existing I2C system bus. The advantages become apparent as this could result in a reduced number of board layers, which in turn minimizes board manufacturing costs. **Figure 2** represents the same system with an I2C GPIO expander. Instead of routing eight lines from the processor to the peripheral devices, the GPIO expander is attached to the existing I2C bus. This reduces the overall routing area as the GPIO expander is now much closer to the peripheral devices.

**Figure 2** Control inputs are controlled through a I2C GPIO expander.

Another key advantage of this approach is scalability. When an infotainment system is based on a common processor platform, the engineer can choose to add or remove peripheral devices based on the target feature set. To have the most optimal cost structure, the engineer must be able to find a processor with enough processing power to specifically address their target end-product needs. A cost-optimized processor often has a lower pin count, which reduces the number of GPIO pins native to the processor. **Figure 3** is a block diagram using the I2C GPIO expander with a low-end processor.
Figure 3 The low-end processor uses a GPIO expander to control the peripheral devices. To maximize system reuse, the GPIO expander can be easily connected to additional peripheral devices. Figure 4 has an audio digital signal processor (DSP) and load switch that have been added for audio processing and power savings, respectively. Adding these devices is relatively straightforward as no additional routing is required back to the processor, thus, the existing board design can be reused.

Figure 4 The GPIO expander is used to control adding an audio digital signal processor and load switch.

As the engineer migrates to a more advanced core processor, the processor’s I/Os may be connected internally to a single voltage domain. For example, the entire I/O bus is connected to a 1.8V supply (often referred to as $V_{\text{I/O}}$). This bus could be connected to a mixture of 1.8-V and 3.3-V peripheral...
devices, which requires adding a general-purpose level shifter as shown in Figure 5.

Several I2C GPIO expanders will be used to consolidate the I/Os as well as perform the level-shifting function to interface back to legacy devices. In Figure 6, a dual-supply GPIO expander is used to level shift from 1.8V to 3.3V, while also consolidating the GPIOs into a single I2C bus. This introduces another level of flexibility as it frees up the processor’s GPIOs that will interface to devices in proximity to the processor.

Figure 5 The mixed voltage system uses level shifters to translate GPIO voltages.

Figure 6 The mixed-voltage system uses a GPIO expander to level-shift and simplify routing.
Re-emergence of buttons

In the last few years, automotive original equipment manufacturers (OEMs) have attempted to migrate the human machine interface (HMI) exclusively to touch screens, gesture controls, and more. However, consumers still prefer having a physical button to control their infotainment system. Being forced to look down at a touch screen and searching for a button could interfere with a critical moment that requires the driver to either avoid a collision or make an abrupt stop. For example, Ford Motor Co. received complaints about its MyFord Touch user interface. Despite its sleek and clean interface, customers still wanted physical buttons to control the system, so Ford added eight control buttons to remedy this.

When adding these buttons, either back into the system or starting from new designs, GPIO expanders are an efficient way for detecting these button presses. This is especially important in infotainment systems where the core voltages are relatively low voltage (5V, 3.3V, 1.8V). Unlike switches that are used typically for body control, these buttons are not connected to the vehicle battery; therefore, a GPIO expander will suffice in detecting and reporting these button presses to a processor. Figure 7 is a common example of a GPIO expander as it is detecting button presses.

![GPIO Expander Diagram]

**Figure 7** Use a GPIO expander for detecting button presses in an infotainment system.

As these infotainment systems increase in complexity, the host processor may not be part of the display. This is often called a “remote display,” where the display basically is a monitor in the car with a completely separate PCB. Fortunately, I2C is such a common protocol that it is included as either a low-speed, auxiliary signal path (as in HDMI) or a back channel (as in flat-panel display link or FPD-Link). HDMI is the de facto video interface standard in consumer electronics, and the FPD-Link has become the most common video standard being used in automotive displays and digital LCD clusters that can be reconfigured. These remote displays still need physical buttons for volume control or configuration. An I2C I/O expander is the preferred solution for these remote applications.
**Figure 8** This button interface using a GPIO expander on a remote display is communicating to the host processor through an I2C back channel.

**Figure 8** shows a typical system using the FPD-link to send video to a remote display. Using the I2C back channel available in most FPD-link SerDes (serializers/deserializers), the GPIO expander is still able to communicate over the FPD-link to the host processor. This is especially useful if the main system is responsible for the content of both the remote display and the cluster. Additional content (such as navigation directions) can be displayed to either the remote display or to the cluster at a click of a button. The nature of the content displayed can be configured dynamically. Physical buttons give the driver a consistent interface to perform these configuration functions.

**Conclusion**

Designers face many challenges when architecting an infotainment system. Whether it is migrating to a lower cost processor or interfacing to legacy systems, an I2C GPIO expander can quickly and efficiently solve the problem without interfering with the existing system architecture. This unobtrusive way of expansion offers scalability for the infotainment design engineer and fast time to market.

Selecting the most optimal GPIO expander depends on the voltage range needed and the number of GPIOs being connected. The **TCA9539-Q1**, a low-voltage, low-power IO expander, is a fully automotive-qualified I2C-controlled GPIO expander with a supply voltage range of 1.65V to 3.6V. **TCA9539-Q1** allows for successful communication to current generation processors as well as future-proofing your design down to 1.65V. The device allows for 16 bits or channel inputs and outputs that can be configured individually. These I/Os can be used to either supplement the I/Os on the processor and detect button presses, or drive status LEDs. An 8-bit version is also available with the **TCA6408A-Q1**, should the engineer need to supplement a design with only eight I/Os. Additionally, the **TCA6408A-Q1** is able to level-shift utilizing the separate I2C voltage rail (VCCI) and P-port voltage rail (VCCP). This added benefit is essential for mixed-voltage systems.

The integrated cockpit is a feature-rich environment that, if not implemented properly, can cause a plethora of distractions while driving. Given our often long commutes to and from work and our desire to remain tethered to the world, these distractions are amplified with information overload. By offering the familiarity of physical interfaces, the driver can keep their eyes on the road while still having access to these features. Clearly, the GPIO expander offers system designers additional flexibility when scoping their next project.

**References**

Additional information

2. Download the TCA9539-Q1 data sheet.

Also see:

- Assessing high-performance automotive radio tuners for infotainment systems
- Automotive 4.0: The new revolution?
- Selecting the right OS for in-vehicle infotainment