

گزارش پروژه ساختار و زبان کامپیوتر

تبدیل پیام به کد مورس و برعکس

اعضای گروه:

غزل شناور، 97101897

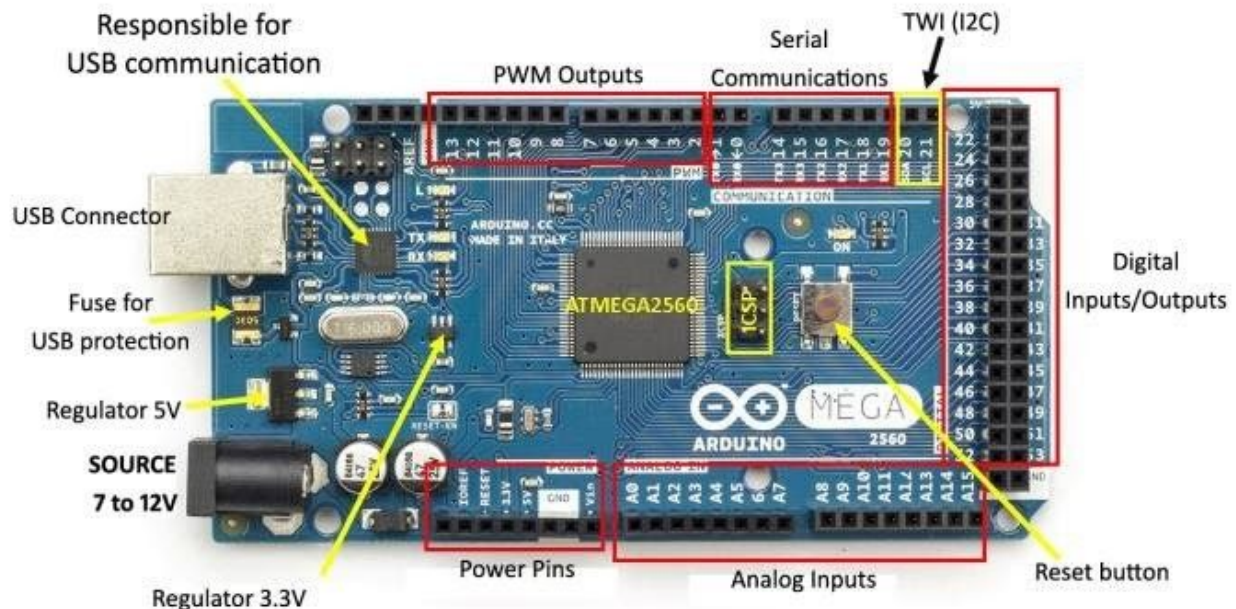
عرشیا اخوان، 97110422

سارا خسروی، 97101586

استاد: دکتر اسدی

ماژول های استفاده شده:

برد arduino mega 2560:



پردازنده ATmega2560 دارای 256kb حافظه و 86 پین ورودی و خروجی **general purpose** می باشد.

4x4 Keypad:

این ماژول هشت پین دارد که چهار پین اول برای ردیفها و چهار پین دوم برای ستونها هستند. هنگامی که یک ردیف و یک ستون انتخاب شوند دکمه ای که در تقاطع آنها قرار دارد انتخاب می شود.

2x16 LCD:

پین اول و دوم منبع تغذیه **lcd** هستند و پین سوم مشخص کننده **contrast** می باشد. پین چهارم، **register select**، انتخاب می کند که کدام یک از دو ثبات **lcd** استفاده شود؛ در صورتی که مقدار صفر به آن داده شود ثبات دستور دستوری را که به **lcd** فرستاده می شود ذخیره می کند و اگر مقدار 1 داده شود داده در ثبات داده ذخیره می شود. پین پنجم (**read/write**) با وصل شدن به زمین در مد نوشتن قرار می گیرد و پین ششم (**enable**) فعال بودن یا نبودن **lcd** را مشخص می کند. پین هفتم تا چهاردهم برای انتقال اطلاعات هستند که ما فقط از چهارتای آخر استفاده کردیم. پین پانزدهم **backlight** و پین شانزدهم کاتد **backlight** می باشد.

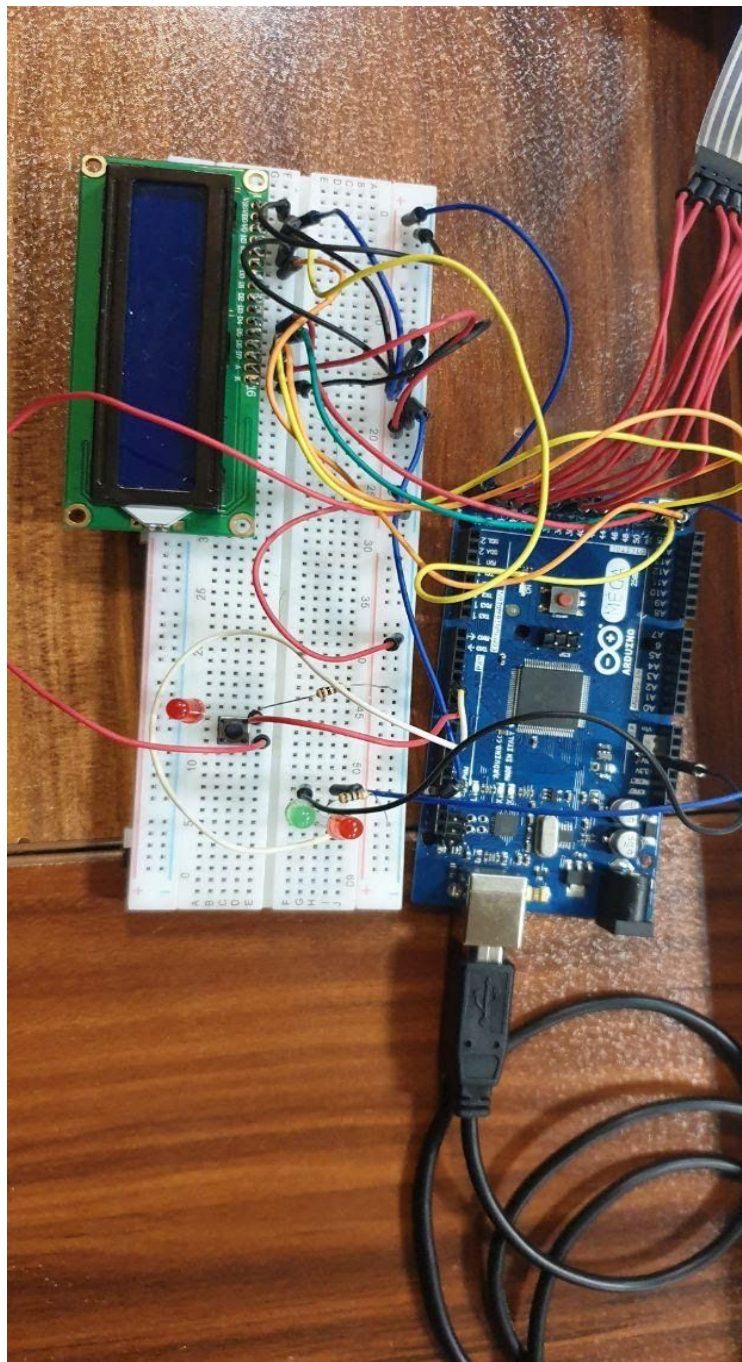
مدار بسته شده:

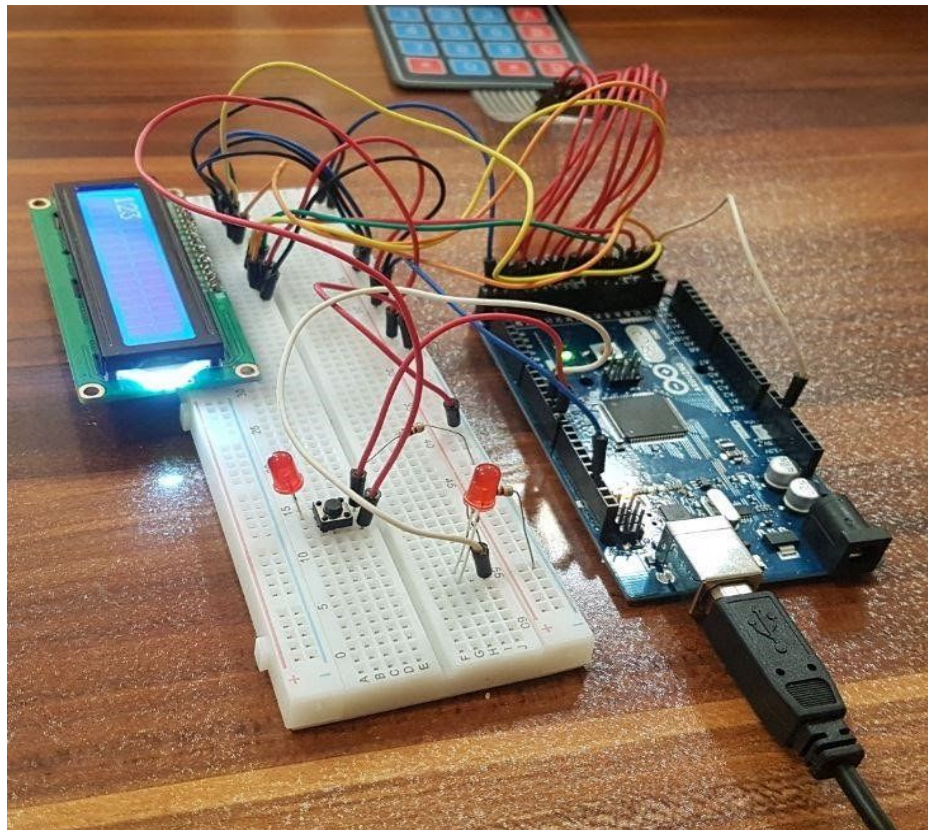
با استفاده از **USB Connector** برد آردوینو را به لپتاپ وصل کردیم و تغذیه برد هم از همان اتصال انجام می شود. سایر ماژولها را با استفاده از **digital input/output** و **pwm outputs** به برد وصل کردیم.

بدلیل نبود سوییچ دو حالتی تغییر مد کاری مدار توسط سیم سفید سمت راست مدار انجام و توسط **led** سبز نمایش داده می شود. در صورتی که به **Gnd** وصل شود **led** سبز خاموش می ماند و ورودی را توسط **Keypad** میگیرد و کد مورس آن را توسط **led**، که به پین پنجم وصل شده، نمایش می دهد. در حالتی که به 5 ولت وصل شود **led** سبز روشن شده و ورودی را به صورت کد مورس از دکمه، که به پین هفت آردوینو وصل شده است، می گیرد و آن را روی **lcd** نمایش می دهد.

ماژول **keypad** را با استفاده از پین های دیجیتال 22، 24، 26، 28، 30، 32، 34، 36 به برد وصل کردیم.

برای اتصال و راه اندازی **lcd** از [این صفحه](#) استفاده کردیم. پین اول به زمین و پین دوم به منبع تغذیه 5 ولت وصل کردیم. پین سوم را بدلیل نداشتن پتانسیومتر مستقیماً به زمین، پین چهارم به پین دیجیتال 48، پین پنجم به زمین، پین ششم به پین دیجیتال 50 و خطوط داده d4 تا d7 را به پینهای 46 تا 40، آند را به 5 ولت و کاتد را به زمین وصل کردیم.





کد:

تمامی کدهای زده شده ضمیمه این گزارش شده است. همچنین کلیه محتوای این فایل در [گیت](#) موجود است.

مرحله تبدیل رشته به کد مورس:

کدهای مورس حداکثر پنج خط/نقطه داشتند پس ما کدها را با شروع از جایگاه کم ارزش عدد و با قرار دادن صفر به جای نقطه و یک به جای خط نوشتیم و هر کدی که کمتر از پنج رقم بود را با 2 ادامه دادیم (مراجعه شود به فایل morse_data، فایل morse.txt) سپس این عدد ها را در مبنای 3 در نظر گرفته و مبنای 10 آنها را در آرایه morseCodes (در فایل morse.ino) قرار دادیم. کتابخانه MultitapKeypad.h را برای keypad استفاده کردیم. هر گاه یک کلید فشرده می شود کاراکتر آن به خانه مربوطه آرایه کد می شود و عدد در مبنای 10 مربوط به آن به تابع داده می شود. در هر مرحله اگر باقیمانده تقسیم آن به سه برابر دو شود تابع تمام می شود و در صورتی که باقیمانده صفر یا یک شود Led روشن می شود (هنگامی که صفر را بخواهد نمایش دهد led برای 0.1 ثانیه و برای یک 0.5 ثانیه روشن می ماند) (keypad.ino تابع handleKey، و توابع morseCode و genMorse در morse.ino). دکمه star کاراکتر قبلی را از buffer پاک می کند و دکمه hashtag بافر را روی lcd چاپ می کند سپس بافر را خالی می کند.

مرحله تبدیل کد مورس به رشته:

ورودی کد مورس از دکمه گرفته می شود. در صورتی که دکمه برای کمتر از 0.3 ثانیه فشرده شود نقطه و بیشتر از 0.3 ثانیه خط در نظر گرفته می شود. عدم فشردن دکمه برای 0.5 ثانیه به منظور تمام شدن این حرف در نظر گرفته می شود. در صورت وجود در آرایه `morseDecodes` حرف معتبر در نظر گرفته شده و روی `lcd` چاپ می شود (تابع `morseDecode` در `morse.ino`) و به ازای هر کد مورس غیر معتبر فاصله چاپ می شود تا کاربر متوجه شود کد نامعتبر وارد کرده است (تابع `handle button` در `button.ino`).

هرگاه `lcd` به انتهای صفحه نمایش خود برسد دوباره از ابتدای صفحه شروع به چاپ کردن می کند. همچنین تمامی بازه های زمانی در نظر گرفته شده قابل تغییر هستند.