Building a Simple HID Peripheral

2.1 Objectives

In this lab, you construct and modify another low-speed USB peripheral around a PIC16C745 microcontroller. This device is a USB keyboard, which conforms to the *human interface device* (HID) class definition.

2.2 HID Class Devices

The human interface device (HID) class includes many familiar devices, such as mice, touchpads, gamepads, and keyboards. Many other devices are part of the HID class as well, some of which have seemingly very little to do with human/computer interfaces, such as thermometers and voltmeters. HID class devices communicate data to the host through predefined forms, called *reports*. In addition to the default control pipe, EP0, which is required of all USB devices, HID class devices must also posess an interrupt IN pipe through which the host receives input reports from the device at regular intervals. Some HID class devices occasionally receive output reports from the host, which the device will use to modify its behavior in some fashion.

In this lab, you will be building a very basic USB keyboard that, once configured by the host, continually types out `foobar` followed by a space as long as it remains plugged in. This keyboard will also have one key (connected to RA0), which the host will interpret as the CAPS LOCK key, and one status LED (connected to RA1), which the host will update to reflect the status of the CAPS LOCK key. The host polls a USB keyboard at a regular interval (e.g., once every 10 ms). In response, the keyboard reports the state of eight modifier keys and the keycodes of up to six keys that are pressed simultaneously through an interrupt IN endpoint (i.e., EP1 IN). If one of the keycodes reported to the host were the CAPS LOCK key, the host would toggle the state of the CAPS LOCK LED send an output report via the default control pipe (i.e., EP0) that indicates the state of all of the status LEDs on the keyboard. Details of the format of the standard USB keyboard reports can be found on p. 56, p. 60, and pp. 62–63 of the *Device Class Definition for Human Interface Devices Version 1.11*. A table of all of the keycodes on a standard USB keyboard is located on pp. 53–57 of the *HID Usage Tables Version 1.11*. Both doucments are available electronically from the USB Implementers Forum. Links to both documents are on the course web page.

2.3 Building a Simple HID Class Peripheral

Obtain the `lab2.zip` firmware kit from the course web site. Create a new MPLAB project file and insert the source files in the appropriate locations. Build the firmware and program

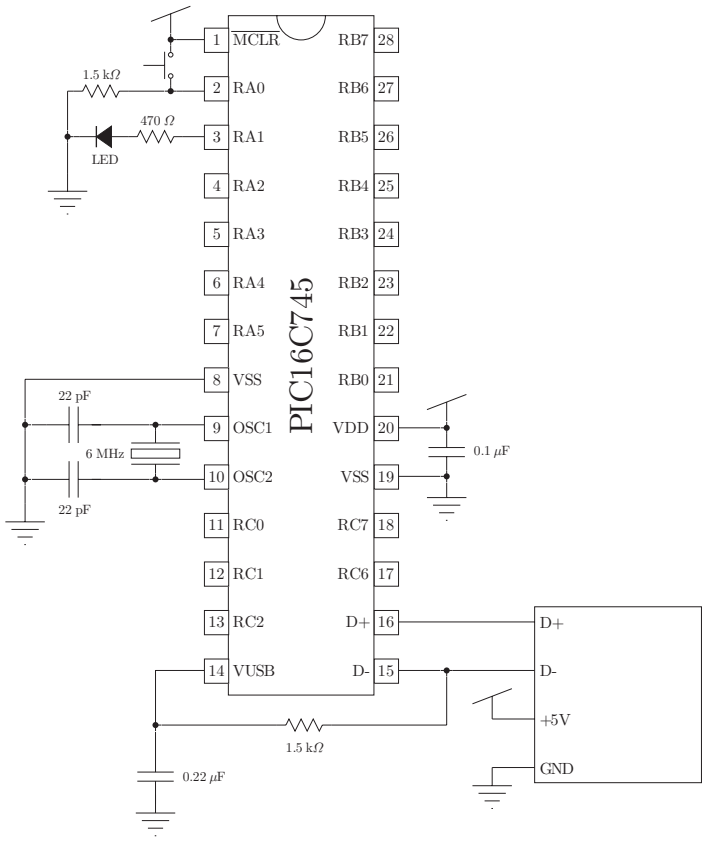

Figure 2.1: PIC16C745 lab 2 circuitry.

a PIC16C745 with it. Construct the circuit around the PIC16C745 shown in Fig. 2.1 in a solderless breadboard. Launch an application, such as Microsoft Word or WinEdt that can capture keystrokes and plug in your newly constructed peripheral. When your peripheral is plugged into a USB port on the host, the host should recognize it as a USB keyboard and it will automatically assign an HID class driver that is appropriate to it. You should see lots of `foobar` in the window of your text editor. If you hit the CAPS LOCK key, you should see the status LED turn on and you should see lots of `FOOBAR` in your text editor.

2.4 Modifying Your Peripheral

Now, you will need to make a modification to the firmware to add a new functionality to your peripheral. You could add another key to one of the PORTA pins, which interrupts the continual stream of `foobar` being sent to the host. You could move the keycode look-up table to a segment of data memory and add a vendor specific request to the firmware to change the keycodes being typed out by the peripheral. If you choose to add a vendor specific request to the firmware, you will have to find the device from your host-side application with the Vendor ID 0x04D8 and Product ID 0x0002; the Product ID for the Lab 1 peripheral was 0x0001.