

***Asignatura:***

Programación de sistemas

***Sustentado por:***

Chelsea Massiell Suazo García – 61911443

**Dirigido:**

Ing. Kevin Cruz

***Asignación:***

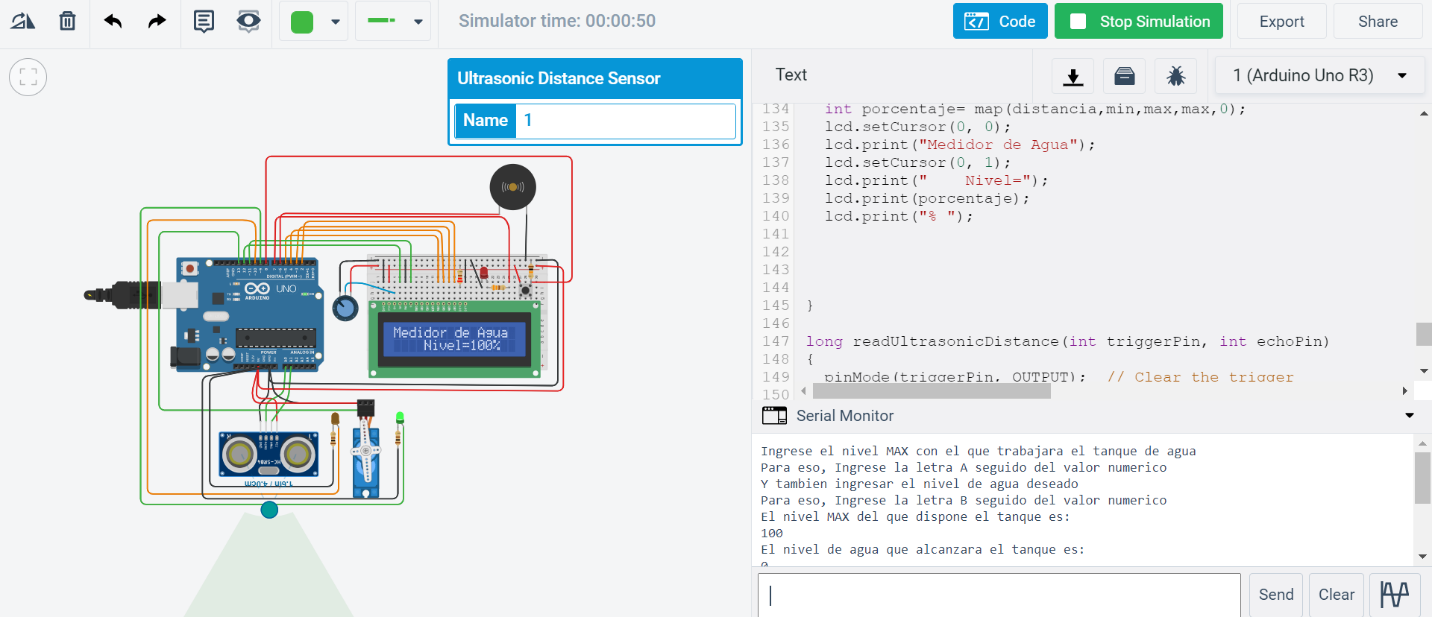
Tarea 4

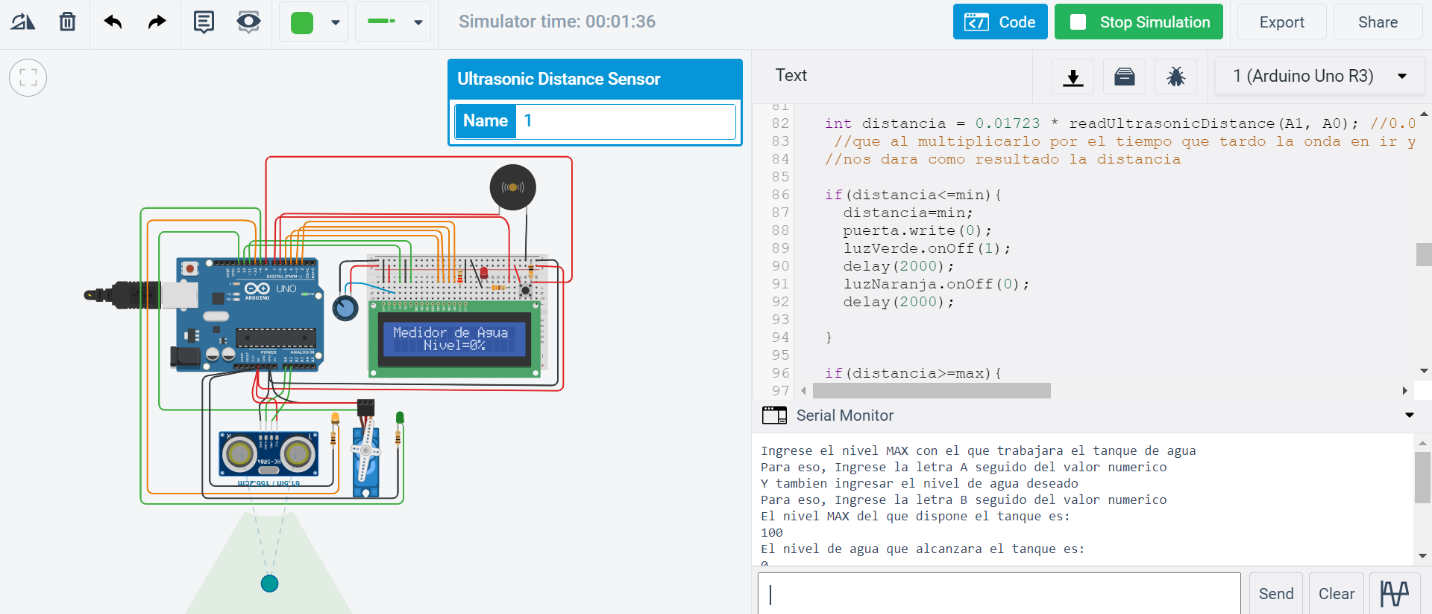
***Campus:***

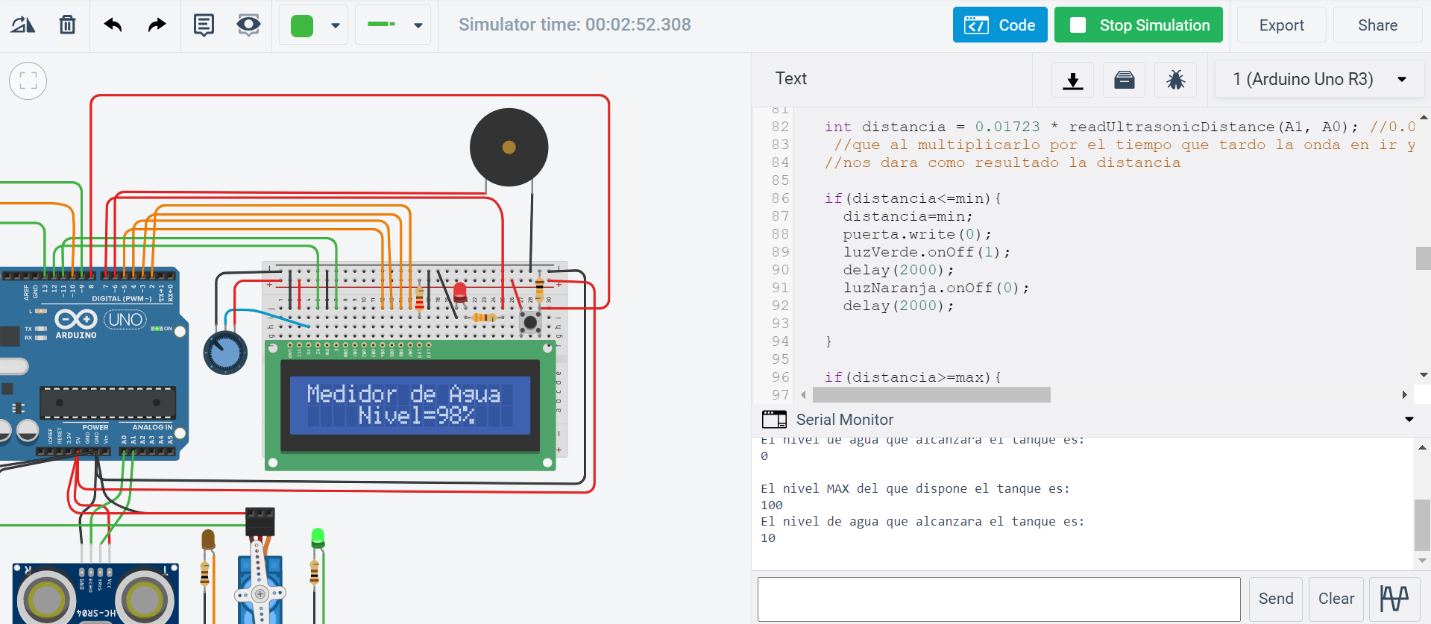
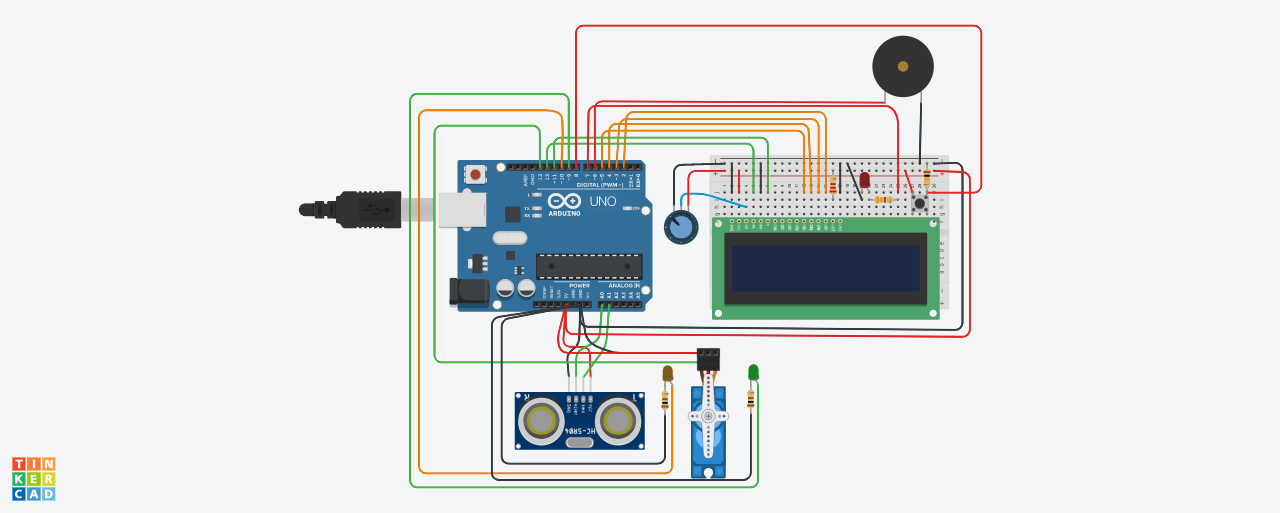
Ceutec Sede Norte

***Fecha:***

23/05/2021







// include the library code:

#include <LiquidCrystal.h>

#include <Servo.h>

Servo puerta;

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int max=0, min=0;

char n;

int ESTADO = HIGH;

//POO

class LED{

public:

LED(byte puerto){

pin=puerto;

pinMode(pin,OUTPUT);

}

void onOff(bool accion){

if(accion){

digitalWrite(pin, HIGH);

}else{

digitalWrite(pin, LOW);

}

}

private:

byte pin;

};

//Objetos de clase creada (LED)

LED luzVerde(9);

LED luzNaranja(10);

void setup() {

pinMode(8,INPUT); // pin 8 como entrada

pinMode(7,OUTPUT); // pin 7 como salida

pinMode(6,OUTPUT); // pin 6 como salida

lcd.begin(16, 2);

Serial.begin(9600);

puerta.attach(13);

Serial.println("Ingrese el nivel MAX con el que trabajara el tanque de agua ");

Serial.println("Para eso, Ingrese la letra A seguido del valor numerico ");

Serial.println("Y tambien ingresar el nivel de agua deseado ");

Serial.println("Para eso, Ingrese la letra B seguido del valor numerico ");

}

void loop() {

while (Serial.available() > 0) {

n=Serial.read();

if(n=='A'){

max = Serial.parseInt();

}

if(n=='B'){

min= Serial.parseInt();

}

Serial.println("El nivel MAX del que dispone el tanque es: ");

Serial.println(max);

Serial.println("El nivel de agua que alcanzara el tanque es: ");

Serial.println(min);

Serial.println();

}

int distancia = 0.01723 \* readUltrasonicDistance(A1, A0); //0.01723 velocidad del sonido

//que al multiplicarlo por el tiempo que tardo la onda en ir y volver

//nos dara como resultado la distancia

if(distancia<=min){

distancia=min;

puerta.write(0);

luzVerde.onOff(1);

delay(2000);

luzNaranja.onOff(0);

delay(2000);

}

if(distancia>=max){

distancia=max;

puerta.write(180);

luzVerde.onOff(0);

delay(2000);

luzNaranja.onOff(1);

delay(2000);

}

if (digitalRead(8) == HIGH){ //evaluo si la entrada esta a nivel alto (Pulsador presionado)

digitalWrite(7, LOW);

analogWrite(6, LOW);

while(digitalRead(8) == HIGH){ //espero a que el pulsador se presione

}

ESTADO = digitalRead(7); //leo estado de led

digitalWrite(7, !ESTADO);//escribo valor contrario

while(digitalRead(8) == LOW){ //antirebote, esto hace es esperar a que el boton deje de estar presionado antes de que el loop vuelva a ejecutar el 1er while

}

}

else if(distancia>=min && distancia<=max)

{

digitalWrite(7, HIGH); //enciendo LED

delay(distancia \* 10);//demora proporcional a la distancia

digitalWrite(7, LOW); //apagoo LED

analogWrite(6, 150); //enciendo buzzer y mando pulso de 150

delay(100);

analogWrite(6, LOW);//apago buzzer

}

int porcentaje= map(distancia,min,max,max,0);

lcd.setCursor(0, 0);

lcd.print("Medidor de Agua");

lcd.setCursor(0, 1);

lcd.print(" Nivel=");

lcd.print(porcentaje);

lcd.print("% ");

}

long readUltrasonicDistance(int triggerPin, int echoPin)

{

pinMode(triggerPin, OUTPUT); // Clear the trigger

digitalWrite(triggerPin, LOW);

delayMicroseconds(2);

// Sets the trigger pin to HIGH state for 10 microseconds

digitalWrite(triggerPin, HIGH);

delayMicroseconds(10);

digitalWrite(triggerPin, LOW);

pinMode(echoPin, INPUT);

// Reads the echo pin, and returns the sound wave travel time in microseconds

return pulseIn(echoPin, HIGH);

}