

Broad Problem Statement

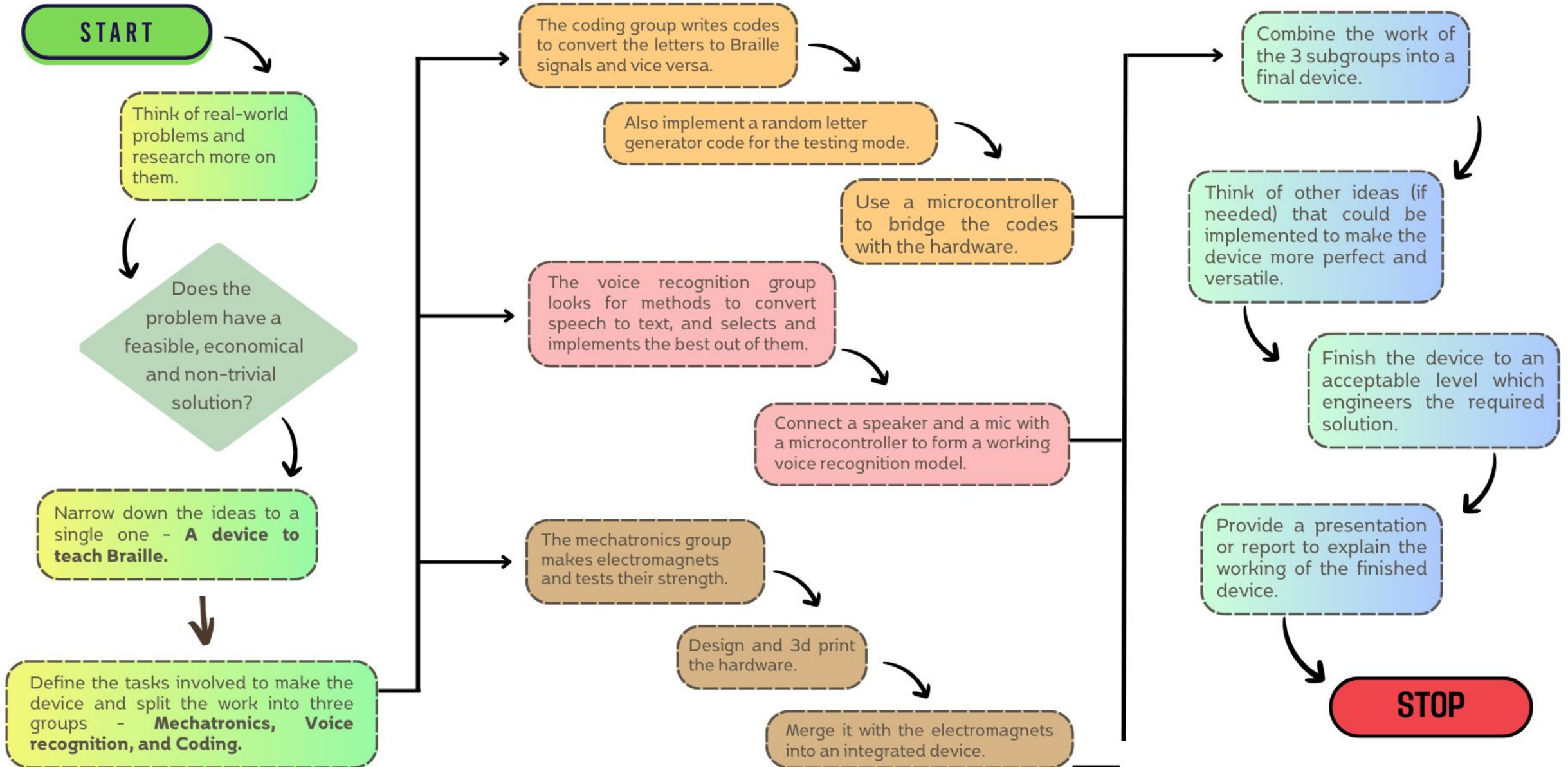
- **Braille** is a vital communication tool that enables blind and visually impaired individuals to read and write independently.
- **The problem** we have identified is the **limited access to braille education and literacy** for visually impaired individuals especially in developing countries like **India**.
- The **significant challenges faced by the visually impaired community**, in accessing Braille education in India are given below:
 - We **don't have enough braille teachers** as a result of which, the underprivileged blind are deprived of a very basic skill of literacy. Learning is possible **only in their presence**.
 - Moreover, the electronic gadgets meant for remote braille training are too **expensive** for many visually challenged people in India to afford (based on this report).

Braille education in India is restricted **only to English** and not to any **native language** in India.

Specific Problem Statement

To develop an interactive and **self-sufficient** device to assist visually-challenged individuals in **learning Braille** digits and letters in English and possibly, local Indian languages.

FLOWCHART



Calculations/Quantifications

List of Components:

- Device frame [3D printed with PLA]
- Cylindrical button shaped extrusions [3D printed with PLA] (6)
- Components of the electromagnet - iron rod, enamelled copper, permanent magnet (6 sets)
- Speaker
- Voice recognition module/Microphone
- Microcontroller (Arduino Uno R3 or Raspberry Pi 3)
- Mode-changing buttons (3), Multipurpose button (1), ON/OFF button (1)

Calculations/Quantifications

$B = \mu n I$, **$n = N/L$** where

N = Number of turns in the electromagnet

L = Length of electromagnet

$\mu = \mu_r \times \mu_0$ where μ_0 = Permittivity of free space

After building a prototype of the electromagnet to be used, we decided on the following specifications:

$N = 200$

$L = 3 \text{ cm}$

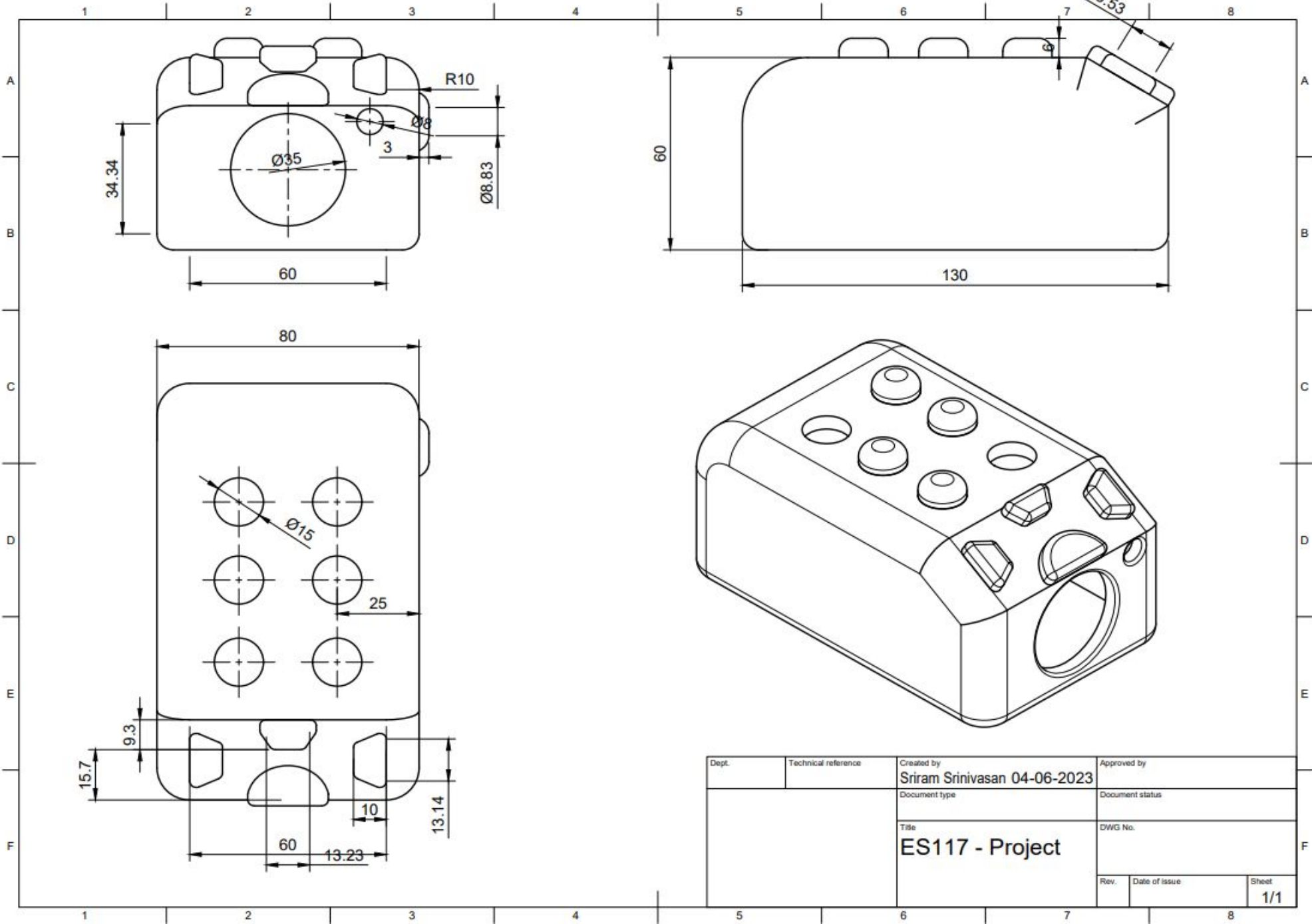
Power source rated at 12V 3A DC

From the above calculations, calculated value of magnetic field turned to be:

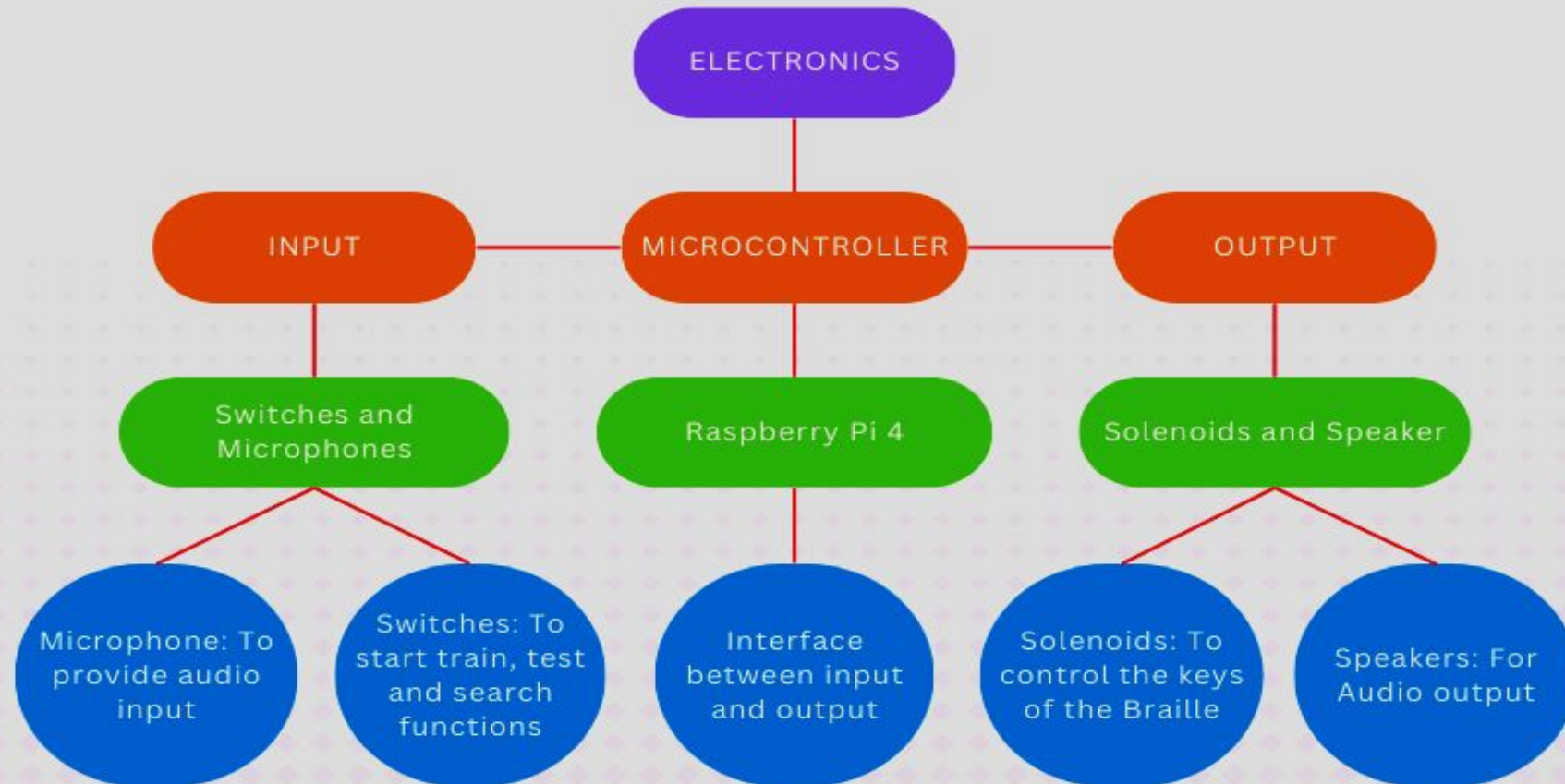
$B = 0.098 \text{ T}$

AutoCAD Model of the Proposed Device

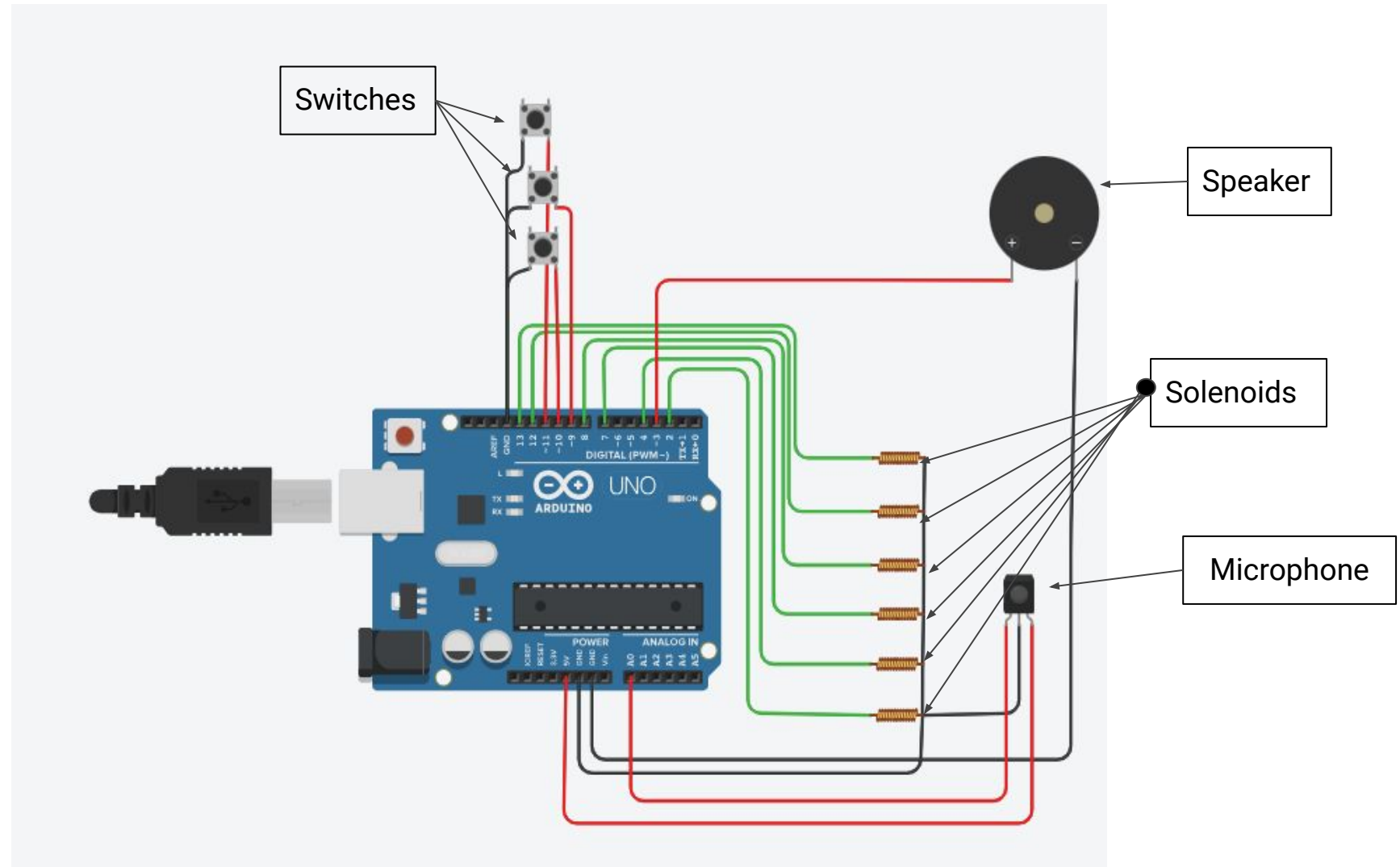




ELECTRONIC COMPONENTS TO BE USED



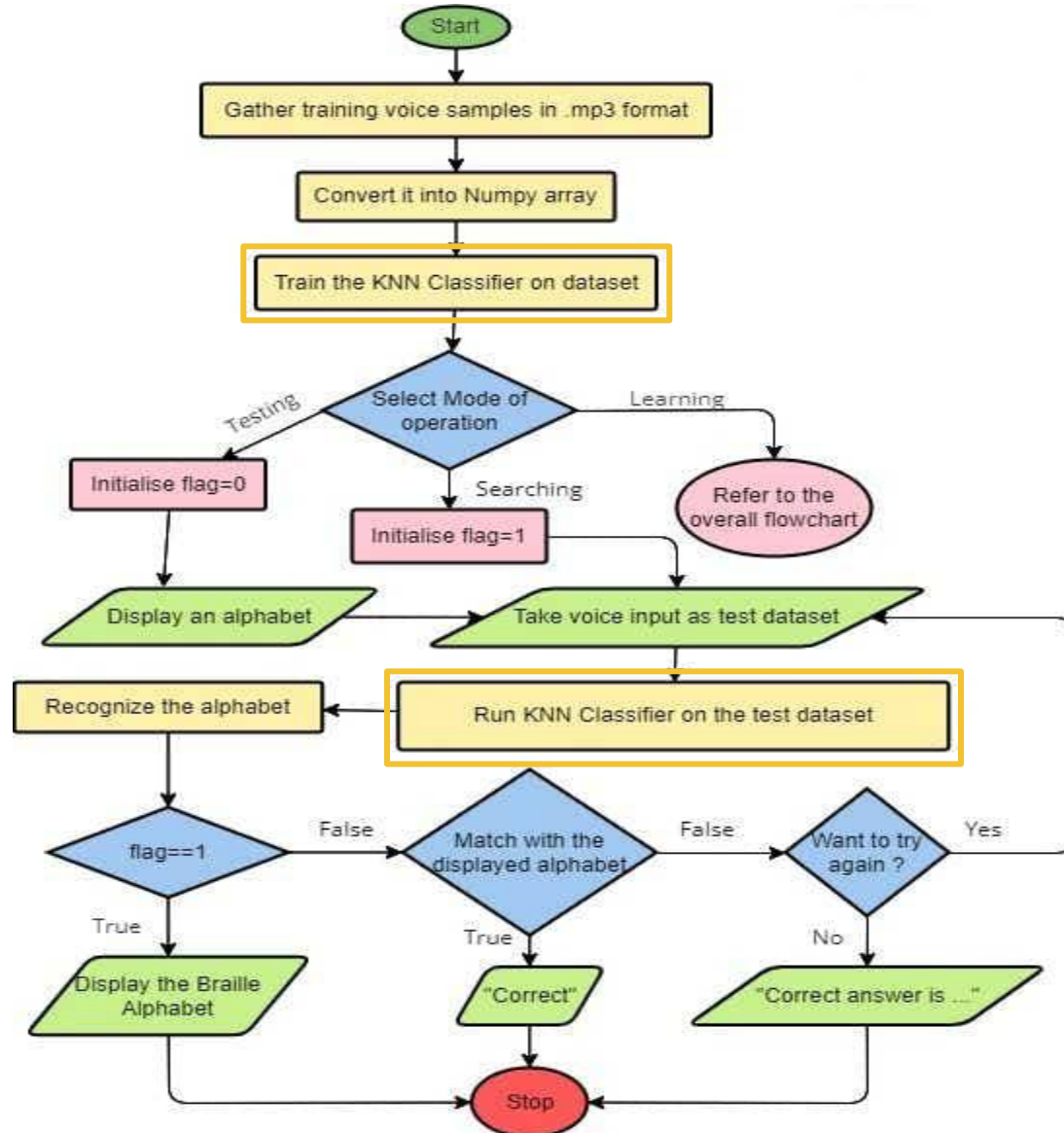
Electronic Circuit Diagram



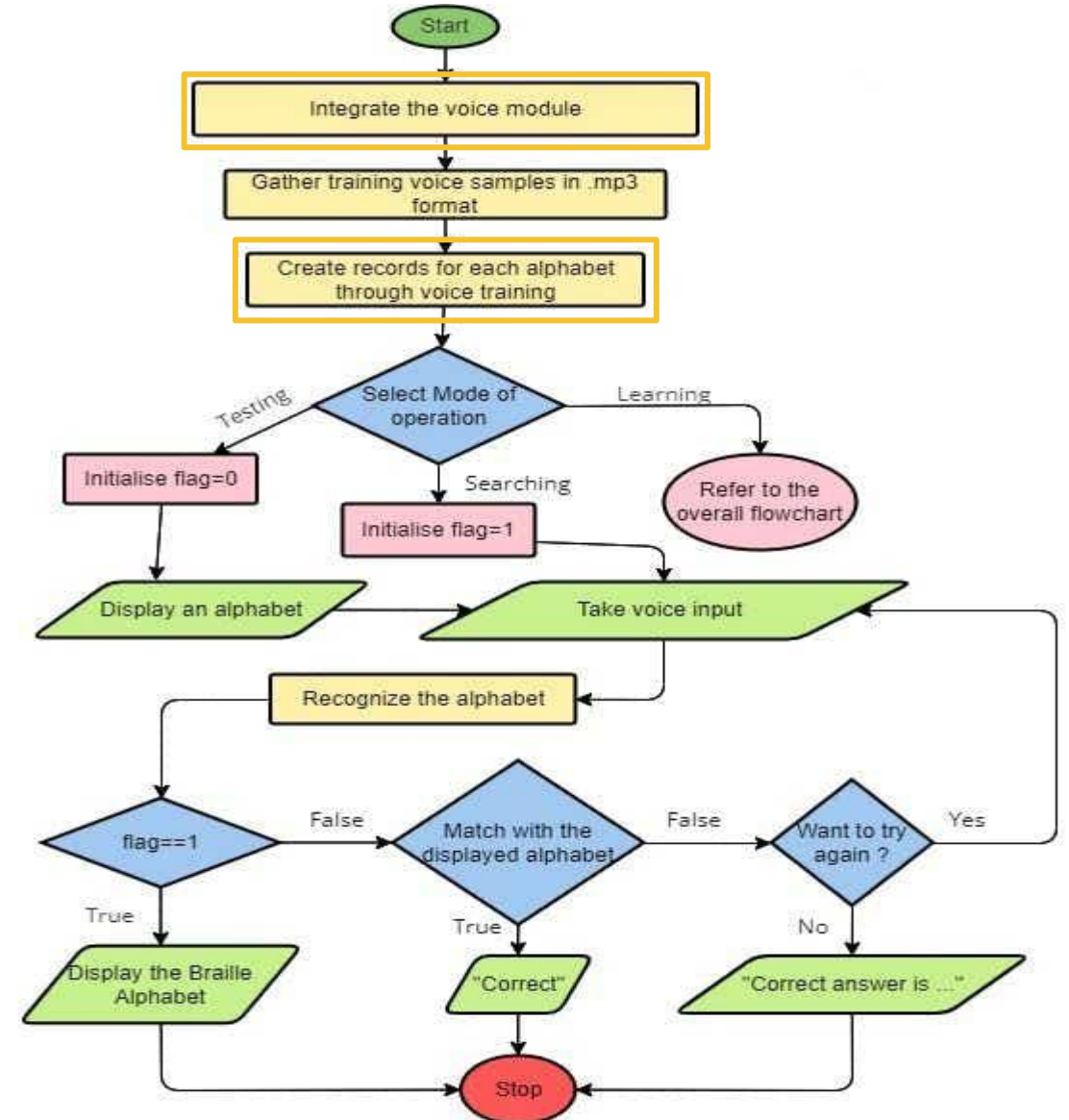
Made on TinkerCAD

Voice Recognition Flowchart

Method 1



Method 2



Pseudo Code

Program Flow:

Setup

Choose Mode

Setup:

Initialize Braille matrix representation for letters A, B, and C

Choose Mode::

Ask user to choose searching, training, or learning mode:

If user chooses searching mode, Enter searching mode

If user chooses training mode, Enter testing mode

If user chooses learning mode, Enter learning mode

Obtain Function:

Obtain voice input

Return input to IC

Learning Mode:

Loop through each letter in Braille representation:

Display Braille representation and Speak letter

Pause and ask if user wants to repeat (Voice Mode)

Searching Mode:

Loop until exited:

Get user's voice input letter

Match input to Braille representation key and display it

Pause to ask if the user wants to repeat (Voice Mode)

Testing Mode:

Loop until exited:

Display a randomly selected letter

Pause to get user's input

If input matches selected letter:

Speak success message

Ask if user wants to continue

If user wants to continue, repeat loop

Voice Function:

Loop until valid input received:

Obtain voice input

If input is "Yes" or "No":

Return True if "Yes", False if "No"

else:

Speak "Didn't quite get you, please repeat"

*The pseudo code on these two pages is the summary of the actual code that we have developed. The next page contains the code that will be implemented.

```
import numpy as np
from time import sleep

def setup():
    a=[1,0,0,0,0,0]
    b=[1,0,1,0,0,0]
    c=[1,1,0,0,0,0]
    braille=np.zeros((3,2))
    di={"A":a,"B":b,"C":c}
    for i in d:
        d[i]=d[i].reshape(3,2)
    return d,braille

d,braille=setup()

def learning_mode():
    print_voice("You are now in learning mode")
    while True:
        for i in d:
            print(d[i])
            print_voice(i)
            sleep(5)
            print(braille)
            sleep(5)
        print_voice("Do you want to repeat ?")
        if voice():
            print_voice("You chose to return to learning mode")
            continue
        else:
            break

def voice():
    while True:
        obtain_voice()
        if voice_input=="Yes" or voice_input=="No":
            if voice_input=="Yes":
                return True
            else voice_input=="No":
                return False
        print_voice("Didn't quite get you please repeat")

def voice2():
    while True:
        obtain_voice()
        if voice_input in list(d.keys()):
            return voice_input
        print_voice("Didn't quite get you please repeat")

def searching_mode():
    print_voice("You are now in searching mode")
    while True:
        voice2()=k
        print(d[k])
        print_voice(k)
        sleep(5)
        print(braille)
        print_voice(q[0])
        sleep(5)
        print(braille)
        sleep(5)
        print_voice("Do you want to repeat ?")
        if voice():
            print_voice("You chose to return to searching mode")
            continue
        else:
            break

def obtain():
    obtain_voice()
    return(voice_input)

def testing_mode():
    print_voice("You are now in testing mode")
    while True:
        q=np.random.choice(list(d.keys()))
        print(d[q[0]])
        print_voice(q[0])
        sleep(5)
        print(braille)
        sleep(5)
        if obtain() == q[0]:
            print_voice("Kudos you got that right")
            print_voice("Do you want to continue")
            if voice():
                continue
            else:
                return
        else:
            print_voice("Oops im afraid thats wrong")
            print_voice("Do you want to try again")
            if voice():
                while True:
                    print(d[q[0]])
                    print_voice(q[0])
                    sleep(5)
                    print(braille)
                    sleep(5)
                    if obtain() == q[0]:
                        print_voice("Kudos you got that right")
                        print_voice("Do you want to continue")
                        if voice():
                            p=1
                            break
                        else:
                            p=2
                            break
                    else:
                        print_voice("Oops im afraid thats wrong")
                        print_voice("Do you want to try again with the same chracter")
                        if voice():
                            continue
                        else:
                            print_voice("Do you want to know the answer")
                            if voice():
                                print_voice(q[0])
                                else:
                                    pass
                            print_voice("Do you want to exit testing mode")
                            if voice():
                                return
                            else:
                                break
            else:
                print_voice("Do you want to exit testing mode")
                if voice():
                    return
                else:
                    break
def choose_mode:
    print_voice("please choose one of the three options search")
    while True:
        obtain_voice()
        if voice_input=="searching":
            searching_mode()
        elif voice_input=="training":
            testing_mode()
        elif voice_input=="learning":
            learning_mode()
        else:
            print_voice("Didn't quite get you please repeat")

setup()
choose_mode()
```

THANK YOU!