PROYECTO PUENTE LEVADIZO MANDO INALÁMBRICO.

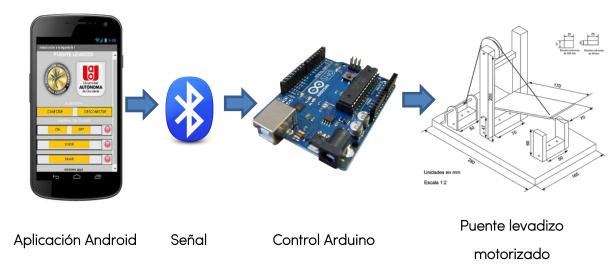
Este proyecto se desarrollado para la clase de introducción a la ingeniería 1 por Gustavo Arango de mirxtrem.apps y es de código abierto para su libre uso y se debe

Resumen.

Se trata de un puente hecho en madera donde se requiere que suba y baje sus plataformas por medio de un motor dc y poleas; este se activará a través de un celular por medio de una aplicación y usando comunicación bluetooth al control del motor.

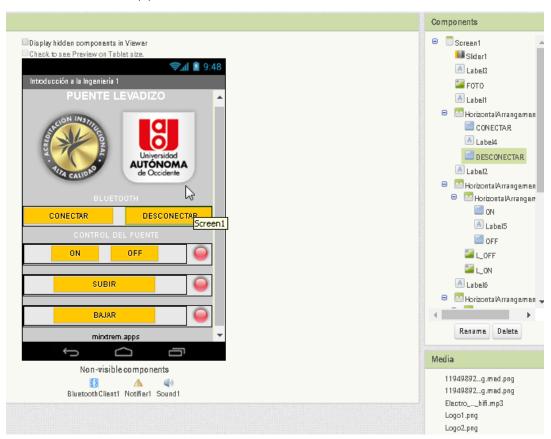
La plataformas tendrán un micro-interruptores finales de carrera para detener el motor y la plataforma cuando lleguen al límite superior e inferior. También se tiene dos led que serán el semáforo.

Estructura de comunicación.



Aplicación móvil android.

Desarrollada en App Inventor 2.



Código Applnvetor2.

Botón CONECTAR: para establecer conexión con Arduino y notifica la conexión.

```
ARDUINO. DEBE ESTAR
when CONECTAR . Click
                                                           PREVIAMENTE ENLAZADA
                                                           DESDE LA CONFIGURACIÓN
do call Sound1 - Play
                                                           DEL CELULAR.
    evaluate but ignore result 📗 call | BluetoothClient1 🔻 .Connec
                                                                ? " (30:14:12:25:04:50) "
    set DESCONECTAR . Enabled . to true .
            ■ BluetoothClient1 →
                                 IsConnected
    then call Notifier1 . ShowMessageDialog
                                                  Se ha conectado al Arduino UNO
                                                  BLUETOOTH "
                                    buttonText
                                                 " OK! "
```

Botón ON: enciende los leds indicadores y habilita los botones SUBIR y BAJAR. Al presionar el botón genera un sonido.

```
when ON r .Click

do call Sound1 r .Play

set L_ON r . Visible r to true r

set L_OFF r . Visible r to false r

set SUBIR r . Enabled r to true r

set BAJAR r . Enabled r to true r
```

Botón SUBIR: enciende los leds indicadores y envia el text 'U' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.

```
when SUBIR Click

do set L_ON . Visible to true viset L_OFF . Visible to false viset L_UP_ON . Visible to true viset L_UP_OFF . Visible to false viset L_DOWN_ON . Visible to false viset L_DOWN_OFF . Visible to true vical BluetoothClient1 . SendText

call Sound1 * Play
```

Botón BAJAR: enciende los leds indicadores y envia el text 'D' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.

```
when BAJAR v Click

do set L_ON v . Visible v to true v

set L_OFF v . Visible v to false v

set L_UP_ON v . Visible v to false v

set L_UP_OFF v . Visible v to true v

set L_DOWN_ON v . Visible v to false v

call BluetoothClient1 v .SendText

text v D v

call Sound1 v .Play
```

Botón OFF: enciende los leds indicadores y envia el text 'S' al Arduino a través de bluetooth. Al presionar el botón genera un sonido.

```
when OFF click

do call Sound1 r Play

set L_ON r . Visible r to false r

set L_UP_ON r . Visible r to false r

set L_UP_ON r . Visible r to false r

set L_UP_OFF r . Visible r to false r

set L_UP_OFF r . Visible r to frue r

set L_DOWN_OFF r . Visible r to frue r

call BluetoothClient1 r .SendText

text rest
```

Botón DESCONECTAR: desconecta la conexión con Arduino y notifica la desconexión. Al presionar el botón genera un sonido.

```
when DESCONECTAR Click
do set SUBIR C. Enabled to false c
set BAJAR C. Enabled to false c
call Sound1 c. Play
call BluetoothClient1 c. Disconnect
set DESCONECTAR C. Enabled to false c
call Notifier1 c. ShowMessageDialog
message control co
```

Sketch Arduino.

// Primero creamos las variables del programa

```
int microSup = 6;
int microlnf = 7;
int subir = 13;
int bajar = 12;
int btonUp = 8;
int btonDown = 9;
int ledVerde = 10;
int ledRojo = 11;
int estadoMicroU = 0;
int estadoMicroD = 1;
int estadoUp = 0;
int estadoDown = 0;
void setup() {
 // Declaramos cuales son las salidas y las entradas y sus estados
pinMode(btonUp, INPUT);
pinMode(btonDown, INPUT);
pinMode(microSup, INPUT);
```

```
pinMode(microInf, INPUT);
pinMode(subir, OUTPUT);
pinMode(bajar, OUTPUT);
pinMode(ledVerde, OUTPUT);
pinMode(ledRojo, OUTPUT);
digitalWrite(subir, LOW);
digitalWrite(bajar, LOW);
digitalWrite(ledVerde, HIGH);
digitalWrite(ledRojo, LOW);
}
void loop() {
 // Acción A: La rampa sube a la señal de subir
estadoUp=digitalRead(btonUp);
 if(estadoUp == HIGH){
 digitalWrite(ledVerde, LOW);
 digitalWrite(ledRojo, HIGH);
 delay(2000);
 digitalWrite(subir, HIGH);
 }
 // Acción B: El motor se detiene para que la rampa deje de subir
estadoMicroU=digitalRead(microSup);
 if(estadoMicroU == HIGH) {
  digitalWrite(subir, LOW);
```

```
// Acción C: La rampa baja a la señal de bajar
 }
estadoDown=digitalRead(btonDown);
 if(estadoDown == HIGH){
 digitalWrite(bajar, HIGH);
 // Acción D: La rampa baja y el semaforo cambia
estadoMicroD=digitalRead(microInf);
 if(estadoMicroD == HIGH) {
  digitalWrite(bajar, LOW);
  delay(2000);
  digitalWrite(ledVerde, HIGH);
  digitalWrite(ledRojo, LOW);
 }
```

Esquema de conexión.

