Universidad Mayor de San Andrés
Facultad de ingeniería

INGENIERÍA ELECTRÓNICA

Informe Final

““INTERFASE DE CONTROL REMOTO DE UN VEHICULO PEQUEÑO””

Asignatura: INTERACCION HARDWARE-SOFTWARE (ETN-1022)
Estudiante: Univ. Diego Alejandro Cruz Torrez
Docente: Ing. Pedro Clifford Paravicini Hurtado

Fecha de presentación: 10/06/2023

Objetivo

Diseñar,programar,probar y desplegar una interfase de control remoto de un vehiculo pequeño,en un plazo de 7 semanas

Aplicaciones

El control remoto de un vehiculo pequeño,tiene una variedad de aplicaciones en las que destacan :

-Transporte de pequeños objetos

-Monitoreo de espacios cerrados

-Juguete

Cronograma

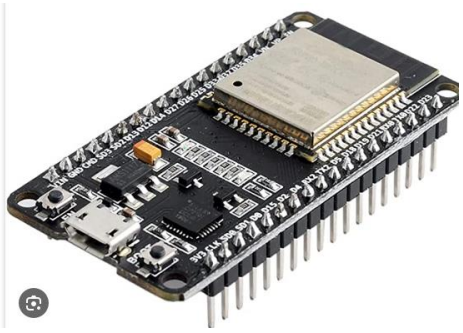
		SEMANAS						
META	ACTIVIDAD	1	2	3	4	5	6	7
1.- Identificar los componentes de hardware y software necesarios para la implementacion de dicho proyecto	Determinar software							
	Deteminar hardware							
	Realizar un presupuesto							
2.-Diseño de la armadura del vehiculo	Posicionamiento de los componentes electronicos							
	Proceso de construccion							
	Proceso de ensamblaje							
3.-Diseño de la interfase de control	Selección de la base de datos							
	Escritura del codigo de la interfase							
	Formato de la interfase							
4.-Pruebas	Carga de software en controlador							
	Pruebas							
5.-Ajustes	Ajustes							
6.-Finalizacion	Toques finales							

Presupuesto componentes Hardware

Componente	Precio
esp32	70
estructura 2WD	70
usb CONECTOR	4
antena	22
driver	22
placa	2,5

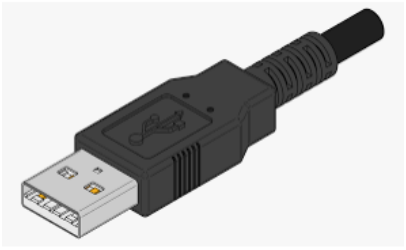
perforada	
Total	190,5

ESP 32



Estructura 2WD

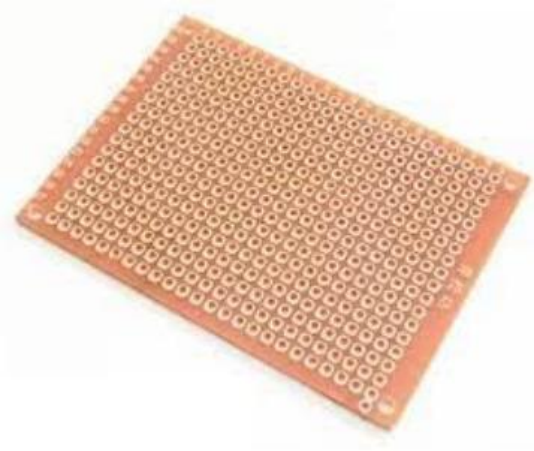
Conector USB



Puente H

Antena

Placa perforada



Planteamiento del problema

-El ESP32 debe recibir información alojada en una base de datos con el fin de controlar bidireccionalmente el funcionamiento de 2 motores

-La base de datos deberá indicar la dirección del movimiento del vehiculo

-Una interface deberá controlar estos cambios de acuerdo al usuario

-La respuesta deberá ser inmediata pues el control es en tiempo real

Construccion Hardware

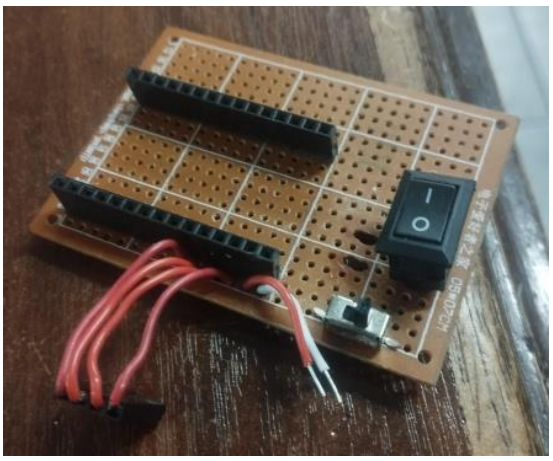
Construcción del circuito

-Esp32

-4 salidas al puente H

-Interruptor alimentación motor

-Interruptor alimentación Esp32

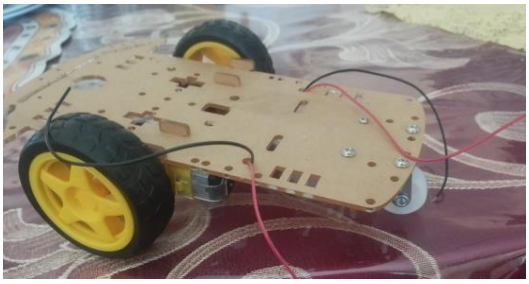


Construcción del vehículo

-2 Ruedas fijas

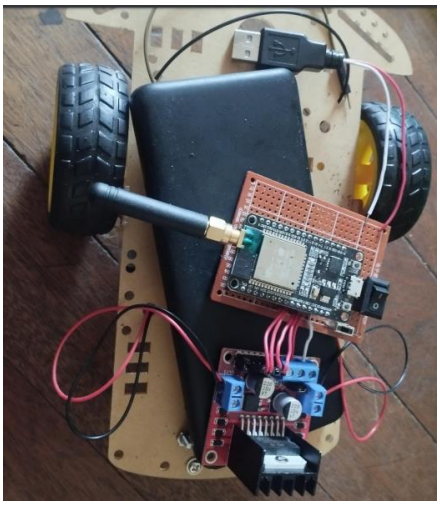
-1 rueda omnidireccional

-2 motores,4 entradas



Ensamblado del vehiculo

Incorporacion del puente H,la antena y la placa perforada con el ESP32



-Se añadió una antena SMA pará una respuesta rápida ante cambios en la base de datos

-La fuente de poder es un cargador portátil con la suficiente potencia para alimentar 2 motores
-Se añadió un switch de alimentación

-Se conecto la placa perforada al modulo puente H

- Se inserto una carrocería proveniente de un auto de juguete

Diseño del software

Programa que permite que el ESP32 reciba información proveniente del los cambios en la base de datos en Firebase

```
#include"Arduino.h"
#include<Arduino.h>
#include<WiFi.h>
#include<Firebase_ESP_Client.h>
#include"addons/TokenHelper.h"
#include"addons/RTDBHelper.h"

// Datos de la red Wifi
#define WIFI_SSID "CASARDA"
#define WIFI_PASSWORD "909225011"

// Firebase project API Key
#define API_KEY "AIzaSyByub4GKNug07U33W_a7-0Vv8alB6NV9iU"

// Usuario y contraseña permitidos
#define USER_EMAIL "diego0987123@gmail.com"
#define USER_PASSWORD "67181239"

// URL de la base de datos
#define DATABASE_URL "https://appif-d66a8-default-
rtdb.firebaseio.com/DACTCTRL"

// Objetos de Firebase
FirebaseData stream;
FirebaseAuth auth;
FirebaseConfig config;

// Ruta de lectura
String listenerPath ="DACTCTRL/";

// Motor A
int state1 =0;
int motor1Pin1 =13;
int motor1Pin2 =12;
// Motor B
int motor2Pin1 =14;
int motor2Pin2 =27;

voidAdelante(){
  digitalWrite(motor1Pin1, 1);
  digitalWrite(motor1Pin2, 0);
  digitalWrite(motor2Pin1, 1);
```

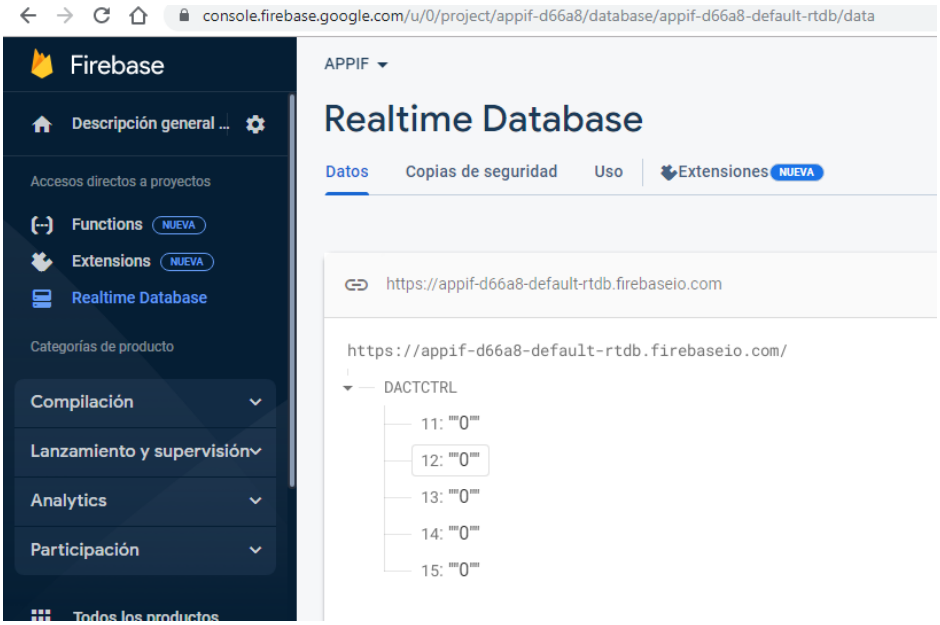
```
  digitalWrite(motor2Pin2, 0);
}
voidIzquierda(){
  digitalWrite(motor1Pin1, 0);
  digitalWrite(motor1Pin2, 1);
  digitalWrite(motor2Pin1, 1);
  digitalWrite(motor2Pin2, 0);
}
voidDerecha(){
  digitalWrite(motor1Pin1, 1);
  digitalWrite(motor1Pin2, 0);
  digitalWrite(motor2Pin1, 0);
  digitalWrite(motor2Pin2, 1);
}
voidAtras(){
  digitalWrite(motor1Pin1, 0);
  digitalWrite(motor1Pin2, 1);
  digitalWrite(motor2Pin1, 0);
  digitalWrite(motor2Pin2, 1);
}
voidStop(){
  digitalWrite(motor1Pin1, 0);
  digitalWrite(motor1Pin2, 0);
  digitalWrite(motor2Pin1, 0);
  digitalWrite(motor2Pin2, 0);
}
voidinitWiFi() {
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to WiFi ..");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print('.');
    delay(1000);
  }
  Serial.println(WiFi.localIP());
  Serial.println();
}

// Funcion que detecta cambios en la base de datos
voidstreamCallback(FirebaseStreamdata){
  Serial.printf("stream path, %s\nevent path, %s\ndata type, %s\nevent
type, %s\n\n",
                data.streamPath().c_str(),
                data.dataPath().c_str(),
                data.dataType().c_str(),
                data.eventType().c_str());
  printResult(data);
  Serial.println();

  // Ruta de cambio
```

```
String streamPath =String(data.dataPath());


String btn =streamPath.substring(1);
int state =data.intData();
state1 =!state1;
Serial.print("Boton: ");
Serial.println(btn);
Serial.print("STATE: ");
Serial.println(state1);
if (state1==1){
  switch(btn.toInt()){
  case11:
  Serial.print("Ad ");
  Adelante();
  break;
  case12:
  Serial.print("Iz ");
  Izquierda();
  break;
  case13:
  Serial.print("sp");
  Stop();
  break;
  case14:
  Serial.print("der ");
  Derecha();
  break;
  case15:
  Serial.print("At");
  Atras();
  break;
  default:
  Stop();
  break;
  }
  }

/* Lectura inicial*/
if (data.dataTypeEnum() == fb_esp_rtdb_data_type_json){
  FirebaseJson json =data.to<FirebaseJson>();

  size_t count =json.iteratorBegin();
  Serial.println("\n---------");
  for (size_t i =0; i < count; i++){
      FirebaseJson::IteratorValue value =json.valueAt(i);
      int btn =value.key.toInt();
      int state =value.value.toInt();
      Serial.print("STATE: ");
```

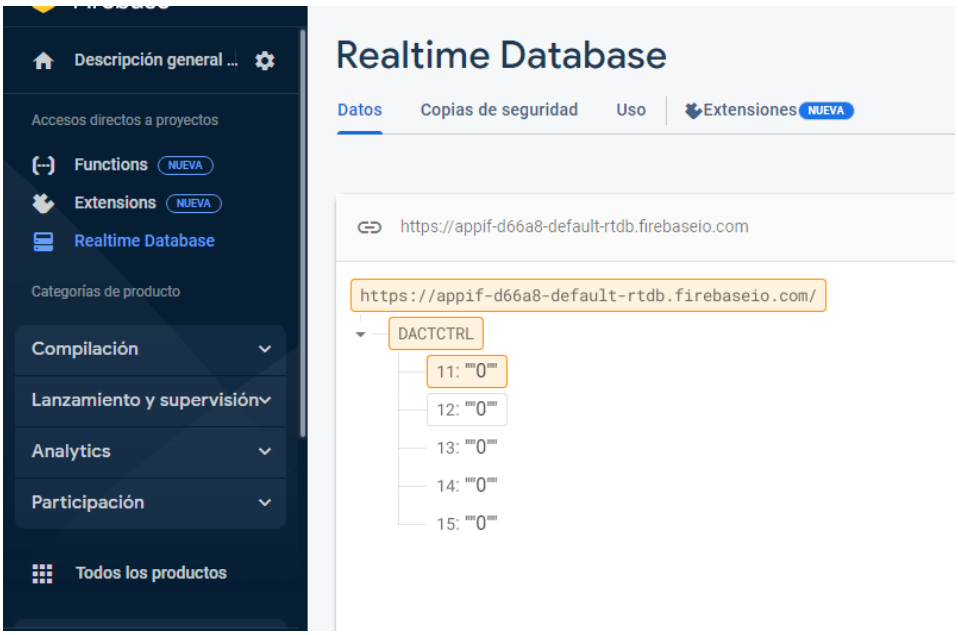
```
        Serial.println(state);
        Serial.print("Boton:");
        Serial.println(btn);
        if (state==1){
          switch(btn){
          case11:
          Adelante();
          break;
          case12:
          Izquierda();
          break;
          case13:
          Derecha();
          break;
          case14:
          Atras();
          break;
          case15:
          Stop();
          break;
          default:
          break;
          }
        }
    }
    json.iteratorEnd();
  }
}

voidstreamTimeoutCallback(booltimeout){
  if (timeout)
    Serial.println("stream timeout, resuming...\n");
  if (!stream.httpConnected())
    Serial.printf("error code: %d, reason: %s\n\n", stream.httpCode(),
stream.errorReason().c_str());
}

voidsetup() {
  Serial.begin(115200);
  initWiFi();
  // Salidas
  pinMode(motor1Pin1, OUTPUT);
  pinMode(motor1Pin2, OUTPUT);
  pinMode(motor2Pin1, OUTPUT);
  pinMode(motor2Pin2, OUTPUT);


  // API key
  config.api_key= API_KEY;
```

```
  // Credenciales
  auth.user.email= USER_EMAIL;
  auth.user.password= USER_PASSWORD;

  //URL
  config.database_url= DATABASE_URL;

  Firebase.reconnectWiFi(true);

  //Conexion */
  config.token_status_callback= tokenStatusCallback; //see
addons/TokenHelper.h

  // 5 intentos
  config.max_token_generation_retry=5;

  // Inicializando Firebase
  Firebase.begin(&config, &auth);

  // Visualizar la ruta de cambio
  if (!Firebase.RTDB.beginStream(&stream, listenerPath.c_str()))
    Serial.printf("stream begin error, %s\n\n",
stream.errorReason().c_str());

  // Deteccion de cambios
  Firebase.RTDB.setStreamCallback(&stream, streamCallback,
streamTimeoutCallback);

  delay(500);

}

voidloop() {
  if (Firebase.isTokenExpired()){
    Firebase.refreshToken(&config);
    Serial.println("Refresh token");
  }
}
```

Base de datos en Firebase



Deteccion de cambios



Interfase creada utilizando MIT APP INVENTOR

Tras presionar un botón se pone un 1 en su respectiva ruta activando la dirección requerida,pulsar stop detendrá el vehiculo independientemente de los otros estados

Implementacion y pruebas

-Tras la instalación se procedió a instalar la aplicación que contiene la interface diseñada

-Se realizaron diferentes pruebas controlando el vehiculo en distintos dispositivos teniendo éxito al control el vehiculo sin inconvenientes

Conclusiones y recomendaciones

-Se logro implementar la interface de un vehiculo a control remoto exitosamente

-Se logro controlar el vehiculo a través de diversos dispositivos

-Se cumplieron los objetivos dados

-Es posible utilizar el ESP-CAM en lugar del ESP32 con el fin de transmitir imágenes desde el punto de vista del vehiculo, mejorando el control del mismo