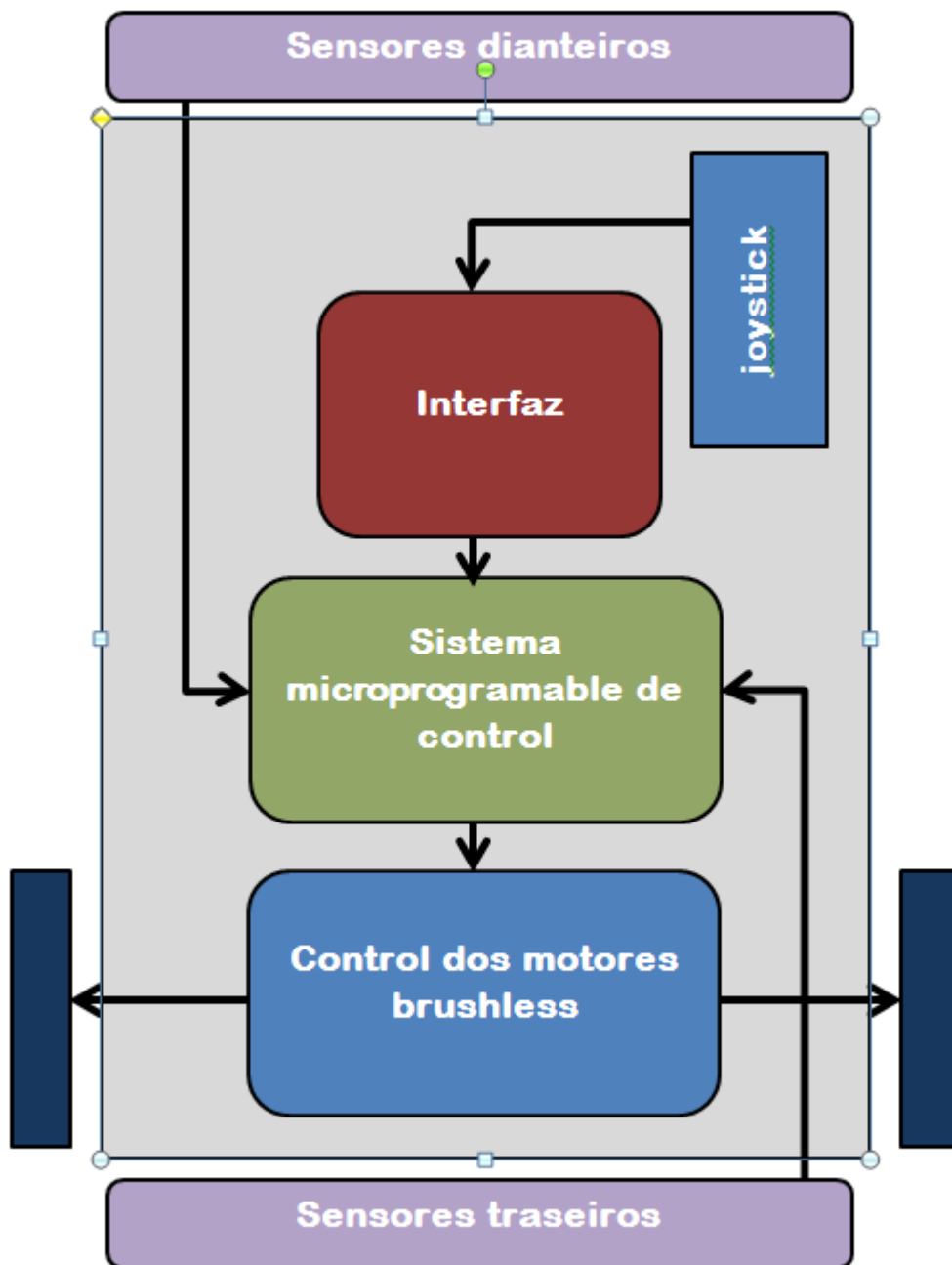


Diagrama de bloques

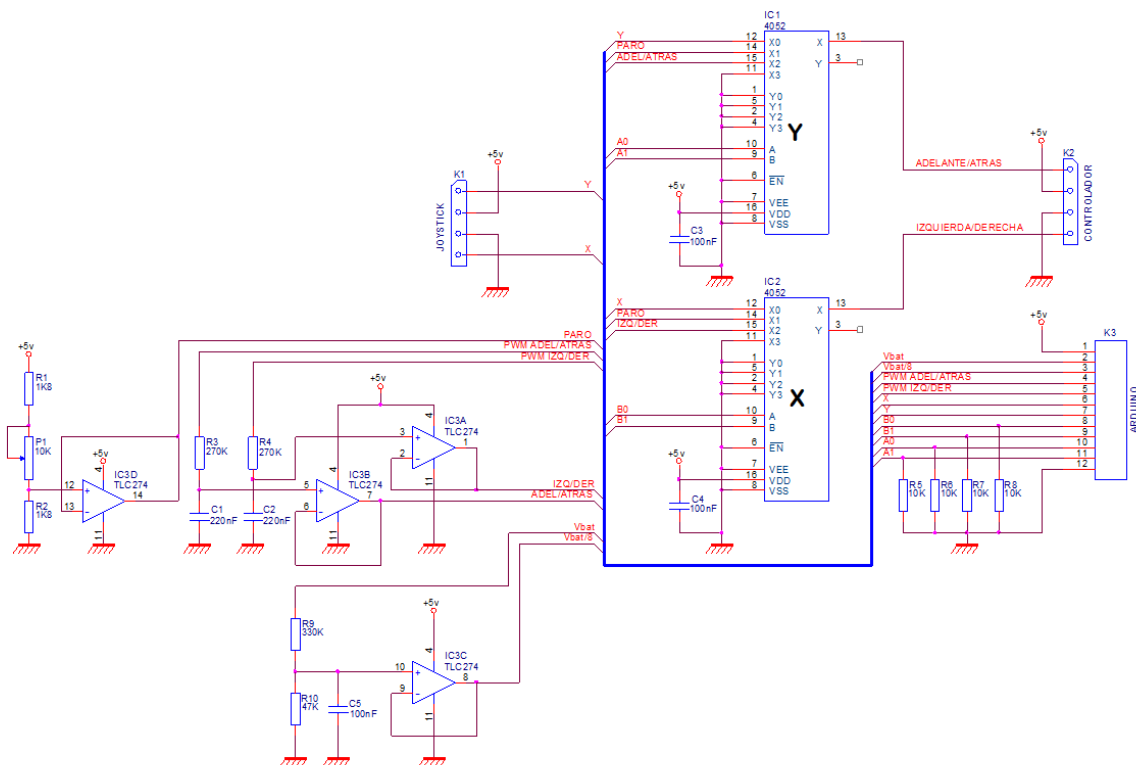


Descrición

Para a tracción do carrito se emprega un par de motores brushless de 180W cada un, situados na parte posterior para maior maniobrabilidade. O guiado contrólase mediante un joystick situado na estrutura do carrito, que xera as ordes de avance-retroceso e dereita-esquerda, as cales, enviadas á etapa de control fan que esta produza os sinais requiridos para que a etapa de potencia faga mover cada un dos motores de maneira adecuada. Sen embargo, estes sinais procedentes do joystick han de ser previamente avaliados e procesados para seren integrados co resto de controis de que dispón o sistema. Para esta integración, dispón dunha placa de interfaz e outra de control, esta última microprogramable.

A circu to de control implem ntase mediante unha placa Arduino Mega, que recibe sinais dos distintos sensores, en funci n dos cales toma as decisi ns oportunas e xera os sinais anal xicos e dixitais adecuados, que env a   placa de interfaz. Esta  ltima adapta e encami a adecuadamente eses sinais producidos polo circu to de control.

Concretamente, o carri o disp n dun sistema anticoli n, que detecta obxectos cun sistema de s nar, mediante uns sensores de ultras ns situados na parte frontal e traseira do carri o. Eses sensores informan   placa de control da distancia   que se atopa o obst culo, para evitaren que o carri o siga avanzando, ou para que o faga m is lentamente. Tam n conta cun sistema de detecc n de escal ns, que impide que se poda precipitar por un eventual escal n.



Esquema do interfaz

PINOUT DIAGRAM



Esquema de conexiónado da placa Arduino Mega

Código do programa do circuito de control sobre Arduino Mega

```

//Variables de control del MUX
const int a0 = 22; //Bit 1 eje Y MUX A
const int a1 = 23; //Bit 2 eje Y MUX A
const int b0 = 24; //Bit 1 eje X MUX B
const int b1 = 25; //Bit 2 eje X MUX B

//Variables de lectura del joystick
const float joyY = A1; // Lectura del voltaje del joystick eje Y
const float joyX = A2; // Lectura del voltaje del joystick eje X
int lecturaY = 0;
int lecturaX = 0;

//Variables de deteccion de obstaculos (parachoques)
const int obstaculoD = 46;
const int obstaculoT = 47;
const int obstaculoR = 49;
const int obstaculoL = 48;
const int buzzer = 13;

int pwm_X = 6;
int pwm_Y = 7;

//Variables de deteccion de obstaculos (ultrasonidos)
long distancia1;
long distancia2;
long distancia3;
long distancia4;

long tiempo1;
long tiempo2;
long tiempo3;
long tiempo4;

//Emisores centrales y laterales de la parte frontal del carrito. Emisor = Trigger
const long emisor1 = 30;
const long emisor2 = 32;
const long emisor3 = 34;
const long emisor4 = 36;

//Receptores centrales y laterales de la parte frontal del carrito. Receptor= Echo
const long receptor1 = 31;
const long receptor2 = 33;
const long receptor3 = 35;
const long receptor4 = 37;

const float potenciometro = A6;

int poten = 0;

void setup()
{
  pinMode(a0, OUTPUT);
  pinMode(a1, OUTPUT);
  pinMode(b0, OUTPUT);
  pinMode(b1, OUTPUT);

  pinMode(pwm_X, OUTPUT);
  pinMode(pwm_Y, OUTPUT);

  pinMode(obstaculoD, INPUT);
  pinMode(obstaculoT, INPUT);
  pinMode(obstaculoL, INPUT);
  pinMode(obstaculoR, INPUT);

  pinMode(emisor1, OUTPUT);
  pinMode(emisor2, OUTPUT);

```

```

pinMode(emisor3, OUTPUT);
pinMode(emisor4, OUTPUT);

pinMode(receptor1, INPUT);
pinMode(receptor2, INPUT);
pinMode(receptor3, INPUT);
pinMode(receptor4, INPUT);

Serial.begin(9600);
}

float leer_analog_y_map (float analog, int minimo, int maximo)
{
    float valor;
    valor = analogRead(analog);
    valor = map(valor, 0, 1023, minimo, maximo);

    return valor;
}

//Subrutinas de los sensores ultrasonicos
void sensor1() {
    digitalWrite(emisor1, LOW);
    digitalWrite(emisor1, HIGH);
    digitalWrite(emisor1, LOW);

    tiempo1 = pulseIn(receptor1, HIGH);
    distancia1 = int(0.017 * tiempo1);

    Serial.print("distancia1:");
    Serial.print(distancia1);
}
void sensor2() {
    digitalWrite(emisor2, LOW);
    digitalWrite(emisor2, HIGH);
    digitalWrite(emisor2, LOW);

    tiempo2 = pulseIn(receptor2, HIGH);
    distancia2 = int(0.017 * tiempo2);

    Serial.print("          distancia2:");
    Serial.print(distancia2);
}
void sensor3() {
    digitalWrite(emisor3, LOW);
    digitalWrite(emisor3, HIGH);
    digitalWrite(emisor3, LOW);

    tiempo3 = pulseIn(receptor3, HIGH);
    distancia3 = int(0.017 * tiempo3);

    Serial.print("          distancia3:");
    Serial.print(distancia3);
}
void sensor4() {
    digitalWrite(emisor4, LOW);
    digitalWrite(emisor4, HIGH);
    digitalWrite(emisor4, LOW);

    tiempo4 = pulseIn(receptor4, HIGH);
    distancia4 = int(0.017 * tiempo4);

    Serial.print("          distancia4:");
    Serial.println(distancia4);
}

```

```

//-----
//SUBROUTINA PARACHOQUES DELANTERO (GRUPO 2)
//-----
int escaneoZonaD() {
    int LED1=0;

    sensor1();
    sensor2();
    sensor3();
    sensor4();

    //Condiciones.
    if(distancia1 < 55 || distancia2 < 100 || distancia3 < 100 || distancia4 < 55)
    {
        LED1=1;
    }
    else
    {
        LED1=0;
        if(distancia1 == 0 || distancia2 == 0 || distancia3 == 0 || distancia4 == 0)
        {
            LED1=0;
        }
    }

    return LED1;
}

void loop()
{
    // joystick eje Y
    int estadoD = 0;
    int estadoT = 0;
    int estadoL = 0;
    int estadoR = 0;
    int obstaculoUltraD;

    lecturaY = leer_analog_y_map(joyY, 0, 5000);
    lecturaX = leer_analog_y_map(joyX, 0, 5000);

    estadoD = digitalRead(obstaculoD);
    estadoT = digitalRead(obstaculoT);
    estadoL = digitalRead(obstaculoL);
    estadoR = digitalRead(obstaculoR);

    obstaculoUltraD=escaneoZonaD();

    poten = analogRead(potenciometro);

    analogWrite(pwm_Y, poten/4);
    analogWrite(pwm_X, poten/4);

    //TENSION JOYSTICK=X0 --> (a0=0-a1=0)
    digitalWrite(a0, LOW);
    digitalWrite(a1, LOW);
    digitalWrite(buzzer, LOW);

    //TENSION DEL DIVISOR DE TENSION PARO=X1 -->(a0=1-a1=0)
    if (lecturaY > 2600 && lecturaY < 5000)    //OBSTACULO DELANTERO
    {
        if (estadoD == HIGH || obstaculoUltraD == 1) //Esto lo hay que hacer con cada zona de deteccion (que aun
        no estan programadas)
        {
            digitalWrite(a0, HIGH);

```

```

    digitalWrite(a1, LOW);

    digitalWrite(buzzer, HIGH);
  }
}

if (lecturaY >= 2000 && lecturaY <= 2600) //PUNTO MEDIO
{
    digitalWrite(a0, HIGH);
    digitalWrite(a1, LOW);
}

if (lecturaY >= 50 && lecturaY < 2000) //OBSTACULO TRASERO
{
    if (estadoT == HIGH)
    {
        digitalWrite(a0, HIGH);
        digitalWrite(a1, LOW);
    }
}

//TENSION JOYSTICK=X0 --> (b0=0-b1=0)
digitalWrite(b0, LOW);
digitalWrite(b1, LOW);

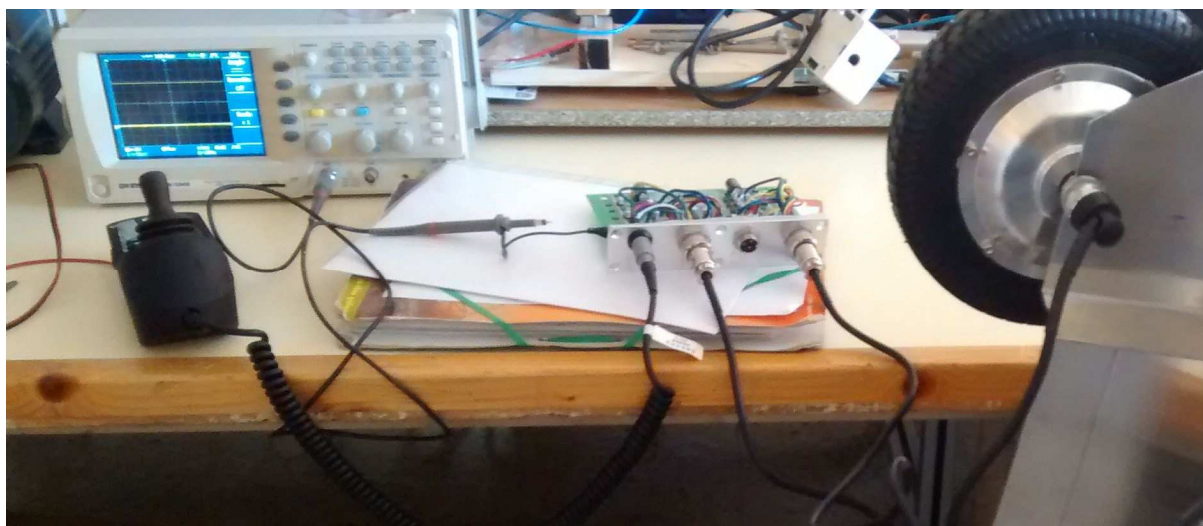
//TENSION DIVISOR DE TENSION PARO=X1 -->(b0=1-b1=0)
if (lecturaX > 2600 && lecturaX < 5000) //OBSTACULO IZQUIERDO
{
    if (estadoL == HIGH)
    {
        digitalWrite(b0, HIGH);
        digitalWrite(b1, LOW);
    }
}

if (lecturaX >= 2000 && lecturaX <= 2600) //PUNTO MEDIO
{
    digitalWrite(b0, HIGH);
    digitalWrite(b1, LOW);
}

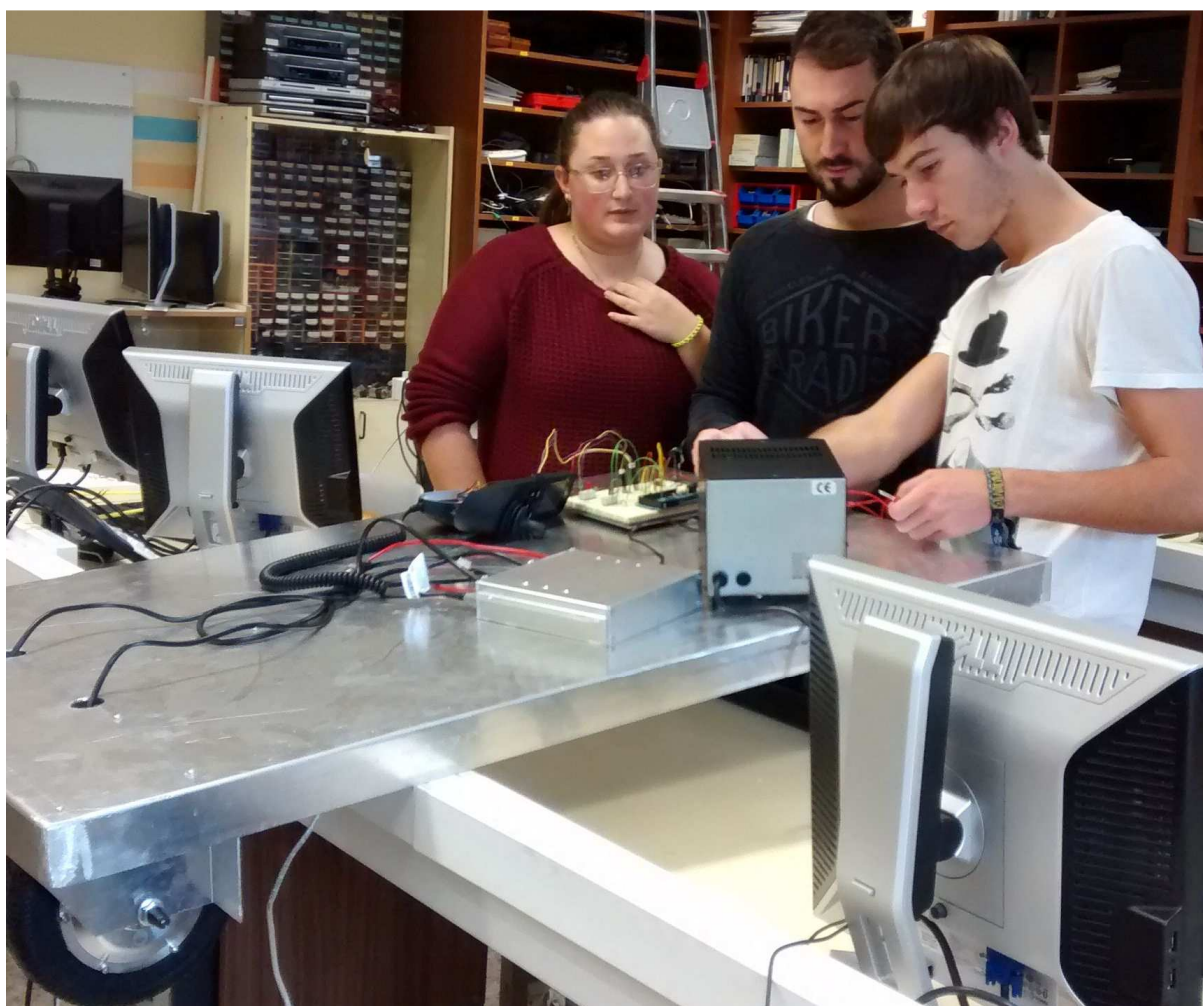
if (lecturaX >= 50 && lecturaX < 2000) //OBSTACULO DERECHO
{
    if (estadoR == HIGH)
    {
        digitalWrite(b0, HIGH);
        digitalWrite(b1, LOW);
    }
}
}

```

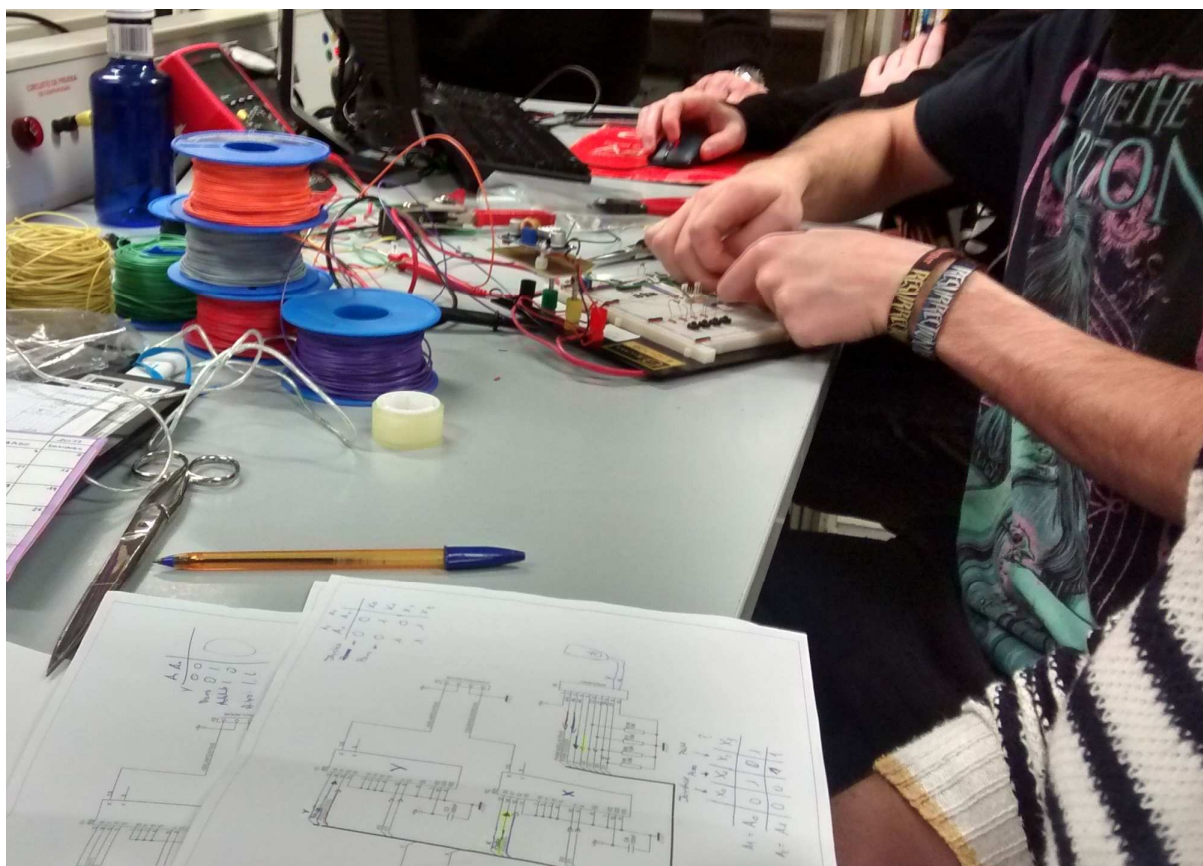
Galería de imaxes



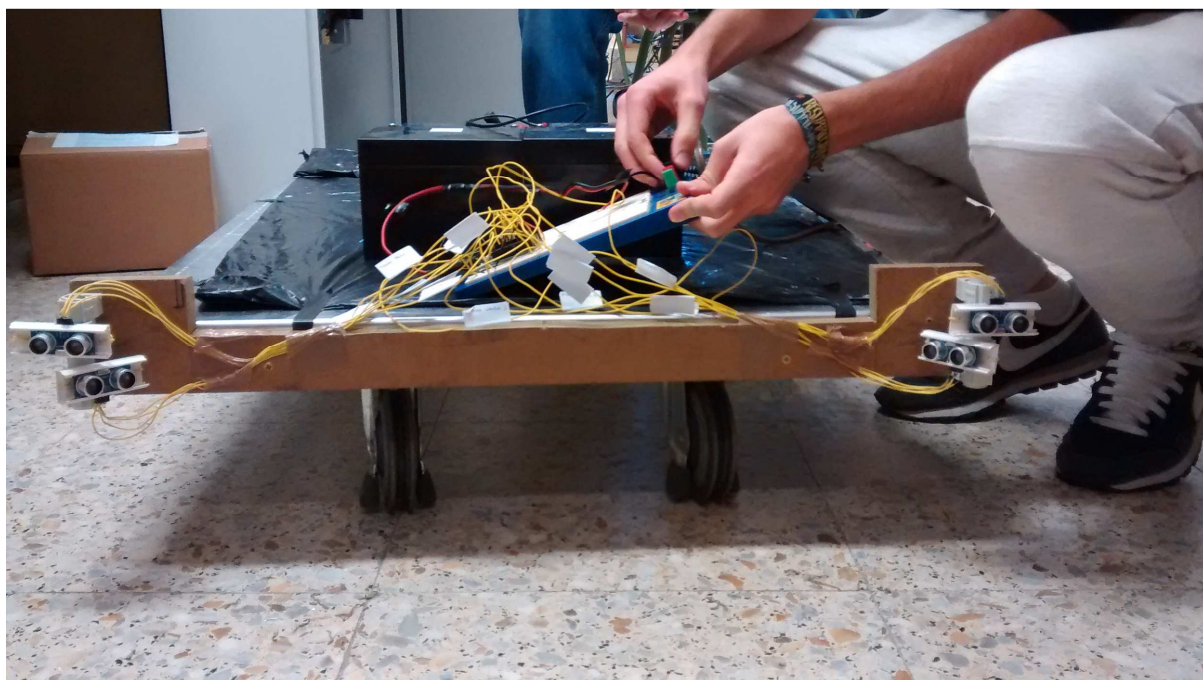
Estudando os sinais de control dos motores brushless de tracción sobre o prototipo do carriño



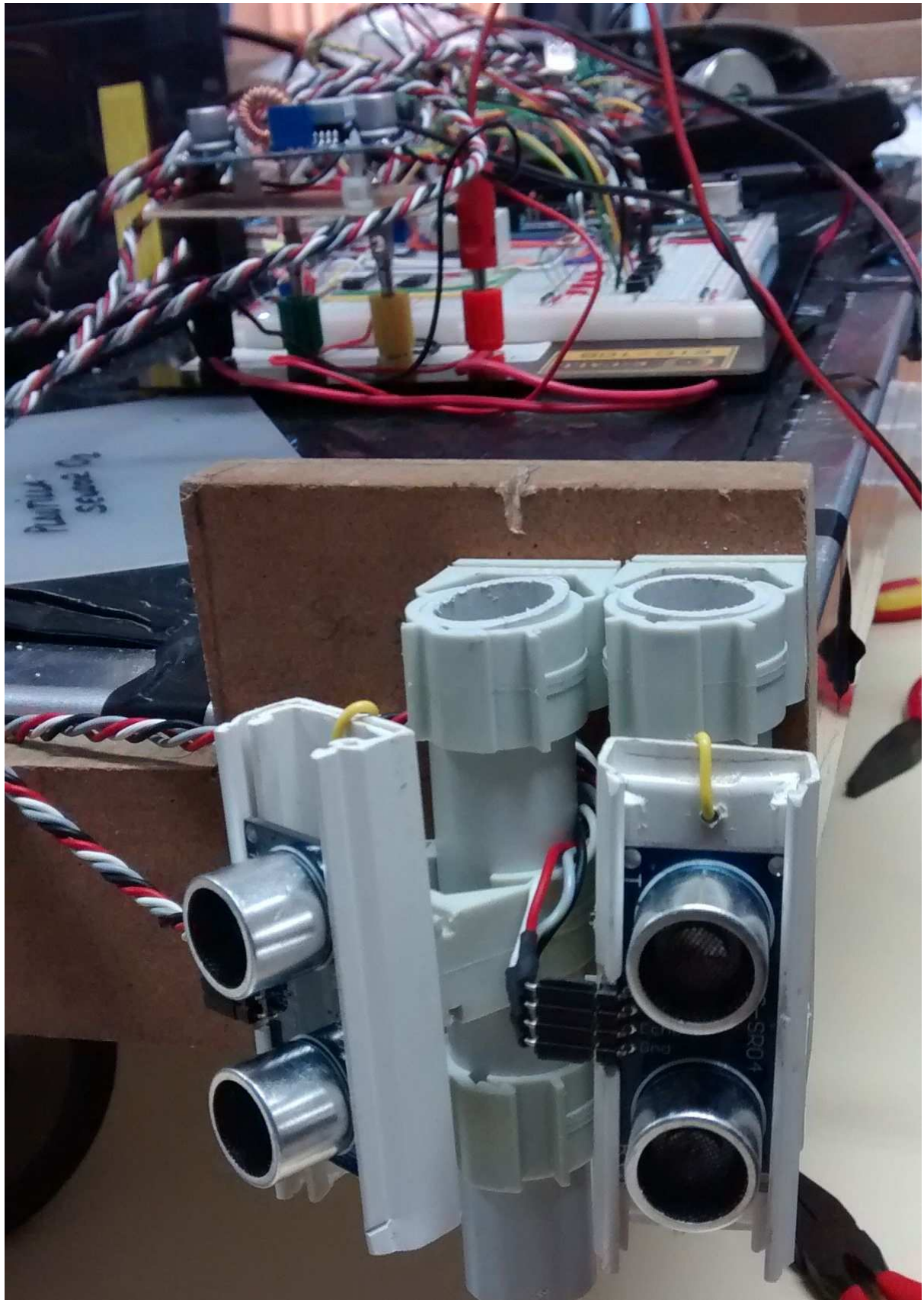
Primeiras probas co prototipo do carriño



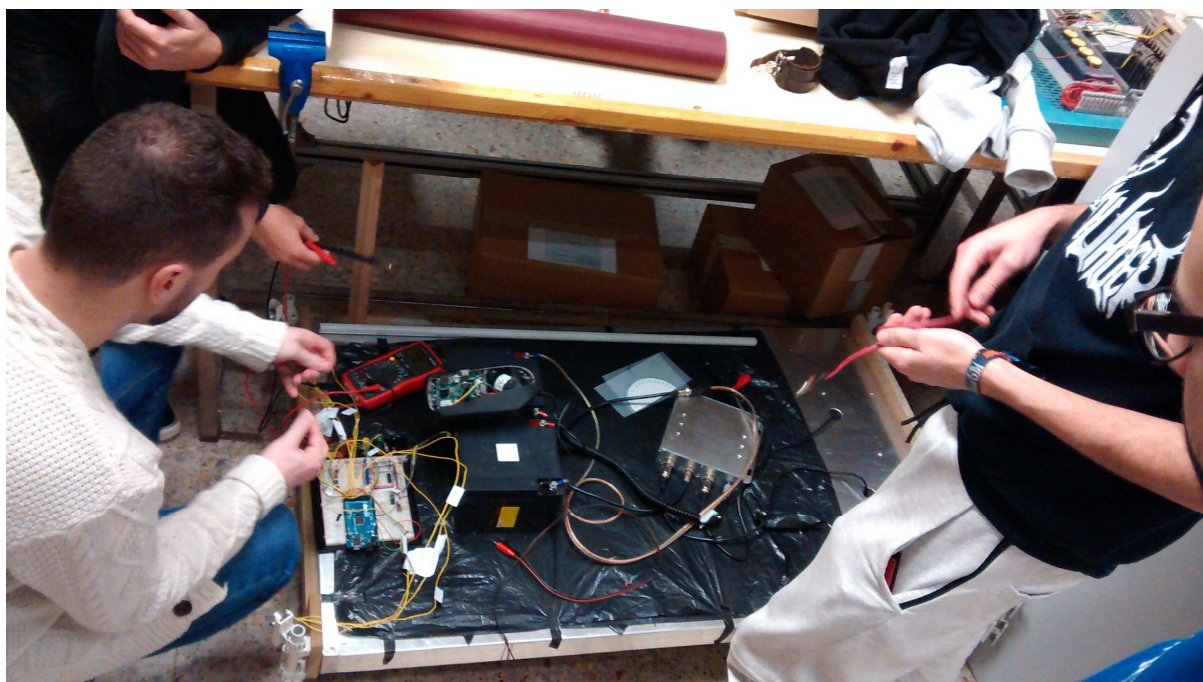
Construindo o interfaz para o control de tracción



Instalando a primeira versión do sistema anticoliión por ultrasóns sobre o prototipo



Sensores de ultrasóns para o sistema anticolidión, unha versión perfeccionada



Integrando o control de tracción e o sistema anticoliación