Obtaining Data from Ping $$\text{Ping } \text{ Ping}$$ Ultra-Sonic Distance Sensors with Parallax Propeller Microcontroller

By Sean Stewart-Moore
March 30th, 2015
ECE 480 Team 4

Abstract

Ultra-sonic sensors are used widely in many applications that are commonly used today. Ultra-sonic sensors can detect objects at certain distances, in which, these distances can be used for computing purposes. The purpose of this application note is to educate how to assemble and establish communication with Parallax’s Ping $$\text{Ping } \text{ Ping}$$ ultra-sonic distance sensors to their Propeller microcontroller board. An introduction to the microcontroller and sensors are given first, followed by key aspects in establishing communication. This application note ends by explaining how to obtain distance readings within the Simple IDE software.
**Table of Contents**

Abstract........................................................................................................................................... 1

Table of Contents .................................................................................................................................. 2

Introduction ............................................................................................................................................ 3

Assembly ............................................................................................................................................... 4

Figure 1: Ping )))) Pin Layout.................................................................................................................. 4

Figure 2: Sensor Assembled to Propeller Microcontroller................................................................. 4

Implementing the Propeller GCC Simple IDE Software ................................................................. 5

Figure 3: Add Simple Library Button.................................................................................................. 5

Figure 4: Library Navigation Setup.................................................................................................... 5

Obtaining data from Ping )))) Ultra-Sonic Sensors ........................................................................ 6

Establish Connection via USB............................................................................................................. 6

Figure 5: USB Interface........................................................................................................................ 6

Implementing the C Code ................................................................................................................... 7

Figure 6: Ping))) Distance Reading C Code....................................................................................... 7

Build and Run..................................................................................................................................... 8

Figure 7: Parallax GCC SimpleIDE Task Bar..................................................................................... 8

Figure 8: Terminal Distance Readings ............................................................................................... 8

Conclusion .......................................................................................................................................... 9

References .......................................................................................................................................... 9
Introduction

Ultra-sonic sensors are great components in detecting objects at a distance. It’s very valuable to know how far objects are for many applications and products. For example, the application in cars that alerts drivers when they’re close to hitting an object while in reverse. In this application note, we will be focusing on the implementation of the Parallax’s Ping))) ultra-sonic distance sensor and Propeller Activity Board microcontroller. Much of this information is applicable to all of the Propeller series of microcontrollers by Parallax. Parallax also provides an Integrated Development Environment (IDE) specifically designed for programming with their Propeller series of microcontrollers.
Assembly

The first step that is needed to be done is the assembly of Ping))) sensor and the Propeller microcontroller. The ping sensor has 3 pins: ground, 5V power supply, and I/O signal. As you can see in figure 1, the left pin is used for grounding and the middle pin is used for an input of 5V for powering the sensor. The right pin acts as an input and output. This pin is used to trigger the sensor to active and let the transmitter send out a wave. Once the receiver on the sensor receives a feedback, it outputs a distance digitally. To assemble the Ping))) sensor to the Propeller microcontroller, first insert the three pins of the sensor into the assembly board located on the micro controller as shown in figure 2. As you can see on the top figure, the activity board has two power rails of 3.3V and 5V. Since the sensor needs 5V to be powered on, connect the middle pin to the 5V power rail. As you can see in the bottom left figure, there is a ground rail. The next step is to connect the left pin on the Ping))) to the ground rail on the activity board. Lastly, for the I/O signal pin, place a 2 k-ohm (red-black-red) between the pin and any port(PX/P1X) that’s located on left of the activity board. In figure 2, the I/O pin is placed in port 15. This concludes the assembly of Parallax’s Ping))) ultrasonic distance sensor and Propeller microcontroller.
Implementing the Propeller GCC Simple IDE Software

The next step in the implementation is to configure the Propeller GCC Simple Integrated development Environment (IDE). The software can be downloaded on Parallax’s website. The IDE is very user friendly as it’s available to numerous platforms, such as Mac, Windows, Linux, and Raspberry Pi.

After downloading and installing the Propeller GCC IDE, open up the program and locate the blue *Add Simple Library* button. Click the button as shown in figure 3. From there, scroll down the “Simple Libraries” folder and navigate through the “Sensor” folder and select the “libping” folder as shown in figure 4. After clicking the Select Folder button, the IDE will add a `#include` directive for your library and to your code and make the necessary additions to the Project. More information adding and creating libraries can be found starting on page 25 in the SimpleIDE User’s guide.

![Figure 3: Add Simple Library Button](image3)

![Figure 4: Library Navigation Setup](image4)
Obtaining data from Ping )) Ultra-Sonic Sensors

Establish Connection via USB

After getting the Propeller GCC IDE updated and having all the parts assembled, it’s now time to establish an interface between the software and hardware. This can be achieved by using an USB-to-USB-mini cable. The Propeller microcontroller can be powered by a 5V USB connection. To establish communication, plug the USB side of the cable into the computer’s USB com port and plug the USB-mini end of the cable into the Propeller microcontroller as shown in figure 5. More description about interfacing with the microcontroller can be found in the Parallax Propeller User Guide.
Implementing the C Code

To implement the code, the `simpletools` and `ping` headers have to be included so that program has the ability to use the functions in those libraries. In figure 6, the `main()` function starts with a repeating while loop. On line 16, the variable `sensor_1`, is created and is equal to the function `ping_cm(15)`. This function allows the user to obtain a distance reading in centimeters by placing the port number that is connected to the signal pin on the Ping))) sensor inside its parameters. On line 17, the print function is used to display the distance reading as the next line allows the distance readings to be displayed every 200 milliseconds.

```c
/*
 * Test Ping Distance.c
 * Measure and display Ping))) Ultrasonic Distance Sensor distance measurements.
 *
#include "simpletools.h" // Include simpletools header
#include "ping.h" // Include ping header

int main()
{
    // main function
    while(1)
    { // Repeat indefinitely
        float sensor_1 = ping_cm(15); // cm to ft conversion
        printf("sensor_1 = %.2f\n", sensor_1); // Display distance
        pause(200); // Wait 1/5 second
    }
}

Figure 6: Ping))) Distance Reading C Code
```
Build and Run

The next step in implementation is to build the code and run it. To build and run the code the microcontroller must be on and connect via USB or errors will occur. In figure 7, the farther button on the right, is the Build and Run button. Clicking this button compiles the code and places program into the RAM of the microcontroller. If there are any errors within the program, the complier will not compile the code and indicate where errors occur. If the code is written correctly, a terminal will pop up on the computer were the distance readings are displayed as shown in figure 8.

Figure 7: Parallax GCC SimpleIDE Task Bar

Figure 8: Terminal Distance Readings
Conclusion

The Propeller Ping )) sensor and Propeller microcontroller are great for determining and analyzing distance as they work very well together. Parallax's IDE gives its users great libraries that create such great interface between the two components. This makes Parallax's Ping )) sensors and microcontrollers highly recommendable for determining and analyzing distance.

References


