\documentclass[12pt,a4paper]{article}

\usepackage{amsmath,amssymb}

\usepackage{graphicx}

\usepackage{geometry}

\usepackage{float}

\usepackage{enumitem}

\usepackage{titlesec}

\usepackage{ulem}

\usepackage{subcaption}

\usepackage{listings}

\usepackage{xcolor}

\usepackage{tcolorbox}

% ---------- COLORS ----------

\definecolor{codebg}{rgb}{0.97,0.97,0.97}

\definecolor{keywordblue}{rgb}{0,0,0.8}

\definecolor{commentgreen}{rgb}{0,0.5,0}

\definecolor{stringorange}{rgb}{0.8,0.3,0}

\definecolor{graynumbers}{rgb}{0.5,0.5,0.5}

% ---------- ARDUINO/ESP STYLE ----------

\lstdefinelanguage{Arduino}{

morekeywords={

pinMode,digitalWrite,digitalRead,analogRead,analogWrite,delay,delayMicroseconds,

pulseIn,Serial,begin,print,println,setup,loop,if,else,for,while,void,int,float,

long,define,include,String,return,OUTPUT,INPUT,HIGH,LOW,setCursor,clear,begin,

attach,write

},

sensitive=true,

morecomment=[l]{//},

morecomment=[s]{/\*}{\*/},

morestring=[b]",

}

\lstdefinestyle{ArduinoStyle}{

language=Arduino,

backgroundcolor=\color{codebg},

basicstyle=\ttfamily\footnotesize,

keywordstyle=\color{keywordblue}\bfseries,

commentstyle=\color{commentgreen}\itshape,

stringstyle=\color{stringorange},

numbers=left,

numberstyle=\tiny\color{graynumbers},

stepnumber=1,

numbersep=8pt,

frame=single,

rulecolor=\color{graynumbers},

breaklines=true,

showstringspaces=false,

tabsize=2,

captionpos=b

}

\tcbset{

colback=gray!5,

colframe=black!50,

boxrule=0.5pt,

arc=3pt,

left=5pt,

right=5pt,

top=5pt,

bottom=5pt

}

\geometry{margin=1in}

\setlength\parindent{0pt}

\sloppy

\raggedright

% Subsection font

\titleformat{\subsection}[block]{\normalsize\bfseries}{\thesubsection}{1em}{}

% Title format

\title{{\Huge \textbf{Lab Report: ESP32 Interfacing Experiments}}}

\author{

V.L.S. Bhargav ; Lanka Kushwanth \\

Table Number: 3 \\

Room Number: 114 \\

Roll Number: 2025102061

}

\date{\today}

\begin{document}

\maketitle

% ------------------- PART A -------------------

\section\*{\Large{\underline{\fbox{PART A: RGB LED Control Using Push Button}}}}

\section\*{\underline{Aim:}}

To control the color of an RGB LED using a push button on the ESP32. Each button press cycles the LED through Red, Green, Blue, and OFF states, demonstrating basic digital I/O control.

\section\*{\underline{Components Required:}}

\begin{itemize}

\item ESP32 Development Board

\item Common Cathode RGB LED

\item Push Button

\item 3 $\times$ 220$\Omega$ resistors

\item Breadboard and Jumper Wires

\end{itemize}

\section\*{\underline{Theory:}}

\begin{itemize}

\item The RGB LED combines red, green, and blue LEDs in one package.

\item Each LED color is controlled via a GPIO pin using digitalWrite().

\item The push button provides user input, read using digitalRead().

\item Debouncing and cycling logic allow changing LED color on each button press.

\end{itemize}

\section\*{\underline{Procedure:}}

\begin{enumerate}

\item Connect RGB LED pins (R, G, B) to ESP32 GPIO pins through 220$\Omega$ resistors.

\item Connect button between GPIO pin and GND (use internal pull-up if needed).

\item Write and upload code using Arduino IDE.

\item Each press should cycle LED through different colors.

\end{enumerate}

\section\*{\underline{Physical Circuit:}}

\begin{figure}[H]

\centering

\includegraphics[width=0.6\textwidth]{IMG\_1924.jpeg}

\caption{Working setup for RGB LED and Push Button on ESP32.}

\end{figure}

\begin{figure}[H]

\centering

\begin{subfigure}{0.32\textwidth}

\centering

\includegraphics[width=\linewidth]{IMG\_1925.jpeg}

\caption{Red Color}

\end{subfigure}

\begin{subfigure}{0.32\textwidth}

\centering

\includegraphics[width=\linewidth]{IMG\_1926.jpeg}

\caption{Green Color}

\end{subfigure}

\begin{subfigure}{0.32\textwidth}

\centering

\includegraphics[width=\linewidth]{IMG\_1927.jpeg}

\caption{Blue Color}

\end{subfigure}

\caption{RGB LED displaying Red, Green, and Blue colors respectively.}

\end{figure}

\clearpage

\section\*{\underline{Code:}}

\begin{lstlisting}[style=ArduinoStyle, caption={RGB LED Control Using Push Button (ESP32)}]

int redPin = 25;

int greenPin = 26;

int bluePin = 27;

int buttonPin = 14;

int buttonState = 0;

int lastState = 0;

int colorState = 0;

void setup() {

pinMode(redPin, OUTPUT);

pinMode(greenPin, OUTPUT);

pinMode(bluePin, OUTPUT);

pinMode(buttonPin, INPUT\_PULLUP);

Serial.begin(115200);

Serial.println("RGB LED Control Started");

}

void loop() {

buttonState = digitalRead(buttonPin);

if (buttonState == LOW && lastState == HIGH) {

colorState = (colorState + 1) % 4;

delay(200); // debounce delay

}

lastState = buttonState;

switch (colorState) {

case 0: digitalWrite(redPin, HIGH);

digitalWrite(greenPin, LOW);

digitalWrite(bluePin, LOW);

break;

case 1: digitalWrite(redPin, LOW);

digitalWrite(greenPin, HIGH);

digitalWrite(bluePin, LOW);

break;

case 2: digitalWrite(redPin, LOW);

digitalWrite(greenPin, LOW);

digitalWrite(bluePin, HIGH);

break;

default:digitalWrite(redPin, LOW);

digitalWrite(greenPin, LOW);

digitalWrite(bluePin, LOW);

}

}

\end{lstlisting}

\clearpage

\section\*{\underline{Observations:}}

\begin{itemize}

\item Button press cycles through Red → Green → Blue → OFF.

\item The LED color changes immediately upon button press.

\item Debouncing ensures single-color transition per press.

\end{itemize}

\section\*{\underline{Conclusion:}}

\begin{itemize}

\item RGB LED successfully controlled using push button with ESP32.

\item Demonstrates basic GPIO input/output operations.

\end{itemize}

% ------------------- PART B -------------------

\clearpage

\section\*{\Large{\underline{\fbox{PART B: Servo Motor Control Using IR Sensor}}}}

\section\*{\underline{Aim:}}

To use an IR sensor to detect an object and control a servo motor’s rotation using the ESP32.

\section\*{\underline{Components Required:}}

\begin{itemize}

\item ESP32 Development Board

\item IR Obstacle Sensor (HW201)

\item SG90 Servo Motor

\item Breadboard and Jumper Wires

\end{itemize}

\section\*{\underline{Theory:}}

\begin{itemize}

\item The IR sensor outputs LOW when an object is detected.

\item The servo motor angle is controlled using PWM (Pulse Width Modulation).

\item The ESP32Servo library provides easy servo control functions.

\end{itemize}

\section\*{\underline{Procedure:}}

\begin{enumerate}

\item Connect the IR sensor output pin to a GPIO input pin of ESP32.

\item Connect servo control pin to a PWM-capable GPIO pin.

\item Power both modules from ESP32 (use 3.3V for sensor, Vin for servo).

\item Upload the code and observe servo movement when object is detected.

\end{enumerate}

\clearpage

\section\*{\underline{Code:}}

\begin{lstlisting}[style=ArduinoStyle, caption={Servo Control Using IR Sensor (ESP32)}]

#include <ESP32Servo.h>

Servo gateServo;

int irPin = 15;

int servoPin = 18;

int irState = 0;

void setup() {

Serial.begin(115200);

gateServo.attach(servoPin);

pinMode(irPin, INPUT);

gateServo.write(0);

Serial.println("IR Sensor Servo Control Started");

}

void loop() {

irState = digitalRead(irPin);

if (irState == LOW) { // Object detected

gateServo.write(90);

Serial.println("Object detected: Servo rotated to 90 degrees");

delay(1000);

} else {

gateServo.write(0);

Serial.println("No object: Servo at 0 degrees");

}

delay(200);

}

\end{lstlisting}

\section\*{\underline{Physical Circuit:}}

\begin{figure}[H]

\centering

\includegraphics[width=0.7\textwidth]{IMG\_1930.jpeg}

\caption{Working setup of ESP32 with IR Sensor and Servo Motor.}

\end{figure}

\clearpage

\section\*{\underline{Observations:}}

\begin{itemize}

\item Servo motor rotates to 90° when IR sensor detects an obstacle.

\item Servo returns to 0° when the obstacle is removed.

\item System response was smooth and reliable.

\end{itemize}

\section\*{\underline{Conclusion:}}

\begin{itemize}

\item IR sensor successfully detected objects and controlled servo motor rotation.

\item Demonstrated effective use of sensors and actuators using ESP32 GPIO and PWM.

\end{itemize}

\section\*{\underline{References:}}

\begin{itemize}

\item Lab Manual

\item Arduino and ESP32 Documentation

\end{itemize}

\end{document}

Prompt:

Ask ChatGPT to give the latex code for any handout by uploading the handout pdf and then give it this latex code for reference.

Do make some changes.