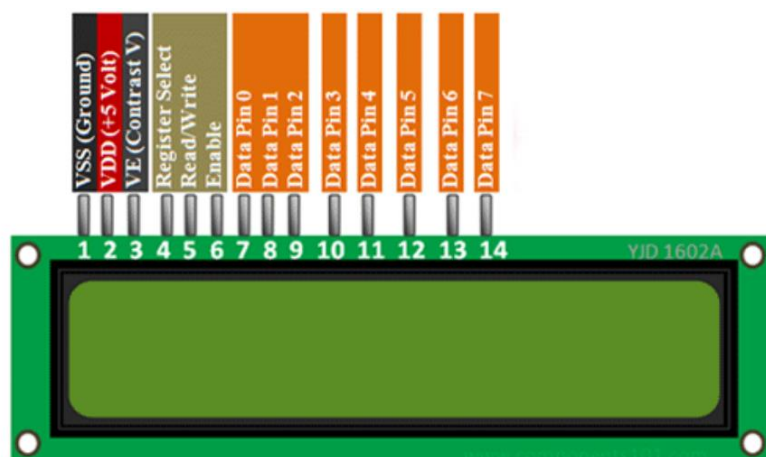


چیپ: LM016L



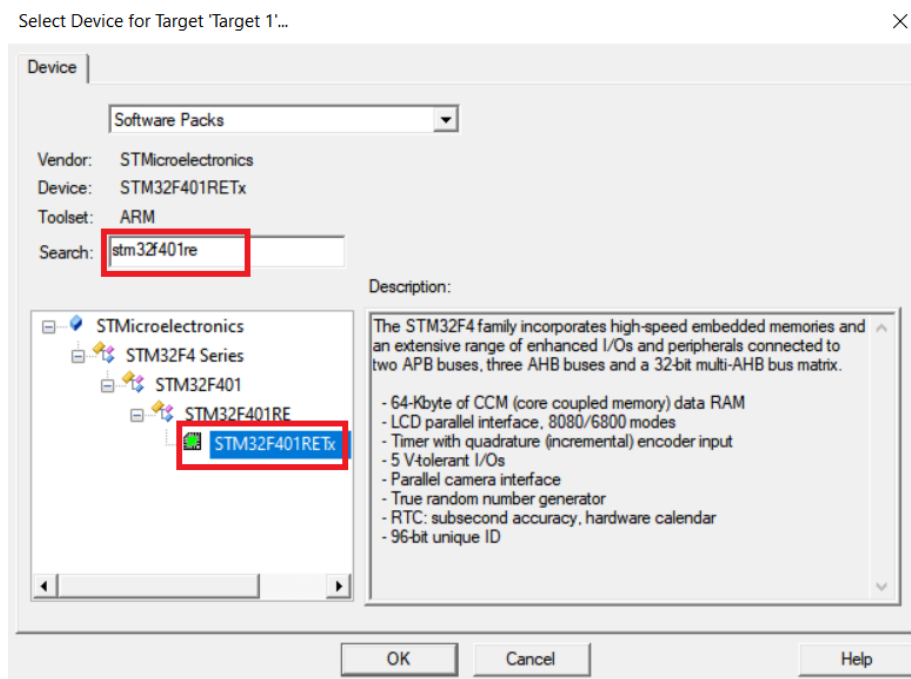
شماره پایه	نماد	کاربرد	اتصال خارجی
۱	V _{SS}	زمین (GND)	پایانه منفی منبع تغذیه
۲	V _{DD}	V _{CC}	ولتاژ تغذیه +۵ ولت
۳	V _{EE}	تنظیم کنتراست	به ولتاژ +۵ ولت متصل می‌شود (برای داشتن کنتراست قابل تنظیم این پایه به پتانسیومتر خارجی وصل می‌شود).
۴	RS	انتخاب رجیستر (داده / دستور)	به پین‌های کنترل‌کننده میکرو کنترلر وصل می‌شود وقتی RS=0 رجیستر دستور و هنگامی که RS=1 رجیستر داده انتخاب می‌شود.
۵	R/W	انتخاب عملیات (خواندن / نوشتن)	به پین‌های کنترل‌کننده میکرو کنترلر وصل می‌شود برای خواندن باید R/W=1 و برای نوشتن مقداری روی رجیستر R/W=0 باشد.
۶	E	فعال‌سازی LCD	به پین‌های کنترل‌کننده میکرو کنترلر وصل می‌شود. با اعمال یک پالس پایین رونده به این پایه تغییرات مدنظر در LCD اعمال می‌شوند.
۷ - ۱۰	DB0 – DB3	چهار خط اول گذرگاه داده (این چهار خط در مد چهار بیتی فعال نیستند)	به پین‌های داده میکرو کنترلر وصل می‌شود
۱۱ - ۱۴	DB4 – DB7	چهار خط دوم گذرگاه داده (این چهار خط در مد چهار بیتی فعال هستند)	به پین‌های داده میکرو کنترلر وصل می‌شود

- دو نوع سیگنال به LCD ارسال می‌شود، سیگنال داده و سیگنال کنترل. این سیگنالها توسط پین RS از هم متمایز میشوند. هنگامی که یک پالس به پایه E داده شود ماژول LCD در لبهٔ پایین رونده پالس، اطلاعات را از روی گذرگاه داده میخواند، اگر اطلاعات دریافتی از نوع داده باشد آن را بر روی LCD نمایش میدهد و اگر از نوع دستور باشد آن دستور را اجرا میکند.

- هنگامی که بخواهیم به LCD دستور خاصی را ارسال کنیم باید کد هگزادسیمال مربوط به آن دستور را بر روی گذرگاه داده قرار دهیم. دستورالعمل های مورد نیاز در دیتاشیت قرار داده شده در کنار تمرین وجود دارند.
- برای سادگی کار شما، در کنار تمرین فایل lcd_prototypes قرار داده شده که در آن یک سری از توابعی که برای کار با LCD به آنها نیاز دارید قرار دارند و بخشی از آنها پیاده سازی شده است. برای کامل کردن توابع از کامنت هایی که در هر تابع نوشته شده کمک بگیرید. خط هایی که کامنت *TODO* دارند باید جایگزین شوند و در صورت نیاز می توانید موارد دیگری نیز به بدنه ی توابع اضافه کنید. همچنین می توانید توابع دیگری نیز برای استفاده از LCD تعریف کنید. (دقت کنید که در بدنه ی بعضی توابع، delay قرار داده شده است، این delay ها برای کار کردن LCD نیاز است و برای سهولت کار شما از قبل قرار داده شده و تست شده اند، بهتر است این خطوط را تغییر ندهید، اما در صورت نیاز می توانید تغییر ایجاد کنید.)

کتابخانه CMSIS:

برای تنظیمات مربوط به میکروکنترلر و دسترسی به رجیسترهای آن، از کتابخانه CMSIS استفاده کنید. برای استفاده از این کتابخانه، در هنگام ساخت پروژه ی جدید در keil، مراحل زیر را انجام دهید:



Software Component	Sel.	Variant	Version	Description
Board Support		MCBSTM32F400	2.0.0	Keil Development Board MCBSTM32F400
CMSIS	<input checked="" type="checkbox"/>			Cortex Microcontroller Software Interface Components
CORE			5.3.0	CMSIS-CORE for Cortex-M, SC000, SC300, ARMv8-M, ARMv8.1-M
DSP		Library	1.7.0	CMSIS-DSP Library for Cortex-M, SC000, and SC300
NN Lib			1.2.0	CMSIS-NN Neural Network Library
RTOS (API)			1.0.0	CMSIS-RTOS API for Cortex-M, SC000, and SC300
RTOS2 (API)			2.1.3	CMSIS-RTOS API for Cortex-M, SC000, and SC300
CMSIS Driver				Unified Device Drivers compliant to CMSIS-Driver Specifications
Compiler		ARM Compiler	1.6.0	Compiler Extensions for ARM Compiler 5 and ARM Compiler 6
Device	<input checked="" type="checkbox"/>			Startup, System Setup
Startup			2.4.2	System Startup for STMicroelectronics STM32F4 Series
STM32Cube Framework				STM32Cube Framework
STM32Cube HAL				STM32F4xx Hardware Abstraction Layer (HAL) Drivers
File System		MDK-Plus	6.13.0	File Access on various storage devices
Graphics		MDK-Plus	5.50.0	User Interface on graphical LCD displays
Graphics Display				Display Interface including configuration for emWIN
Network		MDK-Plus	7.12.0	IPv4 Networking using Ethernet or Serial protocols
USB		MDK-Plus	6.13.7	USB Communication with various device classes

برای استفاده از کتابخانه ی CMSIS باید آن را #include کنید:

```
#include "stm32f4xx.h"
```

سپس میتوانید از امکانات این کتابخانه استفاد کنید.

مثال استفاده از کتابخانه : CMSIS

```
/* Enable the RCC AHB peripheral clock for GPIO port C */
RCC->AHBENR |= RCC_AHBENR_GPIOCEN;
/* GPIOC port PIN 4 mode as output (01) */
GPIOC->MODER |= GPIO_MODER_MODER4_0;
```

به کمک ماکرو های تعریف شده در این کتابخانه می توانیم به راحتی به رجیستر های یک GPIO دسترسی پیدا کنیم، مثلاً در این مثال ابتدا در رجیستر AHBENR مربوط به RCC مقدار RCC_AHBENR_GPIOCEN ریخته شده که باعث فعال کردن کلاک GPIOC می شود و سپس به mode register مربوط به GPIOC دسترسی پیدا کردیم و پین 4 آن را خروجی کرده ایم. برای سهولت می توانید از ماکرو های تعریف شده برای mode هر پین استفاده کنید مثلاً

0_4GPIO_MODER_MODER پین 4 را خروجی می کند و یا می توانید به صورت دستی

مقدار دهید به صورت زیر:

```
RCC->AHBENR |= 0x4  
GPIOC->MODER |= 0x10
```

اینکه در هر رجیستر چه مقداری ریخته شود هم به کمک Reference manual میکروکنترلر قابل تشخیص است.

بقیه ی رجیستر های GPIO کاربردی به صورت زیر قابل دسترسی اند:

GPIO port output data register	: GPIOx->ODR
GPIO port output data register	: GPIOx->IDR
GPIO port bit set/reset register	: GPIOx->BSRR
GPIO port output type register	: GPIOx->OTYPER
GPIO port output speed register	: GPIOx->OSPEEDR
GPIO port pull-up/pull-down register	: GPIOx->PUPDR

برای دسترسی به ماکرو ها و سایر مواردی که در کتابخانه CMSIS تعریف شده اند می توانید در پوشه CMSIS ای که توسط KEIL ساخته می شود وارد شوید و فایل های آن را مطالعه کنید.