www.maxim-ic.com

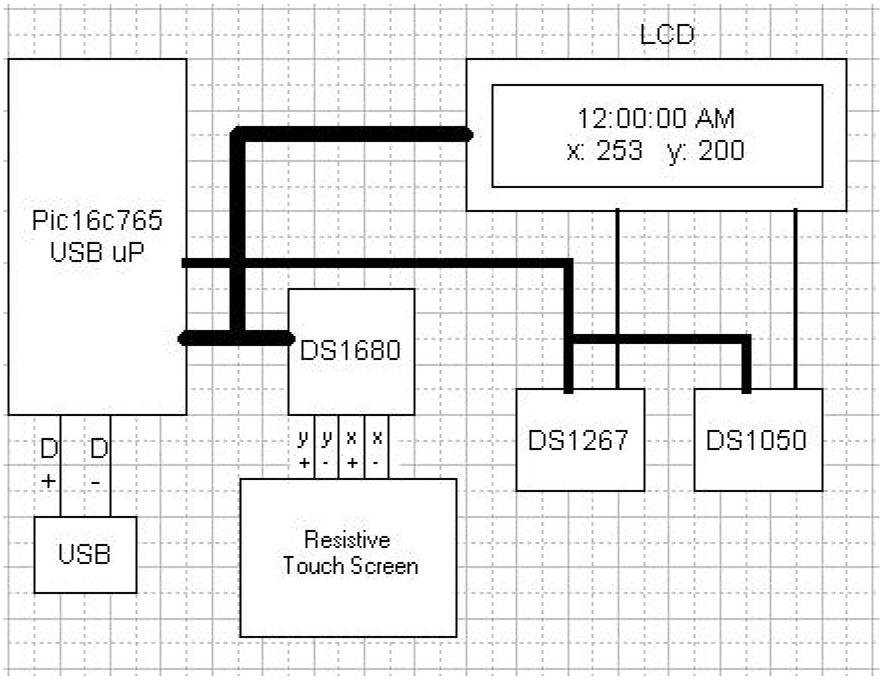
Application Note 513
Integrating the DS1680 Touch-Screen Controller within a USB Device

OVERVIEW

The DS1680 incorporates a touch-screen controller, real-time clock (RTC), a 10-bit analog-to-digital converter (ADC), and other functions needed for low-power products including a power-fail warning and NV RAM controller.

The DS1680 fits well within the Human Interface Device (HID) class of the USB. It can provide user input in the form of mouse functions and as a drawing device for capturing data such as handwriting. This application note demonstrates how to use the DS1680 in a USB system to communicate the X-Y coordinates of the touch screen and RTC information to the USB host. An LCD is used in the system to display the time and touch coordinates locally. The system uses a DS1267 digital potentiometer device to control the contrast of the LCD. A DS1050 5-bit pulse-width modulator is also used for controlling the brightness of the LCD backlit. Through the USB interface, the host controls both the contrast and brightness. The system block diagram shown in Figure 1 shows all of the components used in the system. Following is a description of each of the components shown.

Figure 1. System Block Diagram



DS1680

The DS1680 implements a 10-bit ADC. This converter is multiplexed between two analog inputs (AIN0 and AIN1) and the analog inputs of the touch screen (X- and Y-). A successive approximation technique is used to convert the analog inputs into a monotonic digital output. The output of the ADC is accessed through a byte-wide parallel interface. The DS1680 indicates that data is available when the $\overline{\text{NEW\_DATA}}$ indicator pin is pulsed to a logic 0. To enable data output, the $\overline{\text{COEN}}$ is driven low by the microcontroller. The signals PEN_SELECT and OUT_SELECT are used to decode the wanted data. PEN_SELECT is high to read the touch-screen converted output; when it is low the analog channel is selected. OUT_SELECT should be driven low to output the X coordinate (analog channel 0 if PEN_SELECT low) and should be driven high to output the Y coordinate (analog channel 1 if PEN_SELECT is high). Next, the BHE signal is decoded to select the low or high word of the converted data.

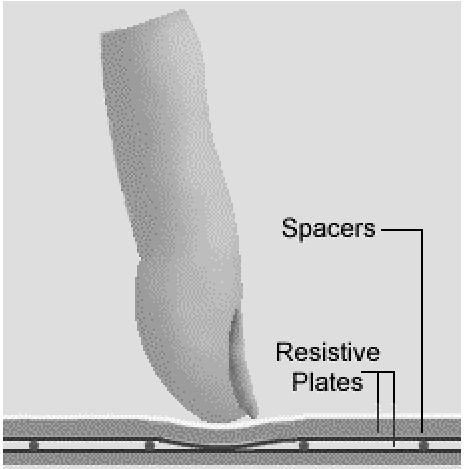
The DS1680 also has an RTC that is used in this system. The clock/calendar is accessed through a 3-wire port. The clock time is shown on the LCD screen and can be read and written from the USB port.

Although not used in this system, the DS1680 supports an RTC alarm, a NV RAM controller, a microprocessor watchdog monitor, and power-fail warning.

Resistive Touch Screen

A 4-wire analog resistive touch screen is used with the DS1680. The resistive touch screen is composed of two plates (X and Y) that are separated by small transparent spacers distributed throughout between the two plates(Figure 2). These spacers prevent the two plates from contacting each other until the screen is touched with a finger or stylus. The interior of the two plates is coated with a thin, uniform, electrically resistive film. When the screen is touched, the X-resistive plate is pushed against the Y-resistive plate making electrical contact. By driving the X-plate and sensing the voltage on the Y-plate, the X-coordinate is resolved. The Y-coordinate is resolved similarly by driving the Y-plate and sensing the voltage on the X-plate.

Figure 2. Touch Screen Diagram



DS1050

The brightness of the LCD is controlled through the DS1050, a 5-bit, programmable, pulse-width modulator that is controlled through a 2-wire interface. Through this interface, the duty cycle can be adjusted from 0% to 100% in 3.125% increments.

For additional information regarding the DS1050 features and functionality, refer to the DS1050 data sheet. Also, there is more information about using the DS1050 for LCD brightness and motor control in *Application Note 163: Using DS1050 Programmable PWMs to Control a DC Motor and an LCD LED Backlight.*

DS1267

The contrast of the LCD is varied with the DS1267 dual digital potentiometer. The DS1267 interfaces through a 3-wire interface. For additional information regarding the DS1267 features and functionality, refer to the DS1267 data sheet.

Pic16c765

The Pic16c765 is a microcontroller that the supports the USB 1.1 low-speed interface. The Pic16c765 was preferred over the Pic16c745 because of 11 more I/O pins. Microchip provides a library of routines to aid in USB development. These routines help with handling device enumeration as well as HID input/output reports. As well as handling USB traffic, the microcontroller polls the DS1680 for new data, displays the RTC and touch coordinates on the LCD, and configures the contrast and brightness for the LCD.

LCD

The system uses an Optrex 16 x 2 character display. The display is configured in software to operate in 4-bit mode.

Firmware

After USB enumeration the firmware enters a loop that does the following:

- Checks if pen is down and X-Y data is available from the DS1680.

  The firmware does this by checking PEN_OFF and $\overline{\text{NEW\_DATA}}$. Two accesses from D0–D7 takes place because the ADC is 10 bits. The 10 bits of data is then reorganized so that they are least significant bit justified within 2 bytes. The data is then sent as a report to the host using 5 bytes (2 bytes for the X coordinate, 2 bytes for the Y coordinate, and 1 byte for pen up/down status).

- Sends a pen-up report if X-Y data is not available.

  This report is only sent once for each consecutive X-Y data stream. A pen-up report consists of 5 bytes where the fifth byte (pen up/down status) is zero.

- Processes any input reports from the host.

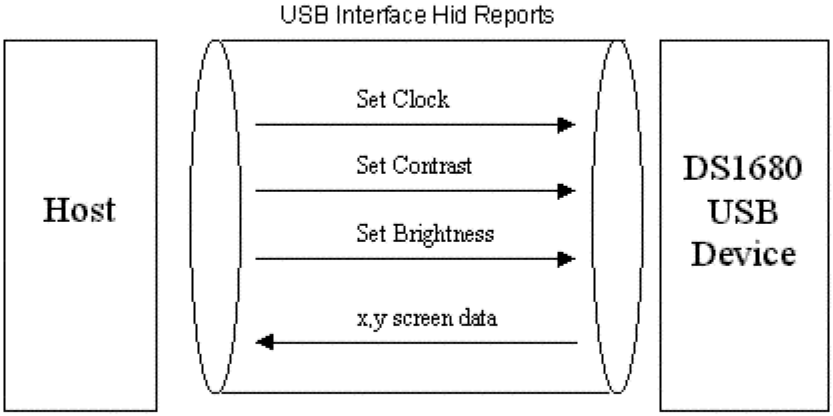
  The USB input buffer is checked for reports from the host. The host can send three requests: set time, set contrast, and set brightness. These reports range in length from 2 to 3 bytes depending on the report received.

- Reads the RTC from the DS1680 and update the LCD if needed.

  A refresh timer is set to refresh the time and date on the LCD screen. When this timer overflows, the new time and data is read from the DS1680 and sent to the LCD.

Figure 3 shows all of the reports in the system and the direction to or from the host. More reports can easily be added to set the RTC alarm, indicate alarm status, or read AIN0/1 analog inputs.

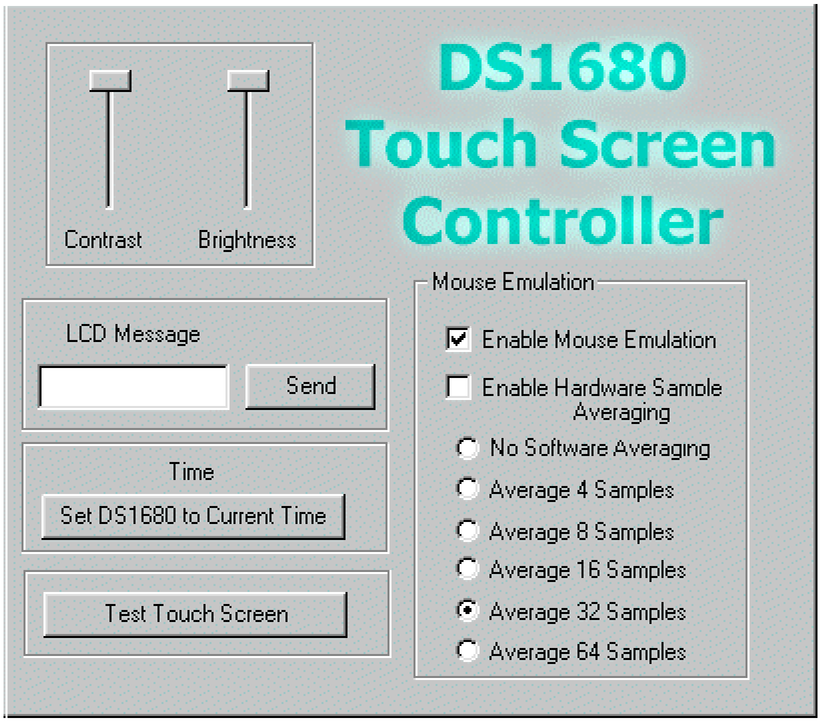
Figure 3. HID Report Flow



Host Software

Figure 4 shows a screen shot of the host software. The program, written in Visual C++ 6.0, provides mouse emulation, software and hardware averaging, time functions as well as contrast and brightness control. Upon startup, the program uses a series of API calls to find a HID device with matching Vendor and Product ID. These calls are wrapped into a useful class called cUsbSupport. The cUsbSupport class is responsible for enumerating the USB device and processing Hid reports. This class is designed to be subclassed and cDS1680_Usb is a subclass of cUsbSupport. cDS1680_Usb adds DS1680 specific members to processes all incoming and outgoing reports. This subclass sends reports to set the time, brightness, and contrast of the hardware and receives all touch-screen reports. cDS1680_Usb class also provides mouse emulation.

Figure 4. Host Screenshot



☑ Enable Mouse Emulation

□ Enable Hardware Sample Averaging

○ No Software Averaging

○ Average 4 Samples

○ Average 8 Samples

○ Average 16 Samples

● Average 32 Samples

○ Average 64 Samples

CONCLUSION

This application note uses an LCD to display info from the DS1680 such as the RTC; if this was omitted then the system is simplified to a two-chip solution (DS1680 and USB controller).

The DS1680 has more valuable features for embedded systems that are not used in this system such as a 1.25V threshold detector for power-fail warning, a microcontroller monitor, and an NV RAM controller for battery backup. The DS1680 also has two analog inputs that are not used in this application note but could easily be used to monitor battery voltage in an embedded system or monitor system temperature.

SCHEMATICS AND SOFTWARE

The schematics and software for this app note is on our ftp site: ftp.dalsemi.com/pub/datalogging/DS1680.