7-Segment Display with Arduino and Push Button

Personal Details

1. BHARATH A. S. - 21BCT0032

2. DESISH D. - 21BCE2299

3. SURAJ JHA - 21BCT0358

4. ARTH P SINGH- 21BCT024

5. KAVI SHAH - 21BCE2083

Aim

To design and implement a simple digital system using an Arduino that displays a number on a 7-segment display, which changes when a push button is pressed.

Objective

- To learn how to interface a 7-segment display with an Arduino.
- To understand how to read input from a push button.
- To implement a basic digital counter that increments or cycles through numbers each time the push button is pressed.

Required Components

- Arduino Uno (or any other compatible Arduino board)
- 1 x 7-segment display (common cathode or common anode)
- 8 x 220-ohm resistors (for current limiting to the segments)
- 1 x Push button
- 1 x 10k-ohm resistor (for the pull-down resistor on the push button)
- Breadboard
- Jumper wires

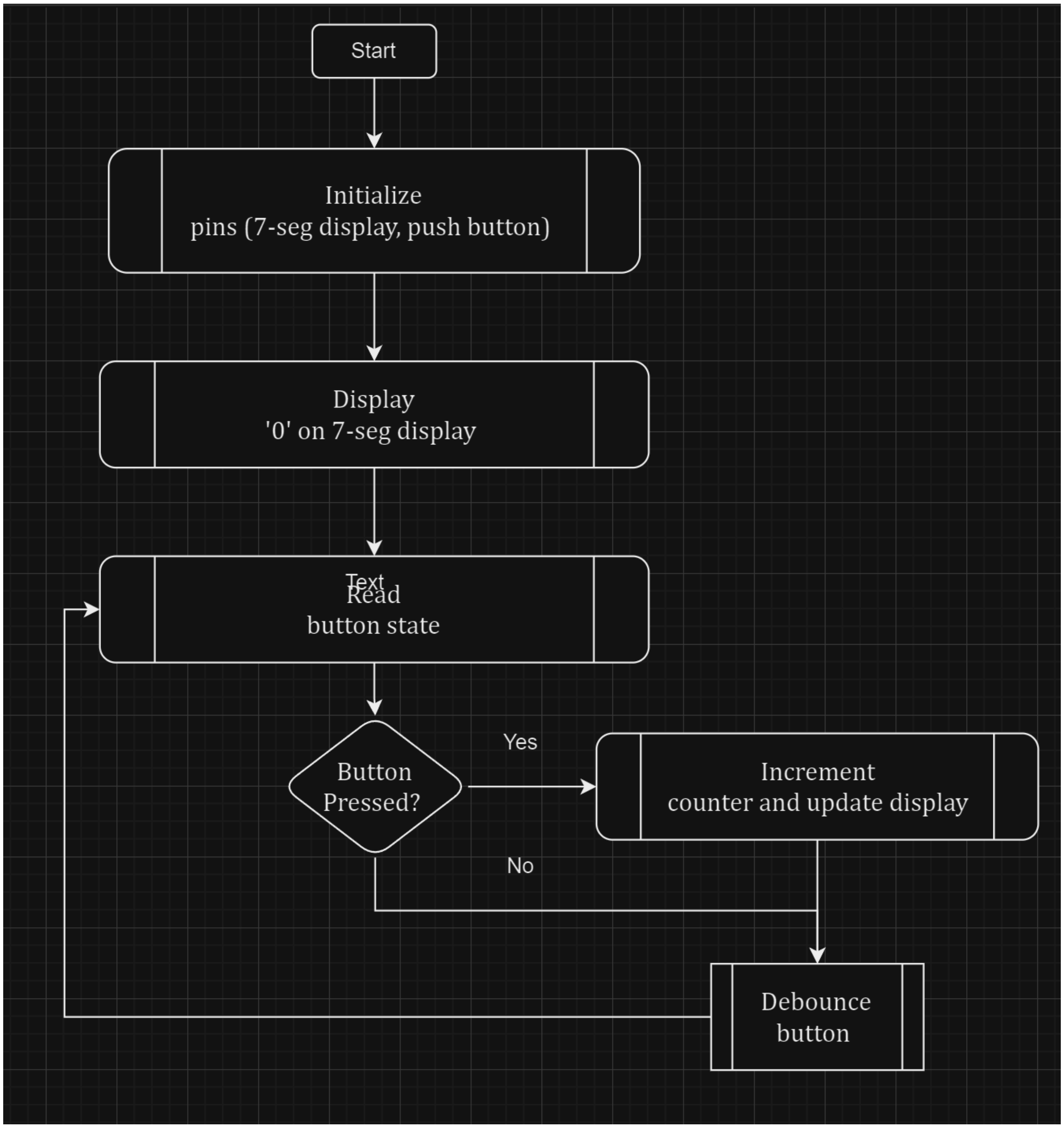
Process and Methodologies

1. Connecting the 7-Segment Display:
- Identify the pins of the 7-segment display.
- Connect the segment pins (A, B, C, D, E, F, G) to digital pins on the Arduino through current-limiting resistors.
- Connect the common cathode/anode pin to ground or Vcc respectively.

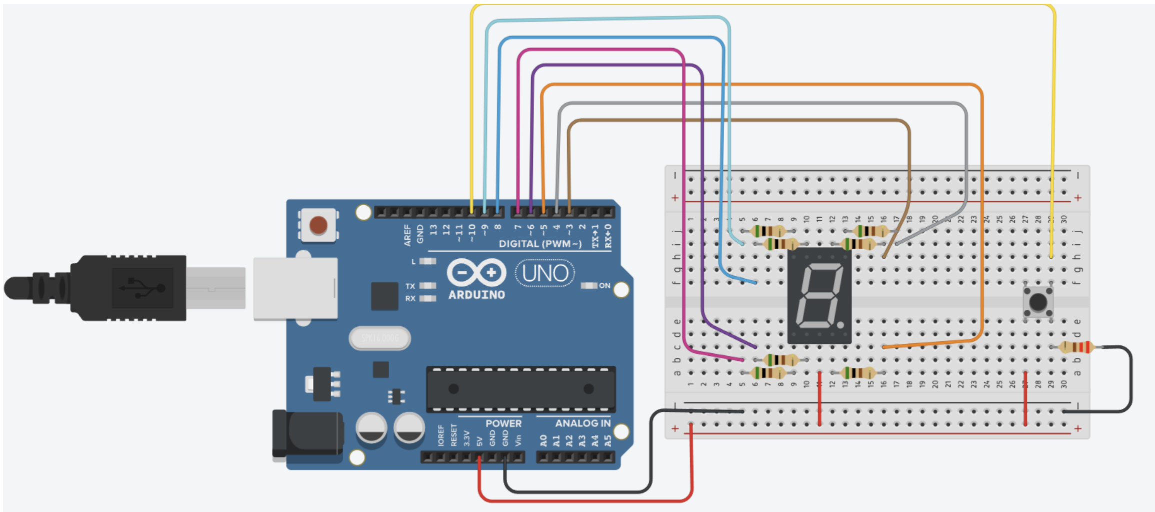
2. Connecting the Push Button:
- Connect one side of the push button to an input pin on the Arduino.
- Connect the other side of the push button to ground.
- Use a pull-down resistor to ensure the input reads LOW when the button is not pressed.

3. Programming the Arduino:
- Write code to initialize the 7-segment display pins as outputs.
- Write a function to display digits on the 7-segment display.
- Implement a loop that reads the state of the push button.
- Each time the button is pressed, the displayed number increments.
- Debounce the push button to avoid multiple counts from a single press.

Flowchart of Process



Circuit Diagram



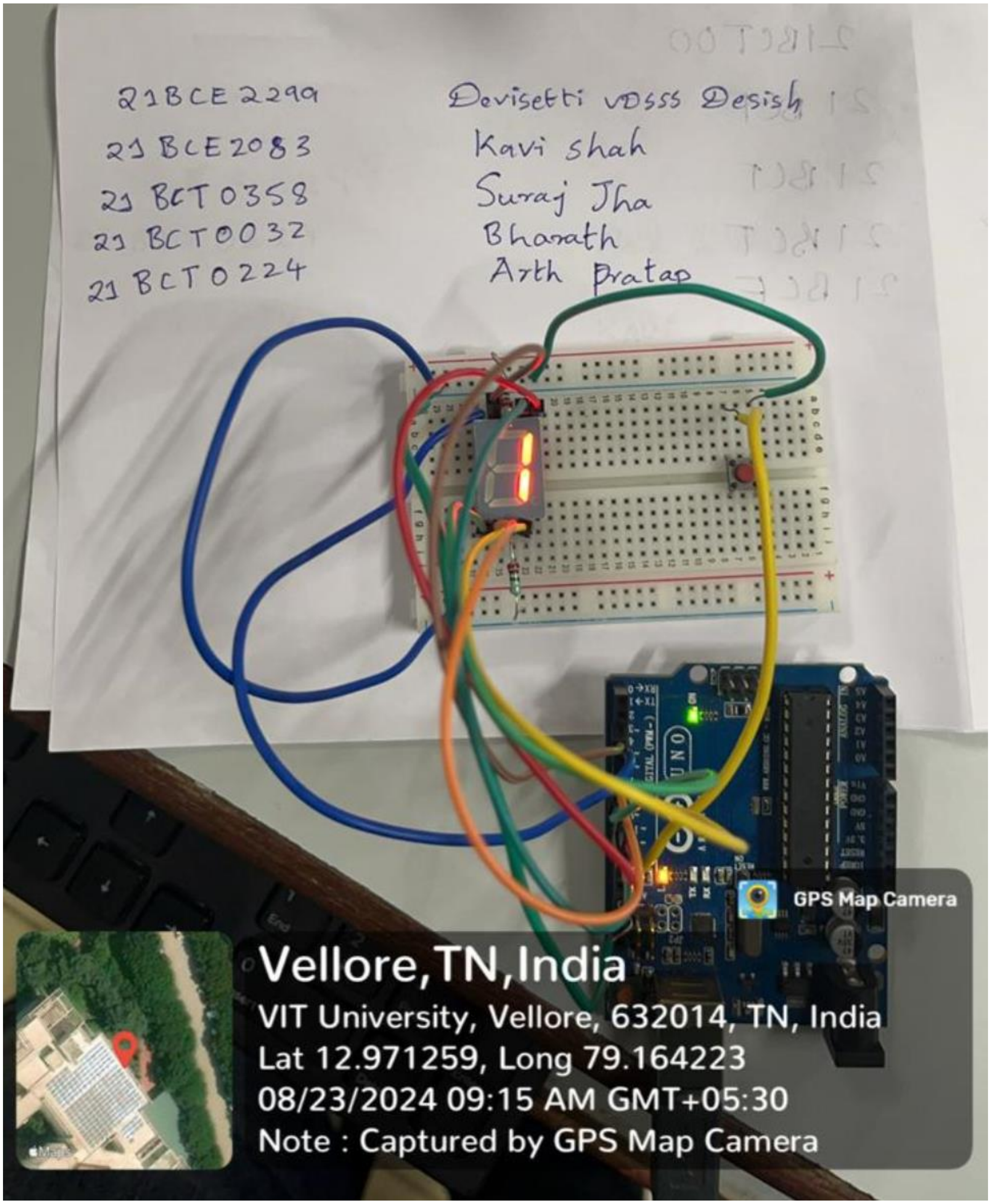
1. 7-Segment Display:
   - Pin A to Digital Pin 2 of Arduino
   - Pin B to Digital Pin 3 of Arduino
   - Pin C to Digital Pin 4 of Arduino
   - Pin D to Digital Pin 5 of Arduino
   - Pin E to Digital Pin 6 of Arduino
   - Pin F to Digital Pin 7 of Arduino
   - Pin G to Digital Pin 8 of Arduino
   - Common Cathode to GND (through a 220-ohm resistor)

2. Push Button:
   - One terminal to Digital Pin 9 of Arduino
   - Other terminal to GND
   - Pull-down resistor (10k-ohm) between the input pin and GND.

Methodologies

- Debouncing: Software debouncing is used to ensure that the push button press is registered once. This can be done by introducing a small delay (e.g., 50ms) after a button press is detected before reading the state again.
- Modular Code: Separate the display logic and the button handling into functions for clarity and reusability.

Practical Implementation Photos and Video


21BCE2299 Devisetti VDSSS Desish
21BCE2083 Kavi shah
21BCT0358 Suraj Jha
21BCT0032 Bharath
21BCT0224 Arth Pratap
GPS Map Camera
Vellore,TN,India
VIT University, Vellore, 632014, TN, India
Lat 12.971259, Long 79.164223
08/23/2024 09:15 AM GMT+05:30
Note : Captured by GPS Map Camera

Google Drive Link:

https://drive.google.com/file/d/1hvIjv0FhhlLTd47H3qQVghmH42Ewbwqu/view?usp=sharing

Arduino Code

```
int segments[] = {2, 3, 4, 5, 6, 7, 8}; // pins connected to the 7-segment display
int buttonPin = 9; // pin connected to the push button
int count = 0;
int digits[10][7] = {

  {1, 1, 1, 1, 1, 1, 0}, // 0

  {0, 1, 1, 0, 0, 0, 0}, // 1

  {1, 1, 0, 1, 1, 0, 1}, // 2

  {1, 1, 1, 1, 0, 0, 1}, // 3

  {0, 1, 1, 0, 0, 1, 1}, // 4

  {1, 0, 1, 1, 0, 1, 1}, // 5

  {1, 0, 1, 1, 1, 1, 1}, // 6

  {1, 1, 1, 0, 0, 0, 0}, // 7

  {1, 1, 1, 1, 1, 1, 1}, // 8

  {1, 1, 1, 1, 0, 1, 1}  // 9

};
void setup() {
  for (int i = 0; i < 7; i++) {
    pinMode(segments[i], OUTPUT);
  }
  pinMode(buttonPin, INPUT);
}
void loop() {
  displayDigit(count);
  if (digitalRead(buttonPin) == HIGH) {
    delay(50); // debounce delay
    count++;
    if (count > 9) count = 0; // reset to 0 after 9
    while(digitalRead(buttonPin) == HIGH); // wait until button is released
  }
}

void displayDigit(int num) {
  for (int i = 0; i < 7; i++) {
    digitalWrite(segments[i], digits[num][i]);
  }
}
```

Inference

Thus implemented a push-button activated 7-segment display LED program to display digits from 0 to 9 sequentially via interfacing with Arduino uno microcontroller board.