

Ain Shams University

Faculty of Engineering

Mechatronics Department – Senior 1



## **CSE211s Introduction to Embedded System**

### **Final Report Project**

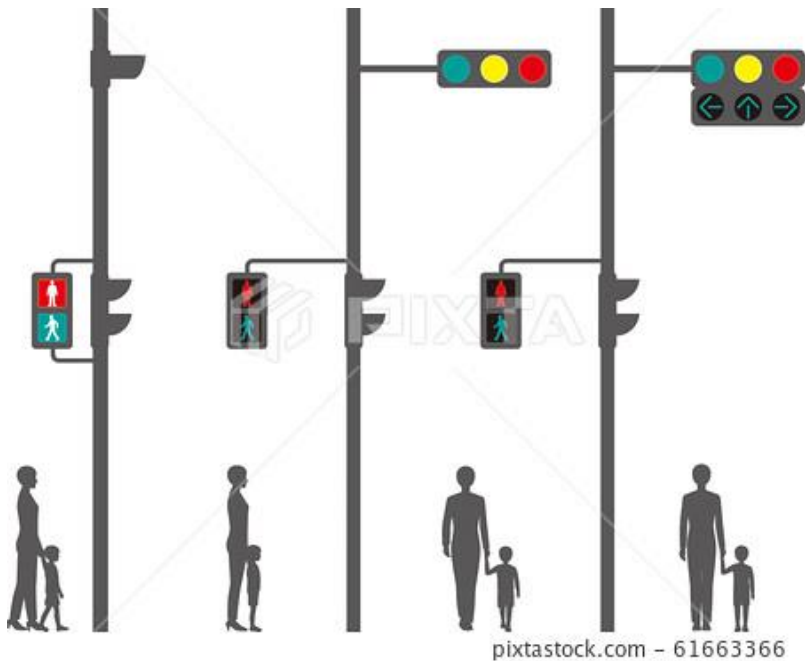
#### **Team (23)**

<b>Name</b>	<b>ID</b>
<b>Mohammed Abdulaziz Farrag Ahmed</b>	<b>1805858</b>
<b>Mohamed Emad Mohamed Hussien Omar</b>	<b>1802245</b>
<b>Mohamed Emad Mohamed Moslem</b>	<b>1806770</b>
<b>Alaa Elsayed Metwally Mohamed</b>	<b>1809361</b>
<b>Ali Abdelhakem Mahmud Hussin</b>	<b>1805347</b>

**Dr/Sherif Hammad**

## Project Description:

In this Project, we will use 2 push buttons for pedestrian to cross the street safely and 6 LEDs for normal traffic sequence as well as 4 LEDs for the pedestrian traffic, we have to arrange one push button with 2 LEDs for each road for the pedestrian as well as 3 LEDs for the car traffic for each road; this circuit always sense the traffic on that particular road, all these connections are interfaced to the microcontroller 'TIVA C'. Based on these connections, controller detects the traffic and controls the traffic system.



## Project Sequence

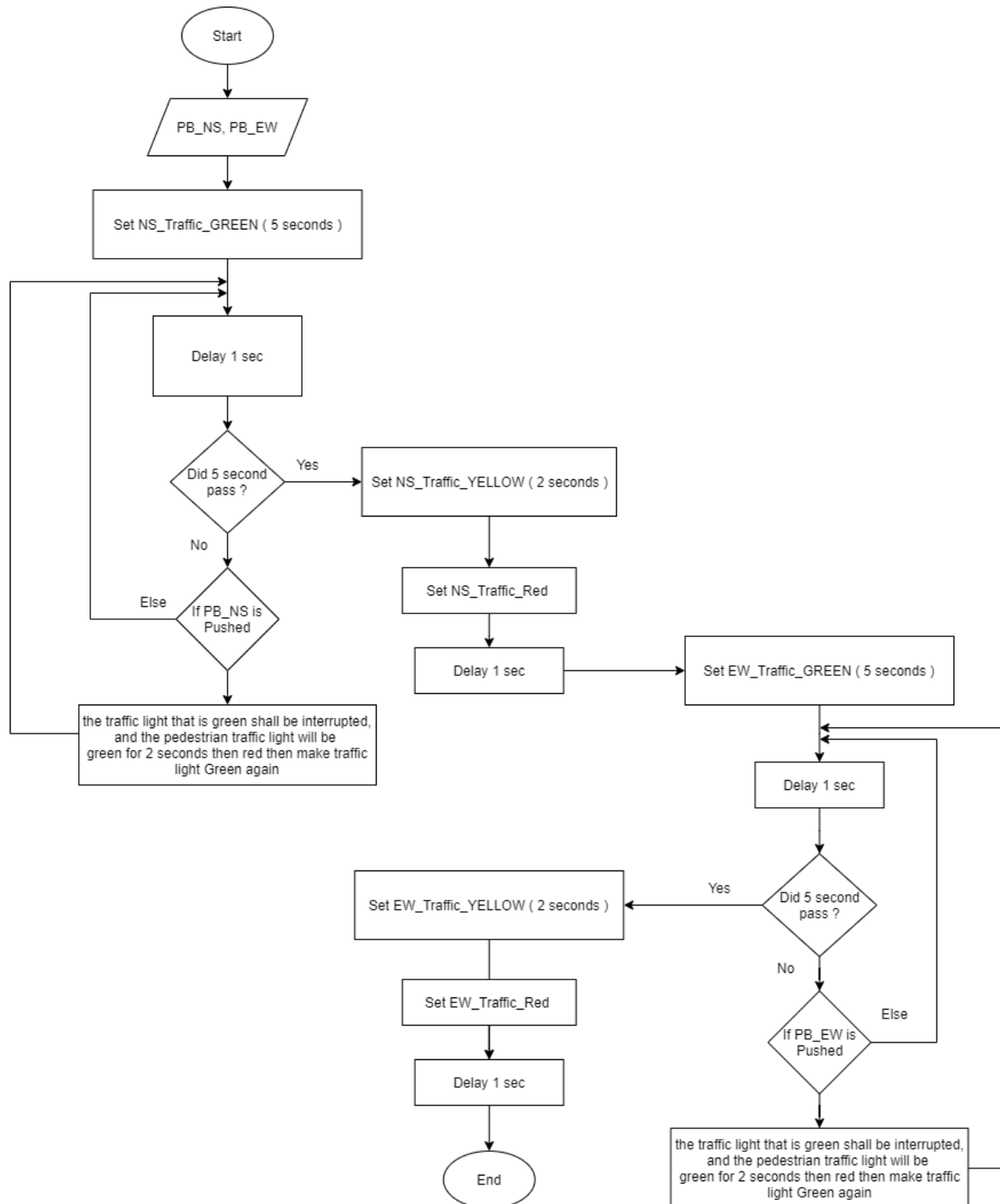
In this system, We Used two push buttons and 4 LEDs. There will be two pedestrian traffics, each with a push button and 2 LEDs: GREEN and RED. Whenever a pedestrian presses the button, the traffic light that is green shall be interrupted, and the pedestrian traffic light will be green for 2 seconds. Then it will be back to RED and the traffic light that was interrupted will resume from when it was paused. Example: If the car traffic light was green for 2 seconds and interrupted it will then resume the remaining 3 seconds to finish the 5 seconds.

Set the normal traffic Light Use one of the timers to have the two car traffic lights working. The traffic light shall stay GREEN for 5 seconds, then YELLOW for 2 seconds, then turns RED. When one of the traffic lights is set to RED the other one has to go GREEN exactly after 1 second.

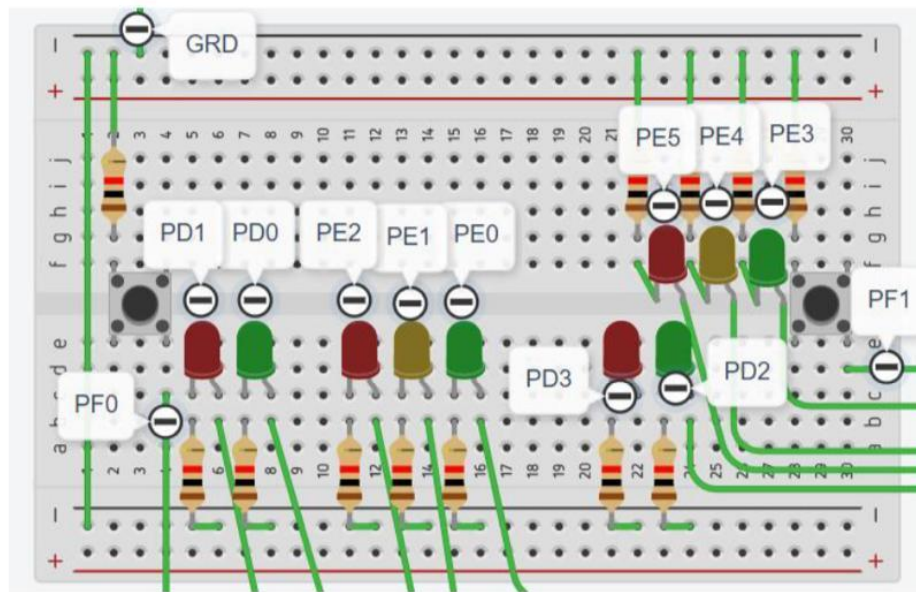
The same sequence then is repeated again:

- GREEN: 5 seconds
- YELLOW: 2 seconds
- RED: Wait 1 second, then start the sequence on the other traffic.

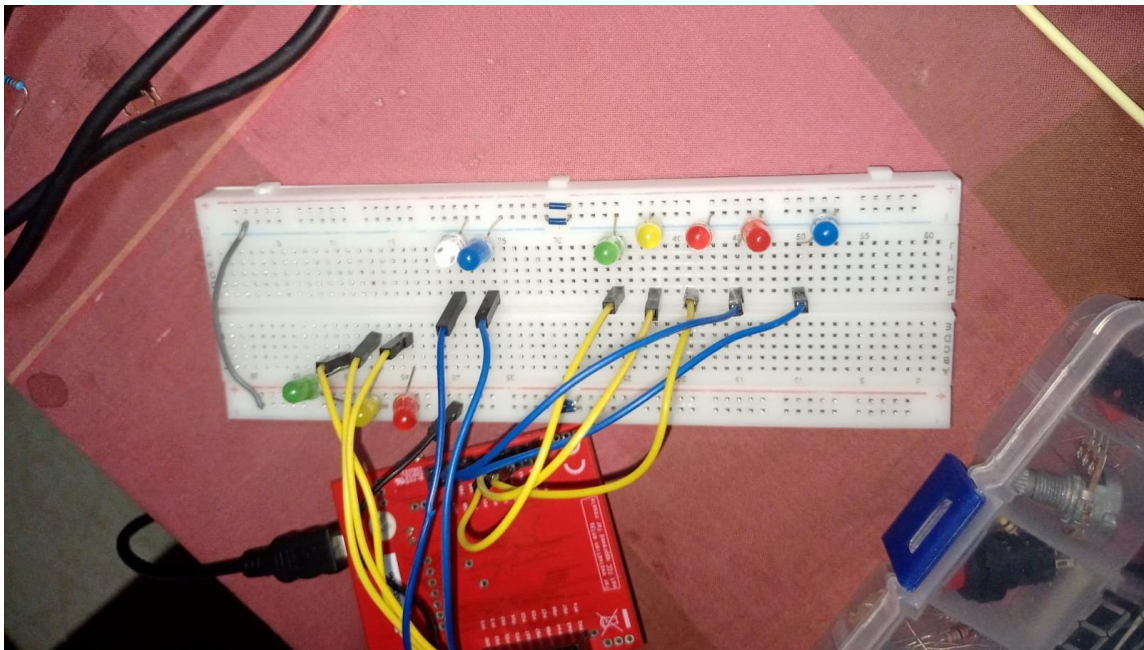
# Flowchart



## Schematic Design



## Real Design



# Code

## 1) Functions

```

/*****
 *
 * Function to initialize Pins in portF as Inputs(switch 1 and switch 2):
 * 1)Enabling Clock for portf pins
 * 2)Configure pins 0,4 as Input pins
 * 3)Enable Interrupts in NVIC for PortF No.30 in Vector table
 *
 *****/
void INIT_PORTF(void)
{
    SYSCCTL_RCGCGPIO_R |= 0x00000020;
    while((SYSCCTL_PRGPIO_R&0x00000020) == 0){};
    GPIO_PORTF_LOCK_R = 0x4C4F434B;
    GPIO_PORTF_CR_R = 0x01;
    GPIO_PORTF_LOCK_R = 0;
    GPIO_PORTF_DIR_R &= (1<<4)|(1<<0);
    GPIO_PORTF_DEN_R |= (1<<4)|(1<<3)|(1<<0);
    GPIO_PORTF_PUR_R |= (1<<4)|(1<<0);
    GPIO_PORTF_IS_R &= (1<<4)|(1<<0);
    GPIO_PORTF_ICR_R |= (1<<4)|(1<<0);
    GPIO_PORTC_IM_R |= (1<<4)|(1<<0);
    NVIC_PRI7_R = (NVIC_PRI7_R & 0xFF00FFFF) | 0x00A00000;
    NVIC_EN0_R = 0x40000000;
}

```

```

/*****
Function responsible for checking interrupt happen by two push Buttons to do a certain function
Case (1) check for interrupt due to SW1
Case(2) check for interrupt due to
 *****/
void GPIOF_Handler(void)
{
    if (((GPIO_PORTF_MIS_R & 0x10) & (GPIO_PORTB_DATA_R & 0x04))!=1)
    {
        // Traffic (1)turn off red, yellow , green
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0 | GPIO_PIN_1 | GPIO_PIN_2, 0);
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 0); //Pedestrian 1 RED led off
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 1); //Pedestrian 1 GREEN led on
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // B is RED
        InterruptDelay(2); // Delay For 2 Seconds
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_4, 1); //Pedestrian 1 RED led on
        GPIOWrite(GPIO_PORTB_BASE, GPIO_PIN_3, 0); //Pedestrian 1 GREEN led off
        GPIO_PORTF_ICR_R |= 0x10;
    }
    else if (((GPIO_PORTF_MIS_R & 0x01)& (GPIO_PORTD_DATA_R & 0x04))!=1)
    {
        // Traffic (2) turn off red, yellow , green
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_6, 0); //Pedestrian 2 RED led off
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_3, 1); //Pedestrian 2 GREEN led on
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // D is RED
        InterruptDelay(2); // Delay For 2 Seconds
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_6, 1); //Pedestrian 2 RED led on
        GPIOWrite(GPIO_PORTD_BASE, GPIO_PIN_3, 0); //Pedestrian 2 GREEN led off
        GPIO_PORTF_ICR_R |= 0x01;
    }
}

```



```

/*****
Function Responsible for initialize systick timer
1)systick Disable
2)reload value 0.5 sec (16MHZ*time in msec-1)
3)Configure Ctl register
*****/
void INIT_SysTick()
{
    NVIC_ST_CTRL_R = 0x00;

    NVIC_ST_RELOAD_R = (((0.5)/(1/16000000))-1);
    NVIC_ST_CURRENT_R = 0x00;
    NVIC_ST_CTRL_R = 0x07;
/*****
*

```

```

/*****
Function responsible for Set Configuration Pins as Input or Output
*****/
void CONFIGURE_PIN(void)
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_0); // Traffic (1) red LED
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_1); //Traffic(1) YELLOW LED
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_2); //Traffic(1) GREEN LED
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_0); //Traffic(2) red LED
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_1); //Traffic(2) yellow
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_2); //Traffic(2) Green
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_3); //Pedestrian 2 GREEN led
    GPIOPinTypeGPIOOutput(GPIO_PORTD_BASE, GPIO_PIN_6); //Pedestrian 2 RED led
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_3); //Pedestrian(1) GREEN led
    GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_4); //Pedestrian (1) RED led
}

```

```

/*****
Function Responsible for Make program that traffic light will work be in Sequence
*****/
void Sequence_Traffic_Light(void)
{
    CONFIGURE_PIN();
    uint8 i = 0;
    while(1)
    {
        if(i == 0){
            GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
            GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // RED light for traffic (1)
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // RED light for traffic (2)
            delay(1000); // delay 1 sec
            i++;
        }
        if(i == 1){
            GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
            GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // RED light for traffic (1)
            GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_2, 4); // GREEN light for traffic (2)
            delay(5000); // delay 5 sec
            i++;
        }
    }
}

```

```

        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // RED light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_2, 4); // GREEN light for traffic (2)
        delay(5000); // delay 5 sec
        i++;
    }
    if(i == 2){
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // RED light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_1, 2); // YELLOW light for traffic (2)
        delay(2000); // delay 2 sec
        i++;
    }
    if(i == 3){
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0, 1); // RED light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // RED light for traffic (2)
        delay(1000); //delay 1 sec
        i++;
    }
    if(i == 4){
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_2, 4); // GREEN light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // RED light for traffic (2)
        delay(5000); // delay 5 sec
        i++;
    }
    if(i == 5){
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);

```

```

        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_2, 4); // GREEN light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // RED light for traffic (2)
        delay(5000); // delay 5 sec
        i++;
    }
    if(i == 5){
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0|GPIO_PIN_1|GPIO_PIN_2, 0);
        GPIOPinWrite(GPIO_PORTB_BASE, GPIO_PIN_1, 2); // YELLOW light for traffic (1)
        GPIOPinWrite(GPIO_PORTD_BASE, GPIO_PIN_0, 1); // RED light for traffic (2)
        delay(2000); // delay 1 sec
    }
    i = 0;
}
}

```

## 2) Main

```

int main()
{
    //Initialize Port f
    INIT_PORTF();

    //Initialize systick timer
    INIT_SysTick();

    // Enable interrupts
    Interrupt_ENABLE();

    // setting output and input pins
    CONFIGURE_PIN();
    // TrafficLight system
    Sequence_Traffic_Light();
}

```



## Video

<https://drive.google.com/file/d/1L2BfB0mVX4iF9ClOAXiVesz3SRqaU1k7/view?usp=sharing>

## Code

[https://drive.google.com/file/d/13XI\\_GWK7ocR960OmymBlph7TDb8aBVZd/view?usp=sharing](https://drive.google.com/file/d/13XI_GWK7ocR960OmymBlph7TDb8aBVZd/view?usp=sharing)