Discussion:

In this part, we want to discuss what we had done with the project. First, we were working with the simulation portion. In the simulation portion, we implemented the MCU, Sensor, LCD display, Oscillator and Capacitor successfully. But our main goal is to connect the buzzer which will indicate us. At first, we connected the buzzer’s VCC pin with the pin no. 37 of the MCU, but the buzzer didn’t working properly. So, we used a PNP transistor which would fixed the problem. We also connected the LCD display with the microcontroller and sensor. We used a POT to reduce the brightness of the LCD but there are some issues with the POT, that’s we remove the POT from the simulation. We connected the VEE port directly onto the ground. There was no option to reduce the brightness.

We burn the code in the MCU using the pickit 3 software. I did not face any problem at time of burning. After burning, we were going to implement the project on the breadboard. Then, we took a breadboard. Then we put the PIC Microcontroller 16F877A and gave the connection of the MCU with the VCC. We used ARDUINO UNO as a power supply for the circuit. We connected the pin no.1 of the MCU with the ARDUINO UNO’s 5V VCC connection.

Next, we connected the pin no. 13, 14 with a crystal oscillator of 8MHz. The Oscillator works as a clock. Next, we connected the Ultrasonic Sensor with the MCU to interact with each other. As, we know that there are four pins in the sensor. We power up the sensor after connecting VCC. We also connected the GND pin with the common ground of the breadboard. Then, I connected the pin no.33 of the MCU with the trigger pin of the Ultrasonic Sensor. I also connected the pin no. 37 of the MCU with the Echo pin of the Sensor. The MCU established a connection through pin no.33 with Trigger pin which produce 10 microsecond pulse for ranging. After finding any obstacles, the Echo pin produced a signal as an output. So, we need to show the output signal, that’s why we used a buzzer for visualizing the signal.

Next, we took a buzzer and connected one portion with the pin no. 37 of MCU and another portion was connected with the common ground. We also connected the 16\*2 LCD display with the microcontroller to see the output but here we faced some issues. In the display, we connected the pin of LCD (D0-D3) with the common ground because we did not need this ports. We connected the pin (D4-D7) of the LCD with the pin of the microcontroller (27-30). We connected the RS, E pin of the LCD with the microcontroller 21 & 22 no. pin. W connected the anode pin of the LCD with VCC and cathode pin with the common ground. We took a POT which helped us to reduce the brightness of the LCD and helped us to consume less power. I connected the VDD pin of the LCD with the VCC pin of the POT and VSS pin of the LCD with the ground pin of the POT. The Connection of the POT was going to the common VCC and another portion on the common ground. After connecting the POT, the display didn’t work properly, so we removed the POT and tried to use the display directly. The display was light up but there were nothing output on the display. We also tried with the buzzer but the buzzer was not working properly. It would produce some error signal.

Our project didn’t run on the first submission. So, we tried with different microcontroller chip and burn it with the same code. But it didn’t run. So, we changed the oscillator and tried with 20MHz, 16MHz, but it didn’t work. Next, we connected LEDs with three VCC to check, if the power has been distributed. The MCU was getting powered because the LEDs were glowing. Then, we also checked our echo and trigger pin from the sensor to the MCU. Then, we suspected that our oscillator’s vibration was not coming to the MCU through the wires. So, we connected our 8MHz oscillator directly to the MCU pins. And then, we connected one LED to the output pin of the buzzer. Then, we also change our code for finding distance. But, we do not need to measure distance. By measuring the distance, we were setting a range so that our buzzer/ LED give us desired output. Then, we burn the code using pickit 3 burner. This time, the LED was blinking when it found an obstacle. So, we connected the buzzer and also the buzzer was working.

We also decrease the range of the ultrasonic sensor from 400cm to 100cm using the code and burn it again. We change the range to 100cm because if we want to implement it for blind people or car parking or finding nearby obstacle. Now, we can test the sensor for nearby obstacle. If any obstacle is in front of the sensor within 100cm, it will beep and show “Obstacle” on the LCD and if there is no obstacle then buzzer won’t sound and LCD will display ‘’Out of Range”.