

Programación en Arduino:

```
proyecto1
1 #include <Servo.h>
2
3 String text;
4 int velocidad, vell, vel2;
5 int Motor1 = 8;
6 int Motor2 = 9;
7 int Motor3 = 10;
8 int Motor4 = 11;
9
10 int R1 = 2;
11 int R2 = 3;
12 int R3 = 12;
13 int R4 = 13;
14
15 int R = 0;
16 int K = 0;
17
18 int valor_salida_pwm;
19
20 int LEDIZQ = 7;
21 int LEDDER = 6;
22 int LEDREV = 5;
23
24
25 /*ULTRASONICO*/
26 int ultrasonic = 4;
27 int distancia;
28
29
30 #define S1 3 //el pin del boton velocidad
31
32
33 void setup() {
34
35     Serial.begin(9600);
36     pinMode (Motor1, OUTPUT);
37     pinMode (Motor2, OUTPUT);
38     pinMode (Motor3, OUTPUT);
39     pinMode (Motor4, OUTPUT);
40
41     pinMode (LEDIZQ, OUTPUT);
```

En esta parte, tenemos el código Arduino. Se incluyo 1 librería para el funcionamiento correcto del código, se definieron los pines conectados del Arduino y un par de variables que se usaran más adelante en la lógica de la programación. Se inicializaron los pines y el puerto serial.

proyecto1

```
58
59 }
60
61 void loop() {
62   ultrason(); //*****
63   while (Serial.available()) {
64     delay(10);
65     char c = Serial.read();
66     text += c;
67   }
68   if(K == 1){
69     analogWrite(LEDIZQ, 250);
70     analogWrite(LEDDER, 0);
71     analogWrite(LEDREV, 0);
72   }
73   }else if(K ==2){
74     analogWrite(LEDIZQ, 0);
75     analogWrite(LEDDER, 250);
76     analogWrite(LEDREV, 0);
77   }else if (K == 3){
78     analogWrite(LEDIZQ, 0);
79     analogWrite(LEDDER, 0);
80     analogWrite(LEDREV, 0);
81   }else if (K == 4){
82     analogWrite(LEDIZQ, 0);
83     analogWrite(LEDDER, 0);
84     analogWrite(LEDREV, 250);
85   }
86
87
88
89   if (text.length() > 0) {
90
91     //Serial.print(text);
92     /*Esto es para saber si esta yendo para adelante o para atras*/
93     if (text == "atr") {
94       R = 1;
95     } else if (text == "ade")
96     {
97       R = 0;
98     }
99   }
```

En el método loop() tenemos la lógica del funcionamiento para saber si el carrito va hacia adelante o hacia atrás, también tenemos una llamada al método que activa el sensor ultrasónico.

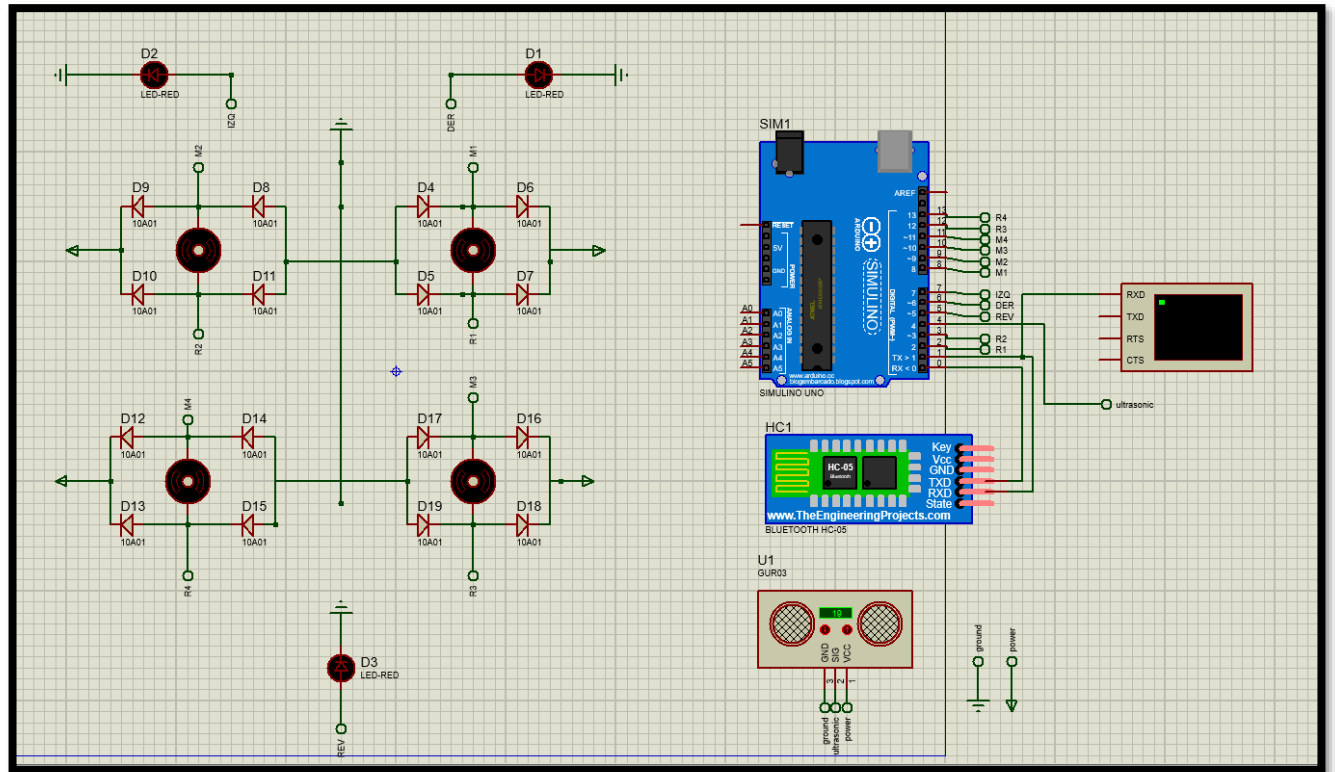
```

proyecto1
195 }
196 }
197
198
199
200
201
202 //***** PARA EL ULTRASONIC
203
204 void ultrason(){
205     distancia = 0.01723 * readUltrasonicDistance(ultrasonic, ultrasonic);
206     if(distancia<19){
207         analogWrite(Motor1, LOW);
208         analogWrite(Motor2, LOW);
209         analogWrite(Motor3, LOW);
210         analogWrite(Motor4, LOW);
211         K = 4;
212         Serial.println("FIN");
213         Serial.println(distancia);
214     }else if(distancia>20){
215         analogWrite(Motor1, vell);
216         analogWrite(Motor2, vell);
217         analogWrite(Motor3, vell);
218         analogWrite(Motor4, vell);
219     }
220 }
221 }
222
223 long readUltrasonicDistance(int triggerPin, int echoPin)
224 {
225     pinMode(triggerPin, OUTPUT); // Clear the trigger
226     digitalWrite(triggerPin, LOW);
227     delayMicroseconds(2);
228     // Sets the trigger pin to HIGH state for 10 microseconds
229     digitalWrite(triggerPin, HIGH);
230     delayMicroseconds(10);
231     digitalWrite(triggerPin, LOW);
232     pinMode(echoPin, INPUT);
233     // Reads the echo pin, and returns the sound wave travel time in microseconds
234     return pulseIn(echoPin, HIGH);
235 }

```

Los métodos ultrason() y readUltrasonicDistance() son los encargados del funcionamiento correcto del sensor que detecta objetos a distancia y hace que los motores del carrito frenen si se encuentra en una distancia $x < 20$.

SIMULACION DEL CIRCUITO



DISEÑO DE LA APP

