DIGISCALE – CONTACTLESS DISTANCE & ANGLE MEASUREMENT DEVICE

Mr. Swapnil Sanjay Bafana[1],
Anarya Shailendra Sonawane [2], Siddhi Sarika Shimpi[3], Vineet Rajesh Wankhede[4]

[1] Lecturer, Department of Electrical Engineering, R. C. Patel Polytechnic, Shirpur-425405, Dist-Dhule Maharashtra, India
[2] Polytechnic Student, Department of Computer Science Engineering, R. C. Patel Polytechnic, Shirpur-425405, Dist-Dhule, Maharashtra, India
[3] Polytechnic Student, Department of Computer Science Engineering, R. C. Patel Polytechnic, Shirpur-425405, Dist-Dhule, Maharashtra, India
[4] Polytechnic Student, Department of Computer Science Engineering, R. C. Patel Polytechnic, Shirpur-425405, Dist-Dhule, Maharashtra, India

ABSTRACT

The project DigiScale is actually self-explanatory. It is a device which can be used by all sorts of engineers, workers and craftsmen of all sorts. It is a device that can measure two of the most important quantities namely Distance and Angle. Measurement technology is an important part of our everyday life, and the limits of what can be measured are increasingly stretched by utilizing various chemical and physical characteristics of material, from nano scale, for the development of transducers, sensors and instruments. Old methods rely heavily on the user to measure the quantities and can be affected by human error. They also lack in accuracy and reliability. Thus, the usage of modern electro-computational devices like "DigiScale".

Keyword: - *Sonar, Gyroscope, OLED, Microcontroller, IIC/I2C, C++.*

1. INTRODUCTION

A specialization in measurement technology requires a good knowledge of the physics behind the different measuring principles, as well as insight into the process to be measured. The DigiScale project is developed in C++ language for the software and arduino based microcontroller to be used as a hardware interface with some additional libraries for, gyroscope and OLED display [1].

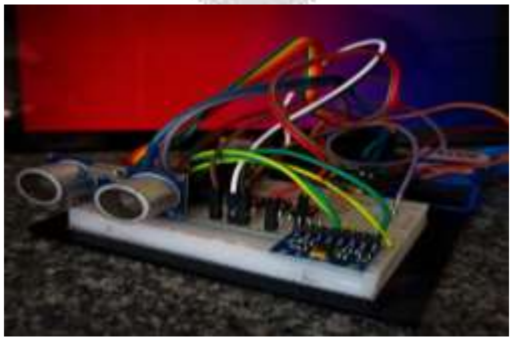
Fig-1 Photo of actual prototype

1.1 Sonar

In order to make the distance measurement possible, we are using sonar technique which is also implemented in submarines and ships to measure water depth and also biologically seen in animals like bats for proximity sensing. It basically uses Ultrasonic waves; these are sound waves beyond our audible range. Thus, we cannot hear these sound waves. There's usually a transmitter and a receiver on such sensors. Transmitter produces ultrasound waves; these waves bounce off the object in front of the sensor. These waves are detected by the receiver. Then the time taken between transmission and detection of the waves is used to calculate the distance.

1.2 Gyro

A gyroscope or simply gyro is an electromechanical device that measures the change in orientation. It uses a rotating mass centered in a Gimbal. A rotating mass is resistive towards changes in rotation and a Gimbal lets the mass rotate in all 3 axes freely. Hence the mass stays in the same place even when the sensor/device is rotated. Hence, we can measure the change in rotation by measuring the rotation on the gimbal's axes.

1.3 Accelerometer

An accelerometer is another piece of technology also used in our sensor to correct the inaccuracies in the Gyro. An accelerometer basically measures acceleration and angular velocity by measuring the changes in centrifugal force. Basically, it cannot measure the orientation or angle of rotation like Gyro, but it can measure the speed/velocity of rotation. Even though this one isn't directly useful to us since we don't want to measure the angular velocity. But it can help us correct the readings from the gyro. There's a certain drift to the rotating mass on the gyroscope since the mass itself cannot fully resist the rotation, because just like everything in the world, it has a practical limit. Also, the gimbal's axes aren't 100% frictionless and these two things will cause the mass to move a bit with the rest of the device if rotated at higher speeds. This will cause a certain bias/inaccuracy in the readings. But, since the amount of inaccuracy changes depending on the rotation speed (Rotating carefully will cause less bias, faster will cause more bias), we can use the data from the accelerometer to determine the rotation speed (angular velocity) and use it to correct the Gyro's motion data.

1.4 OLED

To display all of this information, we are using an OLED (Organic Light Emitting Diode) display. Briefly, this kind of display uses RGB LEDs as pixels instead of the more electrochemical functioning of an LCD panel. These OLED pixels are made multicolor using 6 Organic semiconductor sheets sandwiched on an ordinary LED. The advantage of using an OLED is that the pixels are individually lit. Unlike an LCD where there's a common backlight behind the panel. A traditional LCD causes a lot of problems like backlight bleeding which makes the screen look awfully washed out and unclear because even the parts that are not used are lit. It also causes problems like bad viewing angles which means the text on the screen might disappear when viewing from an oblique angle. Also, at this size, LCD panels don't provide much pixel density due to this, the text appears pixelated.
These might not be huge problems for the TV or PC monitor in your room. But our device won't be in a fixed place on a wall and probably not in an indoor condition. It might be used on a construction site or any other workplace. In that case, visibility might already be low due to dust particles flying around pollution from other equipment and maybe even a glaring sun from above if it's outdoors. And a washed-out, low contrast and pixelated display will only make things worse. Also, the device is supposed to be put in many corners and hard to reach positions, so the user certainly won't be viewing from a perfectly straight angle, so good viewing angles are a must. Not to mention, most of the LCD panels look impossible to use in the sun. So having good contrast and brightness is also important.
An OLED screen will fix most of these problems. Unlike an LCD which has a chemical based function that can be easily disturbed by most of the environmental factors, an OLED is fairly simple, there's an LED for every pixel, it lights up and you see it simple.

1.5 Microcontroller

A microcontroller is an embedded circuit that can govern or perform very specific programmed operations. Kind of like a very purpose-built computer. We are using one to process all of this data and display it on the OLED. Basically, anything that happens after taking input from sensors to printing info on the display happens on the microcontroller.

1.6 I2C communication

I2C stands for the Inter-Integrated Controller. This is a serial communication protocol to connect low-speed devices. It is a master-slave communication in which we can connect and control multiple slaves from a single master. Here the slave is our gyro+accelerometer and the OLED and master is our microcontroller.

2. LITERATURE REVIEW

During the 18th century, the electro-optical distance meter's development has evolved through the techniques of determining the velocity of light. Fizeau [2], who determined the velocity of light in 1840s, and a lot more inventions; E. Bergstrand was then inspired to design the first "Geodimeter" in 1940s [3]. Among the various techniques of noncontact measurement, ultrasonic technique is the best (U. Grimaldi et al., 1995 and P. Purnell et al., 2004) [4] when we need stable and accurate distance of obstacle, no matter what colour it is. It is also usable outside in the Sun.

Heading towards the need of angle measurements Geometry is linked to many other topics in mathematics, including measurement. Angle is an important concept in geometry and in the study of measurement. Measurement is "the process of assigning a number to a magnitude of some attribute of an object, such as its length, relative to the unit" (Clements & Sarama, 2009, p.163) [5]. Understanding angle concepts requires the apperception of the physical properties of angle, including the static (configurationally) and dynamic (moving) aspects (Kieran, 1986; Scally, 1986) [6]. Two strands of geometry are involved: geometry and measurement, each with its own content, procedures and applications. While there is a dichotomy between the two mathematical strands, angle and angle measure are highly intertwined. Early instruments used for the measurement of angles of any size include the astrolabe, planisphere and quadrant. These instruments are all variations of a protractor and are similar to the modern sextant. A sextant is an instrument used to measure the angle of inclination of celestial bodies in order to determine the latitude of the observer. But as the increasing industrialization the demand for angle measurement technology increased, eventually.

3. METHODOLGY

3.1 Components

This project is mostly a combination of various technologies. So, what makes the project possible?

3.1.1 Arduino Uno R3

It is a microcontroller to help the project process data. Arduino is a programmable microcontroller. We can upload a compiled C++ code into it to customize its function. This specific 8-bit board has a RAM of 2KB, Storage space of 8KB for the code. It has multiple I/O pins for all kinds of analog and digital operations.

3.1.2 HC SR 04 Sonar Sensor

It is an Ultrasonic Transceiver Module. It simply has a trigger pin which is connected to the speaker. Thus it creates a wave every time it receives current. Then it has an echo pin which allows current to pass when it detects a wave with additional 2 power pins to run the module.

3.1.3 MPU6050

It is a combined module with multiple sensors like temperature, accelerometer, gyro, etc. it is used as an all-in-one module in drones but we are utilizing it in our project for its gyro-accelero sensing capabilities. It uses I2C capabilities to communicate with the Arduino.

3.1.4 Adafruit IIC OLED display.

Here specifically, we are talking about the 64p SSD1306 version [7]. It is a tiny 1 inch OLED display which has around total 8K pixels which is actually a lot for its size and cost. It has a resolution of 128x64. It is a monochrome panel, so no colors. The pixels will either be off or on. But that's all we need for this project. It again uses I2C communication to receive the complex digital data from arduino.

3.1.5 Prototyping board (Breadboard)

Since this is a prototype, we are going to make the circuit temporary for now. But it can very easily be transferred on a PCB. We are also using jumper cables for connections.

3.2 Computing

3.2.1 External Libraries

Here we are using the library files Wire.h to help us communicate with I2C devices (OLED and Gyro). For the OLED screen, we have the Adafruit_SSD1306.h library which handles low level communication with the module. While Adafruit_GFX.h provides us additional Graphics functions to be used with OLED. The MPU6050.h provides support for the gyro sensor.

3.2.2 Algorithm

Step 1. Create an **MPU6050** object named gyro. (Class reference in MPU6050.h).
Step 2. Define the pins connected to **echo** and **trig** pin of the Sonar sensor.
Step 3. Declare the object called '**OLED'** and provide the necessary info such as the screen resolution, I2C address, etc.
Step 4. Create a **long** and **int** variables for Duration and Distance respectively. These will be used for the Sonar module.
Step 5. Void setup function gains control as soon as the code runs, it runs only once and then transfers control to void loop ().
Step 6. There is basically a serial monitor (**terminal**) when the project is connected to a PC using USB. This is for the developer to communicate with the device for **debugging** purposes. The user has nothing to do with it.
Step 7. clear any previous printings on the display
Step 8. Set the coordinates for the cursor to start printing on the display.
Step 9. Set a text size.
Step 10. Set the text color to be printed in. the actual color can change based on the display since we are dealing with monochrome display, in these displays, WHITE simply means ON state for the pixel.
Step 11. A function Prints a text with the parameters we set previously. "**DigiScale**" is the **splash screen** text that is displayed at startup.
Step 12. Another function gives Display command. All the previous 4 steps didn't actually print a text on the OLED but created a Frame inside the Arduino's memory. Basically, the Arduino has '**Imagined'** the picture but hasn't painted it. It is **cached** on the memory to be specific. This function will send that cached frame to the OLED to be displayed.
Step 13. Add a **delay** between the commands above and below it. Simply, make the Device wait before executing the next commands.
Step 14. Again, print another frame of the splash screen after the 2 second delay.
Step 15. Set the **pinMode** of the **trigpin** and **echoPin** of the Arduino. There are primarily 2 modes INPUT & OUTPUT (and an additional INPUT_PULLUP).
Step 16. 5, 6, 7 are the pins where the buttons are connected. We are simply stating that these pins are supposed to take **input**.
Step 17. The Void Setup () ends here and control is given to void loop (). This function keeps **looping** (repeating the code inside it) until another function is called.
Step 18. Create **Boolean** variable that contains the state of the 3 buttons used. And since a button can only have 2 states, it can be stored as a Boolean value instead of **integer**. (It's better to keep the code as efficient as possible; Arduino has only **2 Kilobytes** memory)
Step 19. **Normalize/calibrate** the gyro and read the Rotation data from the sensor and put them in variables **xl** & **yl**. ('**l**' has no meaning; it's simply a suffix I gave to the names).
Step 20. Give a pulse of voltage to the trig pin of **HC SR 04**. It only turns the pin on for 10 **microseconds**. So that it creates a single sound wave of wavelength **10 µs**.
Step 21. Measure the time it takes for the echo pin to give a pulse (detect the wave) after the initial transmission. This duration is then put in a formula to find out the distance.

Step 22. Speed of sound is **343 m/s**. Convert that into CMs and we get **34,300 CM/s**. and since the duration we get is in **microseconds**. Let's divide it by 1 million (2 times by thousand). We get **0.034 CM/µs**. This means sound travels **1** CM in **0.034** microseconds and if we multiply it by the duration, we get the distance the sound traveled. We still have to divide it by 2 because the sound has to bounce and come back but we only want the distance **one way**.
Step 23. Create a **float** variable and converts this distance into **feet**.
Step 24. An '**if**' statement gets triggered when the button for distance measurement is pushed. (When its Boolean value becomes '1').
Step 25. Clear any previous output from the display.
Step 26. Print formatted texts to show the distance.
Step 27. Print the same output on the **Serial Monitor** for **debugging** purposes. (Not meant to be seen by the user).
Step 28. Print the distance in feet.
Step 29. Finally send above output to OLED & wait 1 second before closing the **If** statement.
Step 30. When the button for angle measurement is pushed, print the angle info for 1 second. The angle of **X** axis from the ground level to be specific.
Step 31. When the button for **Digital bubble level** is pushed, start an **infinite loop** (since there's no condition in the while, just a **logical 1**)
Step 32. **Normalize** the gyro again and read the **button state** of all the 3 buttons. This info is **refreshed** every **cycle** of the loop.
Step 33. Read the X and Y axis from the sensor **again** but this time **adds 64** & **32** to both of these variables respectively. This is because we want the **Crosshair** to be in the centre of the screen when both axes are at Zero (ground level). But the (0, 0) coordinates on the oled are on the top left. To make (0, 0) the center, we simply add the half of the resolutions of the Display (64, 32) which is at the centre.
Step 34. Clear previous frames on the display, and plot a **pixel** on the display depending on its angle.
Step 35. Finally, plot pixels 2px above, below, to the left and right of the Dot to create a crosshair (+ like sign).
Step 36. Then finally print it. Notice that we haven't added any delay on this loop. So above info and the display frame can refreshed as fast as possible for the Microcontroller and display. Arduino's speed varies but the OLED is actually capable of going as fast **500Hz** which is quite fast.
Step 37. A break () statement helps get out of the loop since there is no condition on the While () statement.
Step 38. If the button for one of the other 2 modes is pressed, then this loop closes and it **switches** to that mode.
Step 39. Add a bit of delay before ending the Void Loop () function. It Prevents Button **spamming** causing **stutters** and **bugs**. The delay isn't even noticeable (20ms) but helps a lot with **performance**.
Step 40. Go back to **step 17** and start again, keeps **looping indefinitely**.

4. RESULT

A splash Screen is shown for 2 seconds at startup.

Fig-2 Splash Screen

When put in Distance measurement mode using the dedicated button. The button simply checks the distance to the object directly in front of it and displays is on the OLED in Centimeters and feet.

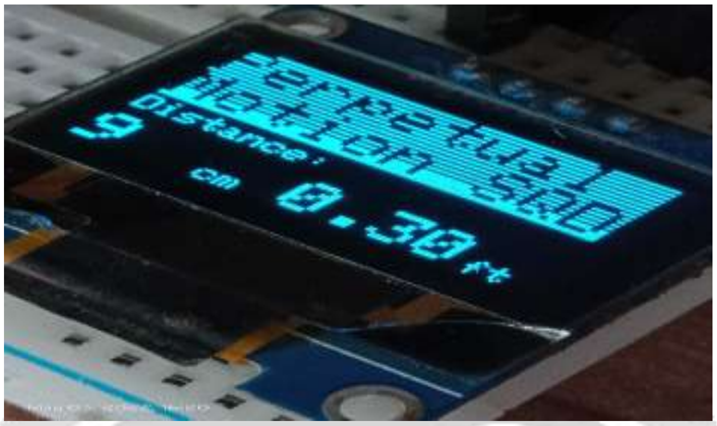

Fig-3 Distance measurement mode.

To check the distance again, the user is expected to press the distance button again. This will refresh the values.

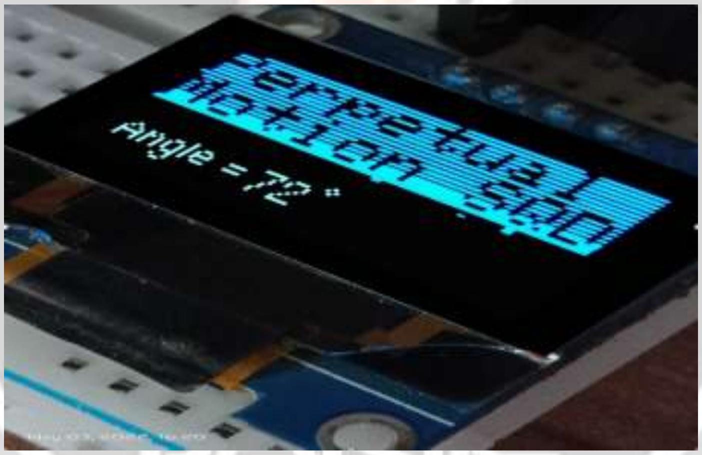

Fig-4 Angle mode

In the angle measurement mode, the device will basically display the current angle between the device and ground. (Ground level can be calibrated by restarting the device).

Fig-5 Digital Bubble level

In the Digital Bubble level mode, the device acts as a spirit level/bubble level device. However, it displays the info digitally. There's simply a crosshair (Plus-like sign) that represents the Bubble as you would see on a traditional bubble level. It moves with respect to the angle. Here the crosshair is in the center because the device is kept on a flat surface. The crosshair will move as we tilt the device.

5. CONCLUSION

To conclude the DigiScale project is simply a digital measurement device which has 3 modes, Distance measurement, angle measurement and a graphical bubble level mode. However, we would like to improve it further by adding improved functionality in the code, like measuring derived quantities like speed using the distance measurement capability or automatically calculating things like hypotenuse if fed with multiple values. We are also working on a multisampling system to extend the distance range beyond 10 feet. This technique will take multiple samples of the data from sonar and perform operations like averaging & removing garbage value, etc. This will help us get accurately estimated values from the sensor beyond its theoretical limit.

6. ACKNOWLEDGEMENT

First of all, we are really thankful to our parents and all family members for providing us an opportunity to study and to work in technical filed. Also, we would like to thank **"The Management, SES"** and **Hon. Dr. N. G. Haswani**, **The Principal** of **R. C. Patel polytechnic, Shirpur** for encouraging us and providing constant motivation along with providing all the facilities for the completion of this project work.

7. REFERENCES

[1]. OLED display library - https://github.com/adafruit/Adafruit_SSD1306
[2]. https://www.aps.org/publications/apsnews/201007/physicshistory.cfm
[3]. https://scholar.google.co.in/scholar?q=u.%20grimaldi%20et%20al%20and%20p.purnell%20ultrasonic%20techniques&hl=en&as_sdt=0&as_vis=1&oi=scholart&authuser=0
[4]. https://en.m.wikipedia.org/wiki/Geodimeter#text=The%20Geodimeter%20(a%20cronym%20of%20geodetic%20Aktiebolaget%20Gasaccumulator)%20company%2020of%20Sweden
[5]. https://scholar.google.co.in/scholar?q=Clements%20and%20Sarama%202009&hl=en&as_sdt=0&as_vis=1&oi=Scholart&authuser=0
[6]. https://link.springer.com/article/10.1023/A:1003938415559
[7]. Extended graphics functions Library (adafruit_GFX.h)- https://www.arduinolibraries.info/libraries/adafruit-gfx-Library