**Implementing a PIC Microcontroller-Based Quiz Buzzer**

STUDENT NAME: G.P.D. THAMARA

STUDENT NUMBER: EC/2021/005

**Source Code**

void main() {

int i = 0;

TRISB = 0b01000011;

PORTB = 0b00000000;

while(1) {

if(PORTB == 0b00000001) {

PORTB = 0b00010100;

while(i == 0) {

if(PORTB.RB6 == 1) {

PORTB = 0b00000000;

i = 1;

}

}

i = 0;

}

else if(PORTB == 0b00000010) {

PORTB = 0b00101000;

while(i == 0) {

if(PORTB.RB6 == 1) {

PORTB = 0b00000000;

i = 1;

}

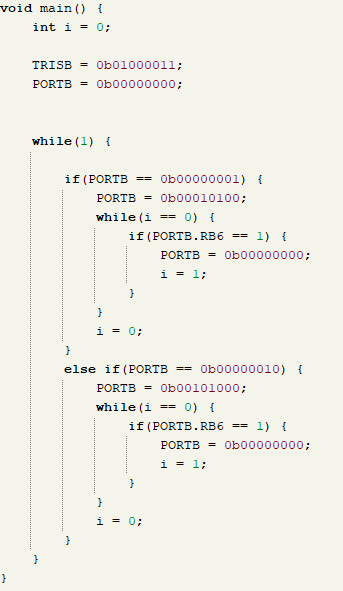
}

i = 0;

}

}

}

****

**Circuit**

A computer circuit diagram with many wires

AI-generated content may be incorrect.

**Observations**

When the first button is clicked then one led is switched on

A computer screen shot of a circuit board

AI-generated content may be incorrect.

When the third button is clicked then currently on LED is switched off when the second button is pressed other led is switched on

A computer screen shot of a circuit board

AI-generated content may be incorrect.

**Discussion**

9 **"Fastest Finger-First"** quiz buzzer systemusing the PIC16F628A microcontroller. The system was designed to detect the first button press among multiple participants, activate a corresponding buzzer or LED, and lock out all other inputs until a manual reset was performed. This approach mirrors real-world competitive environments where quick response identification is essential.

A primary challenge was the microcontroller’s limited current output, which is insufficient to drive external devices like buzzers directly. This issue was resolved by integrating an NPN transistor as a switch, allowing the microcontroller to control higher current loads safely. The transistor provided electrical isolation and current amplification, protecting the microcontroller from damage due to voltage spikes or back EMF.

The experiment successfully demonstrated how digital inputs can be latched to prioritize the first response, ensuring system fairness. The implementation of a reset mechanism allowed the system to be reused for successive rounds efficiently. Overall, the lab reinforced key embedded system principles such as input detection, latching logic, state control, and the importance of driver circuits when working with current-demanding peripherals.