P. R. GLASSEL AND ASSOCIATES, INC.

Engineering Solutions for Your Ideas

A Level Application & Code: A Demonstration for the Microchip PIC18 Starter Kit

Authored by:
Jesse W. Hartley Senior Engineer 22-Jun-10

P.R. Glassel & Associates, 30255 Fir Trail, Stacy MN 55079
Ph: 952-595-PRGA (7742)
Web: www.prga.com

Table of Contents

1 Purpose & Objectives

This document describes the setup, use and design of a demonstration program that implements a 2-axis level using the accelerometer and display of the Microchip PIC18 Starter Kit. The chip uses the USB communication capability of the PIC18® to provide visibility of the real-time data as it is used to display the "bubble" of the level.

This demonstration is built on the USB Device HID Custom Device program provided in the Starter Kit.

2 Usage

The PIC18 Starter Kit is be powered directly from the USB port of the host PC as shown in Figure 1-1 of the MPLab Starter Kit for the PIC18F User's Guide(Ref: 1). The USB connection also provides visibility of the raw data from the various sensors on the Start Kit board.

1. Connect the application USB port of the Starter Kit board to the USB port of the host PC. See Figure 1-1 of the User's Guide (Ref: 1).
2. When the display comes up, press the Menu switch to start the SD Card Bootloader. For explanation of the Bootloader, see the User's Guide sec. 2.1 (Ref: 1).
3. Load and start the "LEVEL.HEX" file in the Demo folder of the SD Card.
4. On the host PC, run the program "HID PnP Demo.exe" found in the "Demos\USB Device – HID-Custom Demos" directory of the Starter Kit. You will see the data streaming from the Starter Kit board in the "Numeric" panel.
5. Press the menu button on the Starter Kit board to see the successive screens of instructions.
6. To calibrate the level, when directed, place the Starter Kit board on a flat surface such as a desk-top and press the L button on the Starter Kit board. Rotate the board 180° with the board still flat on the desk-top, and press the R button.
7. The display of the level's "bubble" appears along with a reference circle and a center dot.
8. To return to the bootloader, press and hold the Menu button.

3 Program Function

The demonstration application features the following:

- A real time display that simulates a traditional 2-axis fluid "bubble" level using the Starter Kit's onboard accelerometer.
- Level sensitivity similar to a carpenter's level. This requires the use of the maximum sensitivity of the accelerometer.
- Filtering the accelerometer output reduce the inherent noise.

3.1 Hardware drivers

3.1.1 The Accelerometer

The accelerometer is a Bosch BMA150 3-axis accelerometer with a 10 bit output. Some pertinent specifications are:

- ±2, 4, or 8g ranges
- Resolution at ±2g range = 1g/256 = 3.9mg/bit.
- on-chip digital filter with bandwidths from 25Hz to 750Hz.
- The stated noise level is 0.5 mg/√Hz.

In this application, the minimum bandwidth of 25 Hz is sufficient to support the dynamics of the displayed "bubble". Using the 25 Hz filter setting, the expected noise level is of the order of 2-3mg rms. This implies that the rms noise level is less than one bit or ~4mg. Any further filtering will improve this number.

The 4mg sensitivity of the accelerometer translates into an angle of ~14min. A good carpenter's level has a sensitivity of about 1/64" over 3 ft (0.4mm over 1 meter) or ~1 degree. A precision machinist's level has a sensitivity of 80-90 sec. or 0.0005"/ft or 0.04mm/m. We expect that the sensitivity of the accelerometer on the Starter Kit board to be better than a carpenter's level but not as sensitive as a machinist's level.

The level can be calibrated by recording the output then rotating the level by 180° on the same surface and recording the output. The difference between these two reading is used to correct for any offsets due to the sensor or the mounting of the sensor on the circuit board.

The driver function, BMA150_ReadByte(), from the Microchip supplied demos was used to read and format the data from the accelerometer. An initialization function, PRGAInitBma150(), was added to set the accelerometer to use the ±2g range and the 25Hz bandwidth.

3.1.2 The Display

The Level display is intended to emulate the bubble of a circular spirit level. As the demo board (and the attached accelerometer) is tilted, the bubble drifts up to the "high" side. The bubble display is a circle whose center is controlled by the accelerometer.

The OLED display is driven by a Sino Wealth SH1101A display controller. To support the desired bubble display, we used the internal display memory exclusively for the image to avoid using internal PIC18F memory.

An alternative approach would be to allocate a shadow image of the display in PIC18F memory. All of drawing would be to the internal memory image followed by a mass update to the display controller.

To use the controller's internal memory directly, some additional read functions were added (PRGAoled.c). In normal operation, the display controller supports sequential reads where each successive read operation gets the next address of the display. The display controller also incorporates a read-modify-write function where the address is not incremented until after the write/end operation. The read-modify-write behavior is discussed in the SH1101A data sheet.

These additional driver functions for the display were added to support this demo:

• OledReadData()	Read display controller memory at the current location.
• OledRMW_Read()	Read via a Read-Modify-Write controller operation
• OledRMW_Write( )	Write and End a Read-Modify-Write controller operation

The read-modify-write operation was split into separate Read and Write functions to facilitate using a variety of data functions (OR, AND, EXOR) without changing the low level code.

The handshake timing is important in the display controller read functions. The first concern is the delay from the processor write that asserts the RD control to the appearance of RD at the display controller. The timing of the PIC18 I/O Port is given in Figure 29-6 and Table 29-13 of the PIC18F46J50 datasheet (Ref: 4). The important time is that the result of an I/O Write appears on the I/O pin up to 150 nsec after the rising edge at the end of the processor Q4 of the write instruction.

The second concern is the delay in the display controller from receiving the RD control to the assertion of the data. The display controller has a maximum of 140nsec between the assertion of the RD control at the controller and the appearance of the data from the controller.

The third timing value is the instruction time of the processor. In an I/O Read at the processor, the pin is sampled on the rising edge at the end of processor Q1 or 20.8 nsec after the beginning of the read instruction. The PIC18F46J50 of the Starter Kit has an instruction clock of 48MHz. This results in a time of 20.8nsec for each of the four phases of the CPU clock or 83.3nsec/instruction.

To read the display controller, the RD control is asserted and followed by a read of the controller data lines. To meet the timing requirements, the display controller requires that a delay be inserted between the assertion of the RD control and the reading of the display I/O port. In the worst case, this is the delay from processor RD assertion to display controller receipt (150nsec) plus the delay of the display controller responding with the data (140nsec) for a total of: 150 + 140nsec=290nsec, or four NOP(); function calls of 83.3nsec each.

3.1.3 The Menu Switch

The menu switch is used through existing Microchip functions. The switch is used to step through the introductory text screens and to restart the SD bootloader.

3.2 Program functions

3.2.1 Filtering

The goal of the filtering is to minimize the noise or variation in the accelerometer signal and therefore noise in the position of the “bubble” in the level display.

The filter chosen is a simple single-pole recursive infinite-impulse-response (IIR) filter with binary coefficients.

$$\boxed{\mathrm{Out}(t) = \frac{\mathrm{In}(t)}{N} + \frac{(N-1)}{N} * \mathrm{Out}(t-1)}$$

In this application, N=16, so the filter coefficients are powers of 2. This allows the use of shifts instead of full multiplies. The frequency response is -3db @ ~0.01 f_{sample}, although here, the sample rate is not controlled. The sample rate is set by the timing of the USB interface. The recursive or IIR filter structure was chosen to minimize the arithmetic and storage required relative to the roll-off characteristics of the filter. Simple averaging and other finite-impulse-response filters don't provide as low a cut-off frequency for the arithmetic required.

One caution: The C-language standard does not specify the behavior of a right shift operation on a signed negative value, but leaves the behavior as implementation dependent. The Microchip compiler behavior is (Ref: 2, sec. B.4):

> "The value is shifted as if it were an unsigned integral type of the same size (i.e., the sign bit is not propagated)."

A function was written to do the right-shifts on signed integers where the sign-bit is propagated. The function inputs are a 'short' for the number to be shifted and an unsigned byte value for the shift count. In the source code, there are two versions of the same function. These two examples illustrate the effects of the algorithm on code length and performance. The first example makes the sign of the number positive before the multiple shift and then restores the sign. In the second (and shorter) example, we use a C union to allow the compiler to use the PIC18 bit-test instructions to determine if the sign-bit needs to be propagated. The PIC18 does not have a multiple bit shift/rotate instruction, so there isn't a disadvantage to 'unrolling' the C right-shift operator.

With low-pass filters, precision of the internal values is important to their function. Here, we want the filter output converge to a constant value for inputs as small as 1 bit. A little study shows that an additional 4 bits of precision in the internal state of the filter is needed to preserve the 1-bit input. This is convenient since the accelerometer input is 10-bits, so the filter output is kept as a 16x value. The 16x value is reduced to 2x when the filter output is use to determine the displayed "bubble" position.

Visually, the accelerometer-driven 'bubble' needs to move smoothly. The additional filtering would reduce the noise of the accelerometer by about a factor of 10. (The bandwidth is reduced by ~0.01, and noise is proportional to the square root of the bandwidth.)

3.2.2 Graphics

The graphics are simple: a border drawn around the edge of the OLED display, a reference circle and a single pixel at the center of the display. The "bubble" is a moving circle drawn on the screen. All of the graphics are redrawn with each update from the accelerometer.

3.2.2.1 Circle Drawing

The circles for the reference and the moving bubble are drawn using the Van Aken algorithm (ref: 3). This algorithm calculates sequence of X,Y coordinates for an octant of a circle. The starting octant coordinates are mirrored to complete the circle.

This algorithm was designed for use on stepping-motor driven plotters and displays, both inherently integer operations. It is robust and is easily implemented using integer arithmetic.

Since the display is small, the circle drawing function is defined using single byte variables.

4 Implementation Notes

1. To create a project that is compatible with the SD bootloader, see section 2.1.3 in the Starter Kit User's Guide (Ref: 1) for instructions.
2. When debugging applications, the bootloader may be erased from the target PIC18. To restore the boot loader, see the readme.txt in Demos/Bootloader directory of the Starter Kit. There is a prebuilt hex file of the bootloader to restore the Starter Kit to its original configuration.

5 References

1. MPLab Starter Kit for the PIC18F User's Guide, Rev. DS5182A, Microchip Technology Inc., 2009.
2. MPLAB C18 C Compiler User's Guide, Rev. DS51288J. Microchip Technology Inc., 2005.
3. Van Aken, J. & Novak, M., "Curve-Drawing Algorithms for Raster Displays" ACM Trans. Graphics, Vol 4,No. 2, April 1985, pp:147-169
4. PIC18F46J50 Family Data Sheet Rev. DS39931C, Microchip Technology Inc., 2009.

6 Appendix A – Header Files Used

The demo specific files are included in the project. Some header files are Microchip supplied files from the Starter Kit in directories:

Demos\Microchip\OLED driver

Demos\Microchip \Soft Start

Demos\Microchip\BMA150 driver

Demos\Microchip\mTouch

Demos\Microchip\Include