UNIVERSITY OF TEHRAN

Instrumentation

Homework 3

Spring 2025

Amir Shahang

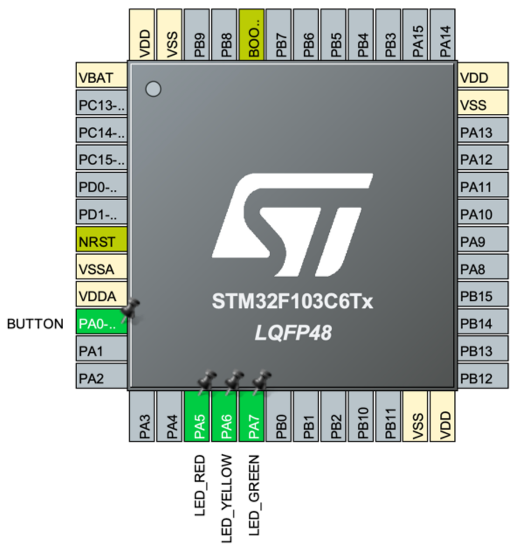
810101448

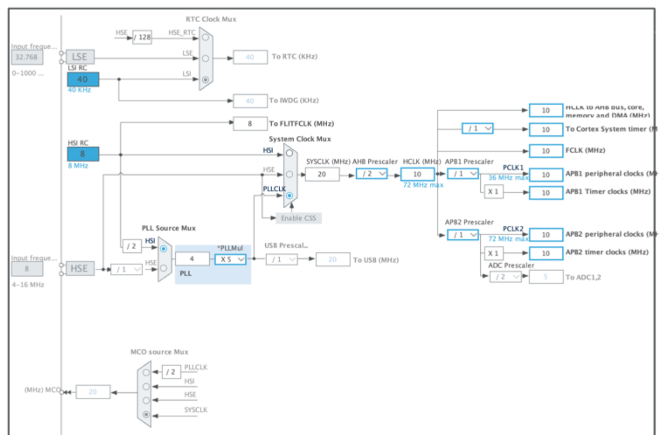
ابتدا با توجه به شماره دانشجویی خود میکروکنترلر و کلاک اصلی آن را انتخاب میکنیم :

SID % 3 = 0 =>   STM32F103C6
SID % 9 = 0 =>   HCLK = 10 MHz

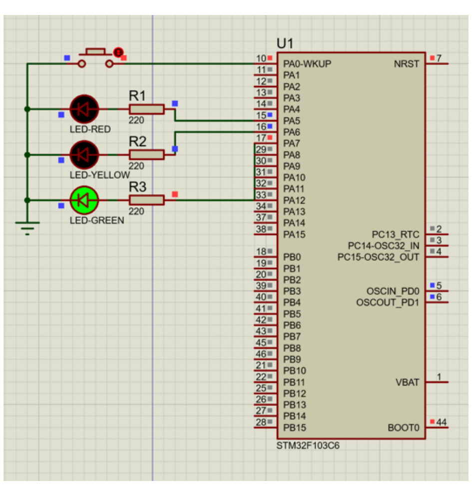
Q 1 :

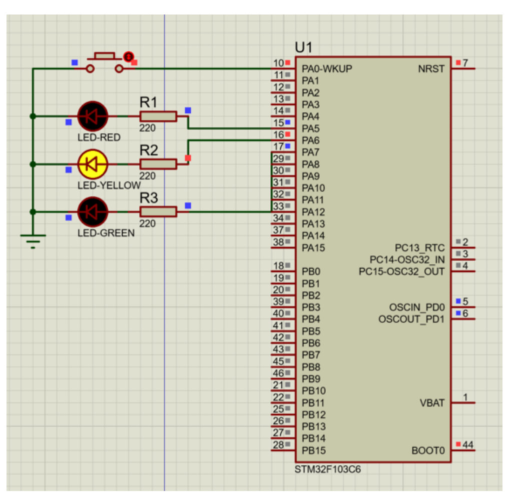
با توجه به توضیحات داده شده در صورت تمرین میکروکنترلر مورد نظر را انتخاب کرده و پایه ی PA0 را برای Push Button در نظر گرفته و آن را به صورت Pull UP قرار میدهیم سپس سه خروجی برای 3 وضعیت مدار قرار داده و Generate Code میکنیم

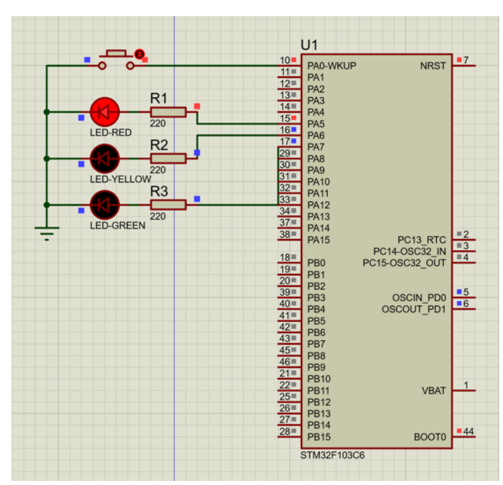




در فایل Main.c ایجاد شده با استفاده از دستورات HAL_GPIO_ReadPin و HAL_GPIO_WritePin کدی نوشته تا با هربار Push کردن رنگ چراغ به وضعیت بعدی برود سپس کد نوشته شده را Build کرده و در نرم افزار Proteus آن را اجرا میکنیم:

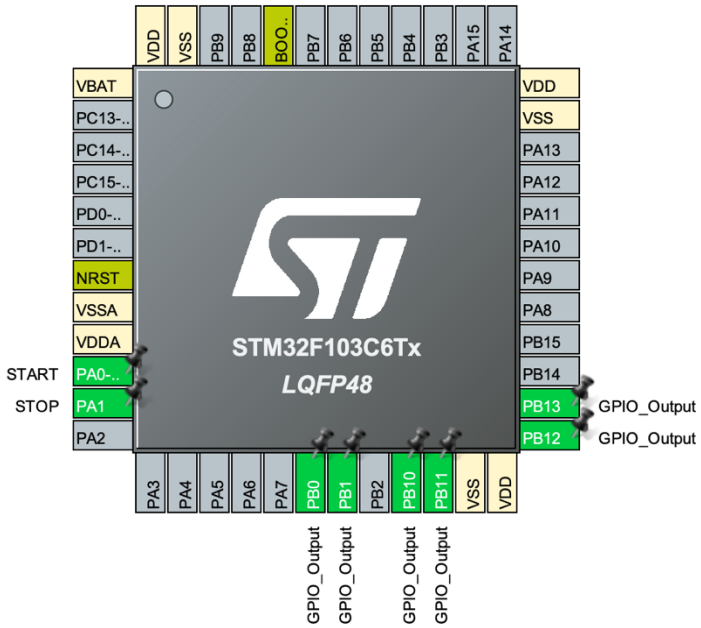


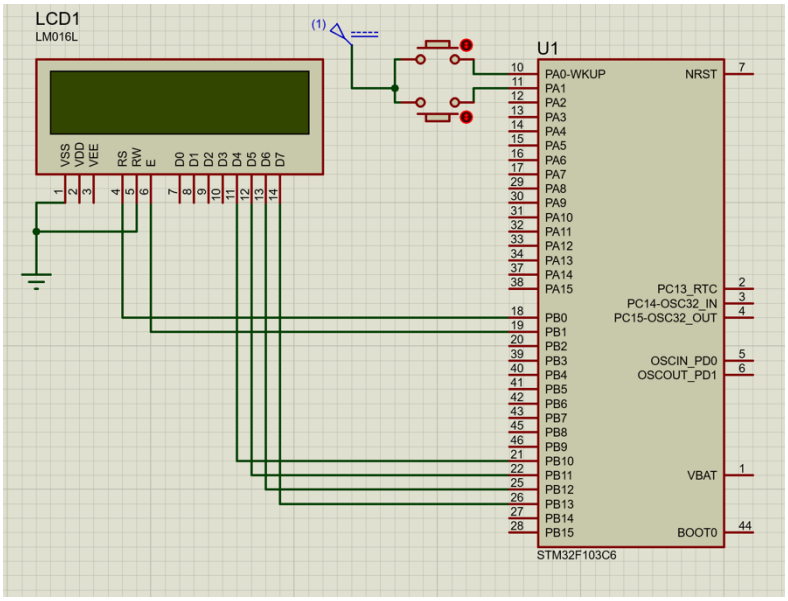




Q 2 :

برای پیاده سازی این سوال دو پایه PA0 و PA1 را برای شروع و پایان تایمر مطابق شکل زیر Push Button میکنیم سپس برای نشان داده تایمر به صورت mm:ss به چهار خروجی نیازمندیم که از پایه های PB10 تا PB13 استفاده شده است. پایه های PB0 و PB1 نیز برای Rs و En تایمر لحاظ میشود





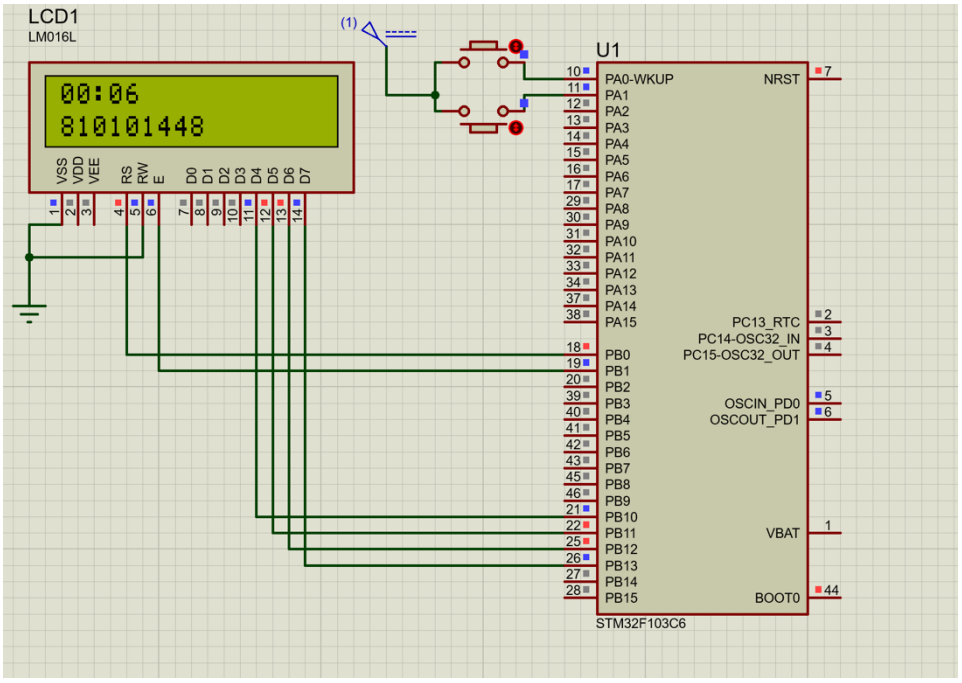
برای پیاده سازی این سوال نیاز به استفاده از Interrupt داریم
Interrupt یا وقفه در میکروکنترلر یه مکانیزمه که وقتی یه رویداد خاص (مثل تموم‌شدن تایمر، زدن دکمه، یا دریافت داده) اتفاق میفته، CPU به‌طور خودکار از برنامه اصلی خارج میشه و میره یه تابع خاص اجرا کنه. در این سوال تایمر تنظیم میشود که هر 1 ثانیه یک بار پر شود (over flow) و وقتی این اتفاق میفتد یک Interrupt ایجاد شده وCPU میرود تابعی به اسم HAL_TIM_Base_Start_IT را اجرا کند.

برای شمردن تایمر نیاز است تا Timer داخلی میکرو را فعال کنیم با توجه به فرکانس کلاک Timer به صورت زیر تنظیم میشود :
Interrupt داخلی تایمر را نیز فعال میکنیم.

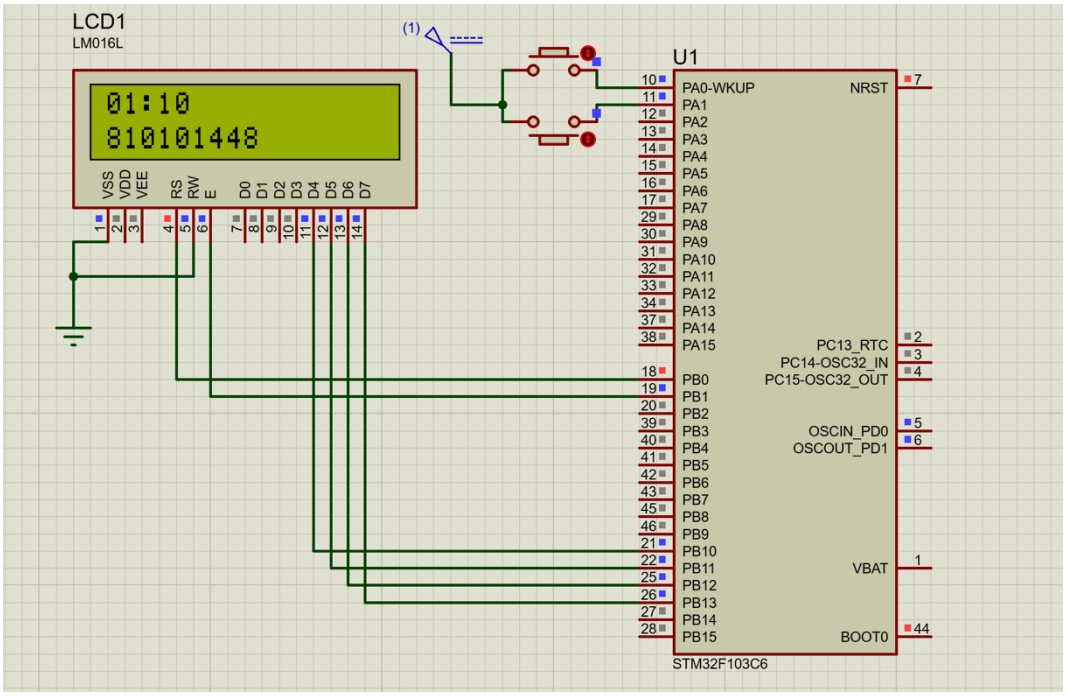
$$\text{Timer} = \frac{\text{HCLK}}{(\text{PSC}+1)(\text{ARR}+1)}$$

برای نمایش زمان نیاز به یک LCD Alphanumeric 16*2 داریم. برای این کار از ماژول LM016l استفاده میکنیم و خروجی را نمایش میدهیم :

با زدن Start Button تایمر شروع به شمردن میکند:



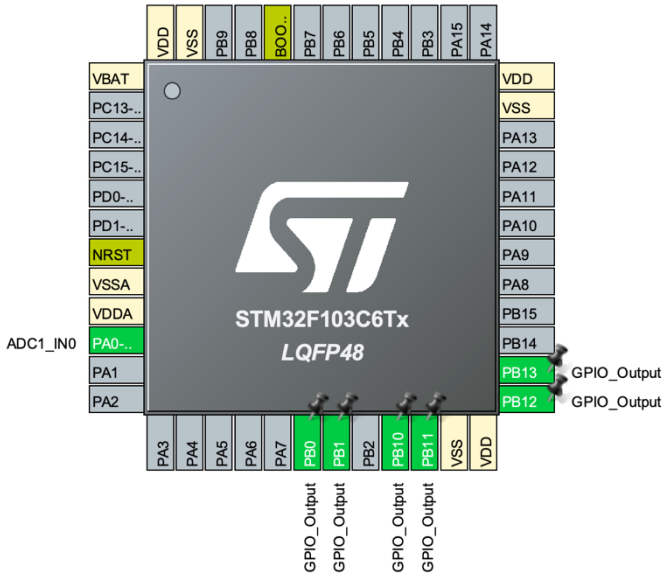
با زدن Stop Button تایمر متوقف میشود :

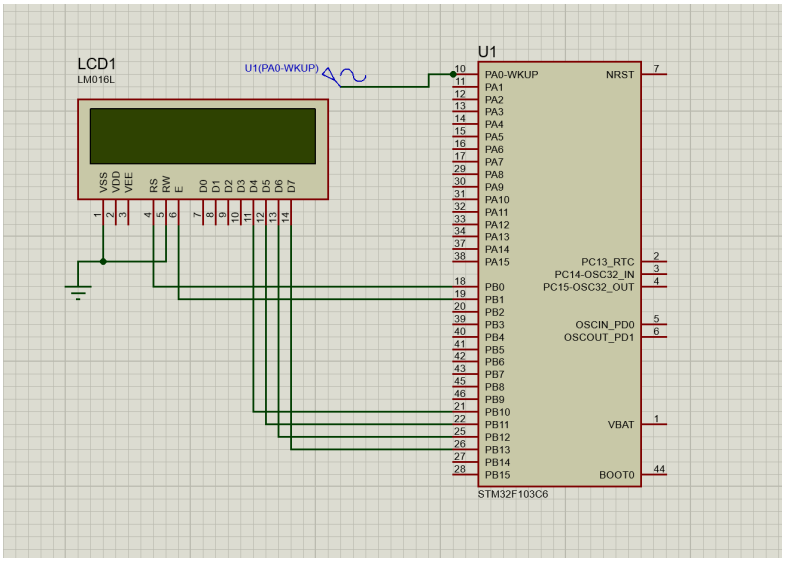


Q 3 :

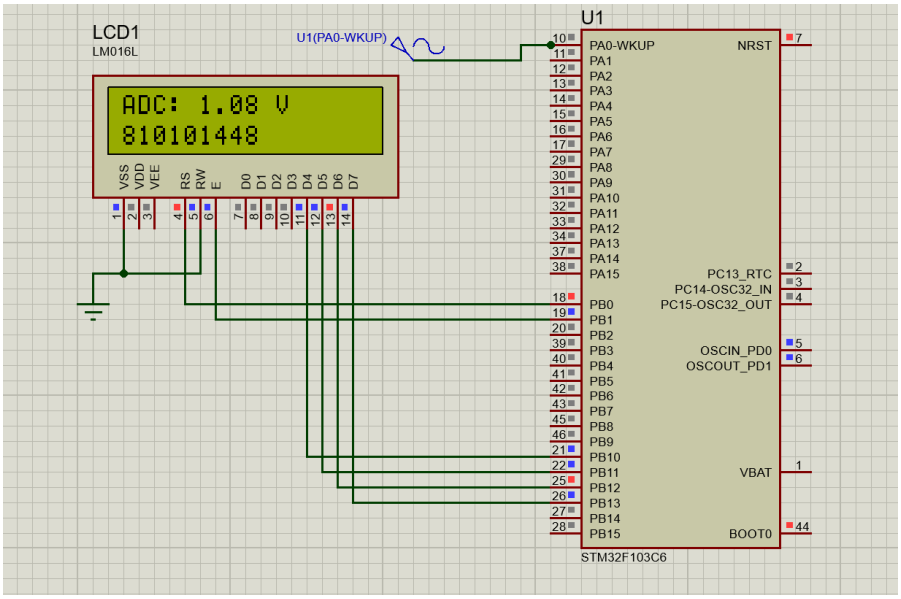
در STM32 واحد ADC به ما امکان می‌دهد که سیگنال‌های آنالوگ مانند ولتاژ متغیر یا خروجی سنسورها را به صورت دیجیتال بخوانیم. این واحد 12 بیتی است و می‌توان آن را از طریق STM32CubeIDE روی پایه‌های مختلفی مانند PA0 (ADC_IN0) فعال کرد.
برای استفاده از آن، ابتدا پایه را روی حالت ADC تنظیم کرده و سپس با توابعHAL مقدار آنالوگ را به عدد دیجیتال تبدیل می‌کنیم

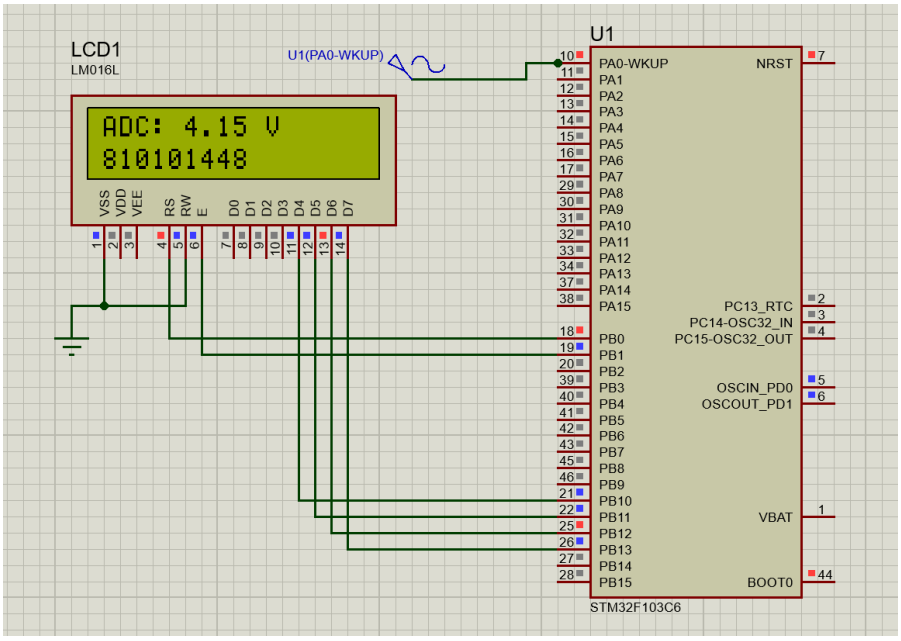
در این سوال لازم داریم یک مبدل آنالوگ به دیجیتال به Stm32 اضافه کنیم و یک پایه برای گرفتن ورودی آنالوگ تعیین کنیم.
پس از انتخاب میکروکنترلر مورد نظراز قسمت Analog > ADC1 مبدل را اضافه کرده و پایه ی PA0 را به آن اختصاص میدهیم
سپس برنامه ای مینویسیم که با استفاده از ADC سیگنال مورد نظر را نمونه برداری کند و مقادیر نمونه برداری شده را با دو رقم اعشار در سطر اول یک
LCD Alphanumeric 16*2 نمایش میدهیم سپس کد نوشته شده را Build کرده و در نرم افزار Proteus آن را اجرا میکنیم. برای این کار سیگنال سینوسی با دامنه 2v و آفست 3v ایجاد کرده تا دامنه بیش از5 ولت نداشته باشد و شامل مقادیر منفی ولتاژ نشود:

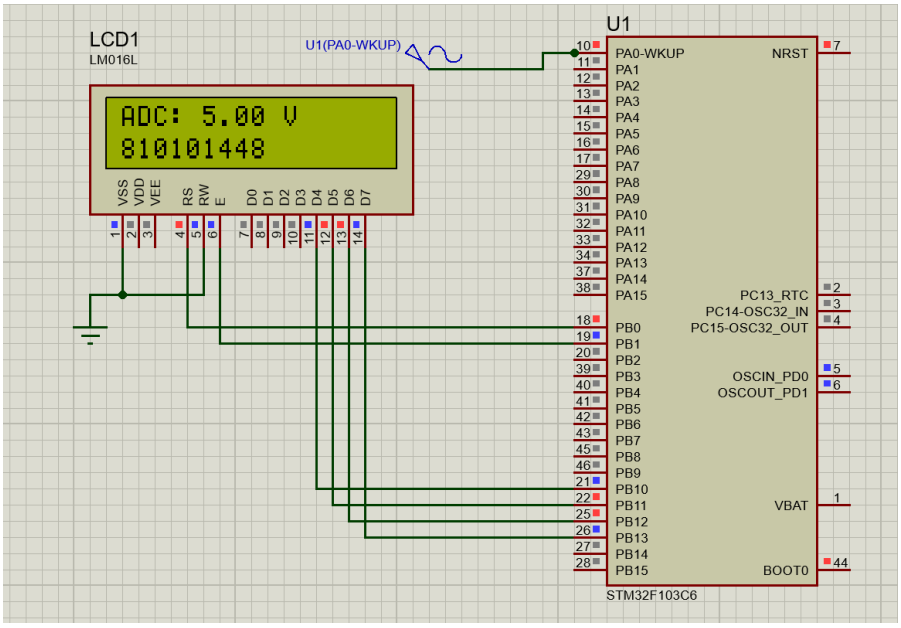




با اجرا کردن فایل hex. تولید شده خروجی های زیر به دست میاید:
مشاهده میشود که خروجی LCD به درستی بین 1 تا 5 ولت قرار میگیرد


LCD1
LM016L
ADC: 1.08 V
810101448
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1(PA0-WKUP)
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6


LCD1
LM016L
ADC: 4.15 V
810101448
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1(PA0-WKUP)
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6


LCD1
LM016L
ADC: 5.00 V
810101448
VSS VDD VEE RS RW E D0 D1 D2 D3 D4 D5 D6 D7
U1(PA0-WKUP)
U1
PA0-WKUP
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PA8
PA9
PA10
PA11
PA12
PA13
PA14
PA15
PB0
PB1
PB2
PB3
PB4
PB5
PB6
PB7
PB8
PB9
PB10
PB11
PB12
PB13
PB14
PB15
NRST
PC13_RTC
PC14-OSC32_IN
PC15-OSC32_OUT
OSCIN_PD0
OSCOUT_PD1
VBAT
BOOT0
STM32F103C6

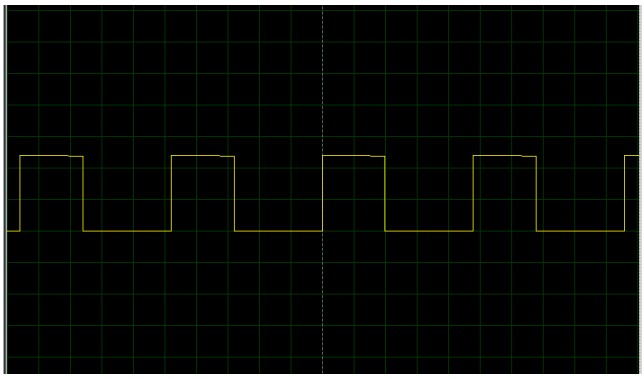
Q 4 :

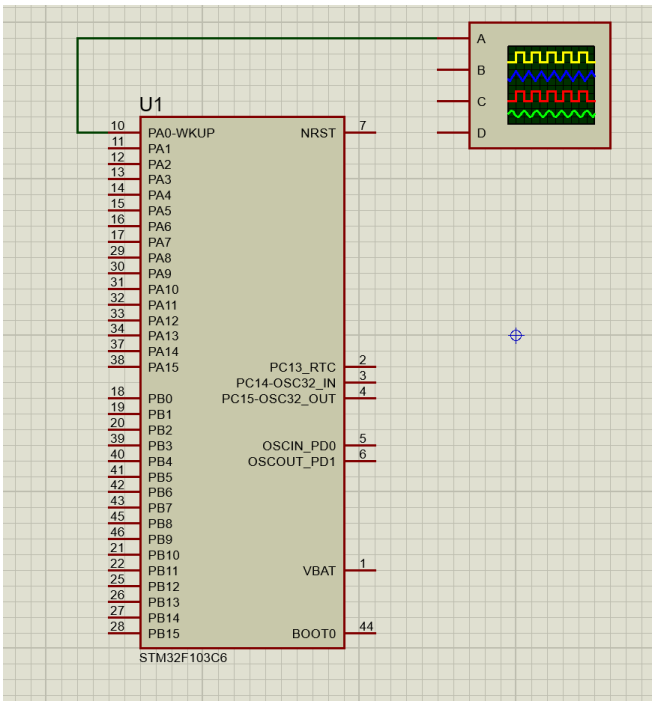
در میکروکنترلرهای STM32 ، واحد TIM یک واحد سخت‌افزاری پرکاربرد است که امکان ایجاد وقفه‌های زمانی، تولید سیگنال PWM ، شمارش پالس، و اندازه‌گیری زمان را فراهم می‌کند. این واحد را می‌توان از طریق نرم‌افزار STM32CubeIDE به پروژه اضافه کرد. برای این کار، پایه مورد نظر را روی حالت تایمر تنظیم کرده و در تنظیمات تایمر پارامترهایی مثل Prescaler ، ARR و Mode را مشخص می‌کنیم. سپس با استفاده از توابع HAL می‌توان تایمر را راه‌اندازی و از آن در برنامه استفاده کرد

با تغییر پارامتر ها تایمر طبق فرمول زیر یک سیگنال PWM با فرکانس 100 هرتز میسازیم :

$$\text{PWM freq} = \frac{\text{HCLK}}{(\text{PSC}+1)(\text{ARR}+1)} \quad => \text{ARR} = 999\,,\text{PSC} = 99$$

سپس CCR1 رو نیز روی 40 قرار میدهیم تا دیوتی سایکل سیگنال برابر 40% شود





سپس با استفاده از ADC و یک پتانسیومتر دیوتی سایکل رو از 10% تا 90% تغییر میدهیم :