# PROJECT REPORT

# Design and Simulation of Circuits and Embedded Systems

## Project Title: RKE (Remote Keyless Entry)


**Submitted by:**

**99007670**

**99006771**

**99006772**

**99007673**

# Content

### i. RKE (Remote Keyless Entry)

**Problem Statement**

Remote keyless entry (RKE) system is a system designed to remotely lock or unlock access to automobiles. RKE functions are printing lock , unlock, alarm activation/deactivation, approach light.As many of us know there were lot of thefts happening in recent times in automobiles either the vehicle or personal things so here is the solution where we can unlock our automobiles with RKE.

# 1. Introduction

The term far off keyless framework (RKS), likewise called keyless section or distant focal locking, alludes to a lock that involves an electronic controller as a key which is enacted by a handheld gadget or naturally by nearness. Broadly utilized in vehicles, a RKS fills the roles of a standard vehicle key without actual contact.

### 1.1 Description

Now a days as we are noticing that many of the automobiles work on this framework it is widely used for remotely lock or unlock vehicles as earlier many of the car manufacturers have gone through this process but at present even many motor vehicles following this framework to keep their automobiles secured. Actually the principle behind this is RKE transmission requires two parts - a transmitter and a recipient. Transmitter - RKE key dandy, other ID gadget with RKE coordinate Recipient - Body Control ECU, other ECU with coordinated RKE. It works by communicating radio waves on a specific recurrence unidirectionally. RKE frameworks execute encryption and moving code calculations to forestall vehicle criminals from blocking and ridiculing the messages. Typical RKE capacities are: Print lock - (Blue switch on-All driven on simultaneously), Print open - (Blue switch press twice All begun simultaneously),Print caution enactment/deactivation - (Blue switch press multiple times-All drove on in clockwise way),Print approach light - (Blue switch press multiple times-All drove on in enemy of clockwise way).This is the modern solution.

# 2. Requirements

2.1 High Level Requirements

| \|id | Description |
|---|---|
| HR01 | Function to print the lock |
| HR02 | Function to print the Unlock |
| HR03 | Function to print alarm activation/deactivation |
| HR04 | Function to print approach light |

2.2 Low Level Requirements

| id | Description |
|---|---|
| LR01 | Press switch once to turn on all led's at a time |
| LR02 | Press switch twice to turn off all led's at a time |
| LR03 | Press switch thrice to blink of all led's in clockwise manner |
| LR04 | Press switch four times to blink of all led's in anticlockwise manner |

## 2.3. SWOT


Fig 2.1 SWOT Analysis

## 2.4. 4W's & 1H

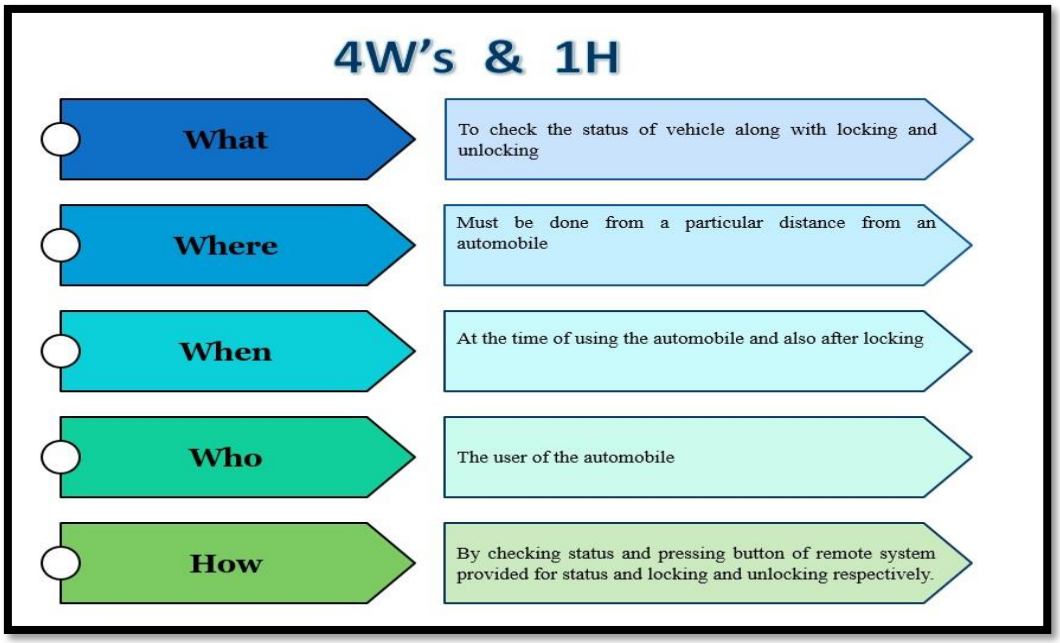


Fig 2.2 4W and 1H

# 3. Architecture

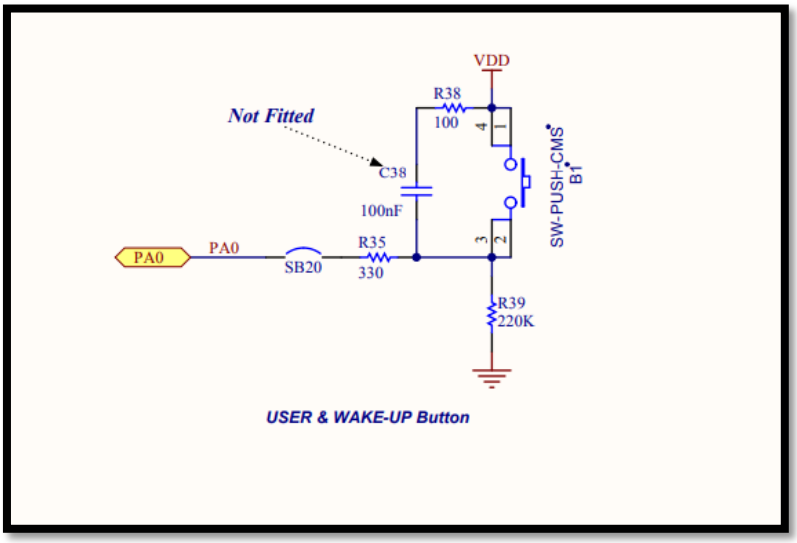


Fig 2.3 User and wake up Button circuit

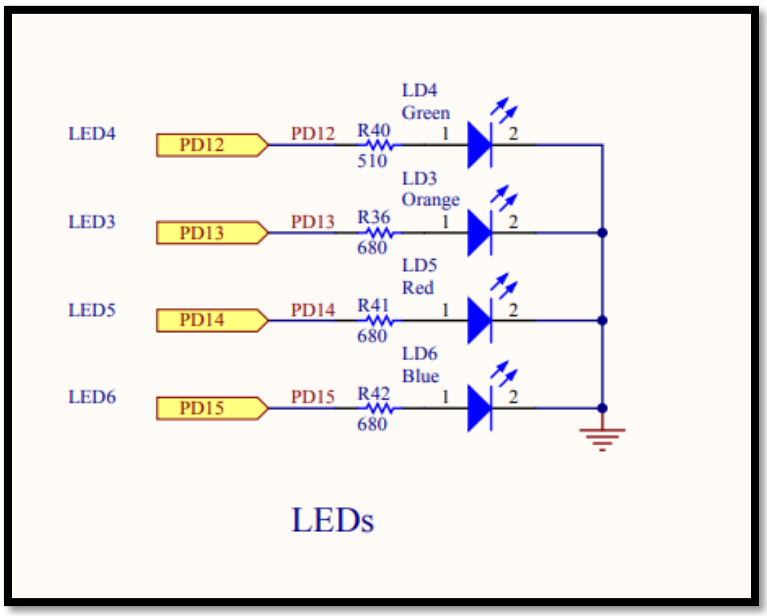


Fig 2.4 LED circuit

# 4. Test plan and Output

## 4.1 High level Test plan

| Test ID | Description | i/p | Expected o/p | Actual o/p | Status |
|---|---|---|---|---|---|
| HL_1 | Function to execute door lock | Press Blue Button once | All Leds ON | All Leds ON | pass |
| HL_2 | Function to execute door unlock | Press Blue Button twice | All Leds OFF | All Leds OFF | pass |
| HL_3 | Function to execute alarm activation/deactivation | Press Blue Button Thrice | All led on in clockwise manner | All led on in clockwise manner | pass |
| HL_4 | Function to execute approach light | press Blue Button four times | All led on in clockwise manner | All led on in clockwise manner | Pass |

# 5. Output

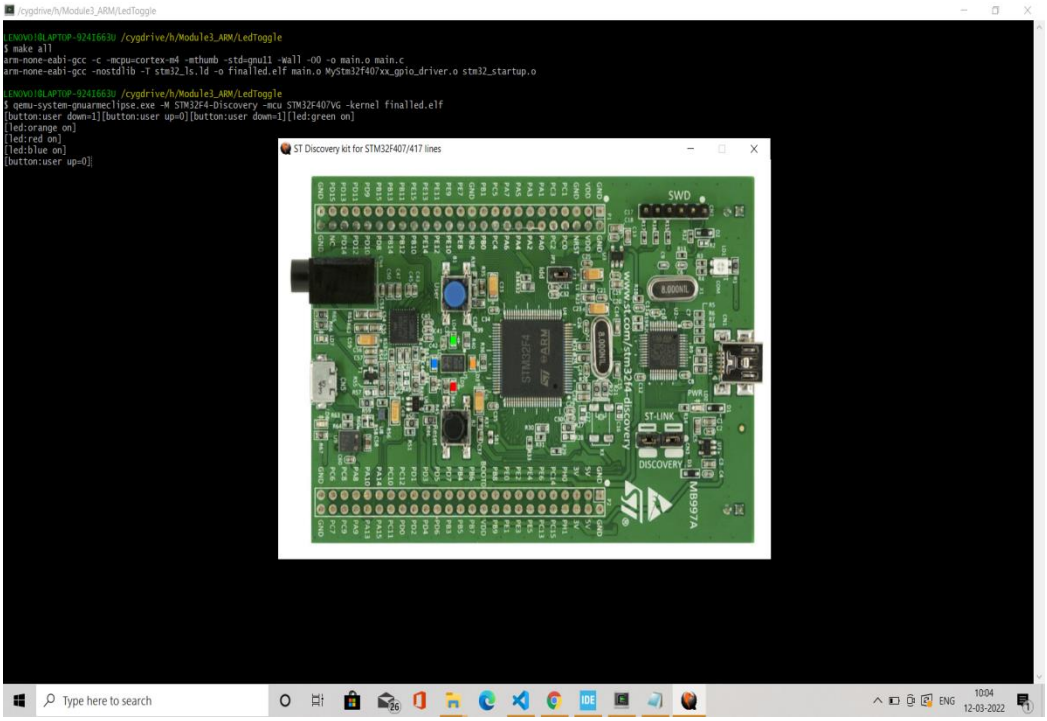


Fig 4.1 LED ON

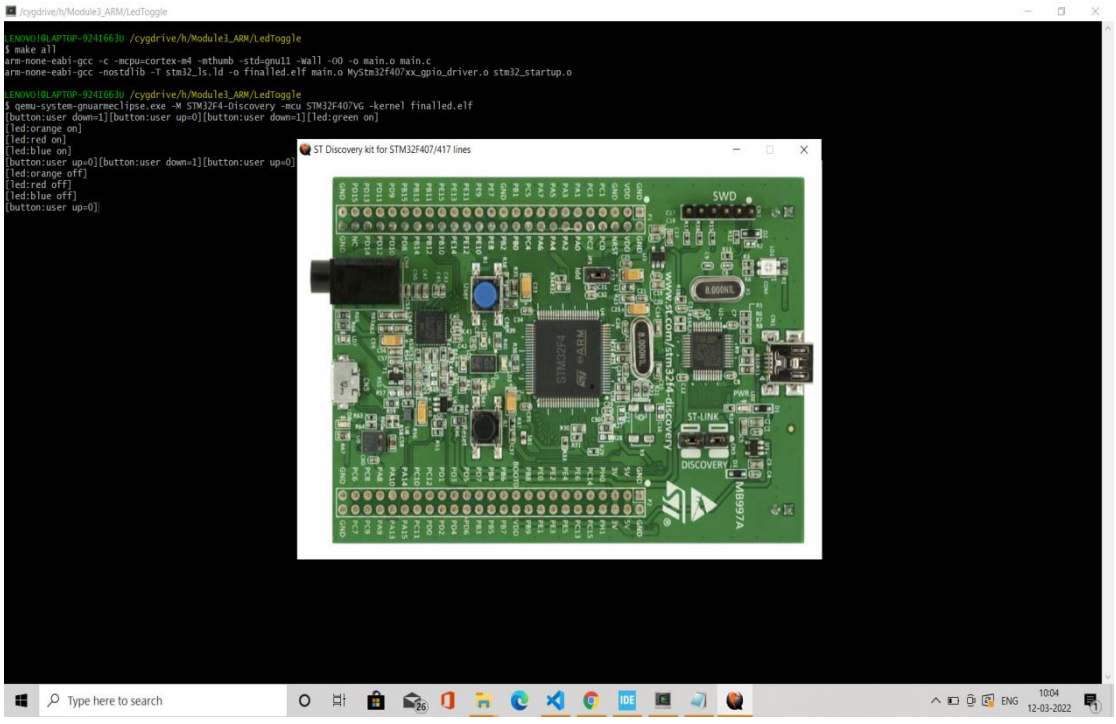


Fig 4.2 LED OFF

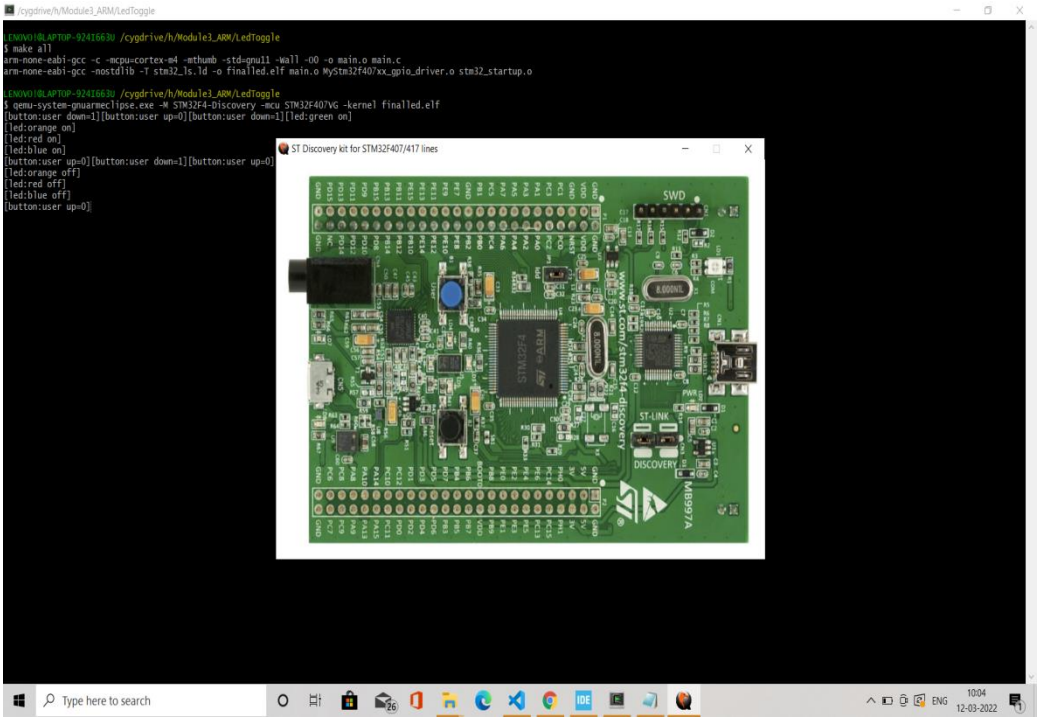
/cygdrive/h/Module3_ARM/LedToggle
LENOVO!@LAPTOP-924I663U /cygdrive/h/Module3_ARM/LedToggle
$ make all
arm-none-eabi-gcc -c -mcpu=cortex-m4 -mthumb -std=gnu11 -Wall -O0 -o main.o main.c
arm-none-eabi-gcc -nostdlib -T stm32_ls.ld -o finalled.elf main.o MyStm32f407xx_gpio_driver.o stm32_startup.o
LENOVO!@LAPTOP-924I663U /cygdrive/h/Module3_ARM/LedToggle
$ qemu-system-gnuarmeclipse.exe -M STM32F4-Discovery -mcu STM32F407VG -kernel finalled.elf
[button:user down=1][button:user up=0][button:user down=1][led:green on]
[led:orange on]
[led:red on]
[led:blue on]
[button:user up=0][button:user down=1][button:user up=0]
[led:orange off]
[led:red off]
[led:blue off]
[button:user up=0]
ST Discovery kit for STM32F407/417 lines
SWD
ST-LINK
DISCOVERY
Type here to search
ENG
10:04
12-03-2022

Fig 4.1 LED