Group members:

Mehmet Zahit Avcı 22040102030

Benan Özkor 23040102067

Elina Mamadalieva 22040102091

İbrahim Melih Karabulut 23040102039

Tunahan Ferramuz Ekelik 22040102034

Kutay Çelik 23040102056

GitHub Link: https://github.com/mehmetzahitavci/TrafficLights.git

Introduction

The aim of our project is to design and implement a traffic management system using microprocessor logic. Our system gives provides real time interactivity through pedestrian request buttons and a visual countdown display. Key features include four-phase vehicle control, hardware-synced pedestrian signals, and an interrupt-driven logic for night mode and pedestrian requests.

Circuit and Hardware Architecture

Master CPU (Vehicle Controller)

The Master CPU is the "brain" of the operation. It is connected to:

Port OA & OB: Control the signal states for North, South, East, and West vehicle lights.

Port OUT (04h): Drives the 7-segment Countdown Display, providing visual feedback to drivers.

Port IN (00h): Receives processed button data and mode switches from the Slave CPU.

Slave CPU (Input Handler)

The Slave CPU's only responsibility is to poll the physical push-buttons (North, East, South, West) and the Night Mode switch. By letting our second CPU handle input polling the Master CPU is never "blocked" by mechanical bouncing or long input cycles, ensuring the timing of the traffic lights are precise and with minimum delay.

To ensure this communication between our CPU's, we used:

Handshaking:

Communication between the Slave and Master CPU's uses a 4-bit parallel handshaking protocol. The Slave CPU continuously reads the four pedestrian button inputs and outputs this compact state to the Master. The Master samples this input continuously to detect button presses that request phase changes.

Polling:

The Master CPU polls input bits to check for both pedestrian button requests and a night-mode switch. This polling is integrated into the main state machine controlling the traffic phases as well as an internal countdown routine.

Pedestrian Integration

To handle pedestrian lights, we used simple and unique feature, connecting pedestrian light to the cross-wired vehicle signal. When turning right pedestrians have priority, depending on this logic we connected the pedestrian lights to the green lights on their left. This reduces code complexity, prevent conflicts even if we occur exception and if we are faced with code error our hardware handles the situation safely.

Software Implementation and Logic

Four-Phase State Machine

The software is structured as a Finite State Machine (FSM) consisting of four main phases (North, East, South, West). Each phase follows a strict sequence:

1.Green Phase: Master CPU sets the corresponding port bits and loads a 9-second duration.

2.Countdown: The COUNTDOWN_LOGIC routine updates the 7-segment display every second.

3.Yellow Transition: A SHORT_DELAY routine handles the transition to ensure safe braking distances before switching directions.

Smart Interruption & Decision Routing

1- The Masking System: During a specific phase (e.g., North Green), the CPU uses a bit-mask (stored in Register H) to ignore the North button but listen for East, West, or South buttons.

2- Skip Logic: If a valid button is pressed on a cross-street, the SKIP_TIMER routine forces the current countdown to zero, accelerating the transition to the requested direction.

Night Mode (Flashing Yellow)

At the start of every cycle, the Master CPU checks Bit 4 of the input port. If the Night Mode switch is active, the system jumps to NIGHT_MODE_ROUTINE. This bypasses the standard FSM and enters a low-power loop where all yellow lights flash via the BLINK_DELAY routine, signaling caution for late-night traffic with minimal delays.