



IMPLEMENTATION OF SENSOR INTERFACING USING ARDUINO

Ayush Kagane
Second Year, Mechanical
Engineering,
Vishwakarma Institute of
Technology, Bibwewadi, Pune,
Maharashtra,

Jason Shaji
Second Year, Mechanical
Engineering,
Vishwakarma Institute of
Technology, Bibwewadi, Pune,
Maharashtra,

Abstract— There are a lot of data security issues while coding in Arduino as the variables are publicly accessible. For a system linked with IoT, data and time-to-time updating of sensor variables is important. This can be done by creating classes and objects. This project involves executing different codes in Arduino IDE. Different concepts of OOP have been applied in this project. Different classes and objects for coding in Arduino IDE and Ardublockly have been declared. These codes are used for Sensor interfacing with O/P devices. Hardware setups and output videos are created for viewing the output of the code.

Keywords— Arduino, Object Oriented Programming, Sensor Technology, Microcontroller Programming, DIY Electronics.

I. INTRODUCTION

Arduino is coded in C/C++ languages. So, by default it supports OOP approach i.e.; it supports the existence of class, objects, data members, member functions. The current technology deals with traditional C programming where there is top-to-bottom execution and the accessibility of variables is public. But the main aim of the project is to create data secure codes and different methods or functions can invoke or access the variables.

Keeping this in mind, different Arduino codes using the concepts of OOP have been created for the project. These codes can interface various sensors and output devices namely, IR sensor, Ultrasonic sensor, Soil moisture sensor, RGB LED, addressable LED, etc.

The rest of the paper is organized as follows. Project algorithm, Class diagram/Flowchart of project are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED ALGORITHM

A. Project Algorithm –

Here, coding in Arduino is done using basics of OOP language. Initially, a class is created and then data members and member functions belonging to respective classes are defined and initialized. To make the program easily understandable, more members functions can be used. They can be divided into Input, Process, Output. This is the general format in which the codes have been developed for the project.

Step1: - Start.

Step2: - Create a class.

Step3: - Declare the variables for storing sensor values.

Step4: - Declare member functions for the processes to be carried out.

Step5: - Declare object to the class globally.

Step6: - Invoke the member functions through the objects in the void loop () function.

Step7: - Stop.

(Also note that this Algorithm is applicable to all the codes created for the project).

B. Class Diagram/Flowchart of the Project –

The Class Diagram contains short description about each program along with names of sensors and output devices used for the respective programs. Detailed information about each program is given in section III.

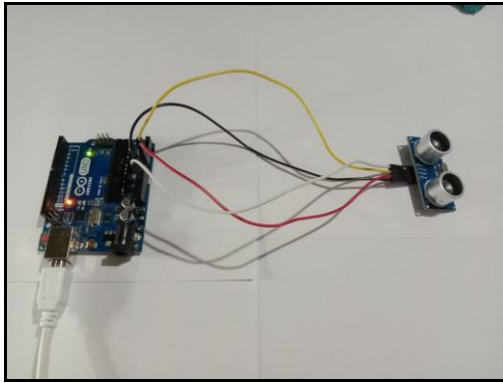


Fig. 7. Hardware Setup of Ultrasonic Sensor

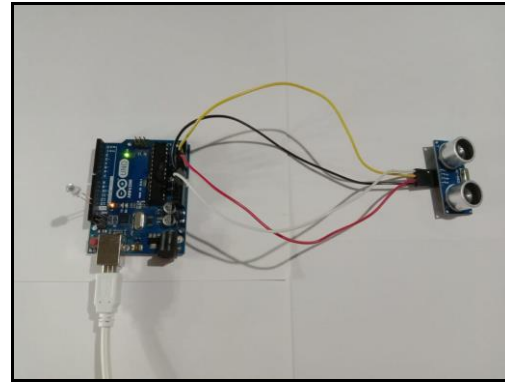


Fig. 10. Hardware Setup of Ultrasonic sensor with LED debounce

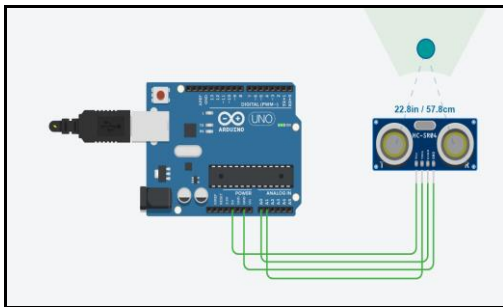


Fig. 8. Tinker CAD circuit of Ultrasonic Sensor

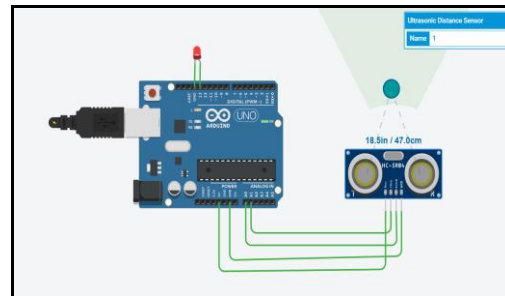


Fig. 11. Tinker CAD Circuit of Ultrasonic sensor with LED debounce

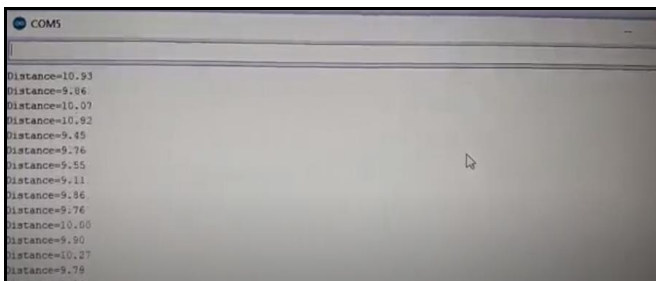


Fig. 9. Output of distance on the Serial Monitor

(Ultrasonic with LED Debounce): - The project consists of Arduino code for interfacing Ultrasonic sensor with a 5V LED. The frequency or blinking speed of the LED changes according to the distance between the Ultrasonic sensor and the obstacle. When the distance between obstacle and Ultrasonic sensor is between 10 and 20 cm, the LED will blink slowly. And when the distance becomes less than 10 cm, the LED will blink rapidly.

(Soil moisture with RGB LED): - The project consists of Arduino code for interfacing Soil-moisture sensor with RGB LED. The RGB LED will turn Green on dry soil condition and it will turn Red on wet soil condition. On the serial monitor the maximum output is $1023(2^{10}-1)$. For dry conditions it shows the reading >600 and for wet conditions it shows the reading <600 .

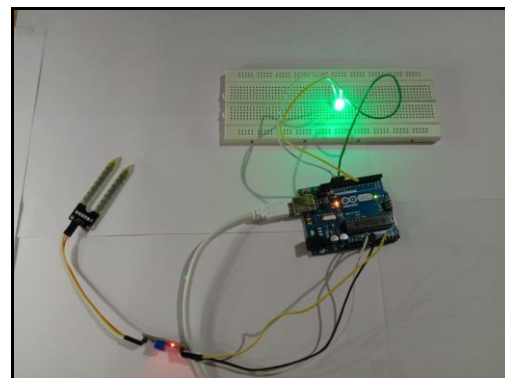


Fig. 12. Hardware Setup of Soil moisture sensor

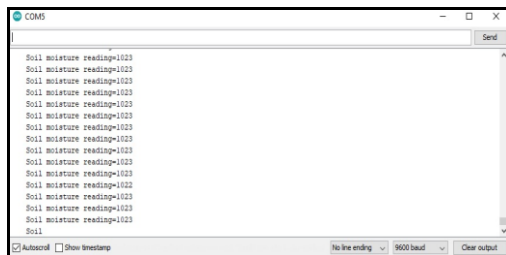


Fig. 13. Output of dry condition of soil

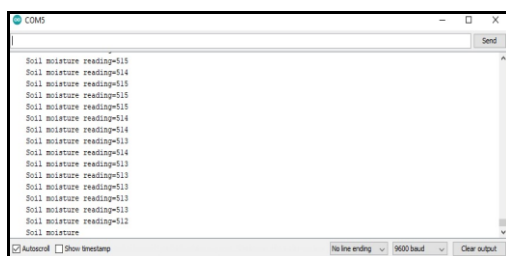


Fig. 14. Output of wet condition of soil

(Addressable LED strip): - The project consists of Arduino code for interfacing WS2812B Addressable LED. The main motive behind creating this code is to show the traditional way of programming in Arduino, that is, with the help of well-defined libraries. The library used for Addressable LED Strip is <FastLED.h>.

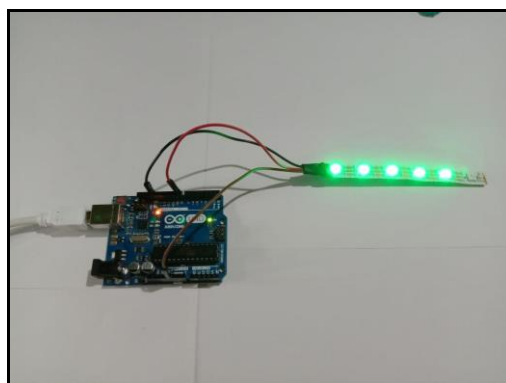


Fig. 15. Hardware Setup and Output of LED Strip

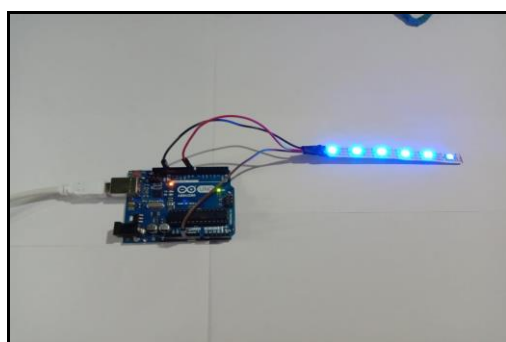


Fig. 16. Hardware Setup and Output of LED Strip

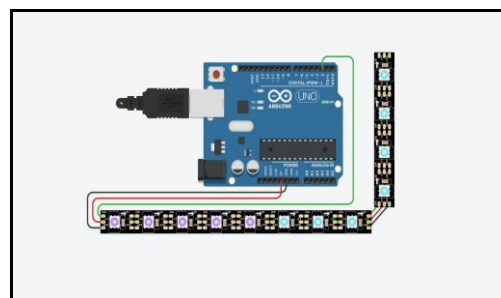


Fig. 17. Tinker CAD Circuit of LED Strip

IV. CONCLUSION

- All the concepts of OOP have been applied to code in Arduino IDE.
- The output of the code can be seen from the hardware setup connected to Arduino Uno.
- In case of Ultrasonic, Soil-moisture and IR Sensors, output can also be checked from the Serial Monitor present in Arduino IDE.
- The output for each code is seen with the help of Videos, Live demo and Serial Monitor.

V. REFERENCE

- [1] Akshey Sharma, Amit Saxena, Govind Varshney: "Arduino- A Brief Review", IJSRMS, Volume 2 Issue 11, pg.: 445-451.
- [2] Leo Louis: "Working Principle of Arduino and using it as a Tool for Study and Research", IJCACS, Volume 1 Issue 2, April 2016.
- [3] Kuldeep Singh Kaswan, Santar Pal Singh, Shraddha Sagar: "Role of Arduino in Real World Applications", IJSTR, Volume 9 Issue 1, January 2020.
- [4] Abdulrahman Alkandari, Areej Alajmi, Sarah Muhammad, Altaf Alshammari: "Ultrasonic sensor gloves for blind people using LilyPad Arduino", IJNCAA, Volume 6 Issue 1, 2016.
- [5] Ankit Kumar, Abhinav Kumar, Dharendra Kumar Verma, Faisal Mansoori: "Arduino Uno based Automatic Plant Watering System", IJSRMS, Volume 2 Issue 12, pg.: 487-492.
- [6] Bakare B, Oguichen T. and Orike S. 2020: "An Improved Microcontroller Based Automated Home Using GSM Phone", European Journal of Electrical Engineering and Computer Science. 4, 4 (Aug. 2020). DOI:<https://doi.org/10.24018/ejece.2020.4.4.171>.
- [7] Joseph Benin and Avinash Srinivasan: "Transition to OOP: An Innovative and Hands-On Approach to Linux,



OOP, Arduinos using EMOro”, US Coast Guard Academy, March 25, 2019.

- [8] Paul Murray: “Arduino the Object Oriented” Way-
<https://paulmurraycbr.github.io/ArduinoTheOOWay.html>
- [9] Bill Earl: “Multi-tasking the Arduino- Part1”-
<https://learn.adafruit.com/multi-tasking-the-arduino-part-1/a-classy-solution>, November 2014.
- [10] RoboticsBackend: “Hardware & Software Tutorial” -
<https://roboticsbackend.com/arduino-object-oriented-programming-oop/>