**Authors:** Harshdeep Singh Chawla ([hxc3427@g.rit.edu](mailto:hxc3427@g.rit.edu) )

Saloni Mahendra Jain([sj7856@g.rit.edu](mailto:sj7856@g.rit.edu))

**Overview**

The objective of this project was to control the position of the ping pong ball in the tube by controlling the duty cycle of the CPU fan attached at the bottom of the tube. We tried to do this by writing the code in C++ using QNX.

**Analysis/Design**

The project consists of a ping pong ball, a standard CPU fan and an ultrasonic sensor. The idea was to maintain the ball at some pre-defined levels in the tube. We used the ultrasonic sensor to collect data regarding the level of the ball in the tube and this data was then analyzed and used to decide the duty cycle of the PWM signal that was applied to the fan. The width of the signal is to be varied always since the height of the ball depends on the air bellow it. Now if the duty cycle is at low for long time the air below the ball will decrease and the ball will fall. And if the width of the signal is long for longer time, the air bellow the ball in the tube will increase hence moving the ball upwards.

**CPU Fan:**

The CPU fan is a PWM controlled fan. It has four pins +12V, GND, sensing and control pin. The is switched on by connecting to the 12V DC power supply and its control pin is connected to the output pin (port C pin 2) of the QNX from which the control signal is obtained by varying width of the PWM signal. The sense pin is not used in this process.

**Ultrasonic Sensor:**

The ultrasonic sensor has four pins a Vcc of +5V, gnd, trigger and echo. The sending and retrieval of the ultrasonic sensor relies on the timing of ‘Triger’ and ‘Echo’ signals.

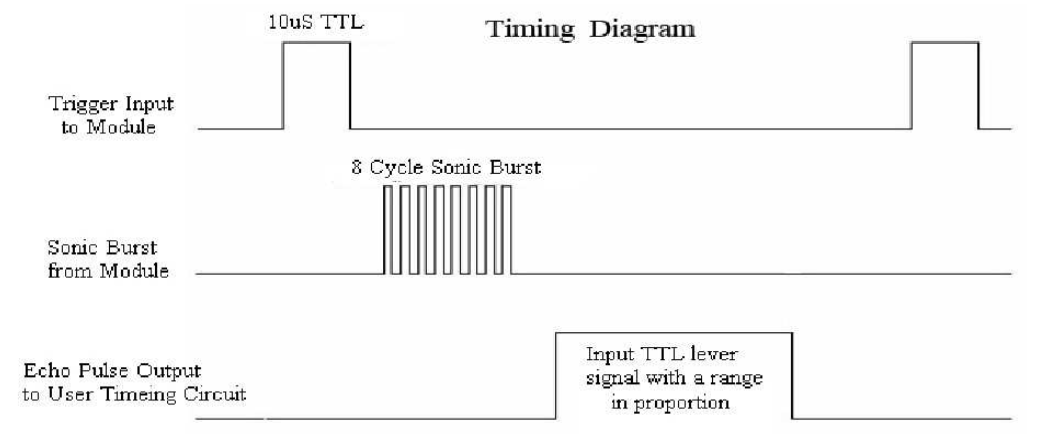


fig: Timing Diagram for HC-SR04

Figure 2 shows the timing diagram of the sequence of operation for HC-SR04. The sensor is activated by a triggering a pulse of which is greater than 10us on the ‘Trig’ pin. The sensor module emits 8 cycle sonic bursts and sets the echo pulse to high. When the trigger is detected back as the burst bounced back because of an object and returned, echo pulse is set to a low. Therefore, the distance of the object is proportional to the pulse width of the pulse seen on the echo pin.

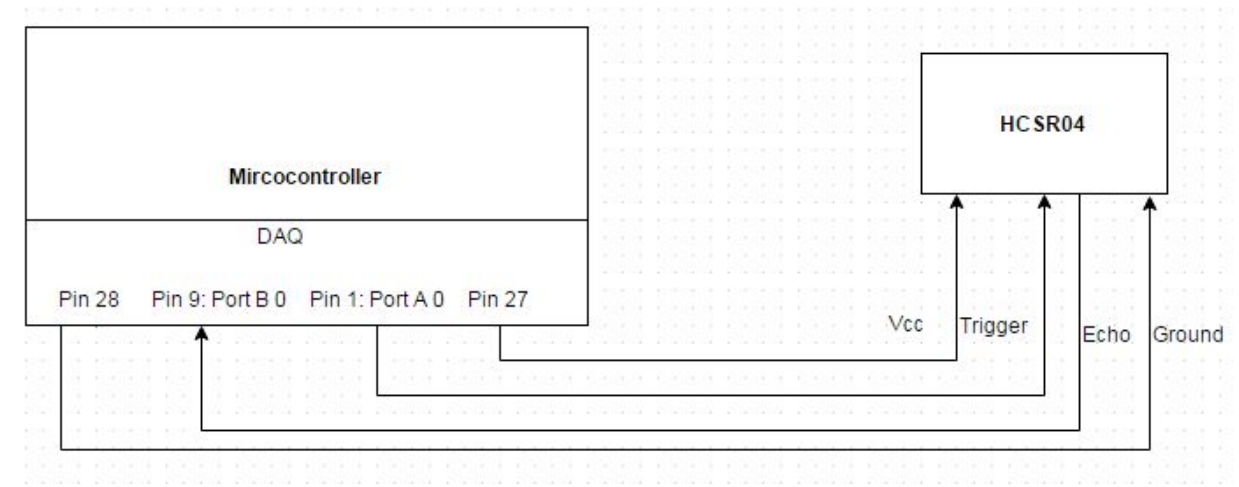


fig: schematic diagram of connection of QNX to ultrasonic sensor.

**Test Plan**

We tested the PWM signal generator part and the Ultrasonic sensor part of the code separately. We did so by varying the duty cycle of the PWM signal generator and then verifying the output using the Oscilloscope. As for the Ultrasonic sensor, we used a measuring scale to verify if the distance displayed on the console is the same as the one obtained by measuring it physically.

The next step was to combine both the codes and then run them.

**Conclusion/Lesson learned**

This project was very interesting and taught us a lot. Initially as we tried to implement this project on the PSoC 5LP we faced a few problems. As we tried to work our way through the old problems we realized that the ground pin of the board was not functioning properly and since we needed to use both the ground pins provided on the board. Furthermore, due to some yet unknown reason one of the PWM pulses that were generated using the PSoC was providing a very low output voltage and so we had to drop the idea of using the PSoC 5LP.

We then moved on to the QNX box for implementing the project. Here we faced a different set of problems. Initially there was an issue with the connectors that we were using after which we faced a major problem. This problem was that our code for the PWM and the Ultrasonic sensor were working perfectly when ran separately however, when combined into a single code they were not running properly.

We tried to debug the code and saw that both the threads were being created. But since QNX is not a bare-metal system, it was hard to synchronize the two threads together.