

گزارش آزمایشگاه سخت افزار

نیم سال اول 1401-1400

گزارش چهارم: نمایشگر دندۀ عقب خودرو
اعضای گروه: متینا

سپهر امینی افشار



دانشکده مهندسی کامپیوتر
مهندزاده

1.0 مقدمه

هدف از این پروژه پیاده سازی سیستمی است که دندۀ عقب خودرو را شبیه سازی کند. این پروژه به این صورت تعریف شده است که یک دوربین فیلم را از محیط می‌گیرد و روی یک LCD نمایش می‌دهد. همچنین زمانی که دوربین به یک شئی نزدیک شود buzzer روشن می‌شود و با فرکانس مناسب با فاصله دوربین و جسم تولید صدا می‌کند. در صورتی که فاصله از حد معینی بیشتر بود صدایی تولید نمی‌شود و زمانی که فاصله کمتر از حد مورد نظر بود این بوق ممتد می‌شود. همچنین فاصله بین جسم و دوربین نیز روی LCD نمایش داده می‌شود.

2.0 محدودیت‌های سخت افزاری

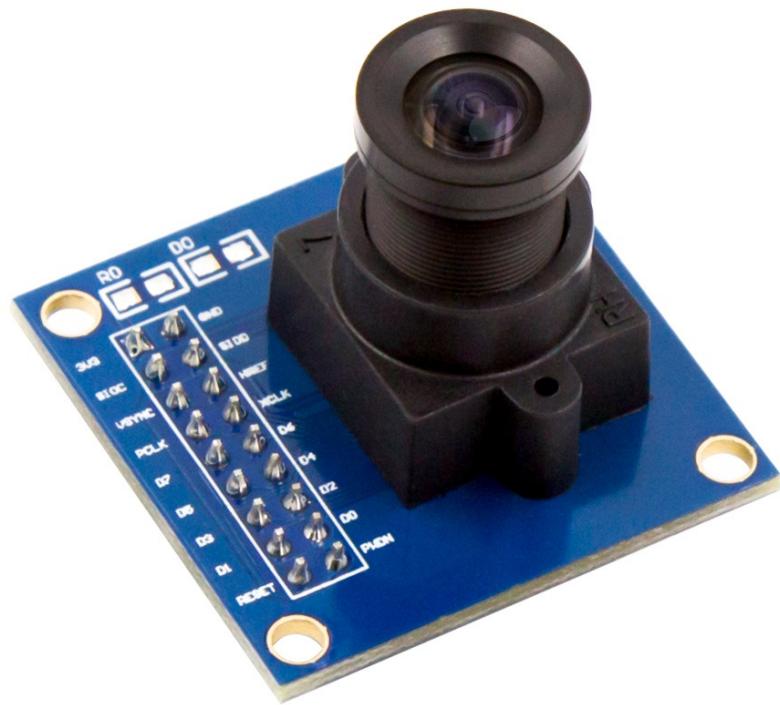
2.1 از آنجایی که این وسیله قرار است با سرباز حمل شود، باید تا حد امکان کم وزن باشد تا مشکلی برای سرباز ایجاد نکند. همچنین باید فضای کمی را اشغال کند تا هم در مسیرهای سخت سالم بماند هم میزان تشخیص پذیری او را افزایش ندهد.

2.2 همچنین چون دوربین‌های موجود در بازار صدا را ضبط نمی‌کنند لازم است تا مژول میکروفون هم به سرباز وصل شود و به همراه فایل فیلم ارسال شود.

2.3 ما از برد Arduino UNO استفاده می‌کنیم و مژول‌های انتخابی باید با این مدل سازگار باشد.

3.0 مژول‌های مورد استفاده

3.1 مژول دوربین اردوئینو مدل OV7670



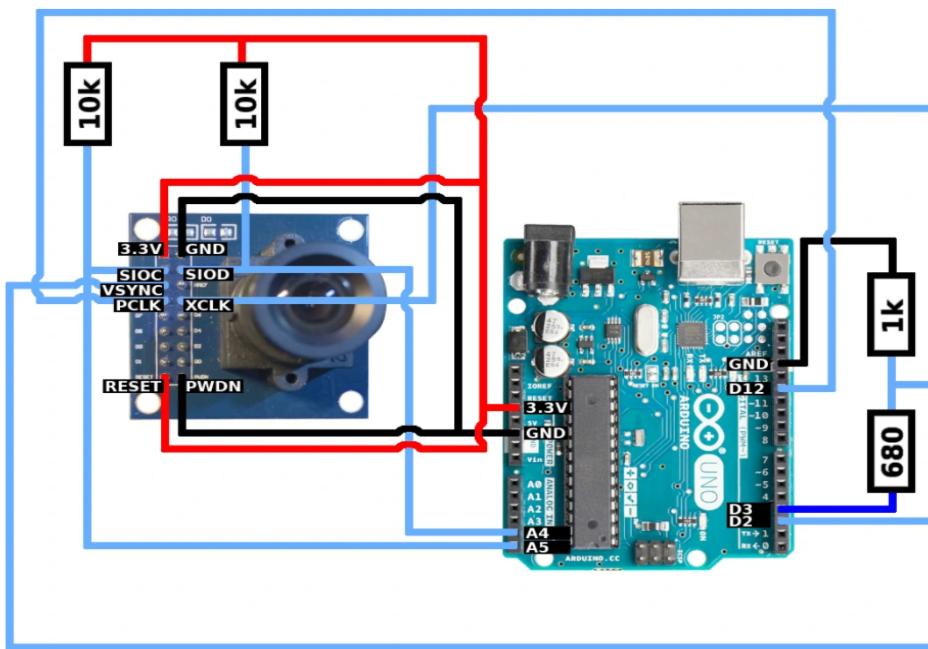
تصویر ۱: مازول دوربین OV7670

3.1.1 ابعاد: $35 * 34 * 29$ میلی متر

3.1.2 وزن: ۳۰ گرم

3.1.3 ولتاژ مورد نیاز: ۳.۳ ولت

3.1.4 نحوه اتصال



تصویر ۲: نحوه اتصال مژول دوربین OV7670 به آردوینو UNO

جدول ۱: لیست پین‌های OV7670

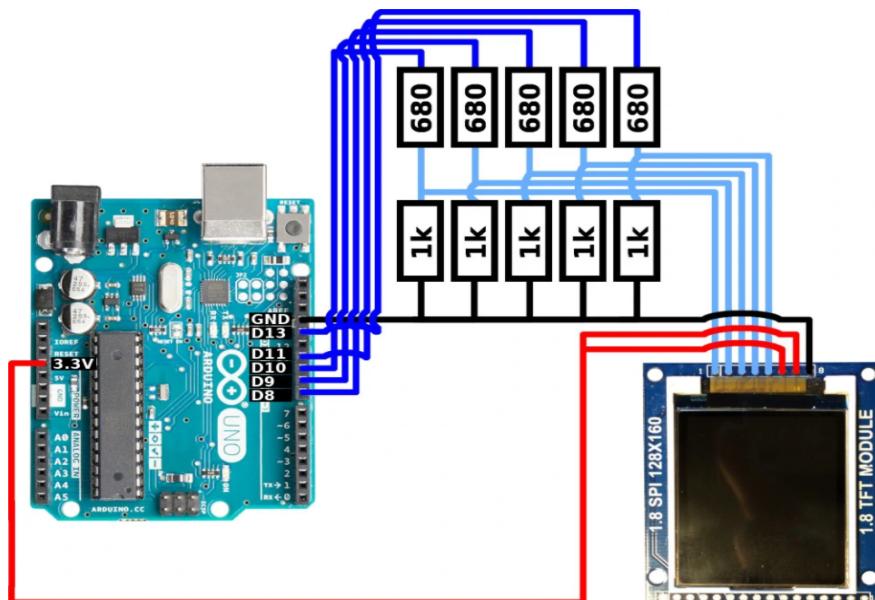
| Pin No. | Pin Name | Type | Description |
|---------|----------|----------------|---|
| 1 | VCC | POWER | 3.3v Power supply |
| 2 | GND | GROUND | Power ground |
| 3 | SCL | input | Two-Wire Serial Interface Clock |
| 4 | Sdata | Bi-directional | Two-Wire Serial Interface Data I/O |
| 5 | VSYNC | output | Active High: Frame Valid; indicates active frame |
| 6 | HREF | output | Active High: Line/Data Valid; indicates active pixels |
| 7 | PCLK | output | Pixel Clock output from sensor |
| 8 | XCLK | input | Master Clock into Sensor |
| 9 | Dout9 | output | Pixel Data Output 9 (MSB) |
| 10 | Dout8 | output | Pixel Data Output 8 |

| | | | |
|----|-------|--------|---------------------------|
| 11 | Dout7 | output | Pixel Data Output 7 |
| 12 | Dout6 | output | Pixel Data Output 6 |
| 13 | Dout5 | output | Pixel Data Output 5 |
| 14 | Dout4 | output | Pixel Data Output 4 |
| 15 | Dout3 | output | Pixel Data Output 3 |
| 16 | Dout2 | output | Pixel Data Output 2 (LSB) |

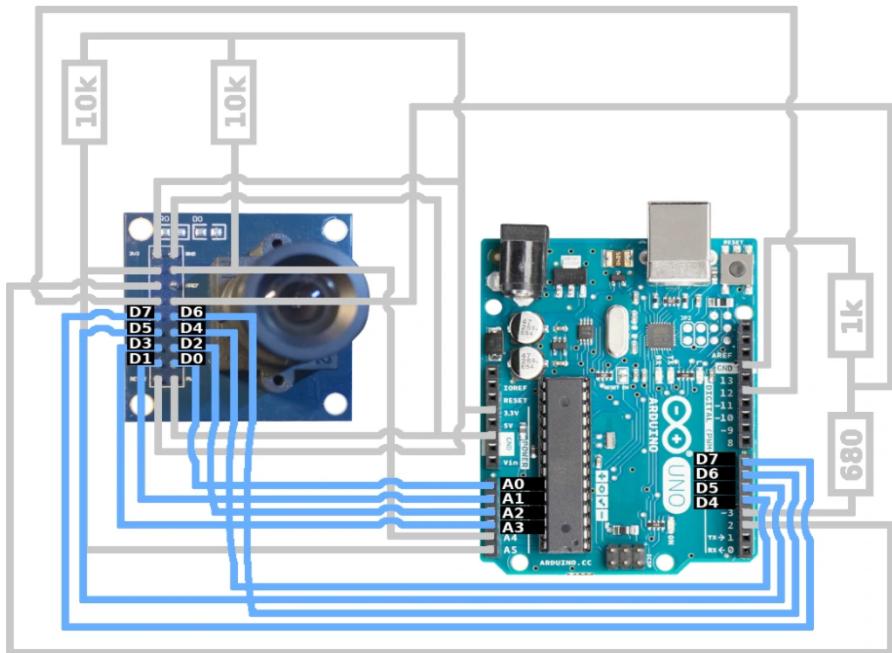
3.1.5 توضیحات: همانطور که مشاهده می کنید این دوربین ۳ پایه دیجیتال و ۲ پایه

آنalog را به خود اختصاص می دهد. از هیچ درایور دیگری لازم نیست استفاده کنیم.

توجه: با تحقیقاتی متوجه شدیم اگر با اتصال این مژول به کامپیوتر دارای محدودیت در نرخ انتقال داده هستیم. کامپیوتر ها از حدی بیشتر نرخ انتقال داده را اجازه نمی دهند. بنابراین اگر بخواهیم نتیجه را در کامپیوتر به صورت زنده مشاهده کنیم، از fps خوبی برخوردار نخواهیم بود. برای رفع این مشکل می توانیم فیلم گرفته شده را در lcd وصل شده به همین بورد به صورت زنده نمایش دهیم.

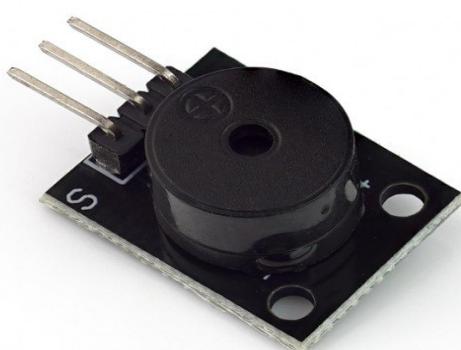


تصویر ۳: نحوه اتصال مژول ال سی دی به مژول آردینو UNO



تصویر ۴: نحوه اتصال پایه های انتقال داده دوربین به برد آردوینو

KY-012 مازول بازr 3.2



تصویر ۵: مازول بازr KY-012

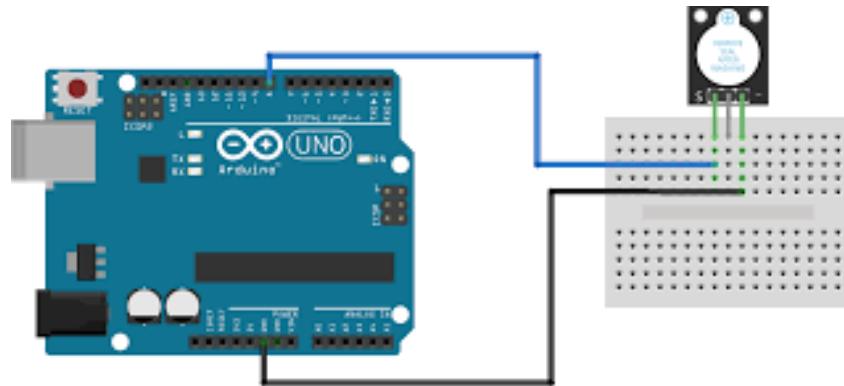
3.2.1 ابعاد: $18.5 * 15 * 15$ میلی متر

3.2.2 وزن: ۱۱ گرم

3.2.3 ولتاژ مورد نیاز: ۳.۳ ولت

3.2.4 نحوه اتصال

Sensor Signal = [Pin 13]
Sensor [N.C] =
Sensor GND = [Pin GND]

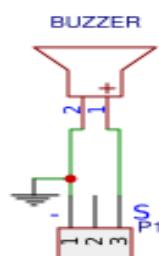


تصویر6: نحوه اتصال به برد آردوینو

3.2.5 توضیحات: این مژول بازر ولتاژ حداقل 3.3 ولت را گرفته و صوتی با فرکانس 2.5KHz را تولید می کند.

3.2.6 تکنولوژی استفاده شده در مژول:

این مژول از یک بازر تشکیل شده است که زمانی که ولتاژ وارد آن می شود با یک فرکانس یکنواخت صدا تولید می کند.



تصویر7: نمای داخلی مژول KY-012

3.2.7 پین های مازول:

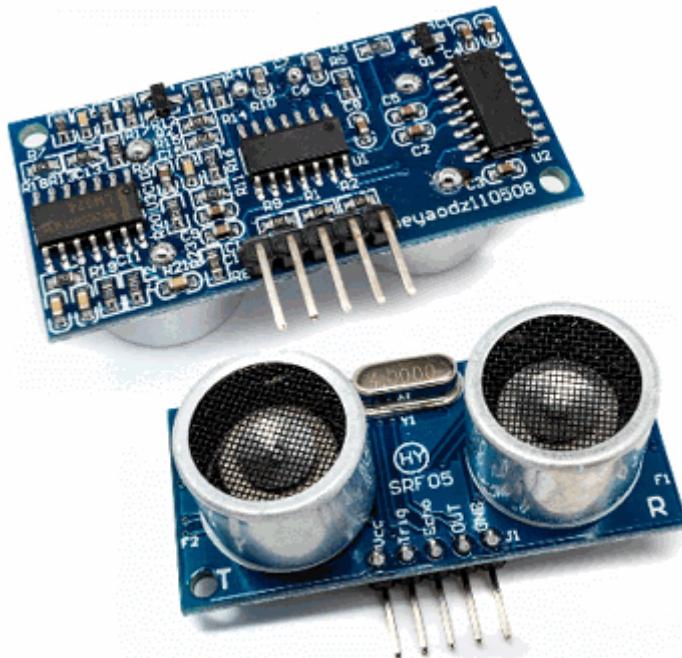


تصویر 8: پین های مازول KY-012

3.2.8 یک نمونه کد برای آزمایش این مازول در زیر آورده شده است:

```
int Buzzer = 13;  
void setup () {  
    pinMode (Buzzer, OUTPUT);  
    // Output pin initialization for the buzzer  
}  
void loop () {  
    digitalWrite (Buzzer, HIGH); // Buzzer will be on  
    delay (4000); // Waitmode for 4 seconds  
    digitalWrite (Buzzer, LOW); // Buzzer will be off  
    delay (2000); // Waitmode for another 2 seconds in which the  
    // buzzer will be off  
}
```

3.3 مازول تشخیص فاصله SRF05 - Ultrasonic Ranger



تصویر 9: مازول تشخیص فاصله SRF05

3.3.1 ابعاد: 43*20*17 میلی متر

3.3.2 وزن: 10 گرم

3.3.3 ولتاژ مورد نیاز: 5 ولت

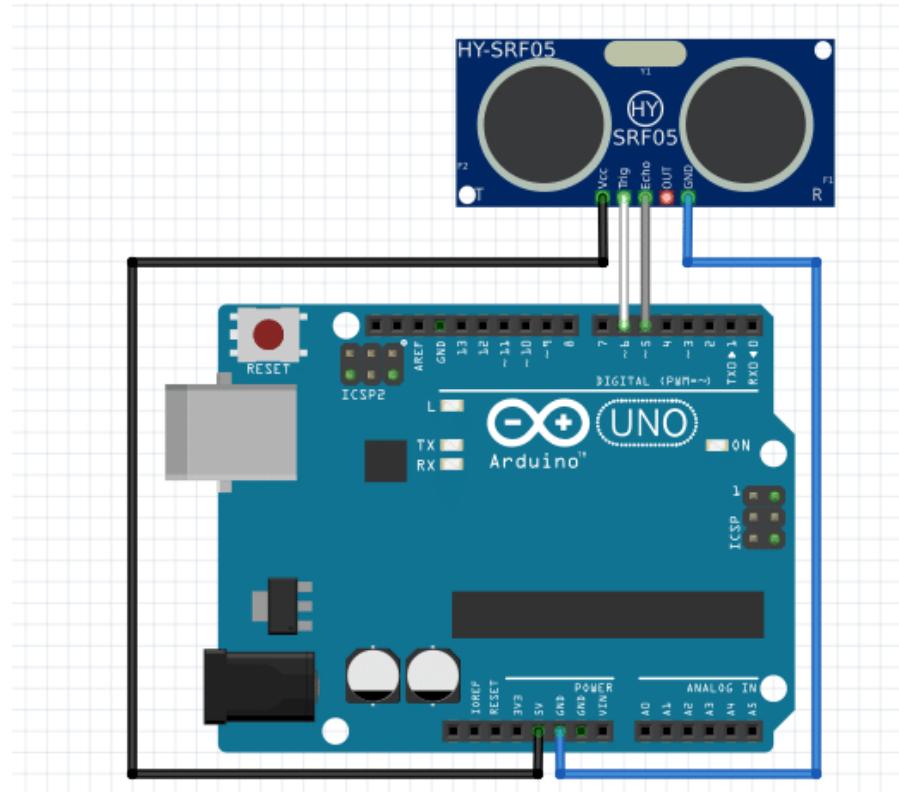
3.3.4 نحوه اتصال

Echo = [Pin 5]

Trig = [Pin 6]

VCC = [Pin 5V]

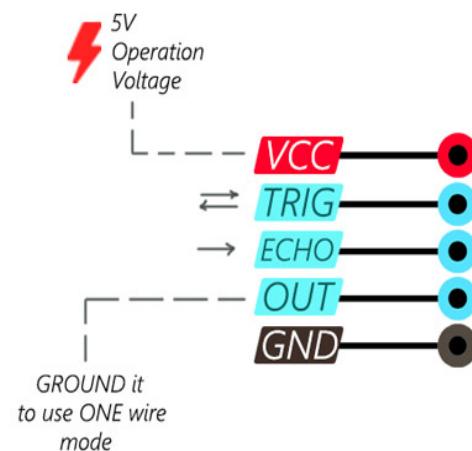
GND = [Pin GND]



تصویر 10: نحوه ای اتصال به برد آردوینو

3.3.5 این مازول تشخیص فاصله توانایی تشخیص جسمی از 2 سانتی متر تا 4.5 متر با دقت 3 میلی متر را دارد. برای استفاده از این مازول باید یک پالس 10uS را به پین trigger وصل کرده و خروجی حاصل را ضبط کنیم. عرض موجی که از پین echo ضبط میشود با فاصله ی جسم با مازول تغییر میکند و با استفاده از آن می توانیم فاصله را تشخیص دهیم.

3.3.6 پین های مازول



تصویر 11: پین های مازول KY-012

3.3.7 یک نمونه کد برای آزمایش این مژول در زیر آورده شده است:

```
#include <NewPing.h>
#define TRIGGER_PIN 12 // Arduino pin tied to trigger pin on the
ultrasonic sensor.
#define ECHO_PIN      11 // Arduino pin tied to echo pin on the
ultrasonic sensor.
#define MAX_DISTANCE 200 // Maximum distance we want to ping for (in
centimeters). Maximum sensor distance is rated at 400-500cm.

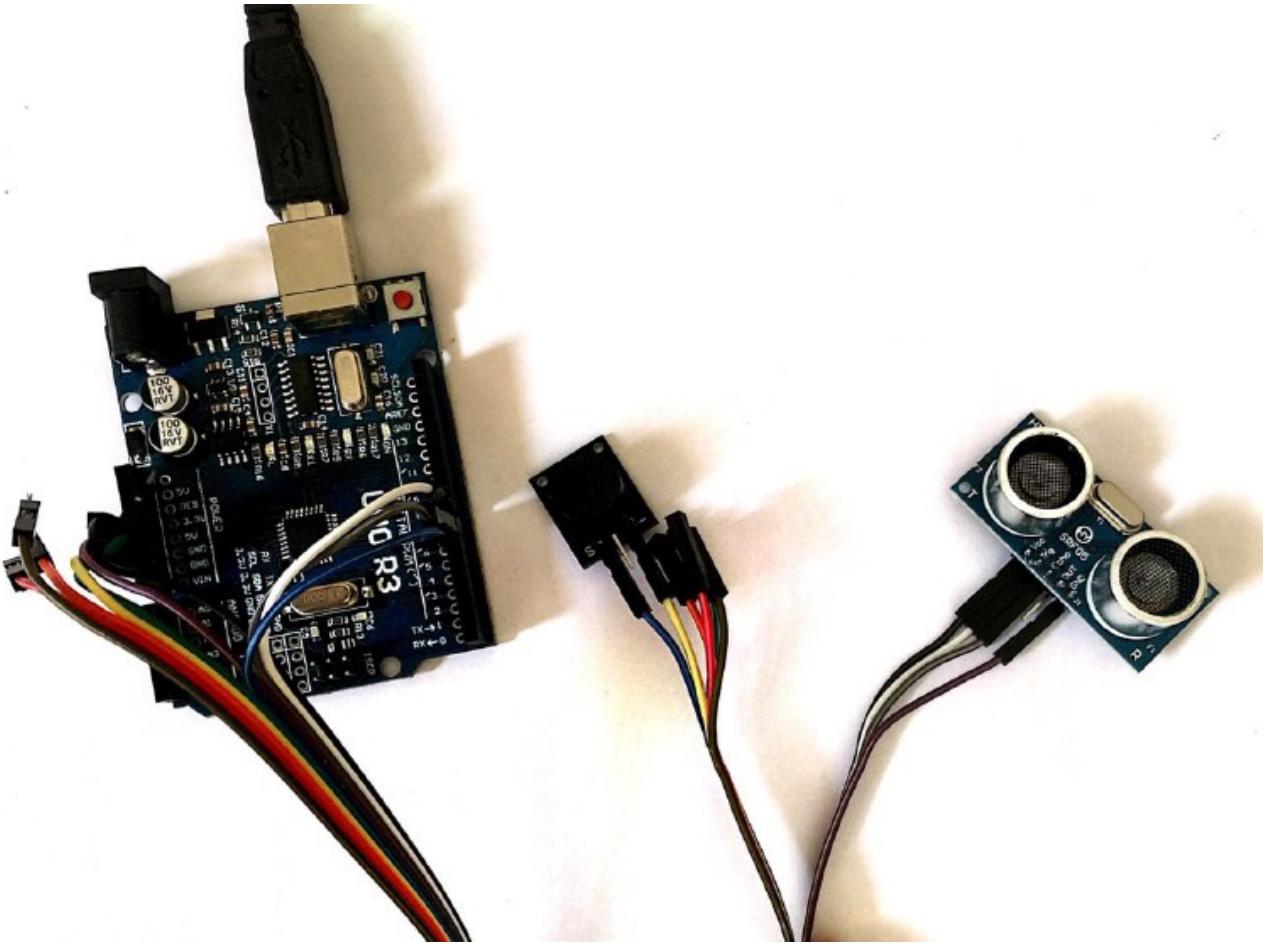
NewPing sonar(TRIGGER_PIN, ECHO_PIN, MAX_DISTANCE); // NewPing setup of
pins and maximum distance.

void setup() {
    Serial.begin(115200); // Open serial monitor at 115200 baud to see
ping results.
}

void loop() {
    delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms
should be the shortest delay between pings.
    unsigned int uS = sonar.ping(); // Send ping, get ping time in
microseconds (uS).
    Serial.print("Ping: ");
    Serial.print(sonar.convert_cm(uS)); // Convert ping time to distance
and print result (0 = outside set distance range, no ping echo)
    Serial.println("cm");
}
```

4.0 طراحی مدار

4.1 در قسمت اول طراحی مدار دو مژول بازر و فاصله سنج را ترکیب کردیم. کد ترکیب این دو به زبان آردوینو است و طوری طراحی شده که با توجه به فاصله‌ی دوربین با جسم، فاصله‌ی صدای تولید شده تغییر کند. همچنین با توجه به این که در این مرحله مدار هنوز به LCD متصل نشده است، فاصله در مانیتور آردوینو و روی پورت 9600 چاپ شده است.



تصویر 12: اتصال دو مژول نهایی به اردوئینو

کد مدار طراحی شده به صورت زیر است:

```
#define trigPin 8
#define echoPin 7
#include <NewPing.h>

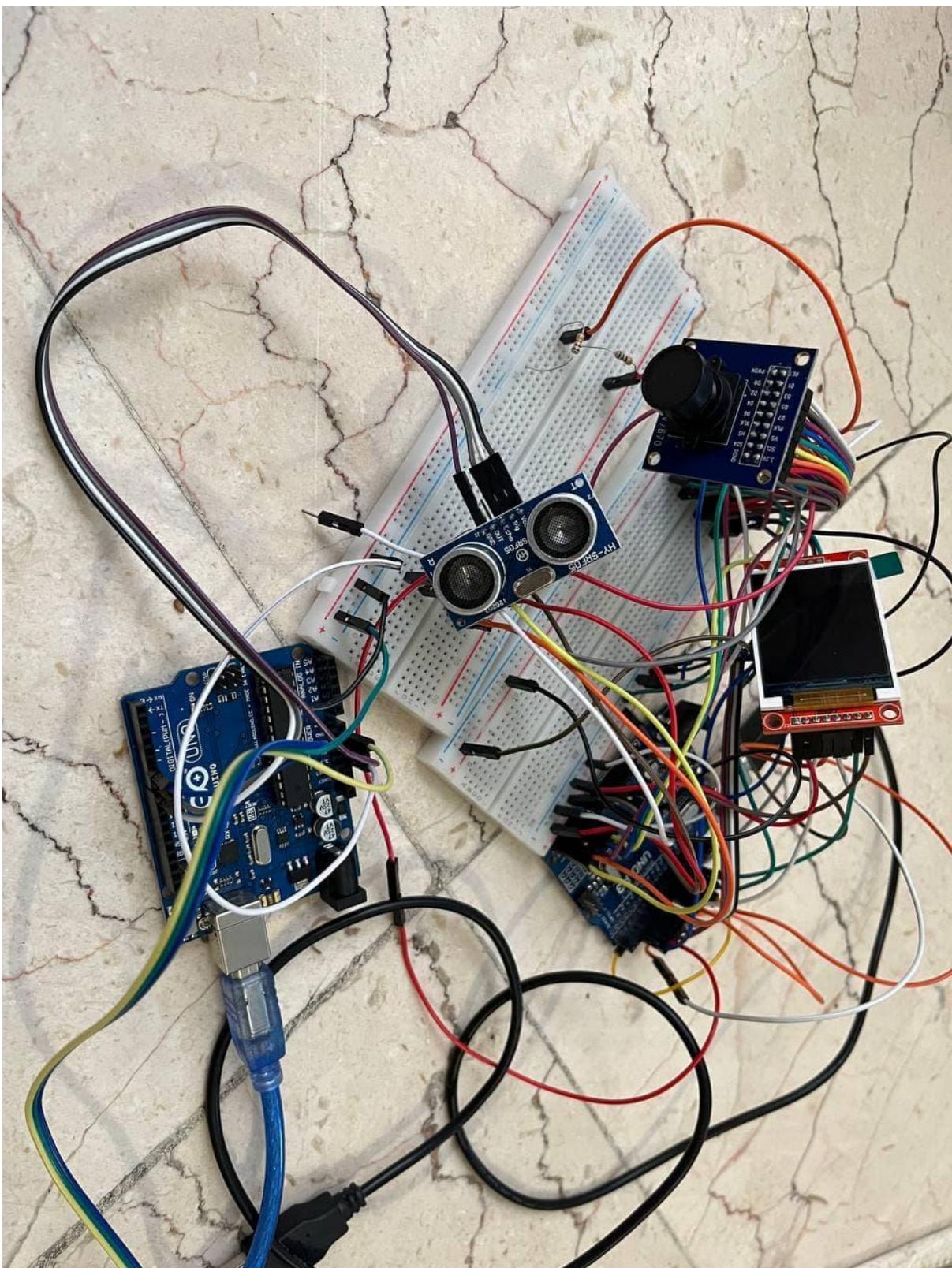
void setup() {
  Serial.begin (9600);
  pinMode(6, OUTPUT);
}

NewPing sonar(trigPin, echoPin, 200);
void loop() {
  delay(50);
  unsigned int uS = sonar.ping(); // Send ping, get ping time in
  Serial.print("Ping: ");
  float distance = sonar.convert_cm(uS);
  Serial.print(distance); // Convert ping time to distance and print result (0 = outside
  set distance range, no ping echo)
  Serial.println("cm");
```

```
If (distance < 100){  
    delay(distance * 100);  
    digitalWrite(6, HIGH);  
    if (distance > 10){  
        delay(distance * 100);  
        digitalWrite(6, LOW);  
    }  
}  
}
```

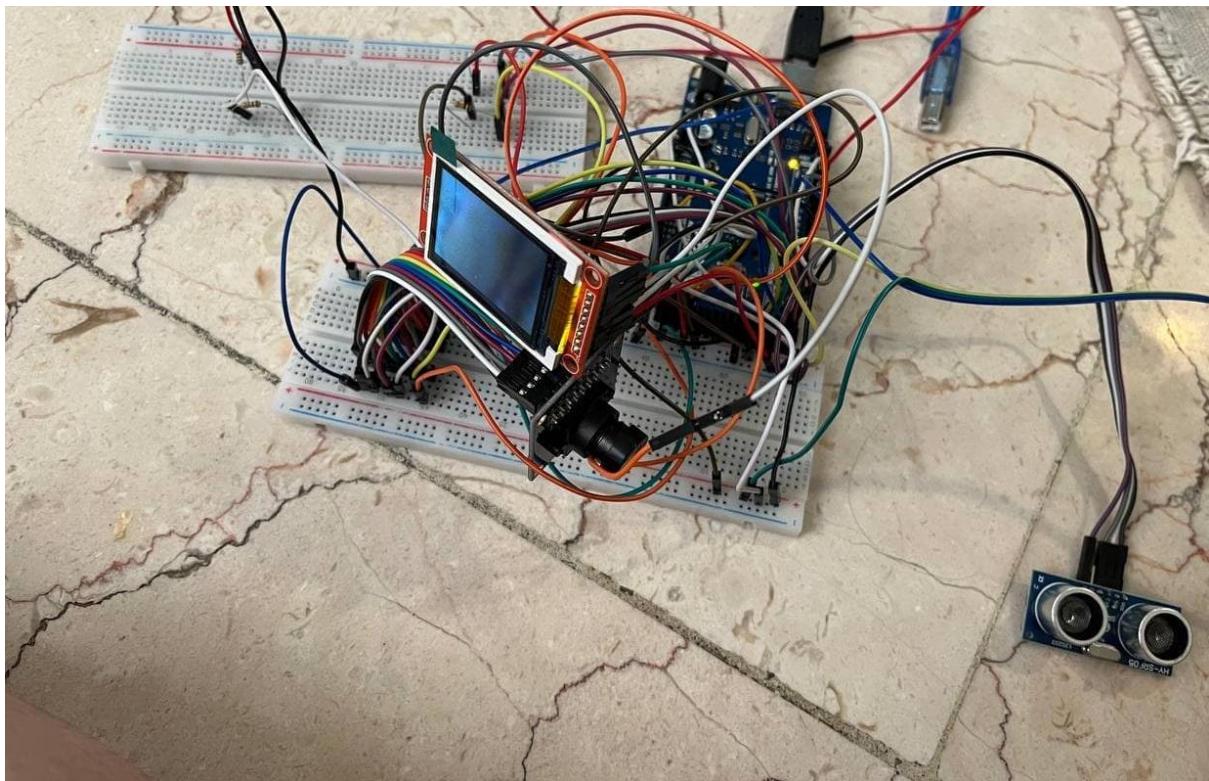
4.2 در قسمت دوم طراحی مدار دو مازول دوربین و LCD را به هم وصل می‌کنیم. دوربین و LCD به طوری به هم وصل شده‌اند که تصویر زنده دوربین روی صفحه نمایشگر نمایش داده شود.

4.3 برای این که فاصله‌ای که مازول SRF05 به دست آورده است، روی نمایشگر نمایش داده شود باید این فاصله از طریق پورت tx آردوینو متصل به فاصله سنج به پورت rx آردوینو متصل به نمایشگر فرستاده شود. سپس این فاصله از در آردوینو مقصد روی نمایشگر نمایش داده شود.



5.0 پیاده سازی نهایی

این دو تا کردن بردهای آردوینو به دلیل شرایط کرونا و در دو شهر مختلف بودن اعضای تیم بود. در پیاده سازی نهایی دو برد ادغام شده و روی برد آردوینو مگا که تعداد پورت بیشتری دارد پیاده سازی شد.



در پیاده سازی نهایی میزان فاصله‌ای که فاصله سنج تشخیص میدارد را به سانتی متر روی صفحه نمایش نشان دادیم.
کدهای مربوط نیز به گیت هاب اضافه شد.

