OCTA BRAILLE DISPLAY

Akash Dakoor (B17032)*, Siddharth Gupta (B17063)†, Shreyansh Kulshreshtra (B17062)‡, Aditya Mantri (B17001)§
Ananya Shukla (B17007)¶, Sharad Shukla (B17061)||

*b17032@students.iitmandi.ac.in
†b17063@students.iitmandi.ac.in
‡b17062@students.iitmandi.ac.in
§b17001@students.iitmandi.ac.in
¶b17007@students.iitmandi.ac.in
||b17061@students.iitmandi.ac.in

***Abstract*—The objective of the project is to build a machine that could translate electronic text into braille and enable the visually impaired people to read, and was affordable. The machine was designed such that it would read a string of text through serial communication of the Arduino UNO microprocessor board and rotate a set of octagonal discs with braille character combinations into the correct configuration. An interface was developed for the user using the Arduino UNO microprocessor board and push-buttons that performed four functions- (a) Printing the next ten characters in the text input (b) Tracing back to the previous set of ten characters (c) Resetting the machine to prepare for shut down (d) Displaying the current time on the machine using an RTC module.**

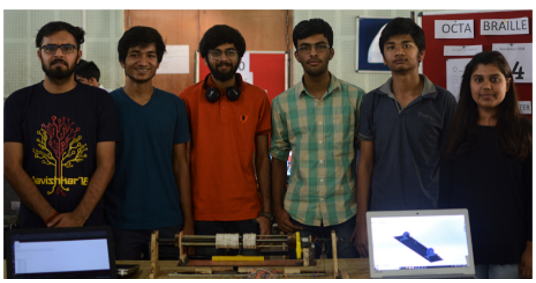

Fig. 1. Team 14 - Octa Braille Display

I. INTRODUCTION

Braille is a writing system used by the blind, comprising raised dots that are read by touch [1]. A set of six dots represent one character. We were inspired to make a project that could benefit the blind community after visiting a school run by the National Association for the Blind. Today, braille books are around 30 percent more expensive than regular books [2] which is a major hindrance in promoting braille literacy. Other refreshable braille displays exist but are extremely expensive [3]. This leads to the blind moving to more convenient options like text-to-speech for information, reducing braille literacy and hence their employment ratios. In order to solve this problem, we decided to make an inexpensive yet convenient refreshable braille display. The Octa Braille Display is a device that converts text input of ten characters at a time from a computer into its braille equivalent and displays it using octagonal discs with small beads used to represent braille dots. This device uses two stepper motors controlled by a micro-controller to permute the discs according to the input given.

II. COMPONENTS USED

S.No	Components Used	Quantity
1	Arduino Uno	1
2	Breadboard/PCB board	1
3	NEMA 17 Stepper Motor	2
4	Motor Driver (A4988)	2
5	DS3231 Real Time Clock Module	1
6	4-Pin Dip Micro PCB Tactile	4
7	9V HI-Watt Battery	2
8	Wooden Octagonal Discs (With Braille Dot Combinations)	20
9	Jumper Wires	~25
10	Slider/Rails	1
11	Wooden Platform	1
12	Arduino Cable	1

A. Arduino Uno

It is an open-source microcontroller board based on Atmega328P(a single-chip microcontroller). It has 14 digital input/output pins and 6 analogue input pins. It is used to control speech production. [4]

Fig. 2. Arduino Uno

B. NEMA 17 Stepper Motor

It is a bipolar stepper motor with 43.2 * 43.2 mm^2 faceplate. It has a step angle of 1.8 degrees. Each phase draws 1.71 A allowing a holding torque of 4.2 kg-cm. It divides a full rotation into a number of equal steps. [5]

Fig. 3. NEMA 17 Stepper Motor

C. Motor Driver (A4988)

Based on Allegros A4988 microstepping bipolar motor driver, it has microstep resolution down to 1/16 step. It operates from 8-35 V and can be used for any stepper motor up to 1 A.

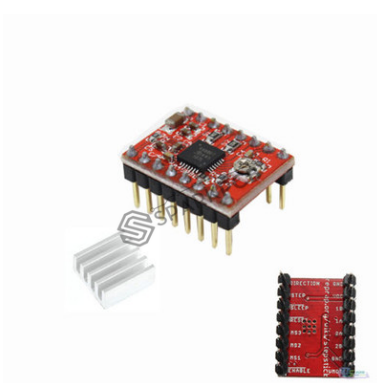

Fig. 4. Motor Driver (A4988) [6]

D. DS3231 Real Time Clock Module

An extremely accurate real-time clock with an integrated temperature-compensated crystal oscillator. The device incorporates a battery input and maintains accurate timekeeping when main power to the device is interrupted.

Fig. 5. DS3231 Real Time Clock Module [7]

E. 4-Pin Dip Micro PCB Tactile

These 6*6*6 mm^3 momentary push buttons have a high duty cycle for a long lifespan and provide just a momentary contact. They normally remain open.

Fig. 6. 4-Pin Dip Micro PCB Tactile [8]

F. 9V HI-Watt Battery

Zinc-Carbon type battery with a metal jacket. It has a nominal voltage equal to 9V and a cut-off voltage equal to 5.4 V. The discharge resistance is 620 Ω.

Fig. 7. 9V HI-Watt Battery [9]

G. *Wooden Octagonal Discs*

The mechanical display is primarily composed of 20 octagonal wooden discs. Each braille character is made by 6 dots, in two columns of three dots each. The device uses two octagonal discs with different permutations of three beads on each face, to create all possible combinations of these 6 dots, making one character. The discs are properly oriented using the motors and the required braille text is obtained.

Fig. 8. Wooden Octagonal Disc

III. Design and Working

A. *Overview*

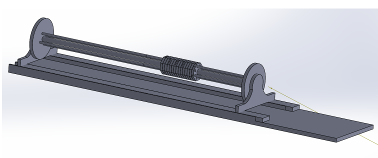

Fig. 9. SolidWorks Model of Octa Braille Display

The Octa Braille Display uses an Arduino UNO microcontroller board, an RTC module, push-buttons, and two stepper motors to perform the necessary functions.
In devising a design for the Octa Braille Display, our team considered different technologies like linear actuators. But since our primary aim was to make the device affordable and portable, we decided not to use them as linear actuators of the size required were extremely expensive. There was also the concern of magnetic interference between two individual actuators. The technologies we used in this device are comparatively cheaper and convenient.
The Arduino UNO microcontroller takes the inputted text and computes the required braille equivalent. It then calculates the desired permutation of the discs and rotates the respective motors accordingly. A Real Time Clock (RTC) was incorporated to keep track of the current time. Push buttons were used to form a physical interface to choose between the four functions- (1) printing the next set of characters, (2) printing the previous line of text, (3) printing the current time, and (4) resetting the machine to neutral before power off.
The display consists of 20 discs, a stepper motor driven rotation mechanism, a holding mechanism, and an axle to mount the display. The stepper motor is used to orient each disc appropriately while the holding mechanism ensures that the orientation once set is not disturbed. Small beads are stuck onto the faces of the octogonal discs in different permutations for different faces to simulate braille.

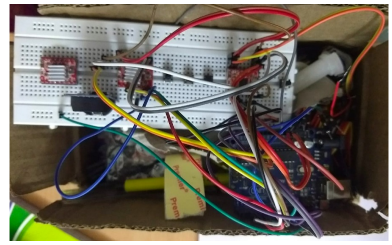

Fig. 10. Circuit

B. *Flowchart and Process*

1) Pseudo code: Below is the pseudo code of the algorithm used-

```

prog OBD

 subprog PRINT_LINE
  STORE pos array values in prevline array
  FOR i from 1 to 10
   READ character from serial port
   CONVERT character to braille
   STORE character in pos array
   CALCULATE position of disc
   ROTATE disc
   UPDATE new position of discs
   MOVE platform forward
  END FOR
  MOVE platform to initial position
 END PRINT_LINE

 subprog RESET
  FOR i from 1 to 10
   STORE blank position in pos array
   CALCULATE position of disc
   ROTATE disc
   UPDATE new position of discs
   MOVE platform forward
  END FOR
  MOVE platform to initial position
 END RESET

 subprog LOOP
  WHILE TRUE
   IF button 1 is pressed
    CALL subprog PRINT_LINE
   ELSE IF button 2 is pressed
    STORE prevline array values in pos array
    CALL subprog PRINT_LINE
   ELSE IF button 3 is pressed
    CALL subprog RESET
   ELSE IF button 4 is pressed
    GET time from rtc module
    CALL subprog PRINT_LINE for time
   END IF
  END WHILE
 END LOOP

END OBD
```

2) Flowchart: Given below is a flowchart representation of the process performed by the microcontroller-

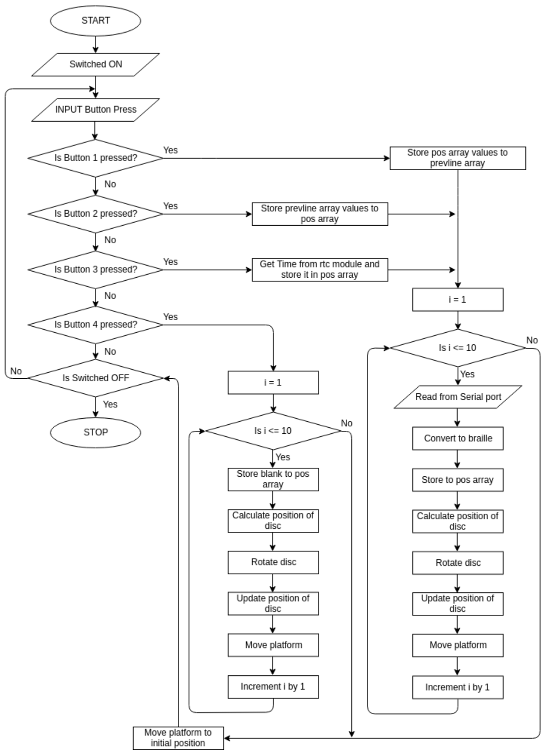

Fig. 11. Flowchart

C. Steps

- The microcontroller continually checks for button presses of the four different buttons forming the interface. Each button corresponds to a different function.
- To print a line, the micrcontroller reads a character through serial communication. It then converts the character to its corresponding braille equivalent and stores the value. The algorithm then calculates the required turning necessary to properly orient the disc.
- This information is communicated to the motor driver corresponding to the motor designated to rotate the discs. The motor is rotated precisely and the disc is aligned appropriately. The motor controlling linear motion then rotates to shift a single disc onto the holding mechanism.
- This process is repeated for ten characters and the platform is moved back to its original position, with no discs in the holding side, in preparation for the next line to be printed.
- The printing process is the same for all four functions. To print the previous line, the microcontroller stores the configuration of the discs corresponding to the printed line before printing a new line. These stored values are then used to calculate the turns required and print the line.
- To print the time, the microcontroller communicates with the RTC module to get the current time through the I2C protocol. This time is then converted to its braille equivalent and printed by the process described above.
- The reset function is implemented to prepare the device for use the next time before powering off. It clears the last printed line and aligns the discs to bring all the blank sides up.

IV. Result

We successfully devised and implemented the algorithm required to print Braille. Our algorithm was tested repeatedly by coding it to print the number of turns required and final position of discs stored, hand simulating the results for sample test cases and then comparing the values obtained. We were unable to reach the amount of mechanical precision required in the given time limit for the project, and the display was not functional. However, only a few minor tweaks are required to fix the issues in the device, like increasing the diameter of all components mounted on the axle to reduce friction during linear motion and correctly positioning the gears between the motor and the rotation mechanism to ensure efficient power transfer.

Making the device affordable was our primary aim and we are satisfied with the amount of money we could build this display in.

V. Conclusions

We have implemented the idea of converting the text fed to the Octa Braille Display into its braille equivalent, enabling the visually impaired to read and comprehend the inputted text. We also incorporated a provision for displaying the current time as a set of braille characters.

Our main aim is to empower them with a tool using which they can experience the might of the words. Though currently, our device takes the input through the microprocessor boards serial communication, we aspire to expand the bases and take the input from the PDF and EPUB files and get the data converted into braille, line by line. The e-books can then be accessible to them that are way better than the bulky and expensive braille books. A feature that we would like to work on is enabling Bluetooth connectivity so that data can be directly transferred to the device from a mobile phone. Another innovation that can be incorporated is using sensors to identify and count different denominations of currency.

VI. Acknowledgement

We would like to acknowledge and express gratitude to Dr. Hitesh Shrimali, Asst. Professor, IIT Mandi, and other faculty members for organising and coordinating Aavishkar, Dr. Shubhajit Roy Chowdhury for helping us with the idea for our project, Mr. Suryavanshi Virendrasingh for mentoring us and guiding us throughout the development of our project, our friends and colleagues for supporting and helping us whenever needed.

References

[1] Definition of Braille
`http://www.dictionary.com/browse/braille?s=t`

[2] Indian Express article
`http://www.newindianexpress.com/pti-news/2018/jan/22/high-cost-of-braille-books-a-dampener-for-tn-studen html`

[3] American Foundation for the Blind
http://www.afb.org/prodBrowseCatResults.aspx?CatID=43

[4] Arduino UNO
https://en.wikipedia.org/wiki/Arduino_Uno

[5] NEMA 17 Stepper Motor
https://en.wikipedia.org/wiki/Stepper_motor

[6] A4988 Motor Driver Image Source:
https://www.sparkpcb.com/development/a4988-stepper-motor-driver-board.html

[7] Real Time Clock D3231
https://datasheets.maximintegrated.com/en/ds/DS3231.pdf

[8] 4-Pin Pushbuttons Image Source:
https://www.amazon.com/Pushbutton-Switches/b?ie=UTF8&nod

[9] 9V H-W battery image source:
https://www.fabtolab.com/Hi-Watt-9V-6F22M-battery

[10] Flowchart prepared using draw.io
www.draw.io