



A.3.1 Actividad de aprendizaje

Circuito sensor de detección de objetos, utilizando un Arduino, un sensor ultrasonico y un Display I2C LCD 16x2.

Instrucciones

- Realizar un sistema simulado, capaz de detectar la presencia y/o ausencia de un objeto, a través de un circuito electrónico, utilizando un Arduino, y un **Sensor Ultrasonico HC-SR04**.
- Toda actividad o reto se deberá realizar utilizando el estilo **Markdown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces, y debe ser nombrado con la nomenclatura **A3.1_NombreApellido_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo .PDF, el cual fue obtenido desde archivo .MD, ambos deben ser idénticos.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
* readme.md
- blog
  - C3.1_TituloActividad.md
  - C3.2_TituloActividad.md
  - C3.3_TituloActividad.md
  - C3.4_TituloActividad.md
  - C3.5_TituloActividad.md
  - C3.6_TituloActividad.md
  - C3.7_TituloActividad.md
  - C3.8_TituloActividad.md
- img
- docs
  - A3.1_TituloActividad.md
  - A3.2_TituloActividad.md
  - A3.3_TituloActividad.md
```

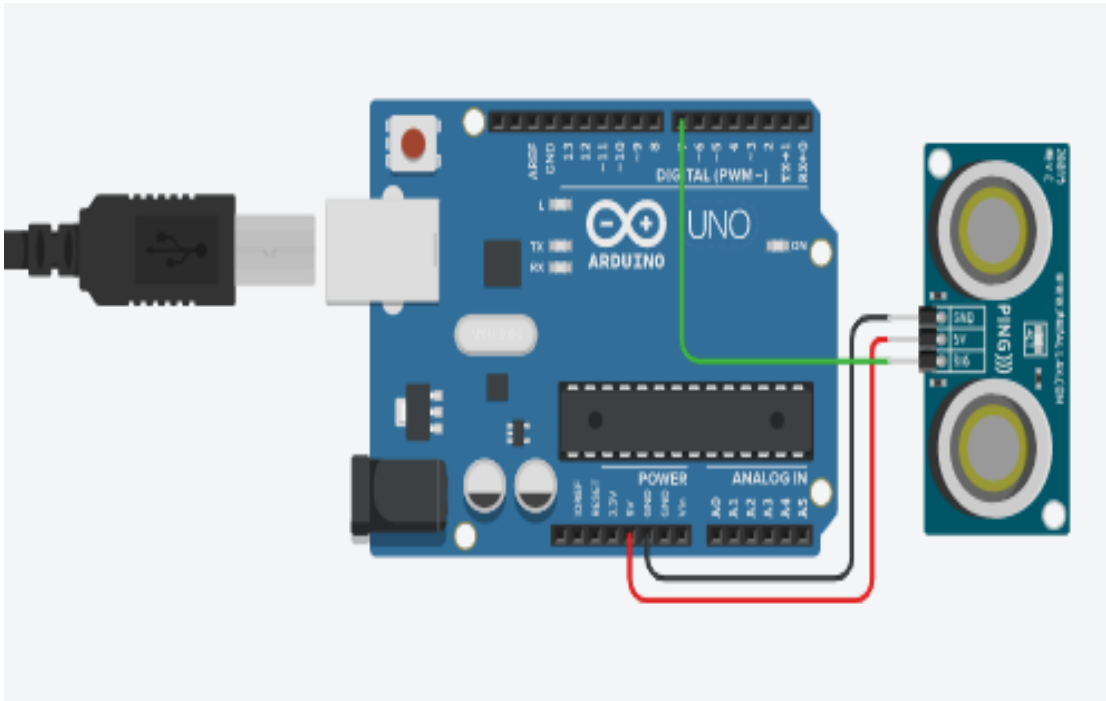
Desarrollo

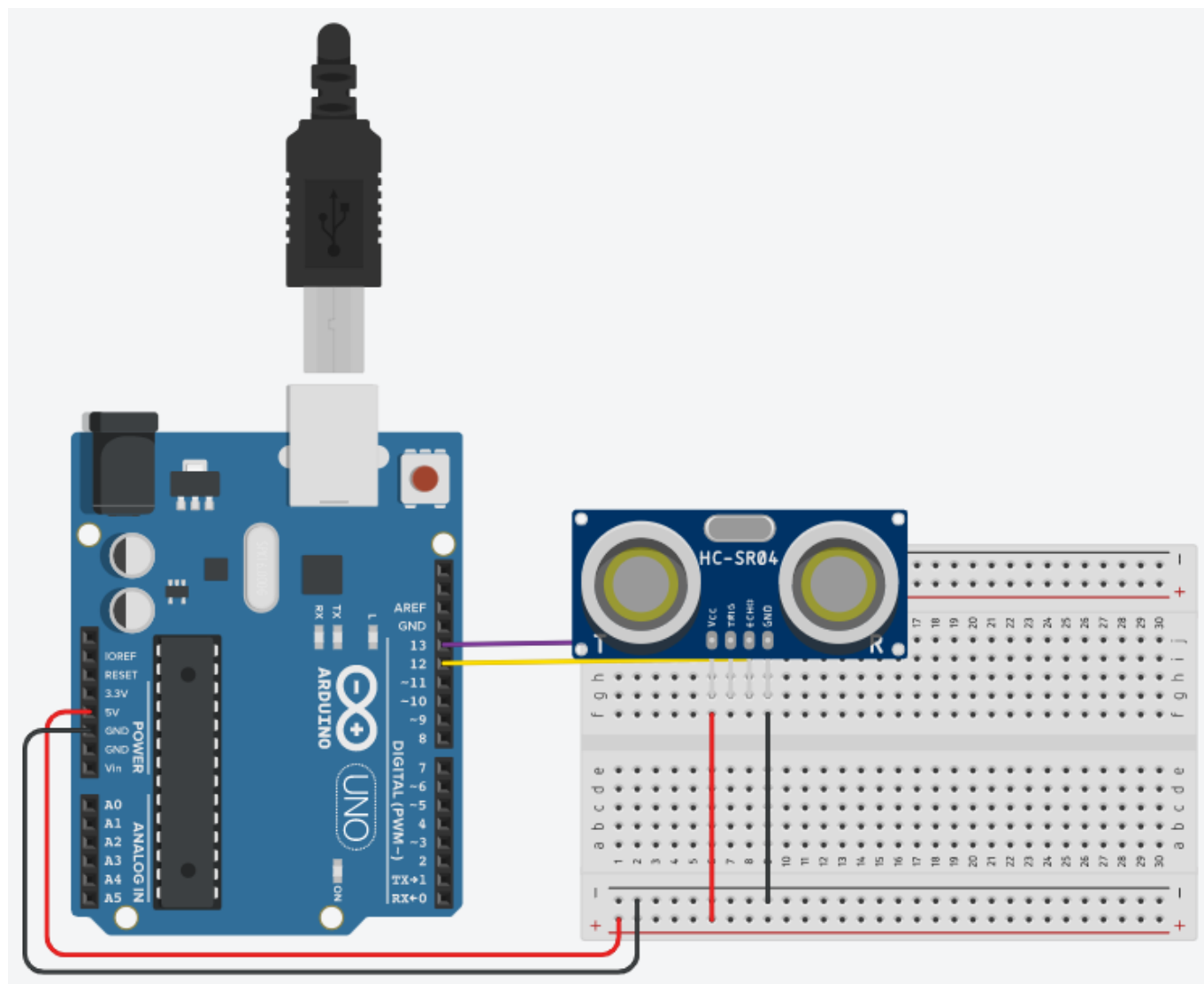
1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

Cantidad	Descripción
1	Sensor Ultrasonico HC-SR04
1	Display LCD de 16x2
1	Fuente de voltaje de 5V
1	1 Potenciómetro 10k
1	Arduino UNO

1. Basado en la imagen que se muestra, ensamble dentro del simulador a utilizar, el circuito electrónico indicado en la **Figura 1**.

Figura 1 Sensor Ultrasonico





2. Realice el programa que permita a través de una de las entradas del Arduino, recibir el valor que registra el **Sensor Ultrasonico** al acercarse un objeto a distintas distancias.

Texto

2 (Arduino Uno R3)

```

1  /*Aquí se configuran los pines donde debemos conectar el sensor*/
2  int TRIGGER_PIN=13;
3  int ECHO_PIN=12;
4  int MAX_DISTANCE;
5  int tiempo;
6
7  /*Iniciar monitor y señalamos nuestras variables como
8  salida o entrada segun el caso */
9  void setup() {
10     Serial.begin(9600);
11     pinMode(TRIGGER_PIN, OUTPUT);
12     pinMode(ECHO_PIN, INPUT);
13
14 }
15
16 /*Calculamos el valor obtenido en cm*/
17 void loop() {
18     digitalWrite(TRIGGER_PIN, HIGH);
19     delay(1);
20     digitalWrite(TRIGGER_PIN, LOW);
21     tiempo = pulseIn(ECHO_PIN, HIGH);
22     MAX_DISTANCE = tiempo/58.4;
23
24     if(MAX_DISTANCE >= 335){
25         Serial.println("NA");
26     }
27
28     else{
29         Serial.print(MAX_DISTANCE),
30         Serial.println(" cm");
31     }
32     delay(1000);|
33 }
34

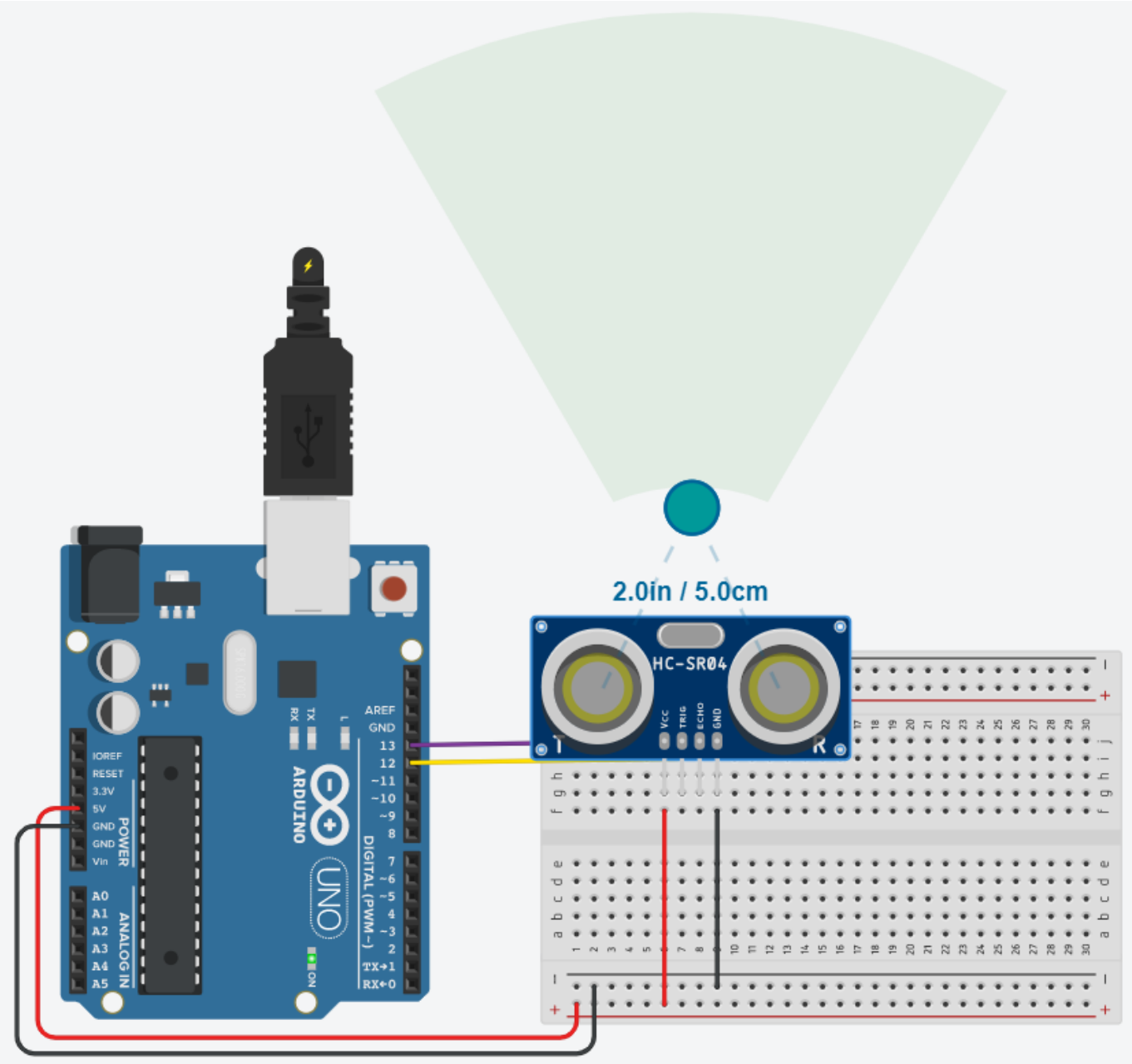
```

3. Considerando que el sensor ultrasonico tiene un rango mínimo y máximo de detección basado en el tiempo de retorno de la señal sonica, que valores se obtienen en la simulación bajo las **siguientes condiciones**:

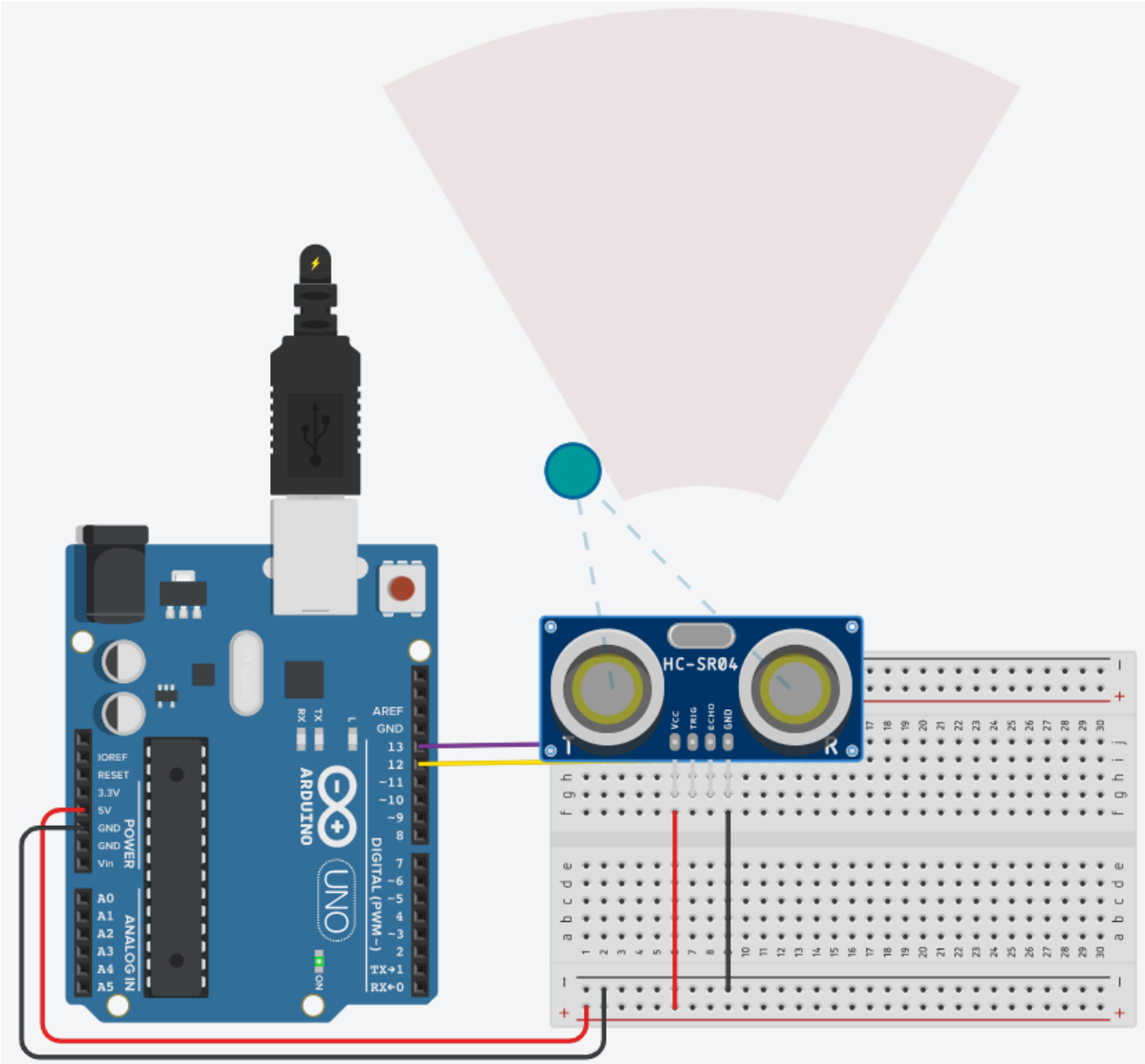
Numero	Condición 1	Condición2	El objeto es detectado?
1	5 cm de distancia al sensor	0 grados al eje perpendicular del sensor	Si
2	50 cm de distancia al sensor	35 grados al eje perpendicular del sensor	No
3	100 cm de distancia al sensor	-35 grados al eje perpendicular del sensor	No
4	5 cm de distancia al sensor	90 grados al eje perpendicular del sensor	No
5	50 cm de distancia al sensor	-60 grados al eje perpendicular del sensor	No

Numero	Condición 1	Condición2	El objeto es detectado?
6	350 cm de distancia al sensor	0 grados al eje perpendicular del sensor	No

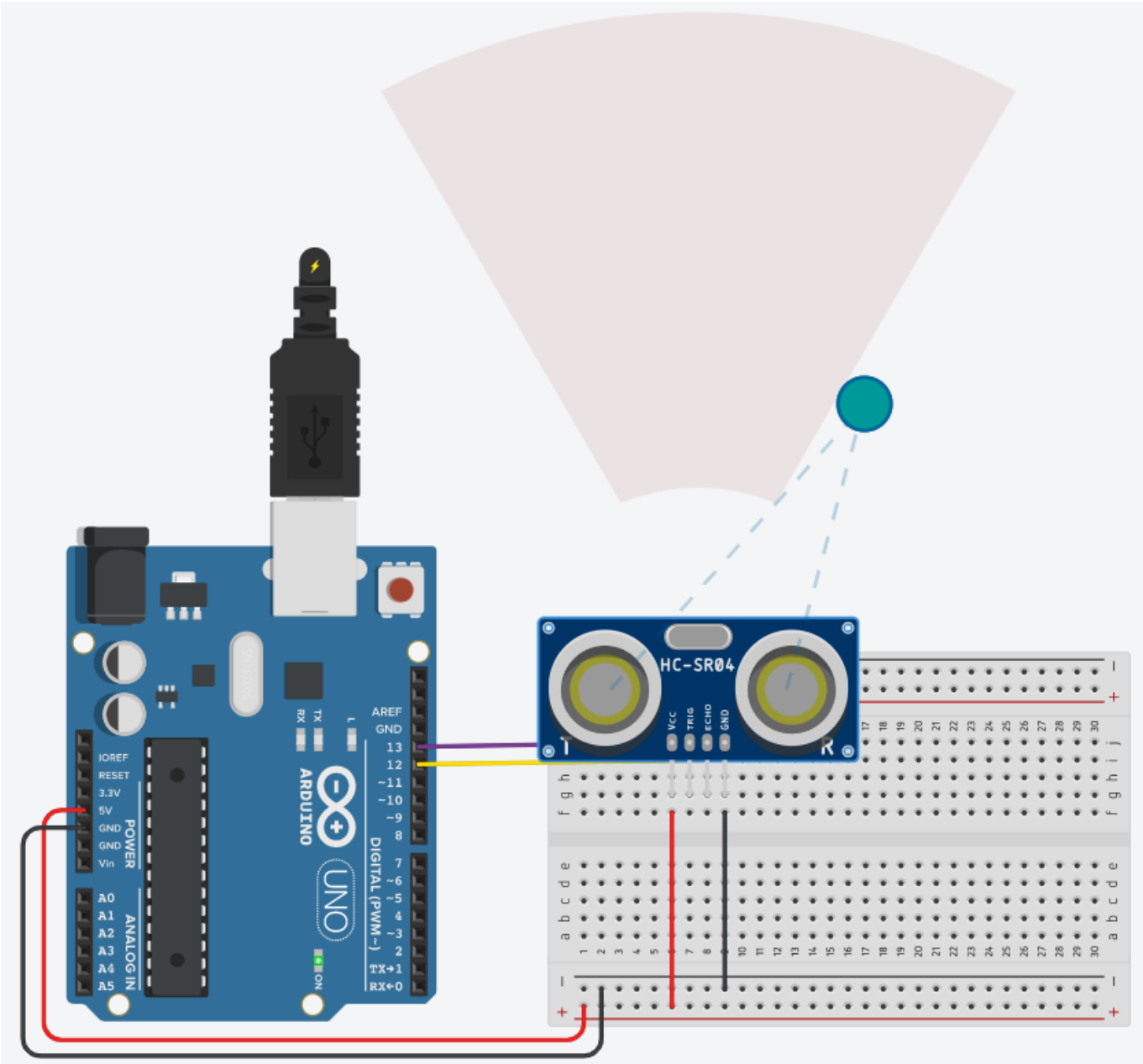
1:



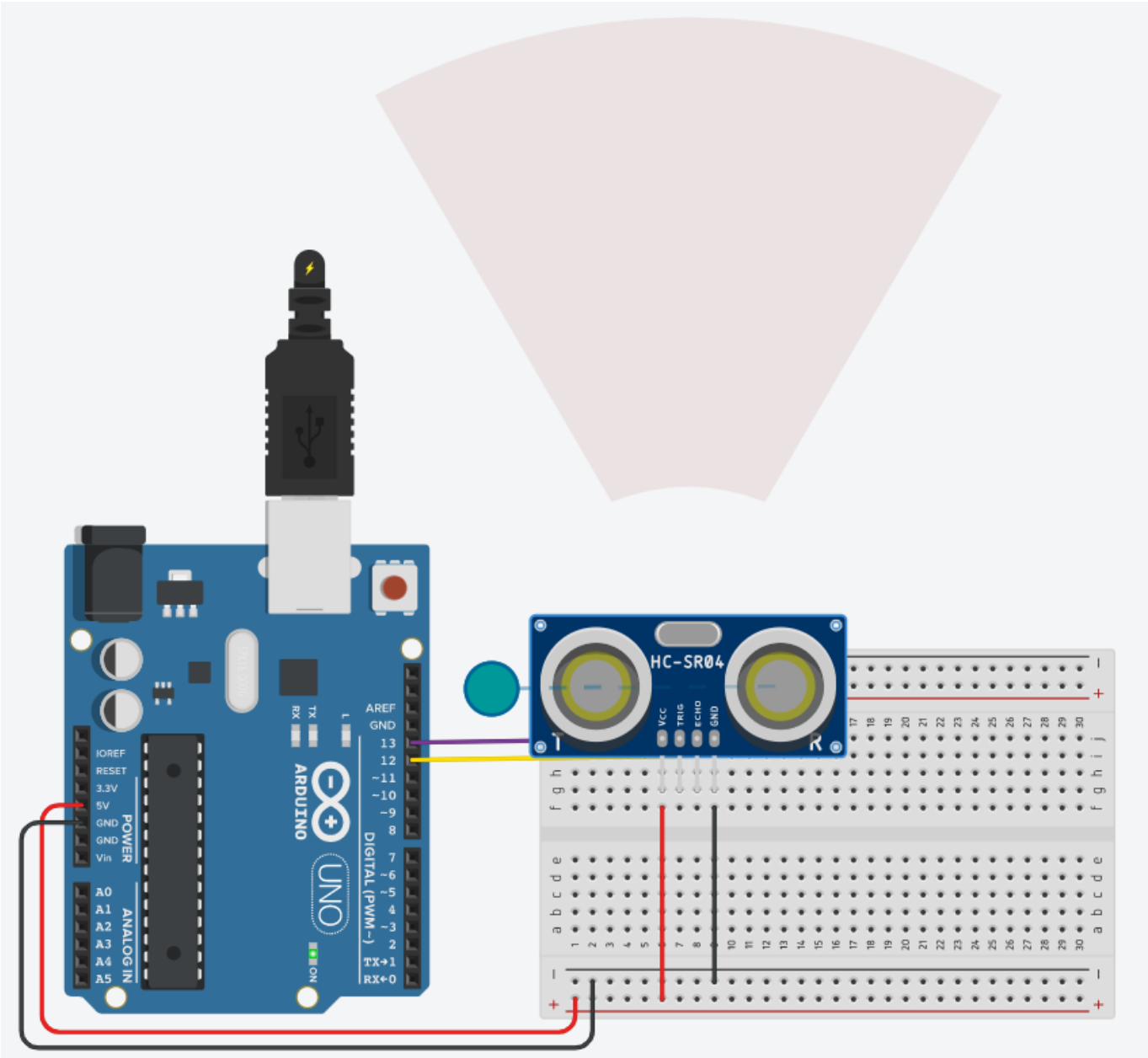
2:



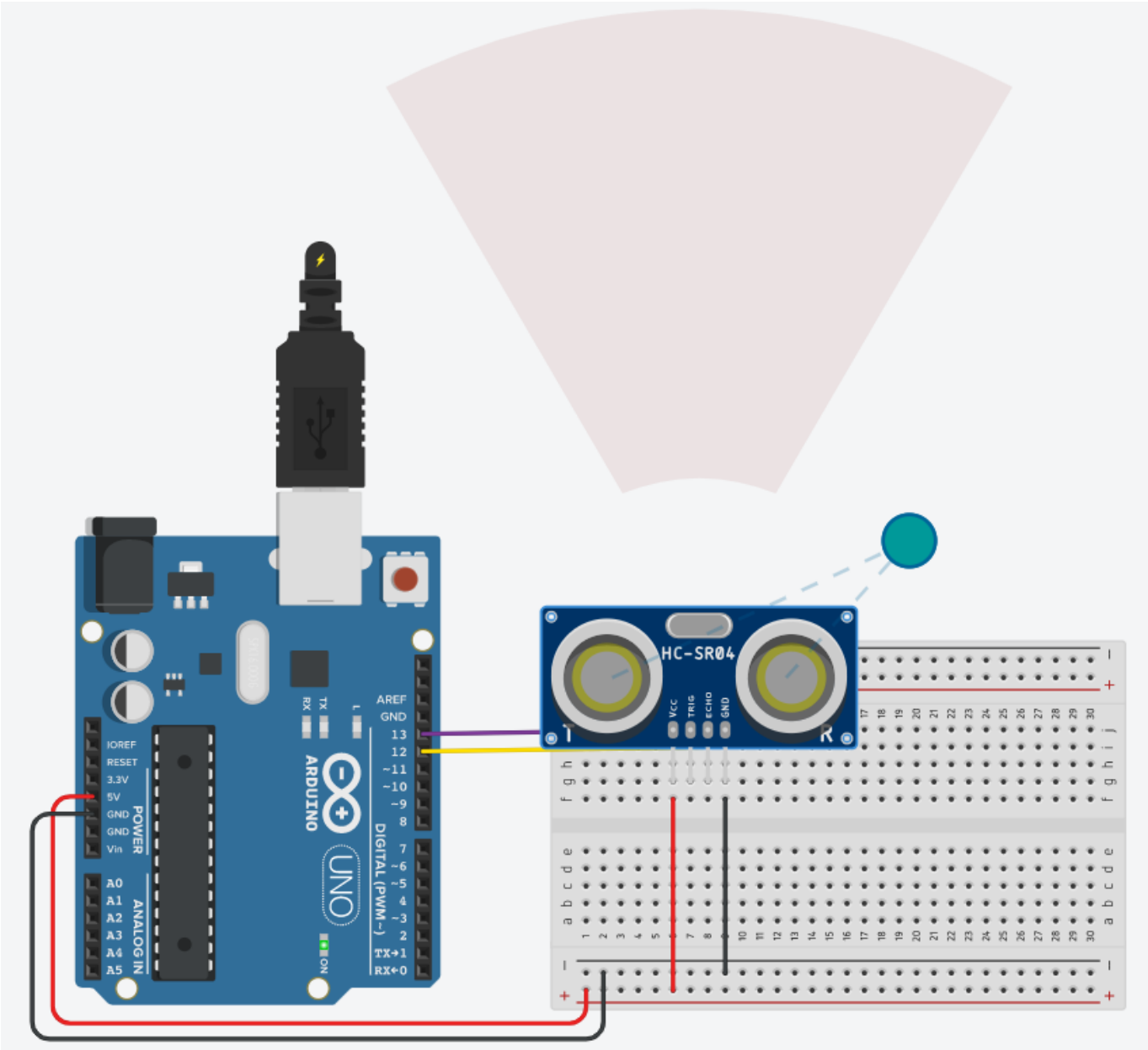
3:



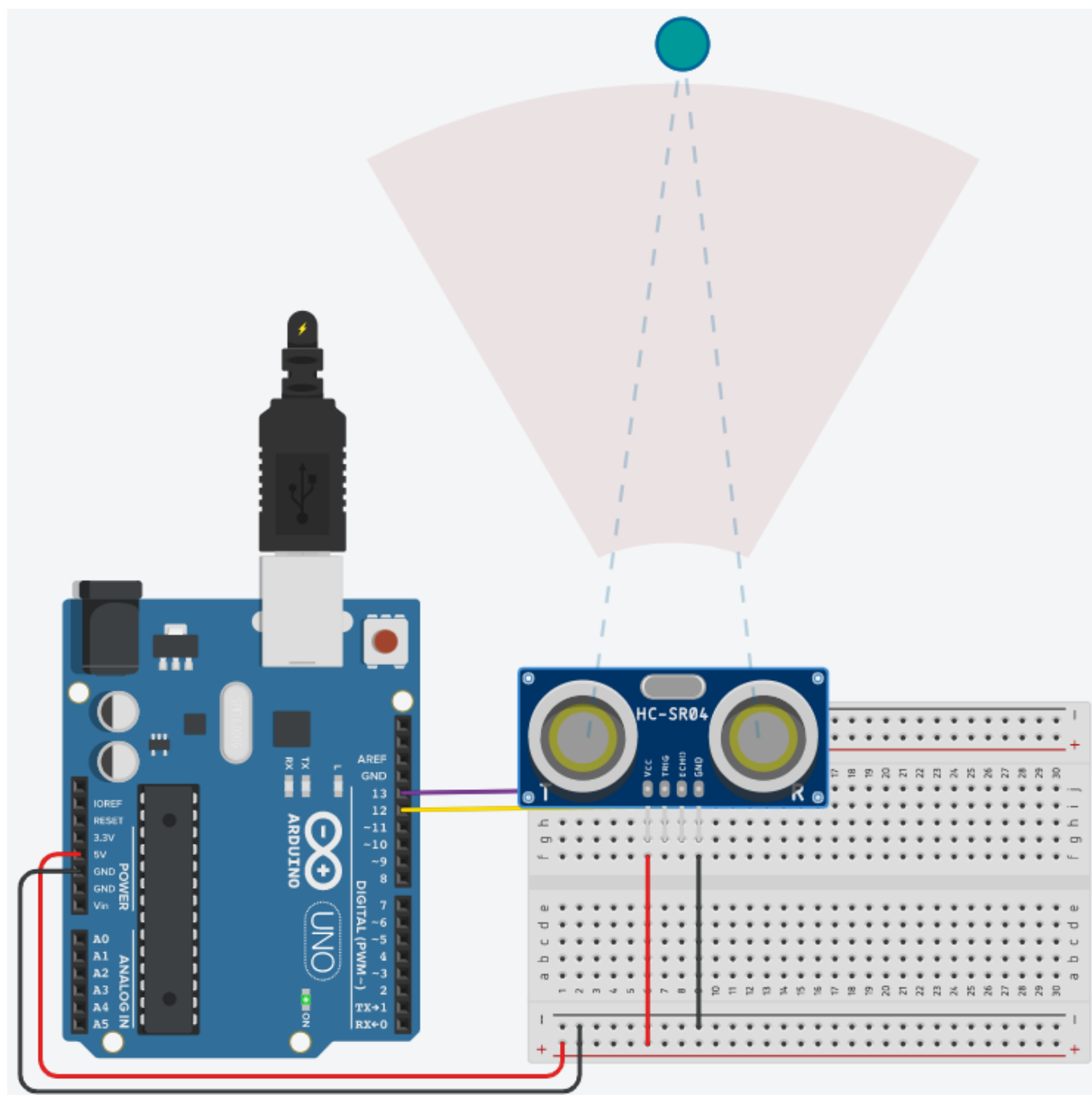
4:



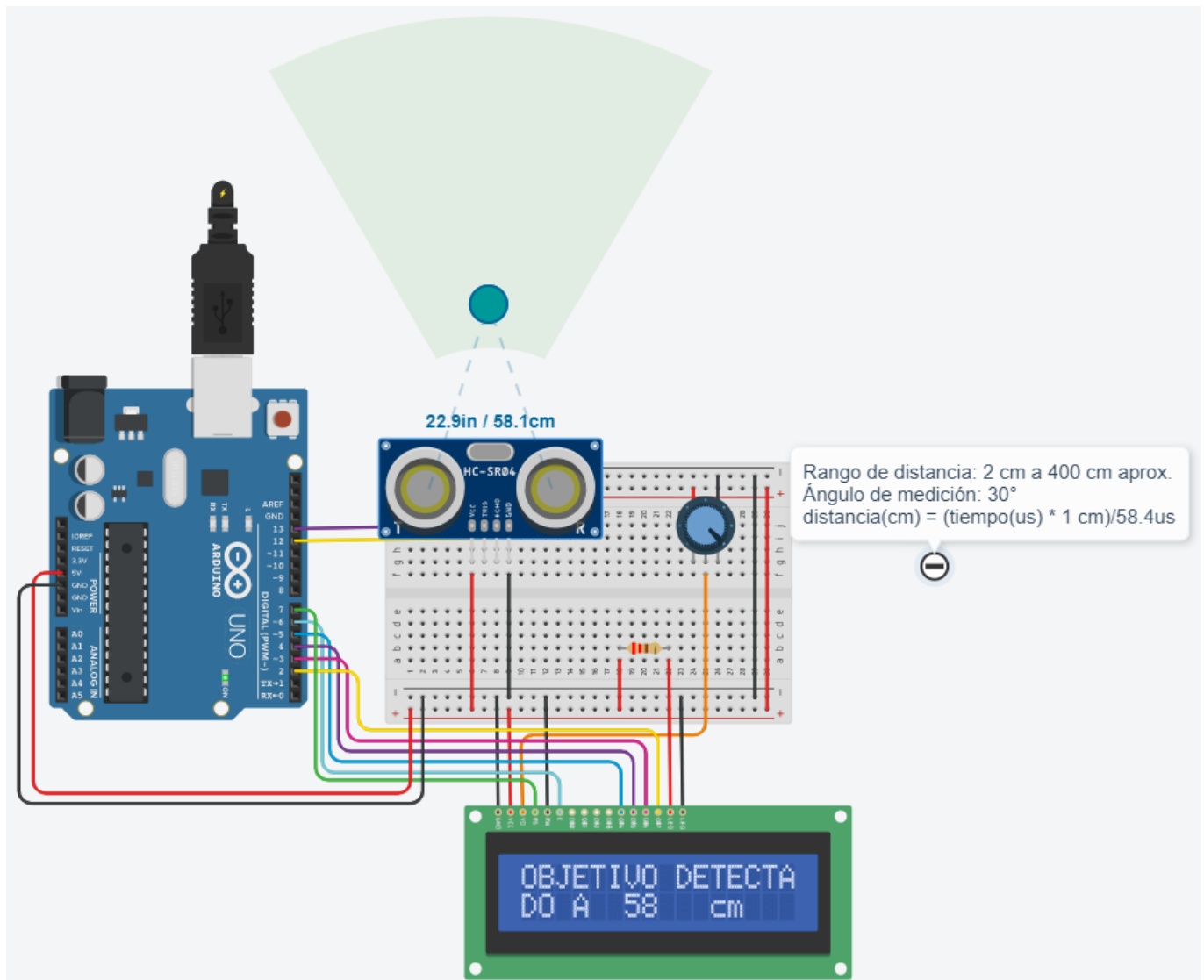
5:



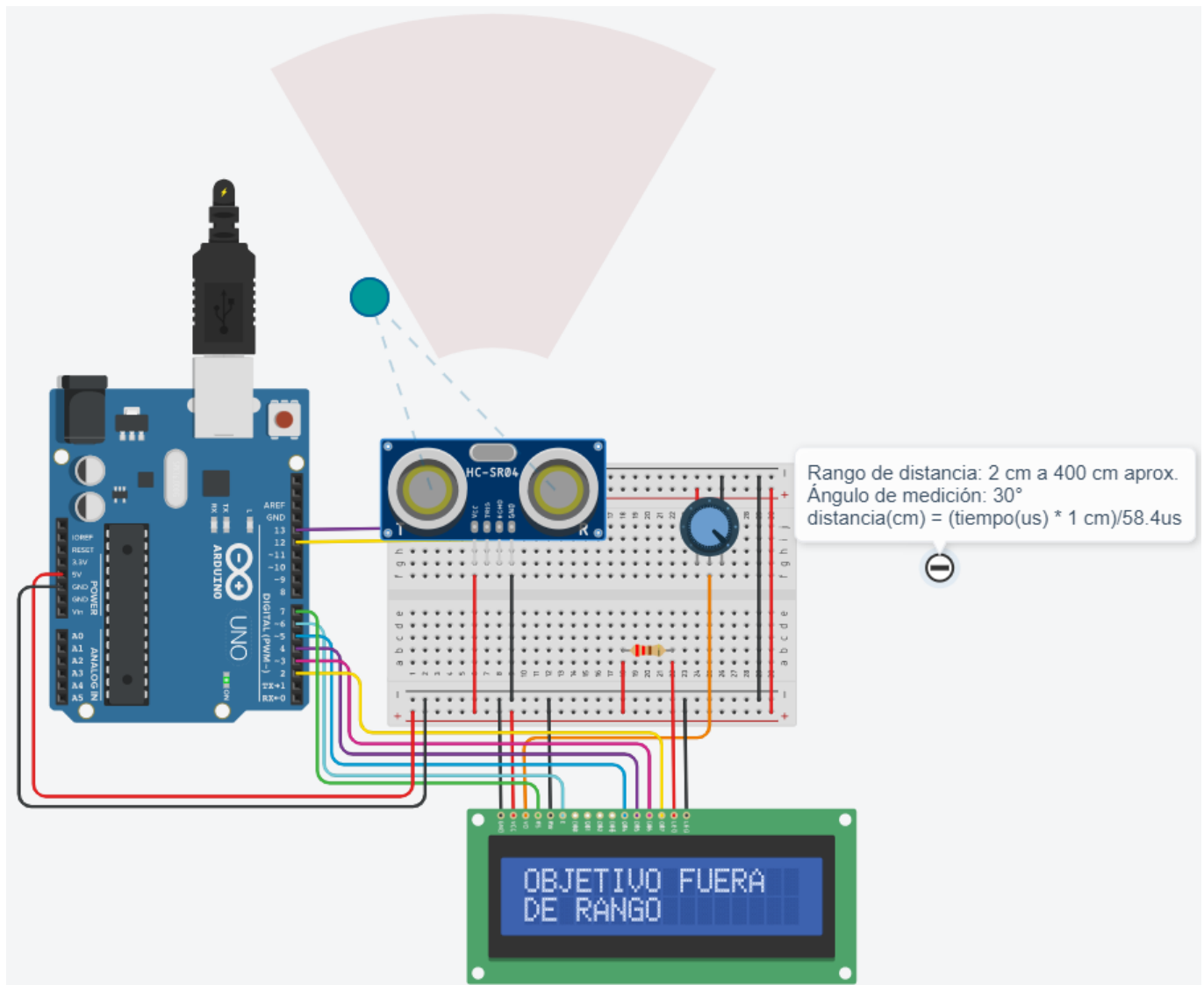
6:



4. Una vez completados los puntos anteriores, agregue a la Figura 1, **un Display I2C 16x2 LCD**, y coloque la imagen del circuito completado hasta este apartado.



- Al haber completado la integración del Display I2C, ajuste el programa que le permita a través del display mostrar el siguiente mensaje, "**Objetivo detectado a ? cm**", y en caso de no lograr la detección indicar el mensaje "**Objetivo fuera de rango**"



Texto

↓

📁

🔍

1 (Arduino Uno R3)

```

1  #include <LiquidCrystal.h>
2  //Se incluye libreria LiquidCrystal
3  LiquidCrystal lcd(7,6,5,4,3,2);
4
5  /*Aquí se configuran los pines donde debemos conectar el sensor*/
6  int TRIGGER_PIN=13;
7  int ECHO_PIN=12;
8  int MAX_DISTANCE;
9  int tiempo;
10
11 /*Iniciar monitor y señalamos nuestras variables como salida o entrada segun el caso */
12 void setup() {
13   Serial.begin(9600);
14   pinMode(TRIGGER_PIN, OUTPUT);
15   pinMode(ECHO_PIN, INPUT);
16 }
17
18 /*Calculamos el valor obtenido en cm*/
19 void loop() {
20   lcd.begin(16,2);
21   digitalWrite(TRIGGER_PIN, HIGH);
22   delay(1);
23   digitalWrite(TRIGGER_PIN, LOW);
24   tiempo = pulseIn(ECHO_PIN, HIGH);
25   MAX_DISTANCE = tiempo/58;
26
27   if(MAX_DISTANCE >= 335){
28     lcd.setCursor(0,0);
29     lcd.print("OBJETIVO FUERA");
30     lcd.setCursor(0,1);
31     lcd.print("DE RANGO");
32     Serial.println("NA");
33   }

```

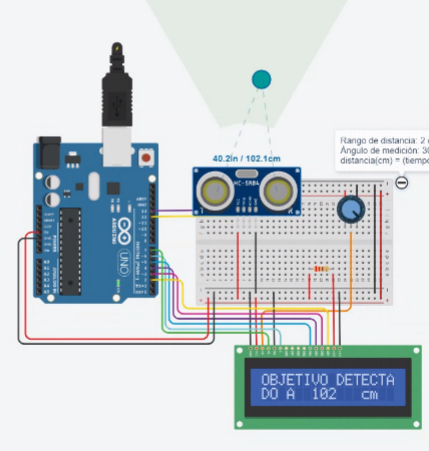
```
34  
35     else{  
36         lcd.setCursor(0,0);  
37         lcd.print("OBJETIVO DETECTA");  
38         lcd.setCursor(0,1);  
39         lcd.print("DO A ");  
40         lcd.setCursor(6,1);  
41         lcd.print(MAX_DISTANCE);  
42         lcd.setCursor(10,1);  
43         lcd.print(" cm");  
44         Serial.print(MAX_DISTANCE),  
45         Serial.println(" cm");  
46     }  
47     delay(1000);  
48 }  
49
```

6. Coloque aqui evidencias que considere importantes durante el desarrollo de la actividad.

Jose Venegas está presentando

Se han guardado todos los cambios.

Horas de simulador: 00:00:10.237



40.2in / 102.1cm

Rango de distancia: 2 cm a 400 cm aprox.
Angulo de medición: 30°
distancia(cm) = (tiempo(us) * 1 cm)/58.4us

OBJETIVO DETECTA DO A 102 cm

```

1 #include <LiquidCrystal.h>
2 //Se incluye libreria LiquidCrystal
3 LiquidCrystal lcd(7,6,5,4,3,2);
4
5 /*Aquí se configuran los pines donde debemos conectar el sensor*/
6 const int trigPin = 9;
7 const int echoPin = 10;
8 int duracion;
9 float distancia;
10
11 /*Iniciamos monitor y establecemos nuestras variables como salida o entrada*/
12 void setup() {
13   Serial.begin(9600);
14   pinMode(TRIGGER_PIN, OUTPUT);
15   pinMode(ECHO_PIN, INPUT);
16 }
17
18 /*Calculamos el valor obtenido en cm*/
19 void loop() {
20   digitalWrite(TRIGGER_PIN, HIGH);
21   delay(1);
22   digitalWrite(TRIGGER_PIN, LOW);
23   tiempo = pulseIn(ECHO_PIN, HIGH);
24   MAX_DISTANCE = tiempo/58;
25
26   if (MAX_DISTANCE >= 229) {
27     lcd.setCursor(0,0);
28     lcd.print("OBJETIVO FUERA");
29     lcd.setCursor(0,1);
30     lcd.print("100 a. ");
31     lcd.setCursor(0,1);
32     lcd.print(MAX_DISTANCE);
33     Serial.print("MAX DISTANCE");
34     delay(1000);
35   }
36   else {
37     lcd.setCursor(0,0);
38     lcd.print("OBJETIVO DETECTA");
39     lcd.setCursor(0,1);
40     lcd.print("100 a. ");
41     lcd.setCursor(0,1);
42     lcd.print(MAX_DISTANCE);
43     Serial.print("MAX DISTANCE");
44     delay(1000);
45   }
46 }

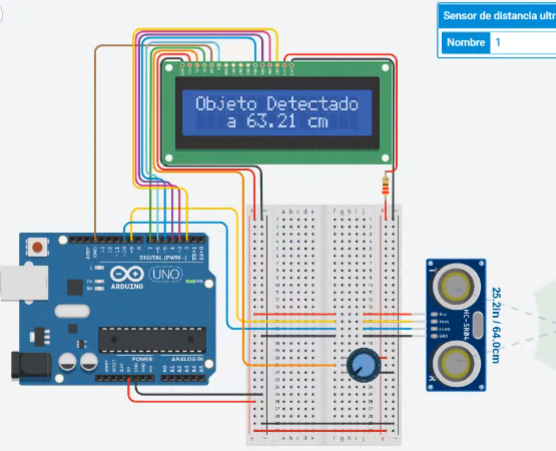
```

Monitor en serie

CESAR MANUEL RAMIREZ CERVANTES está presentando

Se han guardado todos los cambios.

Horas de simulador: 00:00:01.258



25.2in / 64.0cm

Objeto Detectado a 63.21 cm

```

1 #include <LiquidCrystal.h>
2 LiquidCrystal lcd(7,6,5,4,3,2);
3
4 const int trigPin = 9;
5 const int echoPin = 10;
6 int duracion;
7 float distancia;
8
9 void setup() {
10   lcd.begin(16,2);
11   lcd.setCursor(0,0);
12   lcd.print("Objeto Detectado");
13
14   pinMode(trigPin, OUTPUT);
15   pinMode(echoPin, INPUT);
16   Serial.begin(9600);
17 }
18
19 void loop() {
20   digitalWrite(trigPin, LOW);
21   delayMicroseconds(2);
22   digitalWrite(trigPin, HIGH);
23   delayMicroseconds(10);
24   digitalWrite(trigPin, LOW);
25
26   duracion = pulseIn(echoPin, HIGH);
27   Serial.println(duracion);
28   distancia = (duracion * 0.034) / 2;
29
30   lcd.setCursor(0,1);
31   lcd.print(" = ");
32   lcd.print(distancia);
33   lcd.print(" cm");
34 }

```

Monitor en serie

DULCE JASMIN VILLALOBOS PEREZ está presentando

Se han guardado todos los cambios.

1 (Arduino Uno R3)

```

1 #include <LiquidCrystal.h>
2 //Se incluye libreria LiquidCrystal
3 LiquidCrystal lcd(7,6,5,4,3,2);
4
5 /*Aquí se configuran los pines donde debemos conectar el sensor*/
6 int TRIGGER_PIN=13;
7 int ECHO_PIN=12;
8 float MAX_DISTANCE;
9 float tiempo;
10
11 /*Iniciar monitor y sealamos nuestras variables como salida o e
12 void setup() {
13   Serial.begin(9600);
14   pinMode(TRIGGER_PIN, OUTPUT);
15   pinMode(ECHO_PIN, INPUT);
16 }
17
18 /*Calculamos el valor obtenido en cm*/
19 void loop() {
20   lcd.begin(16,2);
21   digitalWrite(TRIGGER_PIN, HIGH);
22   delay(1);
23   digitalWrite(TRIGGER_PIN, LOW);
24   tiempo = pulseIn(ECHO_PIN, HIGH);
25   MAX_DISTANCE = tiempo/58;
26
27   if(MAX_DISTANCE >= 338){
28     lcd.setCursor(0,0);
29     lcd.print("OBJETIVO FUERA");
30     lcd.setCursor(0,1);
31     lcd.print("DE RANGO");
32     Serial.println("HA");
33   }
34   else{
35     lcd.setCursor(0,0);
36     lcd.print("OBJETIVO DETECTA");
37     lcd.setCursor(0,1);
38     lcd.print("DO A ");
39     lcd.setCursor(6,1);
40     lcd.print(MAX_DISTANCE);
41     lcd.print("cm");
42   }
43 }

```

Monitor en serie

5:29 PM 11/9/2020

PEDRO GABRIEL ACEVEDO ENSISO está presentando

All changes saved

1 (Arduino Uno R3)

```

1 #include <LiquidCrystal.h>
2 //Se incluye libreria LiquidCrystal
3 LiquidCrystal lcd(7,6,5,4,3,2);
4
5 /*Aquí se configuran los pines donde debemos conectar el sensor*/
6 int TRIGGER_PIN=13;
7 int ECHO_PIN=12;
8 float MAX_DISTANCE;
9 float tiempo;
10
11 /*Iniciar monitor y sealamos nuestras variables como salida o e
12 void setup() {
13   Serial.begin(9600);
14   pinMode(TRIGGER_PIN, OUTPUT);
15   pinMode(ECHO_PIN, INPUT);
16 }
17
18 /*Calculamos el valor obtenido en cm*/
19 void loop() {
20   lcd.begin(16,2);
21   digitalWrite(TRIGGER_PIN, HIGH);
22   delay(1);
23   digitalWrite(TRIGGER_PIN, LOW);
24   tiempo = pulseIn(ECHO_PIN, HIGH);
25   MAX_DISTANCE = tiempo/58;
26
27   if(MAX_DISTANCE >= 338){
28     lcd.setCursor(0,0);
29     lcd.print("OBJETIVO FUERA");
30     lcd.setCursor(0,1);
31     lcd.print("DE RANGO");
32     Serial.println("HA");
33   }
34   else{
35     lcd.setCursor(0,0);
36     lcd.print("OBJETIVO DETECTA");
37     lcd.setCursor(0,1);
38     lcd.print("DO A ");
39     lcd.setCursor(6,1);
40     lcd.print(MAX_DISTANCE);
41     lcd.print("cm");
42   }
43 }

```

Serial Monitor

48.87 cm
2838.00
40.39 cm
2839.00
49.48 cm
2840.00
2849.00

5:30 PM 11/9/2020

yugiooooooh

Personas (5)

Añadir personas

Jose Venegas (Tú)

CESAR MANUEL RAMIREZ CERVANT...

DULCE JASMIN VILLALOBOS PEREZ

PEDRO GABRIEL ACEVEDO ENSISO (p...)

PEDRO GABRIEL ACEVEDO ENSISO

7. Conclusiones:

- Acevedo Ensiso Pedro Gabriel:**

Dentro de esta práctica armamos un circuito donde detectamos un objeto y calculamos su distancia utilizando un sensor ultrasónico utilizando un simulador, la manera en que armamos este circuito va altamente relacionado con el código, primero instanciamos un objeto de la clase "LiquidCrystal" a la cual llamamos "lcd" a la hora de instanciar utilizamos la siguiente sintaxis: LiquidCrystal(rs, enable, d4, d5, d6, d7), aquí introducimos qué pines utilizaremos dentro arduino y los conectamos en su lugar correspondiente en el display LCD. En cuanto al sensor decidimos utilizar los pines 12 y 13 de nuestro Arduino los cuales serán nuestro trigger y echo respectivamente. En cuanto al resto de los pines simplemente los conectamos a su respectiva alimentación y tierra lógica según lo indique la ficha técnica. Algo curioso que cabe mencionar es que el sensor es capaz de detectar objetos hasta 350 cm

en vida real, pero en el simulador solo detecta objetos a una distancia máxima de 320 cm (aproximadamente) lo cual puede causar resultados curiosos, también hay que mencionar que los ángulos de funcionamiento del sensor ultrasónico son de 15 grados (cono de 30 grados) máximos en vida real, pero el simulador permite detectar objetos que se encuentran a 30 grados (cono de 60 grados).

- **Ramirez Cervantes Cesar Manuel:**

Como podemos observar con el siguiente circuito somos capaces de detectar objetos que se encuentren dentro del rango de funcionamiento del circuito ultrasónico y al ser detectado podemos hacer funcionar diversos mecanismos en respuesta a la distancia encontrada, en este caso usamos un display lcd para mostrar la distancia a la que se encuentra el objeto, aunque en vida real el HC-SR04 tiene un rango de funcionamiento de hasta 400 cm y un ángulo de detección de 15 grados y en el simulador en tinkercad encontramos que tenemos un ángulo de 35 grados y un rango de 330cm, esto ocasionó algunas discrepancias en los resultados comparado con un ultrasonico real.

- **Venegas Medina Jose Alfredo:**

Primeramente, realizamos un circuito en el simulador de Tinkercad, este fue sencillo de hacer, ya que anteriormente habíamos expuesto este tipo de sensor en la actividad A1.3, pudimos comprobar que este sensor manda y regresa sonido, es decir, usa la acústica para poder detectar la distancia de un objeto determinado. Luego de esto, desarrollamos el código para su implementación con Arduino, el cual declarábamos los pines de Tigger como salida y Echo como entrada. Así el Arduino detectara cuanto tiempo tardara el sensor en regresar la señal, una vez con el tiempo establecido, lo dividimos por 58.4 y obtuvimos la distancia. Un detalle que notamos, fue que el sensor tiene un ángulo de 30 grados, es decir de 15 a -15 (perpendicularmente), por lo tanto no deberían poder detectar las distancias 2, 3, 4 y 5, pero por otro lado, las distancias 1 y 6 si deberían poderse detectar según el datasheet del sensor, no obstante, el sensor simulado en Tinkercad solo detecta una distancia máxima de 336 cm aproximadamente y, ya que el sensor debería poder llegar a los 400 cm, podemos concluir que hay una diferencia grande en el simulador. Después agregamos un LCD por medio de la librería "LiquidCrystal", este fue un poco más complicado de controlar, ya que se desconocía su implementación, se agregó un potenciómetro para controlar el brillo de led del display y se programó para que imprimiera a que distancia en cm se encontraba el objeto del sensor, si estaba fuera entonces el LCD mostraría "Objetivo fuera de rango".

- **Villalobos Perez Dulce Jasmin:**

Para la realización de la actividad primero identificamos los materiales que usaríamos en la práctica, y lo simulamos en tinkercad, al realizar la primera parte, medimos la distancia a la que se encontraban los objetos a ciertos grados, para ello escribimos la fórmula para calcular la distasia en el código, luego llenamos la tabla de acuerdo a lo que fuimos observando, a mover el objeto note que se salía del rango de visualización del sensor ultrasónico ya que como sabemos va de 2 cm a 400 aprox., pero dentro de la práctica el simulador solo detecta hasta 335 cm aprox. . Entonces determinamos que ciertas condiciones no eran posible identificar el objeto, luego con ayuda de un video conectamos el display, y añadimos al código la parte de imprimir los mensajes por eso específica las dimensiones (ancho y alto) de la pantalla, pero antes añadimos una librería para el número del pin que está conectado al pin RS e hicimos uso de las funciones.



Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10

EQUIPO VERDE**Acevedo Ensiso Pedro Gabriel:** [Ir a mi Github](#)**Ramirez Cervantes Cesar Manuel:** [Enlace a mi repositorio](#) **Venegas Medina Jose Alfredo:** [Mi Github](#)**Villalobos Perez Dulce Jasmin:**  [ENLACE - MI GITHUB](#)