



Introducción a la programación de microcontroladores con Arduino

Trabajo Final - Auto controlado por bluetooth

Integrantes:

Romero Muñoz, Fernando Mario

Insaurralde, Juan Cruz

Enriquez, Martin Hernan

Materiales utilizados

Componente	Cantidad	Precio unit	Precio total
Tecla encendido	1	\$1.900,00	\$1.900,00
Cable 0,25 (metro)	2	\$280,00	\$560,00
Plancha madera	–	–	\$0,00
Tira pin espadin recto Hembra x40	1		\$1500,00
Tira pin espadin recto Macho x40	1		\$1400,00
Precintos	–	–	\$0,00
Batería de 9V	1		\$4.200,00
Conector BAT 9V	1		\$900,00
Motor con rueda	4	\$4.170,00	\$16.680,00
Puente H L298	2	\$4.190,00	\$8.380,00
Módulo bluetooth HC05	1	\$5.190,00	\$5.190,00
Arduino UNO	1		\$0,00
Soldador y estaño	–	–	\$0,00
Precintos	4		\$0,00
TOTAL			\$40.710,00

Los materiales necesarios para el proyecto fueron adquiridos de varias fuentes:

- **Mercado Libre:** Moto rueda, módulo Bluetooth, y puente H.
- **Casa de electrónica Radián:** Pines macho y hembra.
- **Casa Electrónica RC:** Batería de 9V y conector para batería.
- **Casa Electrónica 80:** Tecla de encendido.
- **Materiales ya poseídos:** Chapadur, precintos, Arduino Uno, soldador, y pegamento.

Código fuente del programa del microcontrolador

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10, 11); // RX, TX
void setup() {
    // Open serial communications and wait for port to open:
    Serial.begin(9600);
    mySerial.begin(38400);
    while (!Serial) {
        ; // wait for serial port to connect. Needed for native USB port only
    }
    Serial.println("Sending AT...");
    mySerial.write("AT");
    if (mySerial.available() > 0) {
        Serial.write(mySerial.read());
    }
    // set the data rate for the SoftwareSerial port
    Serial.println("loop begins");
    LED_setup();
}
void loop() { // run over and over
    if (mySerial.available()) {
        rcd = mySerial.read();
        Serial.write(rcd);
    }
    unsigned long time_now = 0;
    if (rcd == 'O')
    {
        R_LED(1);
        L_LED(1);
        F_LED(1);
        B_LED(1);
    }
    if (rcd == 'o')
    {
        R_LED(0);
        L_LED(0);
        F_LED(0);
        B_LED(0);
    }
    if (rcd == 'X')
    { L_LED(1);
    }
    if (rcd == 'x')
    {
        L_LED(0);
    }
    if (rcd == 'P')
    {
        R_LED(1);
    }
    if (rcd == 'p')
    { R_LED(0);
    }
    if (rcd == 'W')
    { F_LED(1);
    }
}
```

```

if (rcd == 'w')
{ F_LED(0);
}
if (rcd == 'U')
{ B_LED(1);
}
if (rcd == 'u')
{ B_LED(0);
}
if (rcd == '0')
{ spd = 80;
}
if (rcd == '1')
{ spd = 100;
}
if (rcd == '3')
{ spd = 120;
}
if (rcd == '5')
{ spd = 180;
}
if (rcd == 'q')
{ spd = 255;
}
if (rcd == 'F')
{ Forward();
  time_now = millis();
  while (millis() < time_now + 100) {}
}
else if (rcd == 'B')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  Backward();
}
else if (rcd == 'I')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  FdRight();
}
else if (rcd == 'J')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  BkRight();
}
else if (rcd == 'R')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  SharpRight();
}
else if (rcd == 'L')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  SharpLeft();
}
else if (rcd == 'G')
{ time_now = millis();
  while (millis() < time_now + 100) {}
  FdLeft();
}
else if (rcd == 'H')

```

```

{ time_now = millis();
  while (millis() < time_now + 100) {}
  BkLeft();
}
else
  Stop();
}
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif
#define F_LED_PIN      12 //Front LED
#define F_LED_PIXEL    6
#define B_LED_PIN      13 //Back LED
#define B_LED_PIXEL    6
#define R_LED_PIN      7 //Right LED
#define R_LED_PIXEL    1
#define L_LED_PIN      8 //Left LED
#define L_LED_PIXEL    1
Adafruit_NeoPixel F_LED_STRIP(F_LED_PIXEL, F_LED_PIN, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel B_LED_STRIP(B_LED_PIXEL, B_LED_PIN, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel R_LED_STRIP(R_LED_PIXEL, R_LED_PIN, NEO_GRB + NEO_KHZ800);
Adafruit_NeoPixel L_LED_STRIP(L_LED_PIXEL, L_LED_PIN, NEO_GRB + NEO_KHZ800);
void LED_setup()
{
  #if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
    clock_prescale_set(clock_div_1);
  #endif
  F_LED_STRIP.begin();
  B_LED_STRIP.begin();
  R_LED_STRIP.begin();
  L_LED_STRIP.begin();
  R_LED_STRIP.clear();
  L_LED_STRIP.clear();
  F_LED_STRIP.clear();
  B_LED_STRIP.clear();
}
void R_LED(int a)
{
  if(a==1)
  {
    R_LED_STRIP.clear();
    R_LED_STRIP.setPixelColor(0, R_LED_STRIP.Color(0, 255, 255));
    R_LED_STRIP.show(); // Send the updated pixel colors to the hardware.
  }
  else {
    R_LED_STRIP.clear();
    R_LED_STRIP.show();
  }
}
void L_LED(int a)
{
  if(a==1)
  {
    L_LED_STRIP.clear();
    L_LED_STRIP.setPixelColor(0, L_LED_STRIP.Color(0, 255, 255));
    L_LED_STRIP.show(); // Send the updated pixel colors to the hardware.
  }
  else
  {

```

```

        L_LED_STRIP.clear();
        L_LED_STRIP.show();
    }
}
void F_LED(int a)
{
    if(a==1)
    {
        F_LED_STRIP.clear();
        F_LED_STRIP.setPixelColor(0, F_LED_STRIP.Color(255, 255, 255));
        F_LED_STRIP.setPixelColor(1, F_LED_STRIP.Color(0, 0, 50));
        F_LED_STRIP.setPixelColor(2, F_LED_STRIP.Color(0, 0, 50));
        F_LED_STRIP.setPixelColor(3, F_LED_STRIP.Color(0, 0, 50));
        F_LED_STRIP.setPixelColor(4, F_LED_STRIP.Color(0, 0, 50));
        F_LED_STRIP.setPixelColor(5, F_LED_STRIP.Color(255, 255, 255));
        F_LED_STRIP.show();    // Send the updated pixel colors to the hardware.
    }
    else if(a==0)
    {
        F_LED_STRIP.clear();
        F_LED_STRIP.show();
    }
}
void B_LED(int a)
{
    if(a==1)
    {
        B_LED_STRIP.clear();
        B_LED_STRIP.setPixelColor(0, B_LED_STRIP.Color(255, 0, 0));
        B_LED_STRIP.setPixelColor(1, B_LED_STRIP.Color(0, 50, 0));
        B_LED_STRIP.setPixelColor(2, B_LED_STRIP.Color(0, 50, 0));
        B_LED_STRIP.setPixelColor(3, B_LED_STRIP.Color(0, 50, 0));
        B_LED_STRIP.setPixelColor(4, B_LED_STRIP.Color(0, 50, 0));
        B_LED_STRIP.setPixelColor(5, B_LED_STRIP.Color(255, 0, 0));
        B_LED_STRIP.show();    // Send the updated pixel colors to the hardware.
    }
    else if(a==0)
    {
        B_LED_STRIP.clear();
        B_LED_STRIP.show();
    }
}
void Forward()
{
    analogWrite(5, spd);
    digitalWrite(6, 0);
    analogWrite(9, spd);
    digitalWrite(3, 0);
    // delay(50);
}
void Backward()
{
    digitalWrite(5, 0);
    analogWrite(6, spd);
    digitalWrite(9, 0);
    analogWrite(3, spd);
}
void Stop()
{
    analogWrite(5, 0);

```

```

    digitalWrite(6, 0);
    analogWrite(9, 0);
    digitalWrite(3, 0);
}
void FdRight()
{
    analogWrite(5, spd);
    digitalWrite(6, 0);
    digitalWrite(9, 0);
    analogWrite(3, 0);
}
void BkRight()
{
    analogWrite(5, 0);
    digitalWrite(6, 0);
    digitalWrite(9, 0);
    analogWrite(3, spd);
}
void SharpRight()
{
    analogWrite(5, spd);
    digitalWrite(6, 0);
    digitalWrite(9, 0);
    analogWrite(3, spd);
}
void SharpLeft()
{
    digitalWrite(5, 0);
    analogWrite(6, spd);
    analogWrite(9, spd);
    digitalWrite(3, 0);
}
void BkLeft()
{
    digitalWrite(5, 0);
    analogWrite(6, spd);
    analogWrite(9, 0);
    digitalWrite(3, 0);
}
void FdLeft()
{
    digitalWrite(5, 0);
    analogWrite(6, 0);
    analogWrite(9, spd);
    digitalWrite(3, 0);
}
}

```

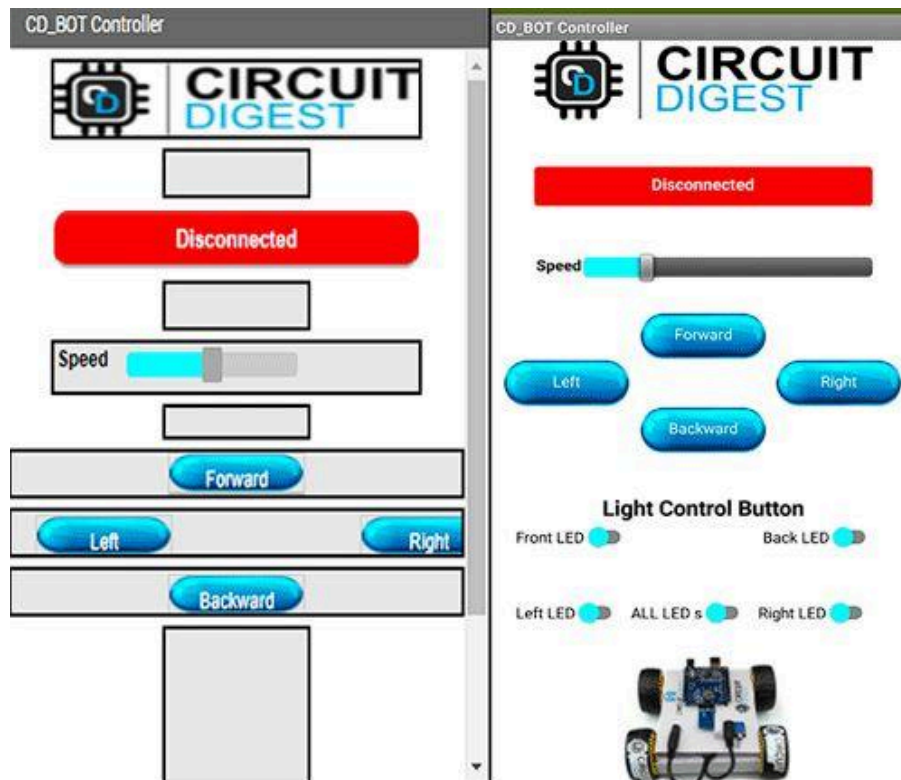
Software necesario para el proceso y sus versiones

- MIT app inventor 2.0
- Arduino IDE 2.0.0

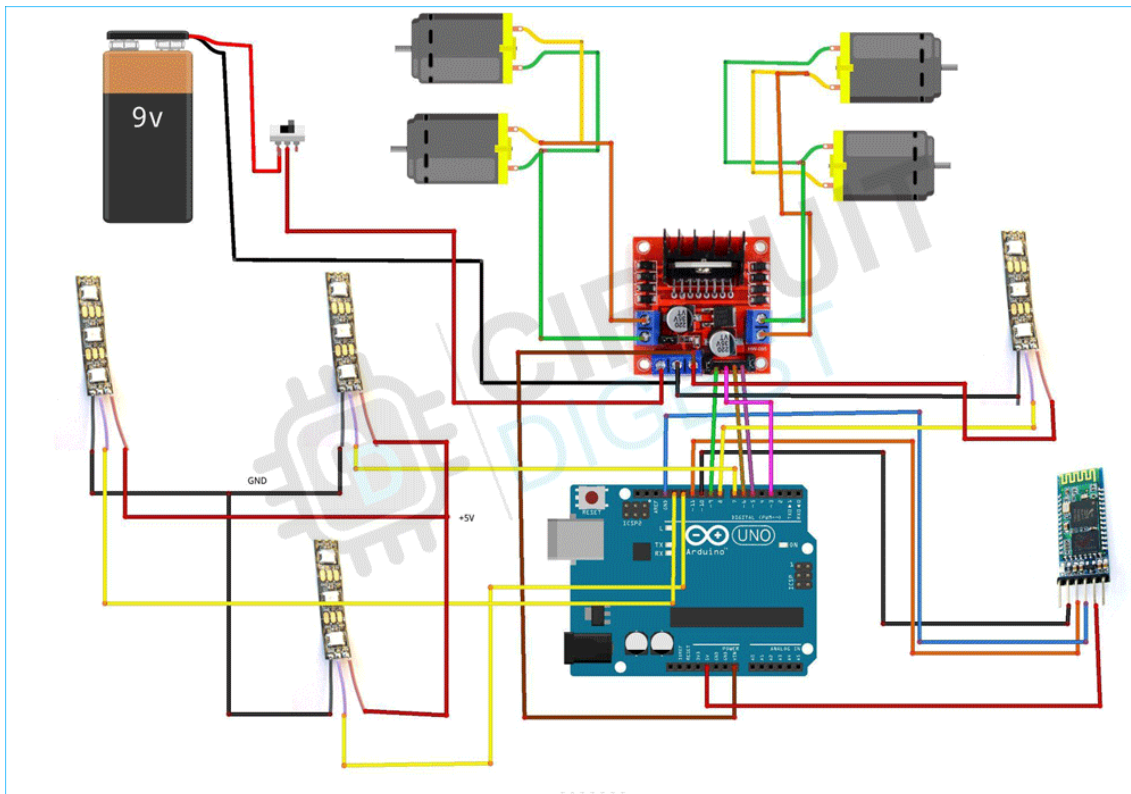
Código fuente del programa



GUI



Esquemático



Problemas que surgieron y cómo se resolvieron

Durante el desarrollo del proyecto, surgieron varios problemas que fueron resueltos de la siguiente manera:

1. **Problemas con la app móvil:** La aplicación móvil que controlaba el automóvil por Bluetooth no funcionaba debido a un problema con uno de los bloques de código. Al ser de código abierto, se pudo modificar el bloque por un evento de clic, solventando el error. Además, la aplicación no solicitaba los permisos necesarios en el celular, impidiendo la conexión. Este inconveniente se resolvió configurando manualmente los permisos desde el sistema Android.
2. **Problemas de ensamblaje:** Durante el ensamblaje del automóvil, las piezas no encajaban de la misma manera que se mostraba en el video debido a las diferencias en los materiales disponibles. Para resolver esto, se serruchó el chapadur para crear un espacio donde colocar el Arduino de cabeza y otro espacio para el botón de encendido, lo que llevó a cambiar la disposición del ensamblado.

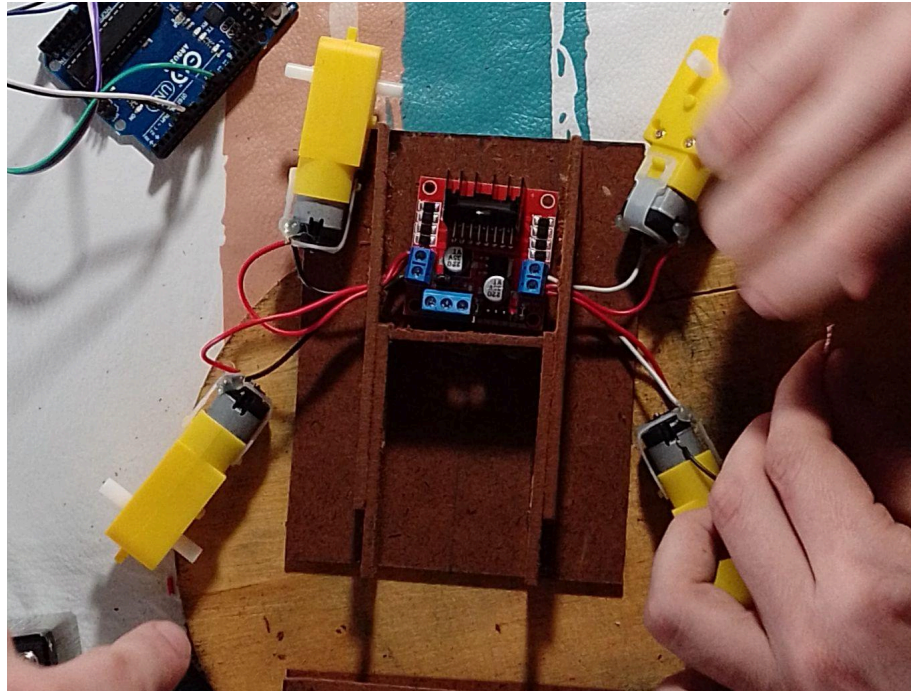
Paso a paso del armado

Para ensamblar el auto, se usaron los siguientes recortes de chapadur:

Corte de las piezas:

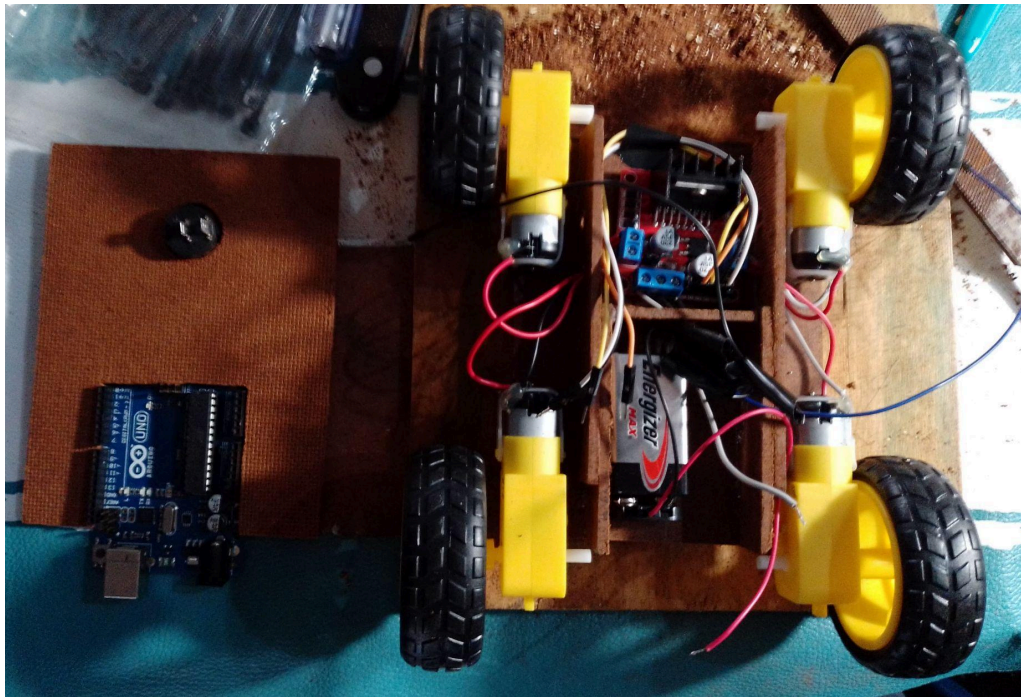
- Piezas rectangulares de 10.5 x 13.5 cm (2 piezas).
- Tiras de 7.5 x 2.3 cm (2 piezas).
- Tiras de 11.5 x 2.3 cm (2 piezas).
- Pieza de 4.3 x 2.3 cm (1 pieza).
- Piezas de 10.5 x 3.5 cm (2 piezas).
- Restos (Durante la dificultad que nos cruzamos usamos restos de recortes)

A partir de una de las piezas rectangulares más grandes se la utilizó como base. Sobre ella en sus extremos se pegaron los motores, en el centro se pegaron paradas las tiras y la pieza pequeña en medio (formando una especie de H con los motores por fuera y dos espacios rectangulares por dentro). Se debió recortar los lados de las tiras para poder pasar los cables de los motores a uno de los espacios rectangulares donde se colocó el módulo del controlador del motor. A su vez también se hizo un agujero en la pieza del medio para poder pasar los cables de la batería la cual se colocó en el espacio rectangular restante.

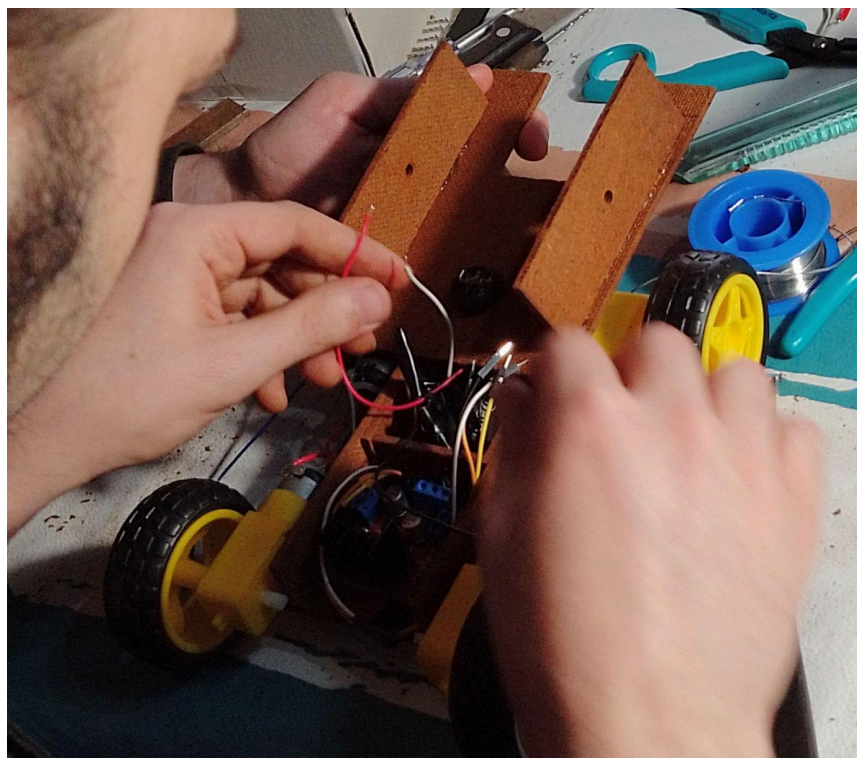


Para colocar el Arduino se tuvo que improvisar un recorte en la tabla restante grande para que este entre dentro volteado de cabeza para hacer las conexiones al interior del coche. También se realizó a la misma pieza de chapadur un corte circular para el botón de encendido que da al espacio de la batería.





Se utilizaron las tiras restantes para dar la altura necesaria para que la pieza recortada pudiese officiar de techo del auto.



Finalmente se conectaron los cables al Arduino, se conectó el módulo bluetooth el cual se colocó dentro del compartimento junto a la batería. Una vez ya realizadas las conexiones se procedió a poner la tabla con el Arduino en posición como el techo del coche. Para dar firmeza a las diferentes partes (techo y base) se precintó el techo a la base.

