



## PROJECT REPORT

จัดทำโดย

กลุ่ม RC6

นายณัฐนันท์ ภู่กาญจน์ 61070507206

นายภูธรนิก อารีย์สว่างกิจ 61070507216

นางสาวสโรชา วงศ์ศิริ 61070507222

นายประกาศิต ชัยเพ็ชร 61070507230

เสนอ

ผศ.สนั่น สระแก้ว

ผศ.สุรพจน์ ตุ่มนาค

อ.นันทิพัฒน์ นาคทอง

รายงานนี้เป็นส่วนหนึ่งของวิชา EMBEDDED SYSTEMS (CPE 328)

ภาคการศึกษาที่ 2 ปีการศึกษา 2563

คณะวิศวกรรมศาสตร์ ภาควิชาคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

### 1) Objective

สร้างอุปกรณ์ LED RGB ที่ได้รับข้อมูลจาก UART Protocol เพื่อควบคุมการทำงานของหลอดไฟ led รหัส APA102 ให้ทำตามคำสั่งที่ input เข้าไป เช่นเปิด-ปิดหลอดไฟ กำหนดความเร็วไฟวิ่งเป็นต้น

### 2) Features

- เปิด-ปิดหลอดไฟแต่ละดวง
- กำหนดสีของหลอดไฟแต่ละดวง
- กำหนดสีของหลอดไฟทุกดวง
- ทำไฟวิ่งตามความเร็วที่กำหนด
- กระพริบไฟตามความเร็วที่กำหนด
- ปิดหลอดไฟทุกดวง
- โหมดไฟวิ่งสีรุ้ง

### 3) Functional Block Diagram



รูปที่ 1 รูปแสดง Functional Block Diagram



---

## Timing diagram

เป็น Frame ที่ใช้สำหรับการคุย สื่อสารระหว่าง APA102 กับ ATmega328 โดยใช้ SPI Protocol

Main

START FRAME	DATA FRAME [LED1]	DATA FRAME [LED2]	DATA FRAME [LED3]	DATA FRAME [LED4]	DATA FRAME [LED5]	DATA FRAME [LED6]	DATA FRAME [LED7]	DATA FRAME [LED8]	END FRAME
----------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	-------------------------	--------------

Start frame

0x00	0x00	0x00	0x00
------	------	------	------

Data frame

0xe0 + Brightness	BLUE	GREEN	RED
----------------------	------	-------	-----

Brightness: 32 level (0 - 31)

RED: 256 level (0 - 255)

GREEN: 256 level (0 - 255)

BLUE: 256 level (0 - 255)

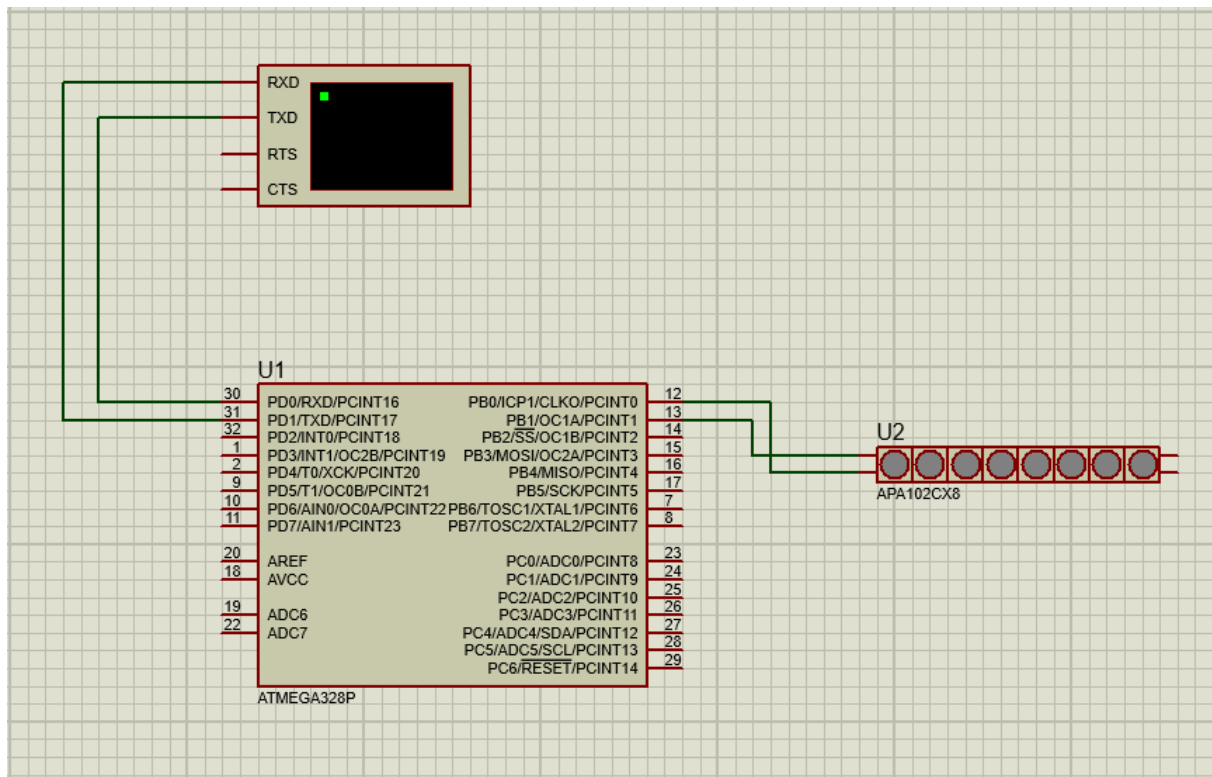
End frame

0xFF	0xFF	0xFF	0xFF
------	------	------	------

#### 4) How it work

1. ใช้ SPI ในการ สื่อสารระหว่าง ATmega328 ไป APA102
2. จากนั้น UART จะส่งข้อมูลต่อไปที่ Atmega328p เพื่อส่งสัญญาณการทำงานต่อไปที่ APA102
3. APA102 ทำตามคำสั่งที่ถูกส่งจาก User

#### Simulation



รูปที่ 2 รูปแสดงจากต่อวงจรจากโปรแกรม Proteus

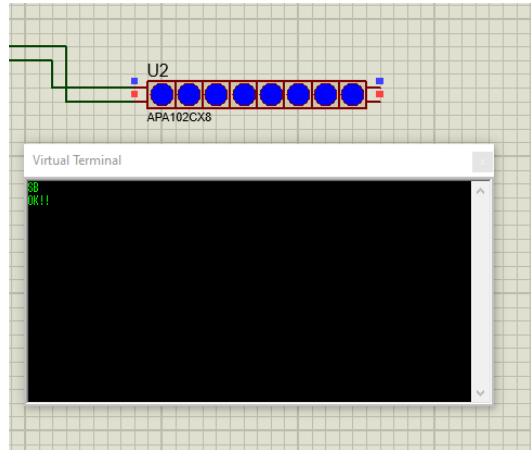


### Simulation from Proteus

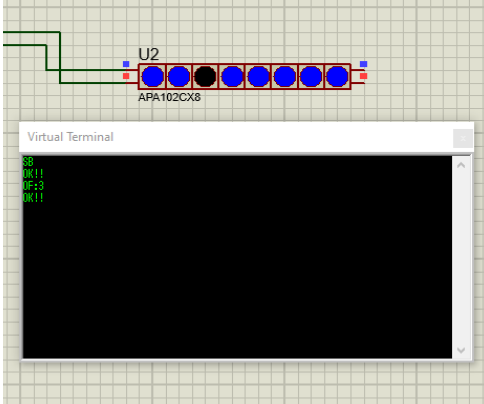
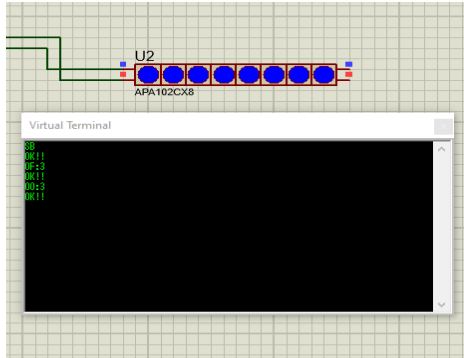
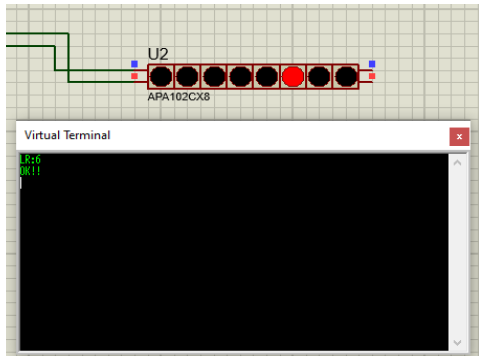
การใช้งานต้องพิมพ์คำสั่งลงไป แล้วกด Enter ถ้าพิมพ์ผิดไม่สามารถลบได้ และ ถ้าหลังจากมีการทำคำสั่งหนึ่งเสร็จแล้ว ต้องการไปคำสั่งใหม่ให้กด Enter จนกว่าจะขึ้นคำว่า WRONG COMMAND แล้วจะสามารถพิมพ์คำสั่งถัดไปได้

นี่คือรหัสสีที่ User สามารถใช้

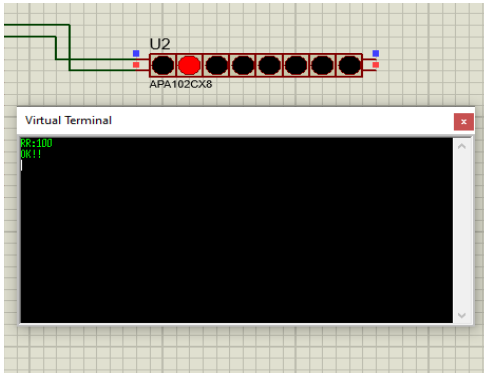
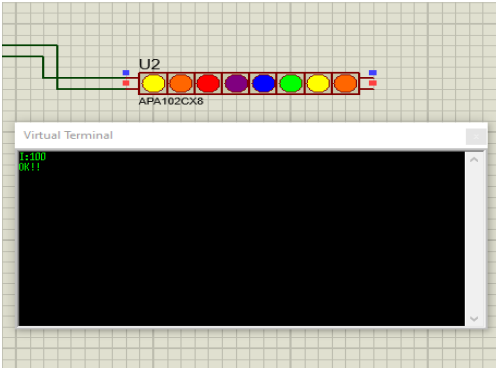
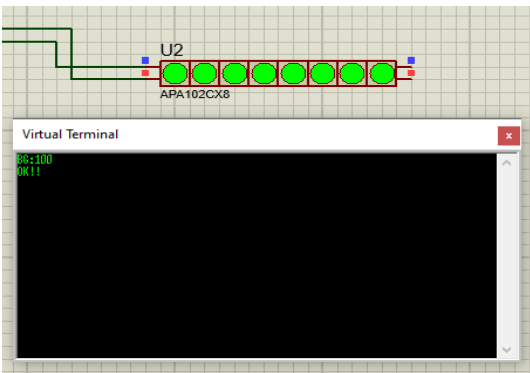
- R stands for Red.
- G stand for Green
- B stands for Blue
- O stand for Orange
- Y stands for Yellow
- P stands for Purple
- ถ้าพิมพ์ไม่ถูกจะได้สี BLUE ตลอด

Name	Command	Example	Output
กำหนดสีหลอดไฟทุกดวง	S<Color> โดยcolor คือสีที่ต้องการ	ต้องการให้ led ทุกหลอดเป็นสี น้ำเงิน	

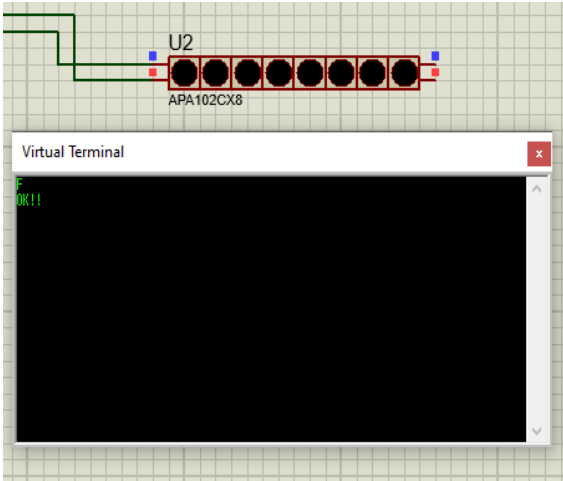


ปิดไฟดวงที่ ต้องการ	OO:<LEDn> โดย LEDn คือ หลอดไฟดวงที่ต้องการ	ต้องการปิด led ดวงที่ 3	
เปิดไฟดวงที่ ต้องการ	OF:<LEDn> โดย LEDn คือ หลอดไฟดวงที่ต้องการ	ต้องการเปิด led ดวงที่ 3	
กำหนดสี หลอดไฟที่ ต้องการ	L<color>:<LEDn> โดย color คือสี และ LEDn คือ หลอดไฟดวงที่ต้องการ	ต้องการตั้งให้ led ดวงที่ 6 เป็นสีแดง	

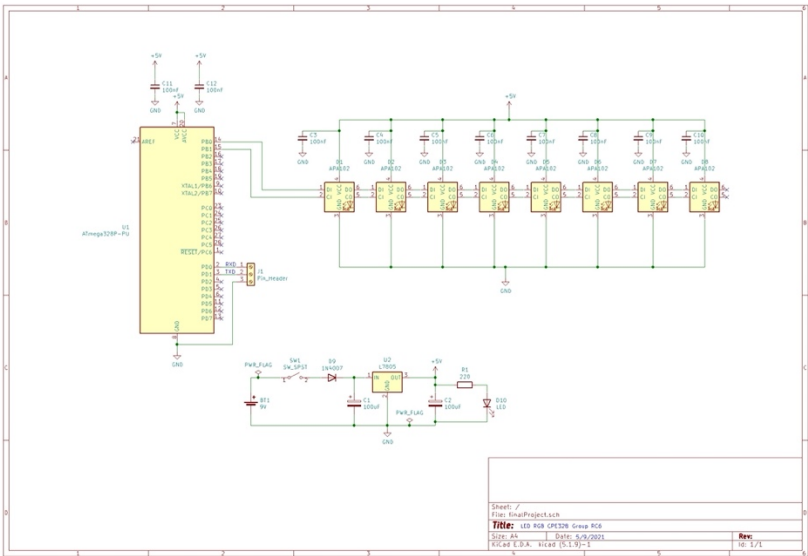


กำหนดไฟวิ่ง	R<Color>:<Delay time> โดย color คือสี และ Delay time คือ ความเร็วที่ต้องการ หน่วยเป็น ms(millisecond)	ต้องการทำไฟวิ่ง สีแดงที่มีความเร็ว 100ms	
กำหนดไฟวิ่งสีรุ้ง	I:<Delay time> โดย Delay time คือ ความเร็วที่ต้องการ หน่วยเป็น ms(millisecond)	ต้องการทำไฟวิ่ง สีรุ้งที่มีความเร็ว 100ms	
กำหนดไฟกระพริบ	B<Color>:<Delay time> โดย color คือสี และ Delay time คือความเร็วที่ต้องการ หน่วยเป็น ms(millisecond)	ต้องการทำไฟกระพริบสีเขียวที่มีความเร็ว 100ms	



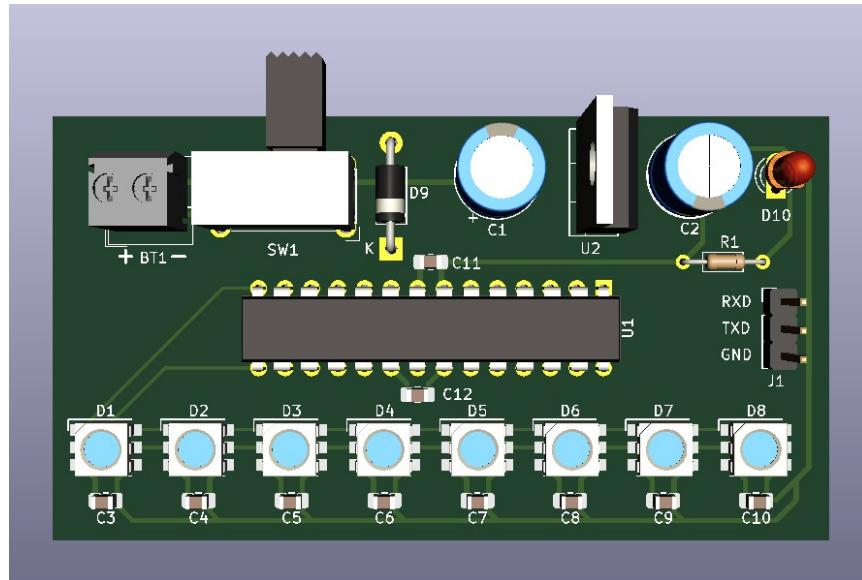
ปิดไฟทุกดวง	F	ปิดไฟทุกดวง	
-------------	---	-------------	---

5) Circuit schema and PCB

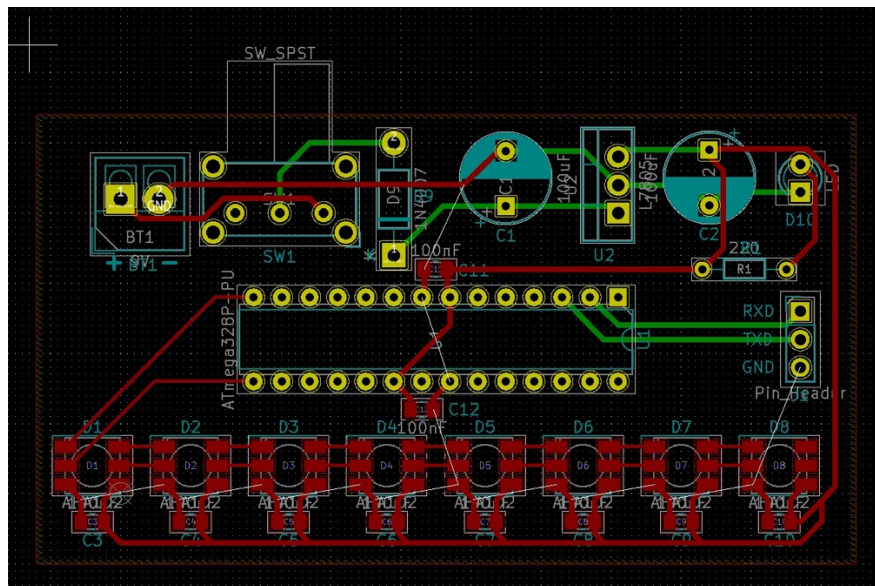


รูปที่ 3 รูปแสดงวงจร Schema จากโปรแกรม KiCad





รูปที่ 4 รูปแสดง Top View ของแผ่น PCB



รูปที่ 5 รูปแสดงบอร์ด PCB ที่ทำการ Routing (ก่อนเท Ground)