



Product Requirement Document (PRD)

Multi-Color LED Pattern Control SW

Case Study Exercise



Table of Contents

1. Introduction	3
1.1 The project	3
2 Use Case Diagram	3
3 Use Case Description	4
3.1 Use case 1 : Configuring MLC	4
3.1.1 Scenarios:	4
3.1.2 Sequence Diagram : Configuring MLC	5
3.2 Use case 2 : Execute LED Pattern	6
3.2.1 Scenarios:	6
3.2.1 Sequence Diagram : Configuring MLC	7
4 User interface description:	8
4.1 Actors and Interface	8
4.1.1 Observer (Actor)	8
4.1.2 Associate Board (Actor)	8
4.1.3 LED Operator (Actor)	8
4.1.4 Configuration Manager (Actor)	8
4.1.5 Console (Interface)	8
4.1.6 RGB LED (Interface)	8
4.1.7 Inter Board Communication (Interface)	8
4.2 UI - User Interface (Console)	8
4.2.1 Home Screen:	9
4.2.2 Slave Home Screen (Master Found Successfully)	10
4.2.3 Slave Home Screen (No Master)	10

1. Introduction

This document describes the Requirements of the proposed case study exercise “Multi Colour LED controller”.

1.1 The project

The project is to Design, develop, validate and demonstrate a suitable SW solution for the “Multicolor LED Controller” (MLC) as specified in the *Case Study Exercise “Multicolor LED Controller”*

The requirement is analysed through Use Case modelling and the corresponding Use Case diagram, Use Case description and User Interface Descriptions are covered in this document.

2 Use Case Diagram

As per the design analysis of the requirement, single binary must be burned into the board so that it acts according to jumper setting

- If jumper present – Slave Mode
- If Jumper Absent – Master Mode

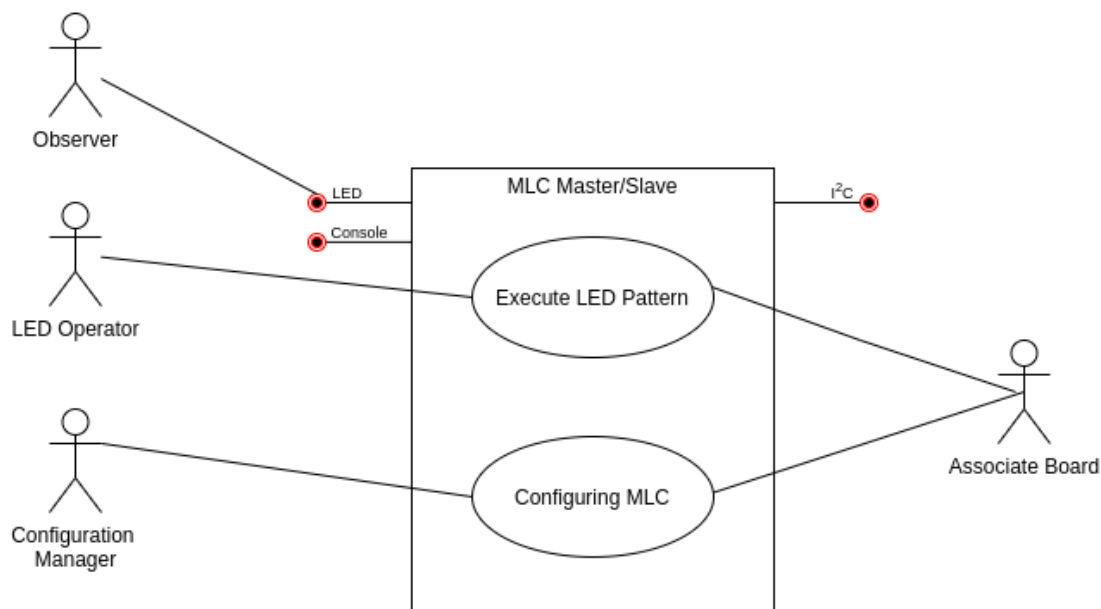


Figure 1.1: Use Case Diagram



3 Use Case Description

According to the use case diagram, two use cases were identified as potentially important to be considered for the design analysis.

- Configuring MLC
- Executing LED Pattern

3.1 Use case 1 : Configuring MLC

The board must be identified as Master/Slave according to the jumper settings and configure that as Master/Slave.

Who: Configuration Manager

Objective: To configure the MLC using user input.

3.1.1 Scenarios:

- If Master found slave
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On the Master console, show the home screen.
 - Display the current Master Home Screen (Default) as shown in UI section
 - Prompt the user to enter the configuration to slave
 - After entering validate the user entered data, if any data is error, alert the user about the error in data entered.
- If Master not found slave
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On the Master console, show the home screen.
 - Display No Slave Found
 - Run Master in default configuration
- If Slave not found master
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On slave console, show home screen of slave
 - Show status *"waiting for master"*



3.1.2 Sequence Diagram : Configuring MLC

Use Case : Configuring MLC

Scenerios

1. Master with slave configured
2. Master with slave not found
3. Master not found

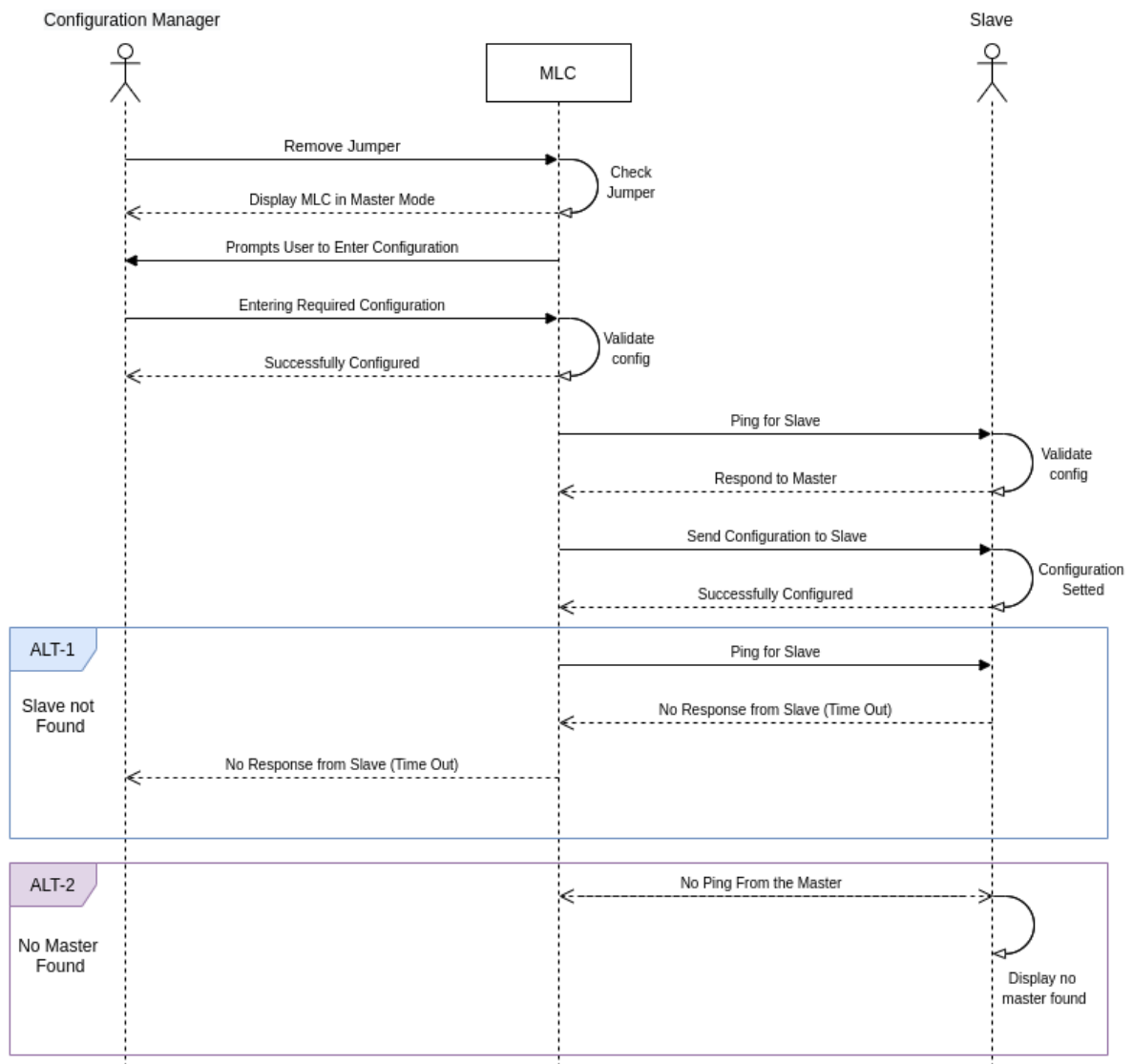


Figure 3.1: Sequence Diagram : Configuring MLC



3.2 Use case 2 : Execute LED Pattern

In this use case, from the user input data, the LED pattern executor will take the start color, end color and other parameters and start execution.

Note:

In **SLAVE MLC**, START Color and END Color must be **reversed** and if MASTER MLC is in **UP** mode then SLAVE MLC must be in **DOWN** mode

Who: LED Operator

Objective: To glow LED in the user configuration

3.2.1 Scenarios:

- If Master found slave
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On the Master console, show the home screen.
 - Display the current Master Home Screen (Default) as shown in UI section
 - Prompt the user to enter the configuration to slave
 - After entering validate the user entered data, if any data is error, alert the user about the error in data entered.
- If Master not found slave
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On the Master console, show the home screen.
 - Display No Slave Found
 - Run Master in default configuration
- If Slave not found master
 - During POWERON, Check the jumper status, If Jumper found Set is as Slave, if jumper not found set is as slave
 - On slave console, show home screen of slave
 - Show status *"waiting for master"*



3.2.1 Sequence Diagram : Configuring MLC

Use Case : Execute LED Pattern

Scenerios

1. Execute pattern in both master and slave in default config
2. Execute pattern in master with no slave in default config
3. Execute pattern in master and slave with user config
4. Execute pattern in master with no slave in default config

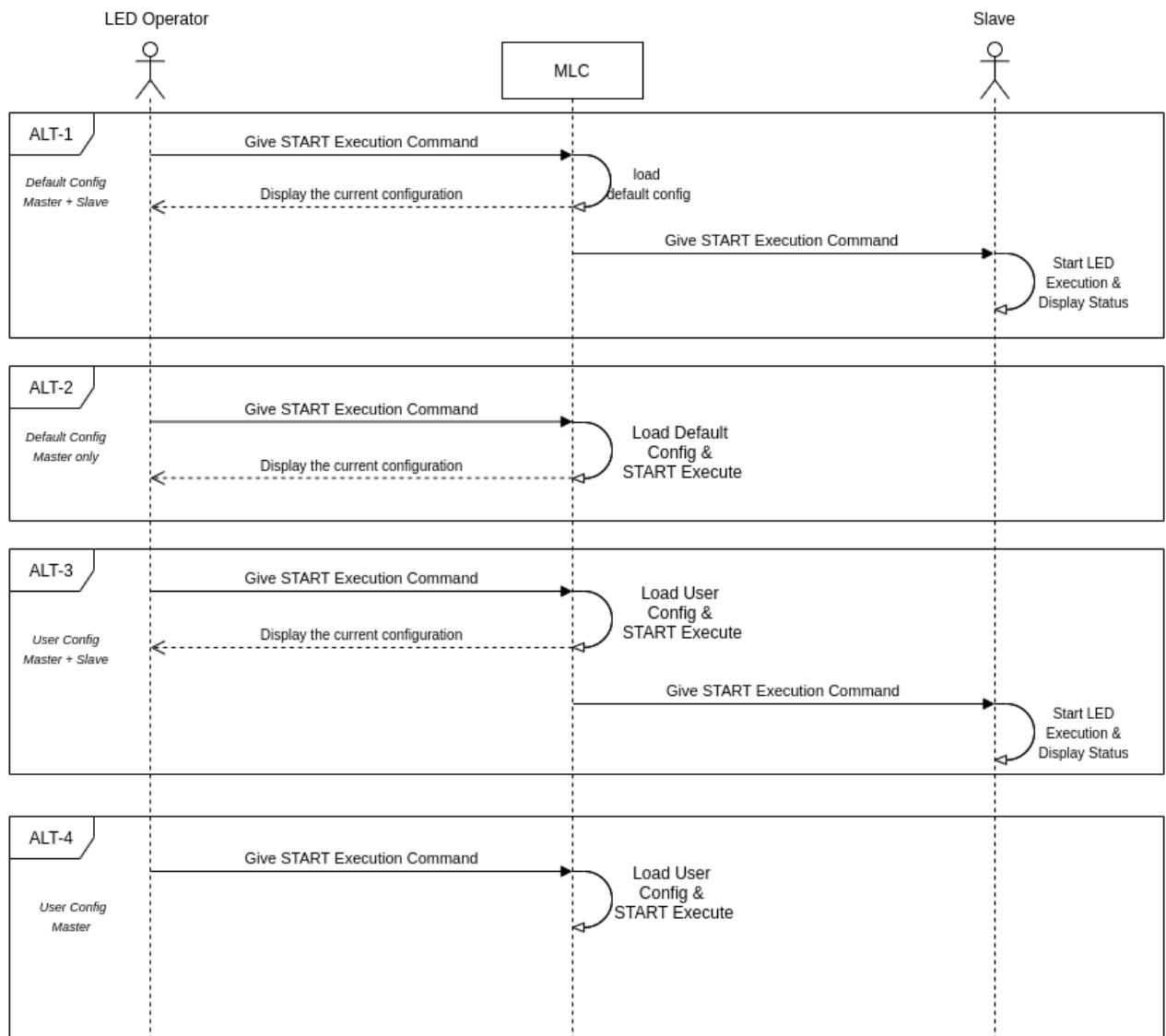


Figure 3.2: Sequence Diagram: Executing LED Pattern



4 User interface description:

The following section describes actors and interface.

4.1 Actors and Interface

4.1.1 Observer (Actor)

The person who verify the current status of the led

4.1.2 Associate Board (Actor)

If the identified board is setted as master then the associate board will be a slave, and vice versa

4.1.3 LED Operator (Actor)

The person who operates or changes the current status of LED

4.1.4 Configuration Manager (Actor)

The person who configures the MLC as Slave or Master

4.1.5 Console (Interface)

Serial communication will be implemented via USB from the MLC to the PC. The MLC acting as slave will be displaying slave console and the one with master configuration will be showing the master console.

Console	Functionality
Master Console	Read, Write
Slave Console	Read Only

Table 4.1: Console Details

4.1.6 RGB LED (Interface)

Since PWM is not available to the on board RGB LED, we need an additional LED adapter board to develop the prototype. PWM channels to headers as listed below

4.1.7 Inter Board Communication (Interface)

Communication between Master and slave can be done using I²C Protocol, the I2C can be available in the pin described below.

Header Pin	Pin Name	Header
SCL	PTE24	J2-20
SDA	PTE25	J2-18

Table 4.2: I²C Headers



4.2 UI - User Interface (Console)

Since the master and slave is allowed to connect to a serial console using a USB, there will be a console for displaying the status, configuration and as well as reading user input. Slave cannot accept the user input therefore ignores all entries by the user. Slave only displays the current status and configuration received from the master, also displays whether a master is detected or not.

In master's console, it will display the status, current executing configuration, details of the slave, provision to accept user configuration etc. The detailed design is illustrated below

4.2.1 Home Screen:

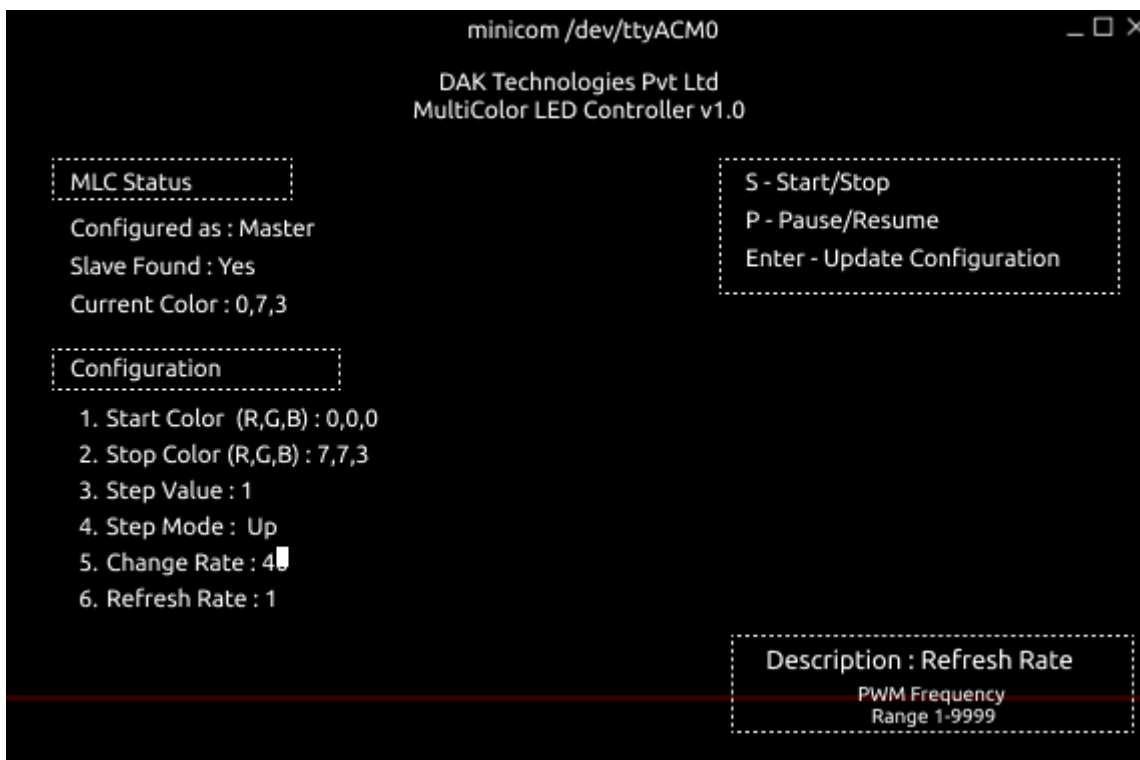


Figure 4.1: Master Console



4.2.2 Slave Home Screen (Master Found Successfully)

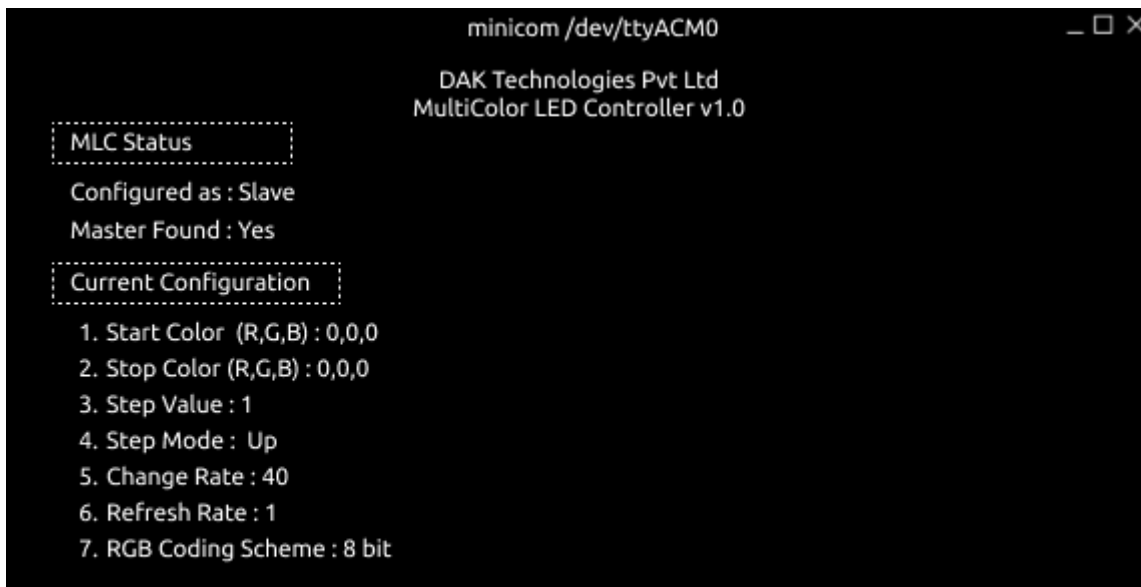


Figure 4.2: Slave Console

4.2.3 Slave Home Screen (No Master)

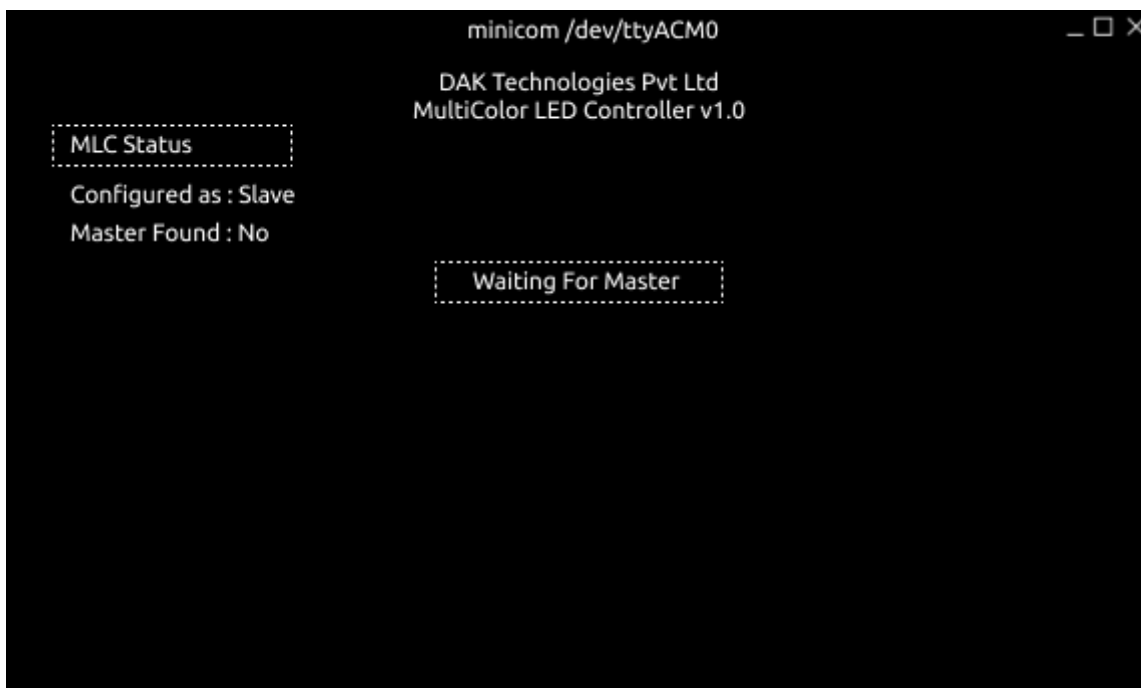


Figure 4.3: Slave Console - (No Master)