Capturing and processing graphical images requires manipulating data into a form that you can use. This Design Idea describes an imaging system using a USB (Universal Serial Bus) image-capturing system that uses OmniVision’s 640×480-pixel, 8-bit-color OV7660 image sensor. The CY7C68013A-128AXC from Cypress Semiconductor provides a USB interface between a PC and the image sensor (Figure 1). The control software is written in LabView from National Instruments.

To avoid losing data from the image sensor, the system employs a data buffer in the image-processing algorithm. The buffer uses system memory for data storage. The queue ensures that the system will not lose data regardless of how much time it takes to process each row in an image. This technique is useful in measurement systems in which the speed of data acquisition and data processing may differ.

Figure 2 shows the programming flow chart. After the system starts, you must set the driver to NI-VISA (Virtual Instrument Software Architecture), a software layer that provides a common programming interface across many types of measurement instruments and software drivers. Once you set the driver, you can initiate the USB device. LabView provides a driver wizard that helps you to build drivers. The LabView code for this graphic-system design can easily implant USB data transmission and its applications. Request a copy of the LabView code here.

After initializing the USB device, the software allocates system memory in a FIFO (first-in/first out) configuration to become the data buffer. A memory endpoint sets the input buffer’s size to 4 kbytes. The software then reads the image in rows and stores data from the sensor in the buffer memory. After reading the data from the buffer, the system image uses two threads to process the data.

Figure 3 shows the LabView programming diagram for USB data transmission. The program includes for-loop procedures for storing the image in the buffer memory, reading and processing image data, and performing state checking.

The main processing algorithm obtains and displays red, green, and blue data of each pixel. Figure 4 shows the test result. The element in the buffer shows that the system processed 614,400 pixels. The actual amount will vary based on the PC’s performance. A powerful PC can smoothly run this program, whereas a weak PC will cause the data to accumulate in the buffer.