

Universidad de San Carlos de Guatemala -USAC-
Centro Universitario de Occidente -CUNOC-



Tema
Reporte de construcción del circuito

Integrantes

Carlos Benjamin Pac Flores - 201931012
Fernando José Rodríguez Ramírez - 202030542
Dulce Pilar León Monzón - 201930139

Quetzaltenango, Mayo del 2023

Componentes	3
Código	4
Anexos	11
Circuito digital -base del circuito real	11
Imagenes de evidencia del progreso en la maqueta	12

Componentes

Cdad	Nombre	Propiedades	Aplicación
1	Arduino Mega	permitir la interconexión y programación de partes electrónicas para ser integradas en un circuito funcional y objetivo	En la manipulación y dirección de los elementos electrónicos empleados para llevar a cabo el experimento realizado por Galileo Galilei
1	Sensor ultrasónico - hc-sr04	Emisión de ondas ultrasónicas, para permitir la medición de medidas a partir de dichas ondas.	En la medición de las alturas a las que se encuentre la rampa, cuando el motor la haga elevar o descender, para obtener el angulo deseado.
1	Servo motor básico	Controlar el movimiento que es generado a partir de su eje, para permitir la impulsación o movimiento de ciertos objetos	Sostenedor e iniciador del movimiento del sujeto de prueba al inicio del experimento
1	Pantalla LCD	Entregar resultados de manera visual a partir de caracteres, generados a partir de acciones realizadas a partir de los elementos que intervinieron en el proceso.	Muestra parcial de los resultados, para dar a conocer lo que está sucediendo en un momento exacto durante el experimento.
8	Diodo LED	Emitir señales de luz para dar a entender que cierto proceso se está llevando a cabo y dio un resultado específico	En parejas para permitir obtener el resultado de un módulo de infrarrojos
1	Motor Bipolar de 4 hilos	Permitir el movimiento a partir de la rotación generada por la polaridad que se encuentra inmersa en este aparato.	En el movimiento de la rampa, para permitir que esta forme un ángulo específico al momento de alcanzar una altura en específico.
1	Keypad 4x4	Permitir la entrada de datos, de una manera más sencilla e interactiva para el usuario	Enviar una opción u otra para saber qué procedimiento desea el usuario realizar en ese momento.
4	Resistor 1K	Alterar o modificar el paso de la corriente y evitar en la mayoría de	Evitar el daño en componentes del circuito.

		los casos, la sobrecarga y por lo tanto daño en componentes electrónicos más complejos	
4	Resistor 10kΩ	Alterar o modificar el paso de la corriente y evitar en la mayoría de los casos, la sobrecarga y por lo tanto daño en componentes electrónicos más complejos	Modular la corriente recibida por los LED, para obtener resultados más precisos, a partir del ordenamiento PULL UP.
4	Resistor 330Ω	Alterar o modificar el paso de la corriente y evitar en la mayoría de los casos, la sobrecarga y por lo tanto daño en componentes electrónicos más complejos	Modular la corriente recibida por los LED, para obtener resultados más precisos, a partir del ordenamiento PULL DOWN.
1	H-bridge	Invertir el sentido de la corriente a partir de los transistores por los cuales se encuentra formado	Permiso del funcionamiento adecuado del motor bipolar de 4 hilos, al aplicar la secuencia correcta de impulsos a las bobinas o fases del motor.
4	Transistores npn 2222A	Amplificar o conmutar corrientes pequeñas (no mayores a 1 Watt)	Para la amplificación de las ondas emitidas por los elementos infrarrojos

Código

```
//libraries of modules
#include <Keypad.h>
#include <Stepper.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <Servo.h>

//Configuraciones servo
#define SERVO 26
Servo servoMotor;
```

```

//Params of dc motor
#define STEPS 96
#define SPEED 100

//Motor configuration
Stepper stepper(STEPS, 14, 15, 16, 17);

//HSR04 module params
const int Trigger = 2;
const int Echo = 3;
double time;
double distance = 0.00; //distance

//highs on centimeters
const double H1 = 10; //15.53
const double H2 = 20;
const double H3 = 38; //42.43
double errorPercentage = 2.0;

//Entradas de los sensores infrarrojos
#define REF1 22
#define REF2 23
#define REF3 24
#define REF4 25

//Configuracion de pantalla LCD
LiquidCrystal_I2C lcd(0x3F, 16, 2);

//Array de tiempos
unsigned long tiempos[] = { 0, 0, 0, 0 };
double distancias[] = {0.16,0.18,0.16};

//Matricial keyboar configs
byte rows[4] = { 13, 12, 11, 10 };
byte columns[4] = { 9, 8, 7, 6 };
char opciones[4][4] = {
    { '1', '2', '3', 'A' },
    { '4', '5', '6', 'B' },
    { '7', '8', '9', 'C' },
    { '*', '0', '#', 'D' }
};
Keypad teclado = Keypad(makeKeymap(opciones), rows, columns, 4,

```

```

4);
char boton;

void setup() {
    //Config of serial port
    Serial.begin(9600);
    //HSR04 module pins
    pinMode(Trigger, OUTPUT);
    pinMode(Echo, INPUT);
    digitalWrite(Trigger, LOW);
    //Motor speed
    stepper.setSpeed(SPEED);
    //Inicializacion de los pines de entrada leds infrarojos
    pinMode(REF1, INPUT);
    pinMode(REF2, INPUT);
    pinMode(REF3, INPUT);
    pinMode(REF4, INPUT);
    //Init del servo
    servoMotor.attach(SERVO);
    servoMotor.write(180); //180 sujetta la pelota
    // Inicializacion de la pantalla LCD
    lcd.init();
    lcd.backlight();
    opcionesLCD();
}

void loop() {
    //Calcular la distancia de la plataforma
    calcDistance(); //Serial.println(distance);
    //Obtener el valor del boton del teclado
    boton = teclado.getKey();
    //stepper.step(-12);
    switch (boton) {
        case '1':
            to15Degrees(((distance < H1) ? 1 : -1));
            boton = " ";
            break;
        case '2':
            to25Degrees(((distance < H2) ? 1 : -1));
            boton = " ";
            break;
        case '3':

```

```

        to45Degrees(((distance < H3) ? 1 : -1));
        boton = " ";
        break;
    case '4':
        servoMotor.write(150); //150 suelta la pelota
        iniciarMedicion();
        boton = " ";
        servoMotor.write(180); //180 sujetta la pelota
        break;
    }
}

void calcDistance() {
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10); //Enviamos un pulso de 10us
    digitalWrite(Trigger, LOW);

    time = pulseIn(Echo, HIGH); //obtenemos el ancho del pulso
    (tiempo)
    distance = time / 59; //Obtenemos distancia en cm
}

void to15Degrees(int dir) { //dir = 1/-1 => rotate direction
    toDegrees(dir, H1);
}

void to25Degrees(int dir) {
    toDegrees(dir, H2);
}

void to45Degrees(int dir) {
    toDegrees(dir, H3);
}

//It is ok, but you are going to change from
// sign, because if the current high is
// - more than the needed - izq (negative numbers-bajar)
// - less than needed - der (positive numbers-subir)
// It will be applied to the multiplier of the motor

void toDegrees(int dir, double high) { //dir = 1/-1
    calcDistance();
}

```

```

if (dir == -1) {
    //Baja la plataforma
    Serial.println("Baja plataforma");
    Serial.println(high);
    while (!isWithinError(high, distance, errorPercentage)) {
        stepper.step(-12);
        calcDistance();
        Serial.println(distance);
    }
} else {
    //Sube la platarforma
    Serial.println("Sube plataforma");
    Serial.println(high);
    while (!isWithinError(high, distance, errorPercentage)) {
        stepper.step(12);
        calcDistance();
        Serial.println(distance);
    }
}
}

bool isWithinError(double val1, double val2, double
errorPercentage) {
    double diff = fabs(val1 - val2);
    double avg = (fabs(val1) + fabs(val2)) / 2.0;
    double error = avg * (errorPercentage / 100.0);
    return diff <= error;
}

void iniciarMedicion() {
    lcd.clear();
    lcd.print("Inicio Medicion");
    int s1 = digitalRead(REF1);
    int s2 = digitalRead(REF2);
    int s3 = digitalRead(REF3);
    int s4 = digitalRead(REF4);
    bool bandera = true;
    do {
        s1 = digitalRead(REF1);
        s2 = digitalRead(REF2);
        s3 = digitalRead(REF3);
        s4 = digitalRead(REF4);

```

```

//Si leemos 1 en el sensor 1 gurdamos el tiempo
if (s1 == 1) {
    lcd.clear();
    lcd.print("Sensor 1");
    lcd.setCursor(0, 1);
    lcd.print("Activado");
    tiempos[0] = millis();
}

//Si leemos 1 en el sensor 2 gurdamos el tiempo
if (s2 == 1) {
    lcd.clear();
    lcd.print("Sensor 2");
    lcd.setCursor(0, 1);
    lcd.print("Activado");
    tiempos[1] = millis();
}

//Si leemos 1 en el sensor 3 gurdamos el tiempo
if (s3 == 1) {
    lcd.clear();
    lcd.print("Sensor 3");
    lcd.setCursor(0, 1);
    lcd.print("Activado");
    tiempos[2] = millis();
}

//Si leemos 1 en el sensor 4 gurdamos el tiempo
if (s4 == 1) {
    lcd.clear();
    lcd.print("Sensor 4");
    lcd.setCursor(0, 1);
    lcd.print("Activado");
    tiempos[3] = millis();
    //Detenemos el ciclo de ejecucion
    bandera = false;
}

} while (bandera);
lcd.clear();
lcd.print("Fin Medicion");
String resultado =
String(tiempos[0])+", "+String(tiempos[1])+", "+String(tiempos[2])+
", "+String(tiempos[3]);
lcd.setCursor(0, 1);
lcd.print(resultado);

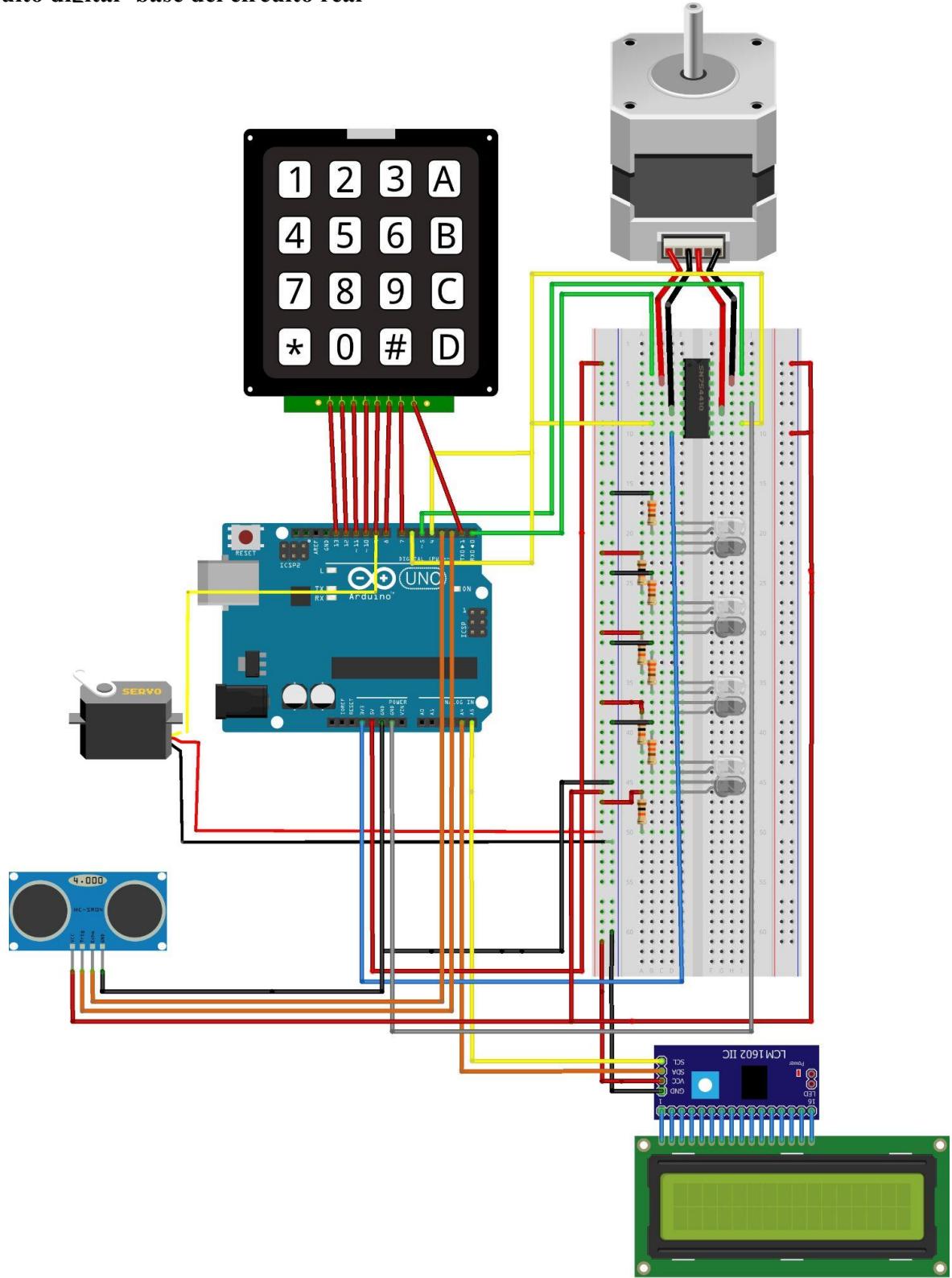
```

```
Serial.println(resultado);
delay(4000);
lcd.clear();
opcionesLCD();
}

void opcionesLCD(){
    lcd.setCursor(0, 0);
    lcd.print("Op: ");
    lcd.print("1:15' ");
    lcd.print("2:20' ");
    lcd.setCursor(0, 1);
    lcd.print("3:45' ");
    lcd.print("4:init");
}
```

Anexos

Circuito digital -base del circuito real



Imagenes de evidencia del progreso en la maqueta

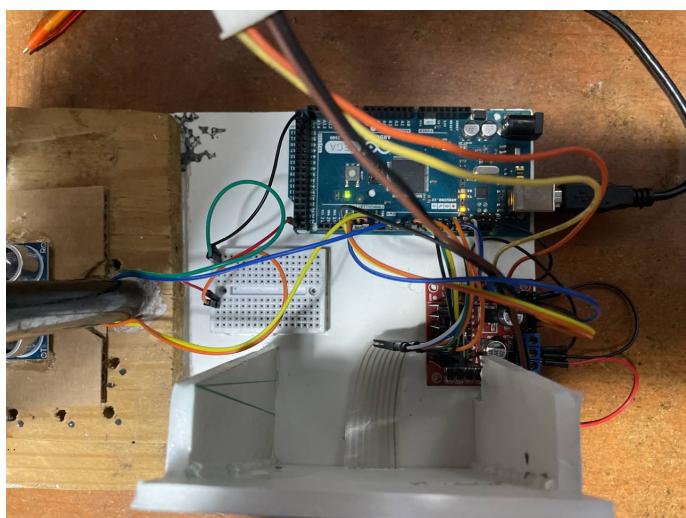


Imagen 2: Arduino mega y concesiones iniciales

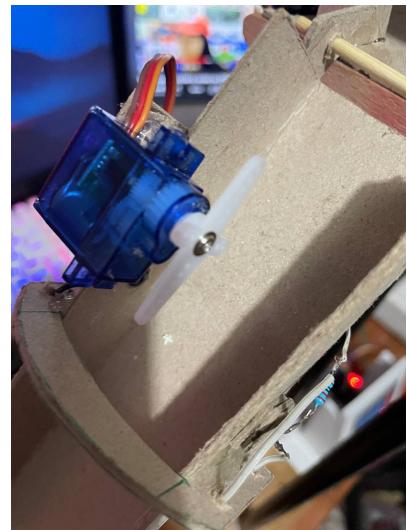


Imagen 3: Servo e infrarrojo

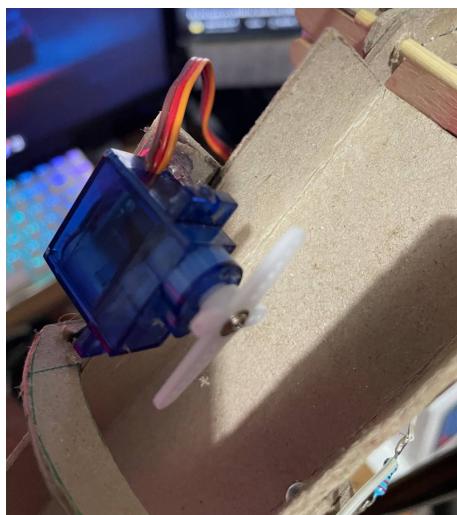


Imagen 4: Imagen agrandada del servo

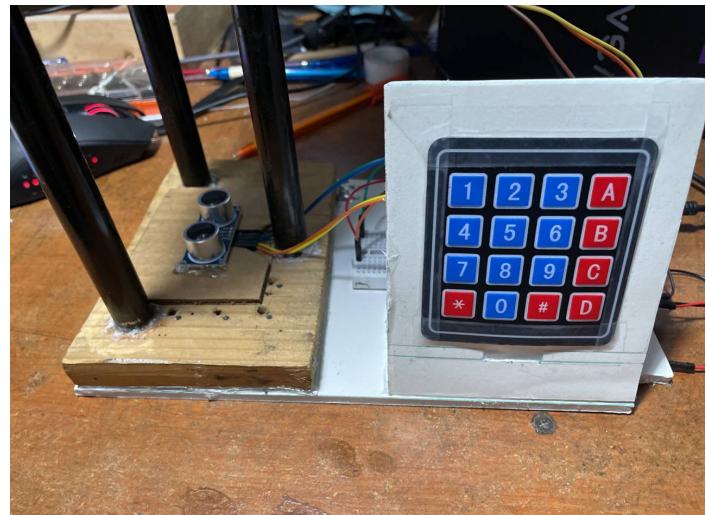


Imagen 5: Keypad 4x4



Imagen 6: Motor impresora

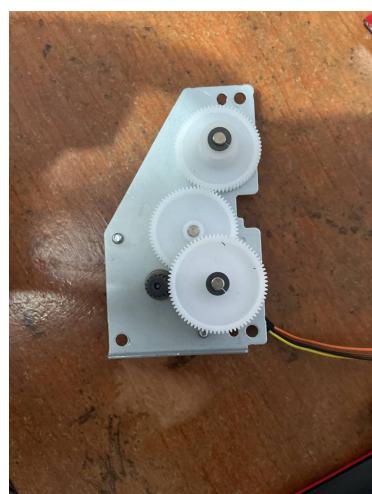


Imagen 7: Parte delantera del motor



Imagen 8: Receptor infrarojo

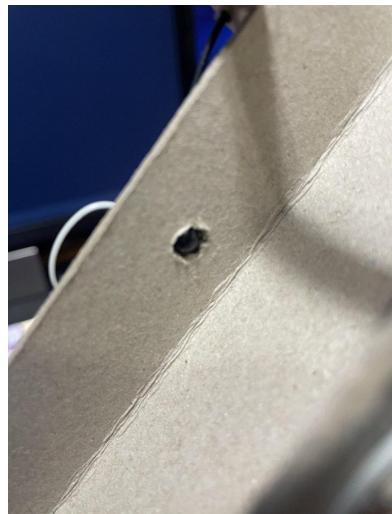


Imagen 9: infrarojo emisor



Imagen 10: Sensor HSR04



Imagen 11: Vista frontal de rampa



Imagen 12: Vista lateral de la rampa