



## راه اندازی میکروکنترلر STM32f407VGT6 و کار با GPIO

هدف از این تمرین راهاندازی بخشهای اولیه و اصلی میکروکنترلر و آشنایی با نحوه کار با GPIO میباشد.

برای این منظور یک پروژه هی سیستم آبیاری خودکار به عنوان مثال تعریف شده است. این سیستم برای سنجش میزان رطوبت خاک و آبیاری خودکار به کار خواهد رفت. سیستم مذکور متشکل از چندین بخش خواهد بود. این بخش شامل ماژول رطوبتسنج، میکروکنترلر،

keypad، LCD و شیر آب میشود.

### بخش ۱) ماژول رطوبت سنج خاک

برای سنجش میزان رطوبت خاک انواع مختلفی از ماژو لها وجود دارند. ماژول مورد استفاده در این تمرین، YL-69 خواهد بود (ماژول TE215 هم مشابه YL-69 بوده و تفاوت چندانی با آن ندارد). دلیل انتخاب این ماژول، قیمت کم آن و مناسب بودن برای پروژه های ساده میباشد. شکل زیر این ماژول را نشان میدهد:

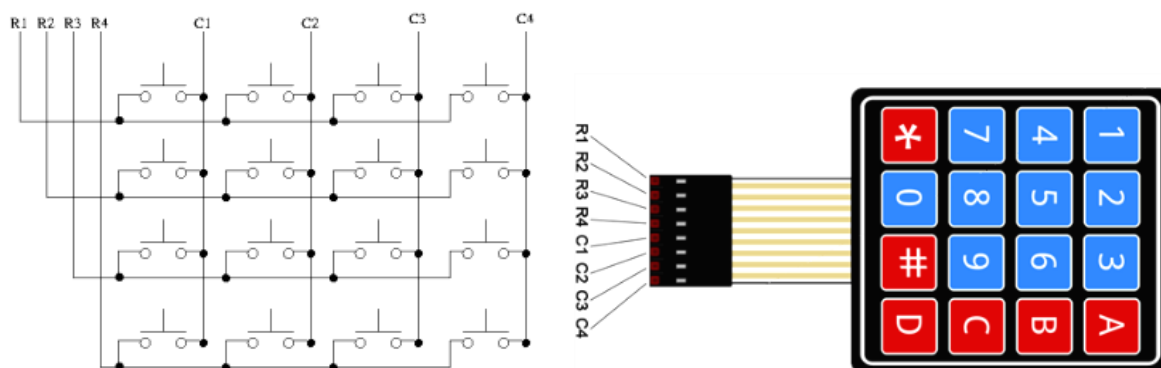


این ماژول دارای چهار پین VCC، GND، AO و DO است. پین AO مربوط به خروجی آنالوگ و پین DO مربوط به خروجی دیجیتال میباشد. خروجی آنالوگ ولتاژی متناسب با میزان رطوبت خاک تولید میکند و به منظور اندازه گیری این ولتاژ نیاز به راهاندازی واحد ADC میکروکنترلر است که در این تمرین ما با این پین کاری نخواهیم داشت. خروجی دیجیتال بدین شکل است

که اگر رطوبت از حد معینی کمتر باشد، مقدار ۱، و اگر از آن حد معین بیشتر باشد مقدار ۰ به خود میگیرد. این حد هم توسط پتانسیومتر آبی رنگ که روی ماژول قرار دارد، قابل تنظیم است. پین DO را به پین PB1 میکروکنترلر وصل کنید.

## بخش ۲) کیپد (keypad)

کیپدها در واقع ماتریسی از push button ها هستند که در یک پد قرار گرفته اند. شکل سمت راست زیر یک کیپد 4\*4 و شکل سمت چپ نحوه ی اتصالات درون آن را نشان میدهد.



مزیت کیپد این است که تعداد پین کمتری برای اتصال نیاز دارد (مثلا برای کیپد فوق ۸ پین به جای ۱۶ پین). اما خواندن وضعیت کلیدها به سادگی یک push button نیست. الگوریتمهای مختلفی برای این کار ارائه شده است. یکی از این الگوریتم ها شیفت دادن یک 0 بین سطرها کیپد و خواندن وضعیت ستونهای آن است.

برای انجام این کار یک لینک برای دسترسی به کتابخانه مورد نیاز برای تشخیص باتن فشار داده شده قرار داده می شود :

[https://stm32f4-discovery.net/download/tm\\_stm32f4\\_keypad/?wpdmdl=1787](https://stm32f4-discovery.net/download/tm_stm32f4_keypad/?wpdmdl=1787)

<https://stm32f4-discovery.net/download/tm-stm32f4-gpio-library/?wpdmdl=2237>

<https://stm32f4-discovery.net/download/defines-h/?wpdmdl=1769>

سه لینک بالا حاوی سورس و هدر فایل های مورد نیاز برای راه اندازی یک keypad را در اختیار ما می گذارد. این کتابخانه برای تنظیم پایه ها از کتابخانه gpio نوشته شده توسط نویسنده استفاده شده. هدر defines.h برای انجام define های جدید استفاده می شود و قابل حذف است اما می توانید به جای انجام define در این فایل، پایه های پیش فرض اتصال پایه های keypad را در tm\_stm32f4\_keypad.h تغییر دهید. این کتابخانه قابلیت تشخیص یک ورودی جدید هر ۱۰۰ میلی ثانیه را داراست که می توانید مقدار پیش فرض آن را با تغییر KEYPAD\_READ\_INTERVAL انجام دهید. همچنین باید به این دقت شود که این کتابخانه قابلیت کار با keypad های 4\*3 را نیز دارد. برای تنظیم آن به صورت 4\*4 بایست در تابع init ورودی به صورت TM\_KEYPAD\_Type\_Large باشد. برای توضیحات بیشتر و دسترسی به برنامه نمونه به لینک زیر مراجعه کنید.

<https://stm32f4-discovery.net/2014/09/library-32-matrix-keypad-stm32f4xx>

در برنامه نمونه ارائه شده در لینک بالا بین هر **case** که نشانه ی فشردن یک کلید است می توانید عملیات مربوط به آن کلید را انجام دهید. همچنین به جای استفاده از **void TM\_DELAY\_1msHandler(void)** که نیاز به **include** کردن **"tm\_stm32f4\_delay.h"** است که نیاز به اضافه کردن هدر و سورس فایل های بیشتری است ، از

**void HAL\_SYSTICK\_Callback(void)** استفاده کنید (این تابع به وقفه ی **systick** سرویس می دهد) و به جای **TM\_DELAY\_Init()** از **HAL\_SYSTICK\_Config(1)** استفاده کنید که در سورس فایل **stm32f4xx\_hal\_cortex.c** وجود دارد. (زیرا ما از توابع **HAL** استفاده می کنیم). کار این تابع تنظیم فاصله بین دو وقفه ی **systick** است که می خواهیم یک میلی ثانیه باشد. **Systick** یک تایمر جانبی ارائه شده در ساختار **ARM cortex** است که در **cubemx** به صورت پیش فرض برای ایجاد تاخیر استفاده می شود.

**TM\_KEYPAD\_Read()** همیشه حاوی کلید فشرده شده است. پس می توانید از **if** هم به جای **switch** استفاده کنید.

اگر با وقفه آشنایی ندارید به این صورت عمل می کند که در صورت رخداد یک وقفه (مثلا فاصله های یک میلی ثانیه که با تابع **HAL\_SYSTICK\_Config(1)** تنظیم کردیم). اعمال در حال انجام در **while(1)** رها می شوند و به زیر برنامه سرویس دهی وقفه می رود و آن اعمال را اجرا می کند. (برای این مثال اعمالی که در تابع **void HAL\_SYSTICK\_Callback(void)** برنامه نویسی می کنیم)

### بخش ۳) شیر آب الکتریکی:

باز و بسته شدن شیر آب با توجه به میزان رطوبت، (در صورت صفر بودن پایه ی **DO**) و میزان باز ماندن شیر آب با توجه به زمانی که توسط کاربر (با استفاده از کیپد) وارد میشود، تعیین خواهد شد.

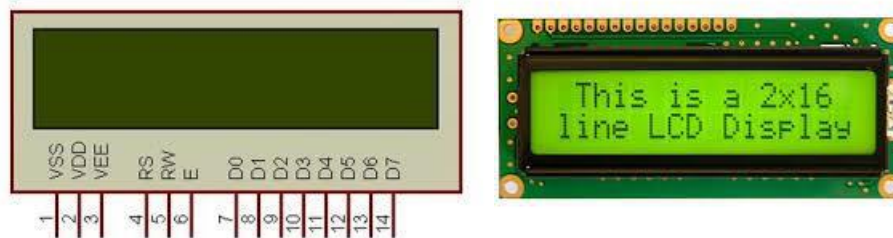
نکته ی قابل توجه این است که شیر آب را نمی توان مستقیما به یکی از پین های میکروکنترلر وصل کرد. چون پین های میکروکنترلر جریان دهی محدودی دارند و نمی توان جریان زیادی از آنها کشید. از طرفی شیر آب جریان نسبتا زیادی هنگام باز بودن نیاز دارد. لذا باید مداری طراحی شود تا این جریان را تامین کند. حتی در بیشتر موارد نیز ولتاژ کاری شیر از **۳.۳** ولت بیشتر است. به عنوان نمونه یک شیر برقی سلونوئیدی **24V,1.2W,DC** در زیر مشاهده میشود. این شیر در حالت عادی بسته بوده و با اعمال ولتاژ لازم به دو پایه اش مسیر آب را باز میکند.



برای درایو نمودن شیر آب فرض کنید از یک ماسفت نوع n به عنوان درایور استفاده می کنیم. (در کد نویسی اثری نخواهد داشت)

#### بخش ۴) LCD

برای نمایش اطلاعات از LCD میتوان کمک گرفت LCD. ها در دو نوع کاراکتری و گرافیکی وجود دارند. LCDهای گرافیکی قابلیت نمایش اطلاعات با دقت بیشتر را دارند. ولی را هاندازی پیچید هتر و قیمت بیشتری نیز دارند. همچنین هر دو LCD کاراکتری و گرافیکی در انداز ههای مختلف ی تولید میشوند. ما از LCD کاراکتری 16\*2 استفاده خواهیم کرد.



شکل سمت راست LCD و شکل سمت چپ پایه های آن را نشان میدهد. پایههای VSS و VDD که مربوط به تغذیه هستند. پایه ی VEE هم برای تنظیم کنتراست کاراکترهای نمایش داده شده به کار میرود. این تنظیم کنتراست معمولاً توسط یک پتانسیومتر انجام میشود. پایههای RS و RW و E پایه های کنترلی و پایه های D0 تا D7 مربوط به دیتا هستند. امکان استفاده از دیتا به دو صورت ۸ بیتی و ۴ بیتی وجود دارد. مزیت روش ۴ بیتی استفاده از چهار بیت کمتر نسبت به حالت دیتای ۸ بیتی است. ما نیز از مد چهار بیتی برای اتصال به میکروکنترلر استفاده خواهیم کرد. در این حالت فقط چهار پایه از پایه های دیتا مورد استفاده قرار میگیرند و به چهار پایه ی دیگر کاری نداریم. نحوه ی اتصال پین های lcd در لینک زیر آمده است. همچنین در این لینک برنامه نمونه ای برای بررسی نحوه ی استفاده از این کتابخانه وجود دارد.

<https://deepbluembedded.com/stm32-lcd-16x2-tutorial-library-alphanumeric-lcd-16x2-/interfacing>

برای استفاده از کتابخانه لینک بالا باید علاوه بر لینک زیر :

[https://github.com/Khaled-Magdy-DeepBlue/STM32\\_Course\\_DeepBlue/tree/master/ECUAL/LCD16x2](https://github.com/Khaled-Magdy-DeepBlue/STM32_Course_DeepBlue/tree/master/ECUAL/LCD16x2)

لینک زیر را هم به عنوان کتابخانه به پروژه اضافه کنید :

<https://drive.google.com/file/d/1x6N19YExuAgTW3Cooq95-RqU2xL4RV2H/view>

این لینک حاوی تابع `delay` نوشته شده توسط نویسنده است که از آن در کتابخانه `lcd` استفاده کرده ، زیرا `HAL_Delay` فقط قادر به ایجاد تاخیر های ۱ میلی ثانیه ای است و نویسنده به تاخیر های در اندازه میکرو ثانیه نیاز دارد .

برای استفاده از کتابخانه تاخیر می توانید از لینک زیر کمک بگیرید :

<https://deepbluembedded.com/stm32-delay-microsecond-millisecond-utility-dwt-delay-timer-/delay>

همچنین برای استفاده یک تایمر برای ایجاد تاخیر هم کتابخانه ای در بالا قرار داده شده (`Timer_Delay`) که در صورت نیاز به تابع تاخیر دیگر می توانید از آن استفاده کنید . ( به صورت پیش فرض از `TIM4` (تایمر شماره چهار) برای ایجاد تاخیر استفاده می کند .)

هر دو کتابخانه `lcd` و تاخیر بالا برای `stm32f1` نوشته شده اند که می توانید با تغییر `#include "stm32f1xx_hal.h"` در هدر فایل ها به `#include "stm32f4xx_hal.h"` آن را برای `stm32f4` تنظیم کنید.

همچنین تعیین محل اتصال پایه های `lcd` را (تغییر پیش فرض برای عدم تداخل در استفاده پایه ها ) در سورس فایل `LCD16x2_cfg.c` می توانید انجام دهید.(در صورت عدم تداخل نیازی نیست)

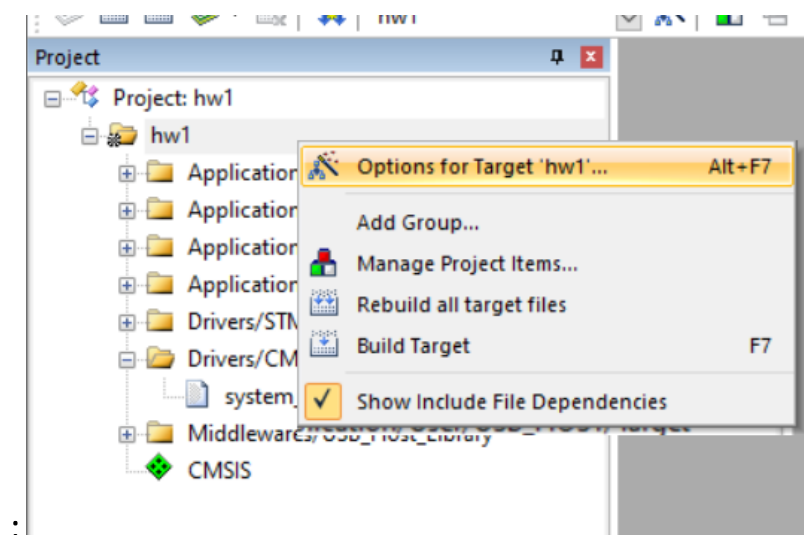
## بخش ۵) میکروکنترلر

میکروکنترلر دائماً وضعیت پین `PB1` (که به خروجی دیجیتال رطوبت تسنج وصل شده است) را چک میکند و در صورتی که رطوبت خاک کم شود، پین `PB2` (که به شیر آب وصل است) را به وضعیت `high` برده (بازشدن شیر آب ) و به مدت زمان `t_open` آن را در وضعیت `high` نگه داشته و سپس دوباره آن را در وضعیت `low` قرار دهد `LCD` . نیز در حالت عادی، در سطر اول عبارت `"Moisture: Wet"` یا `"Moisture: Dry"` را با توجه به میزان رطوبت خاک نشان دهد و در سطر دوم عبارت

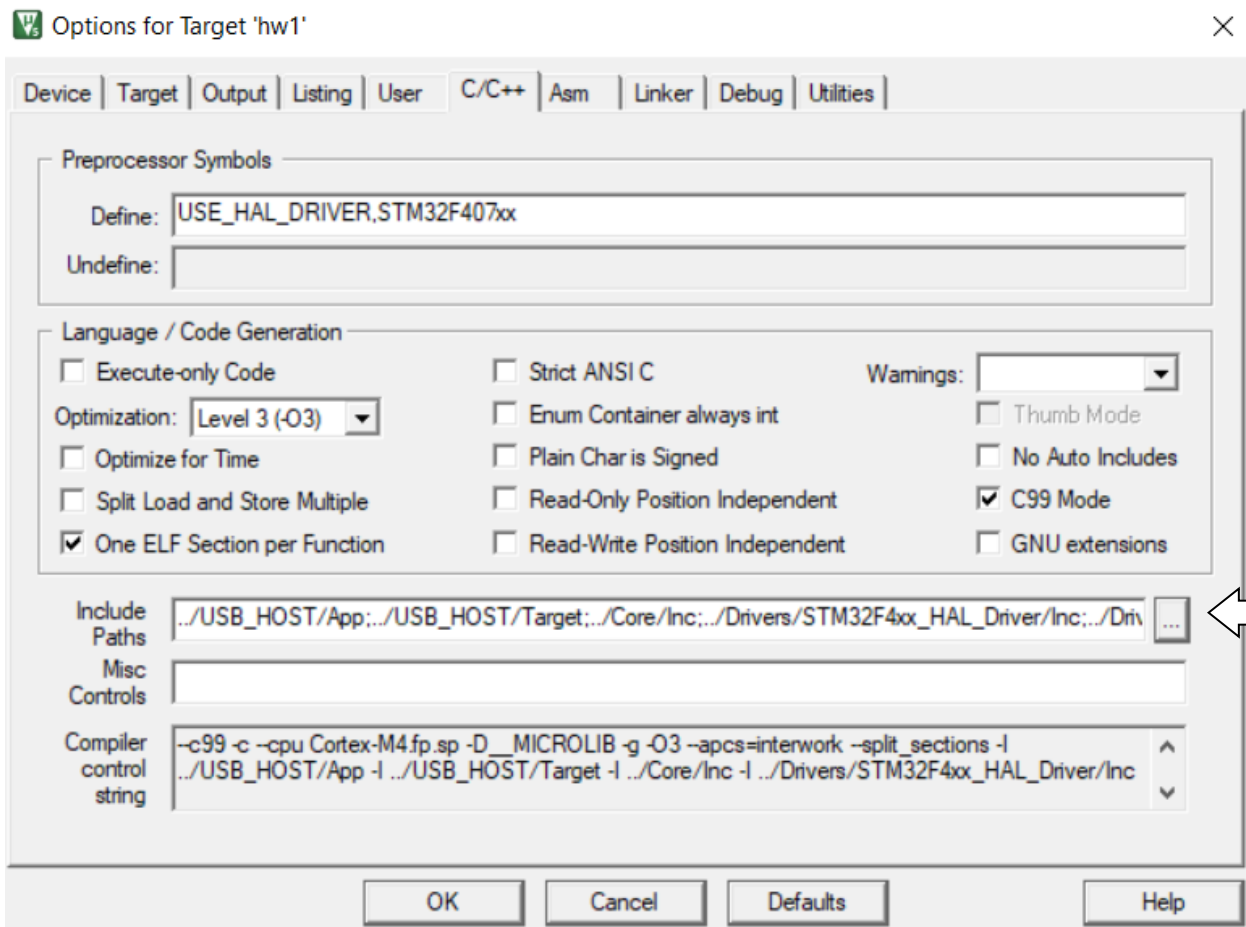
“Open Time: 1 Min” یا “Open Time: 2 Min” یا “Open Time: 3 Min” را با توجه به مقدار  $t_{open}$  نشان دهد .  
 $t_{open}$  یک متغیر است که مقدار باز ماندن شیر آب را در هر بار باز شدنش بر حسب دقیقه نشان میدهد. مقدار اولیه  $t_{open}$  برابر ۱ میباشد. با فشردن کلید A در کپد، LCD عبارت “Choose Time from” را در سطر اول و عبارت “{1,2,3} Min:” را در سطر دوم نشان داده و سپس منتظر فشرده شدن کلید باشد. اگر کلید فشرده شده یکی از سه کلید ۱ یا ۲ یا ۳ باشد، آن را در متغیر  $t_{open}$  ذخیره کرده و حالت نمایش LCD را به حالت قبل برگرداند. اگر کلید دیگری فشرده شود هیچ اتفاقی نیفتد و همچنان منتظر فشرده شدن کلید بماند.

برای تعیین زمان سپری شده از باز ماندن شیر آب می توانید به دو صورت عمل کنید .

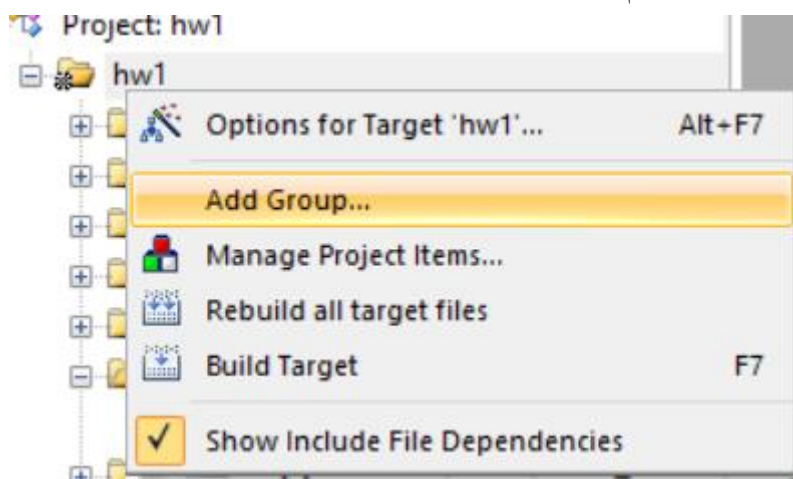
۱. در `while(1)` تابع `HAL_Delay` را با مقادیر کوتاهی استفاده کنید (مثلاً ۱۰ میلی ثانیه) (برای عدم ایجاد اشکال در خواندن keypad) سپس به یک متغیر اضافه کنید . متناوباً (در صورت باز بودن شیر) مقدار آن را چک کنید تا به زمان مورد نظر برسید (برای یک دقیقه به ۶۰۰۰ برسد) سپس شیر آب بسته شود .
  ۲. در همان `HAL_SYSTICK_Callback(void)` که برای keypad استفاده کردیم به یک متغیر اضافه کنید. (فاصله ها به صورت ۱ میلی ثانیه است پس باید متغیر به ۶۰۰۰ برسد) و در `while(1)` مقدار آن را متناوباً (در صورت باز بودن شیر) بررسی کنید.
- برای اضافه کردن هدر فایل ها در navigator به صورت زیر کلیک راست کنید



سپس بر روی `options for target` کلیک کنید . بر روی پنجره ی باز شده بر روی تب `C/C++` بروید و محل هدر فایل ها را اضافه کنید .

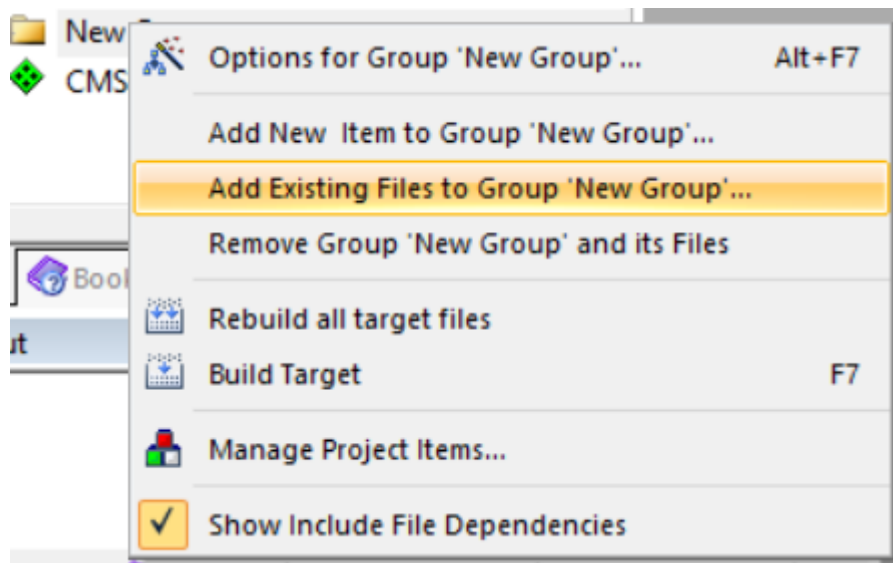


برای اضافه نمودن سورس فایل ها را هم می توانید بر روی navigator کلیک راست کنید و یک گروه جدید بسازید مانند



زیر:

سپس با کلیک بر روی 'add existing files' سورس فایل ها (\*.c) را اضافه کنید.



بعد از انجام پروژه به صورت کامل با زدن بر روی build target پروژه را بیلد کنید و در صورت وجود error یا warning آن ها را رفع کنید .

### فایل‌های تحویلی شامل

۱. شماتیک مدار بسیار ساده کشیده شده بر روی یک کاغذ

۲. پروژه ه ی نرم افزار Keil uVision

۳. گزارش کار در فرمت PDF باشد.

در گزارش کار خود این سوال را پاسخ دهید:

۱. کدام روش تعیین زمان سپری شده از باز بودن شیر آب بهتر است و چرا ؟

۲. برای نشان دادن زمان باز ماندن شیر آب بر روی LCD به صورت معکوس شمار و به ثانیه چه تغییری در کد نیاز است.

۳. تحقیق کنید روش pooling و روش interrupt در استفاده از keypad چه تفاوتی دارد و کدامیک بهتر است ؟

موفق باشید