

VART

Vehículo Autónomo para
Registro de Temperaturas



INTRODUCCION

Este Proyecto desarrolla un vehículo micro controlado para registrar temperatura en determinados puntos, con un control de trayectoria remoto desde una app IOS donde se van a reportar los datos de la temperatura.

¿QUE?

¿CUANDO?

¿QUIENES?

¿Que?

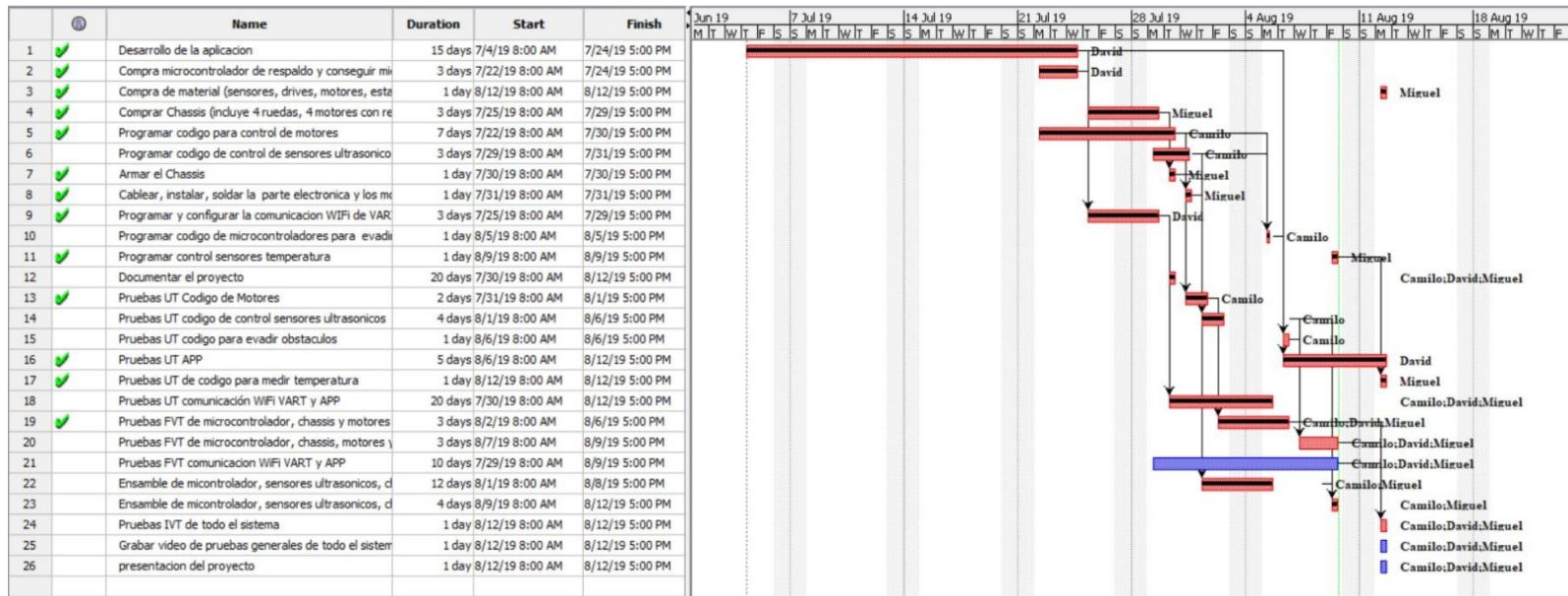


Se usa una plataforma (chassis) conocida y desarrollada previamente para un vehículo de 4 ruedas, 4 motores, 2 sensores (ultrasónicos para evadir obstáculos).

Tambien se incorpora el uso de un microcontrolador con un modulo WiFi incluido.

¿Cuando?

Dentro de la materia Arquitectura de microcontroladores



**Junio 26 - Agosto 12
2019**

¿Quienes?

El equipo.....

1 Camilo Espinosa

2 David Osollo

3 Miguel Torres

VART

Vehículo Autónomo para
Registro de Temperaturas



OBJETIVOS

- Desarrollar e implementar un vehículo microcontrolado capaz de evadir obstáculos y registrar temperaturas

• Desarrollar e implementar una app en IOS para enviar comandos hacia un vehículo microcontrolado, sobre las 4 direcciones: izquierda, derecha, adelante, atrás, detener y el comando de medición de temperatura.

• Recopilar información sobre la temperatura en distintos puntos dentro del área de trabajo del vehículo microcontrolado.

Entregables

Entregables

SW

Una app para IOs que puede controlar el vehículo desarrollado

PLATAFORMA

Plataforma mecánica de un vehículo de 4 ruedas, controlado por un ESP8266, con adaptaciones de sensores ultrasónicos(2) y de temperatura (1)

DRIVERS

El código del microcontrolador para ejecutar tareas de avance, retroceso, giro a la derecha, giro a la izquierda, medición de temperatura y envío de datos.

VART

Vehículo Autónomo para
Registro de Temperaturas

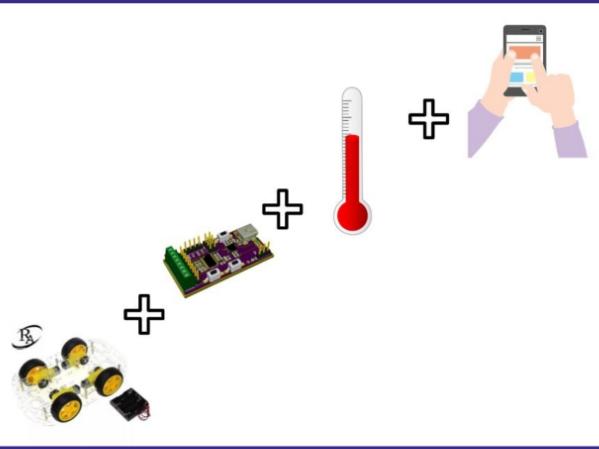


MATERIALES

- ESP8266
- XCODE framework
- Arduino framework
- PlatformIO
- Sensor de temperatura LM35
- Sensor Ultrasonido HC-SR04
- Chassis
- Motores DC, servos.

DISEÑO

Requerimientos



Requerimientos

Alto, Medio, Bajo

- | | |
|--|---|
|  Incluir un sistema microcontrolado que maneje puertos I/O y comunicación |  Vehiculo microcontrolado con 4 ruedas, en lugar de 3, para lograr estabilidad |
|  Comunicación entre sistema microcontrolado y app IOS |  Medición de temperatura |
|  Desarrollar una app en IOs |  Evadir obstaculos |
|  Video del vehiculo funcionando | |

VART

Vehículo Autónomo para
Registro de Temperaturas



Desarrollo

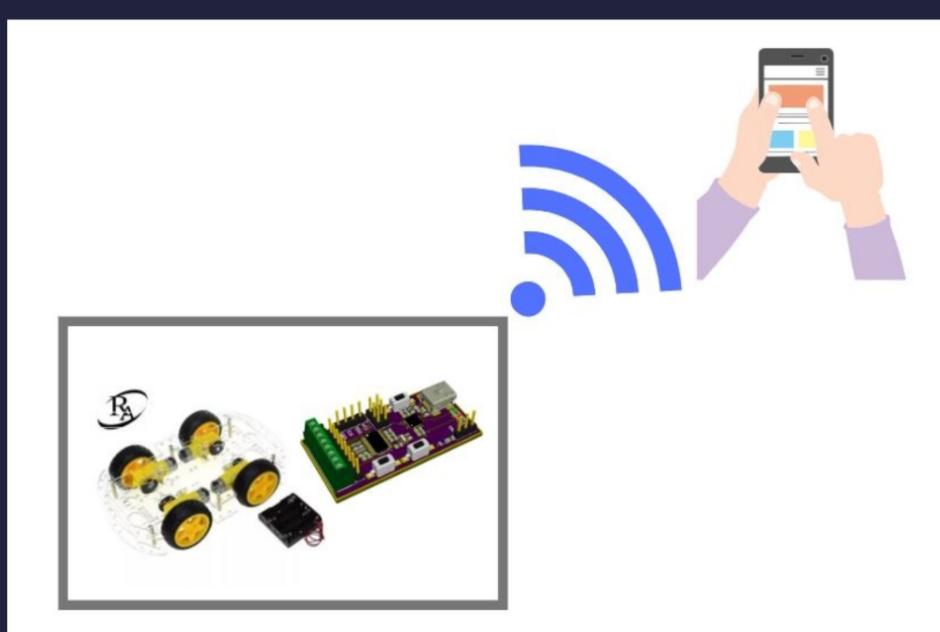
- Se desarrollaron 3 grandes bloques:
 - El vehículo incluyendo el chassis y el driver de los motores.
 - Los sensores, que incluyen sensores ultrasónicos y de temperatura.
 - La App que utiliza el protocolo WiFi para comunicarse con el vehículo.

Vehículo

Sensores

WiFi

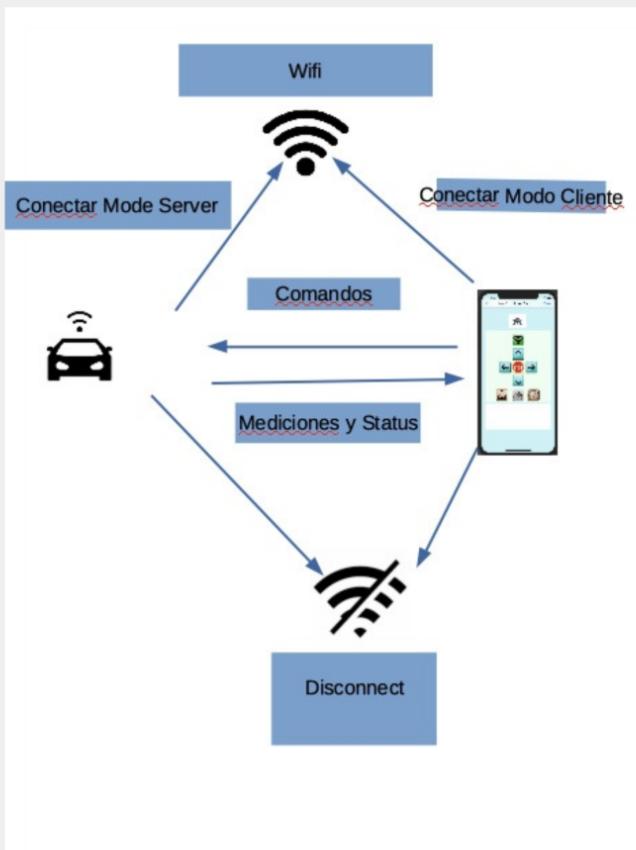
Comunicación WiFi



La comunicación entre el vehículo y la app es WiFi, se envían comandos y se recibe el dato de temperatura

```
//Commands from móvil
#define CMD_STOP      '0'
#define CMD_FORWARD   '1'
#define CMD_BACKWARD  '2'
#define CMD_LEFT      '3'
#define CMD_RIGHT     '4'
#define CMD_AUTO      '5'
#define CMD_MEASURE   '6'
```

Comunicación WiFi

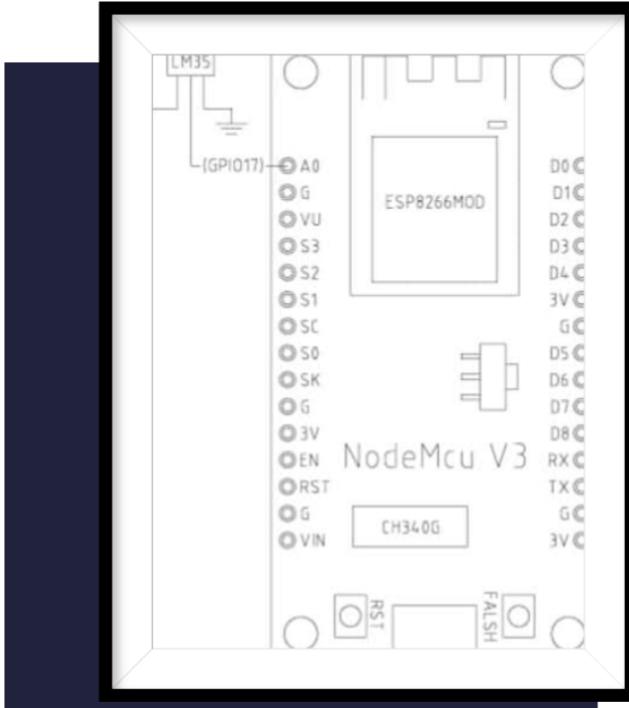


La app establece conexión con la red donde esta conectado el vehículo, se envían comandos en valores ASCII, la medición se recibe como un valor float.



Sensores

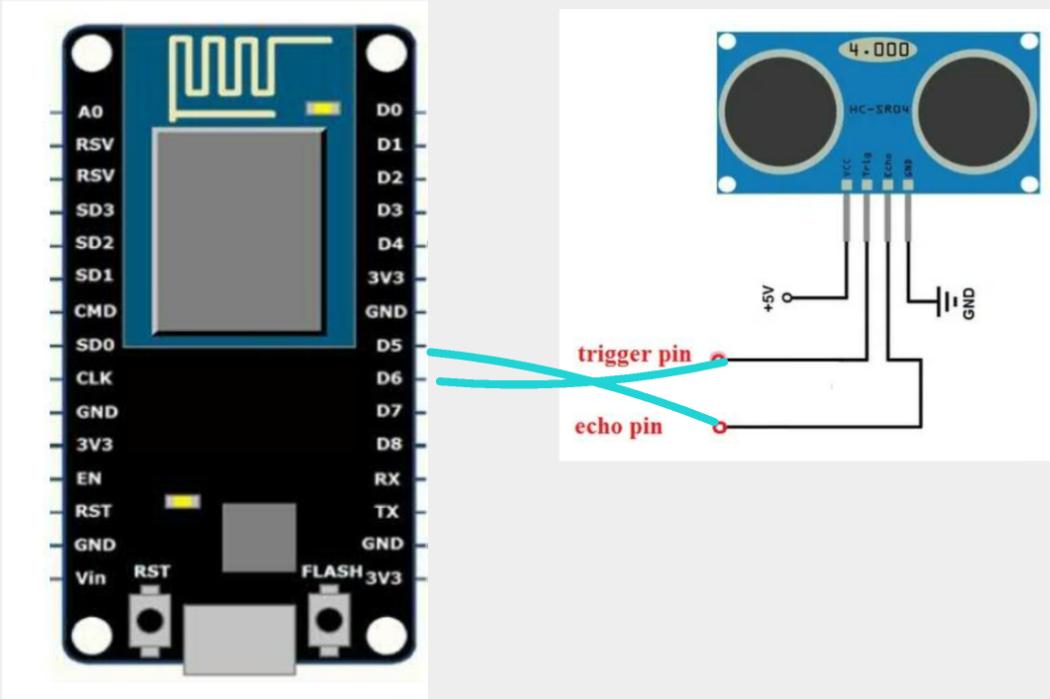
Sensor de Temperatura



Next

Sensores

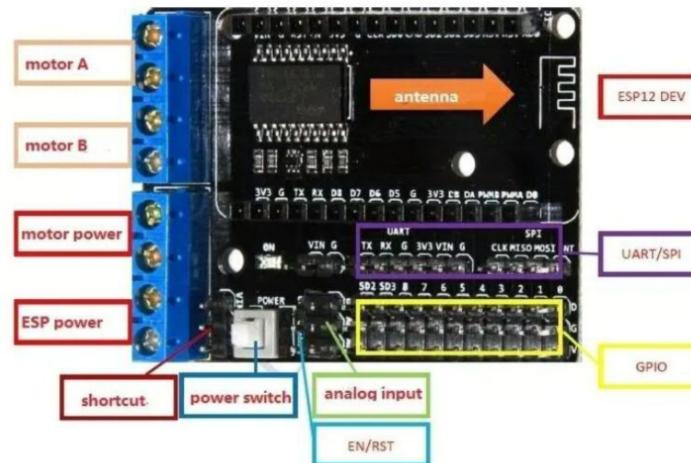
Sensor de Ultrasonido



- Cada segundo se verifica la distancia, debe ser mayor a 50cm.
- Si se detecta un obstáculo se hace un giro a la derecha de 90 grados, se avanza 1 seg, y se gira a la izquierda.
- Se vuelve a avanzar y se repite el ciclo.

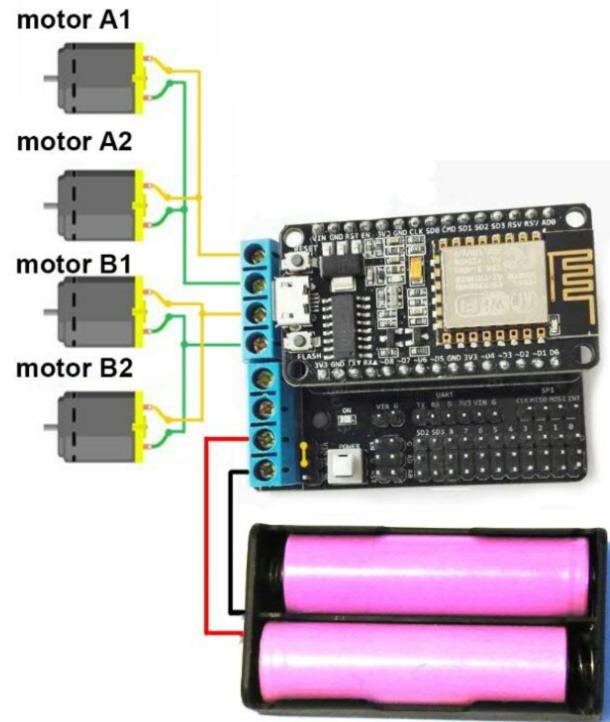
Vehículo

Conexiones, driver, código.



Código

```
#define PIN_D1 5 // gpio5 = D1 PWM_A  
#define PIN_D2 4 // gpio4 = D2 PWM_B  
#define PIN_D3 0 // gpio0 = D3 DA (A- A+)  
#define PIN_D4 2 // gpio2 = D4 DB (B- B+)
```



```
bool forwardmotor(){
//forward
Serial.println("starting forward");

digitalWrite(PIN_D1, HIGH); // PWM_A HIGH
digitalWrite(PIN_D3, HIGH); // DA HIGH
digitalWrite(PIN_D2, HIGH); // PWM_B HIGH
digitalWrite(PIN_D4, HIGH); // DB HIGH
delay(1000); // wait
Serial.println("forward");
return true;
}
```

```
bool turnrightmotor(){
//turn right

digitalWrite(PIN_D1, HIGH); // PWM_A HIGH
digitalWrite(PIN_D2, HIGH); // PWM_B HIGH
digitalWrite(PIN_D3, HIGH); // DA HIGH
digitalWrite(PIN_D4, LOW); // DB HIGH
//digitalWrite(PIN_D4, HIGH); // DB HIGH
delay(600); // wait
Serial.println("Right");

return true;
}
```

```
bool forwardautomotor(){
//forward
Serial.println("starting forward auto");
Serial.print("Distance in auto: ");
Serial.println(distance);
while(client.read()!= CMD_STOP)
{
    while(distance >50 )
    {
        digitalWrite(PIN_D1, HIGH); // PWM_A HIGH
        digitalWrite(PIN_D3, HIGH); // DA HIGH
        digitalWrite(PIN_D2, HIGH); // PWM_B HIGH
        digitalWrite(PIN_D4, HIGH); // DB HIGH
        delay(1000); // wait
        Serial.println("forward auto");
        distanceobst();
    }
    stopmotor();
    delay(800);

    //turing right 90 degrees
}
```

```
void distanceobst() {

    // Clears the trigPin
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);

    // Sets the pingPin on HIGH state for 10 micro seconds
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(pingPin, LOW);

    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);

    // Calculating the distance
    distance= duration*0.034/2;
    // Prints the distance on the Serial Monitor
    Serial.print("Distance: ");
    Serial.println(distance);
    delay(200);
}
```

VART

Vehículo Autónomo para
Registro de Temperaturas



Evidencia 1

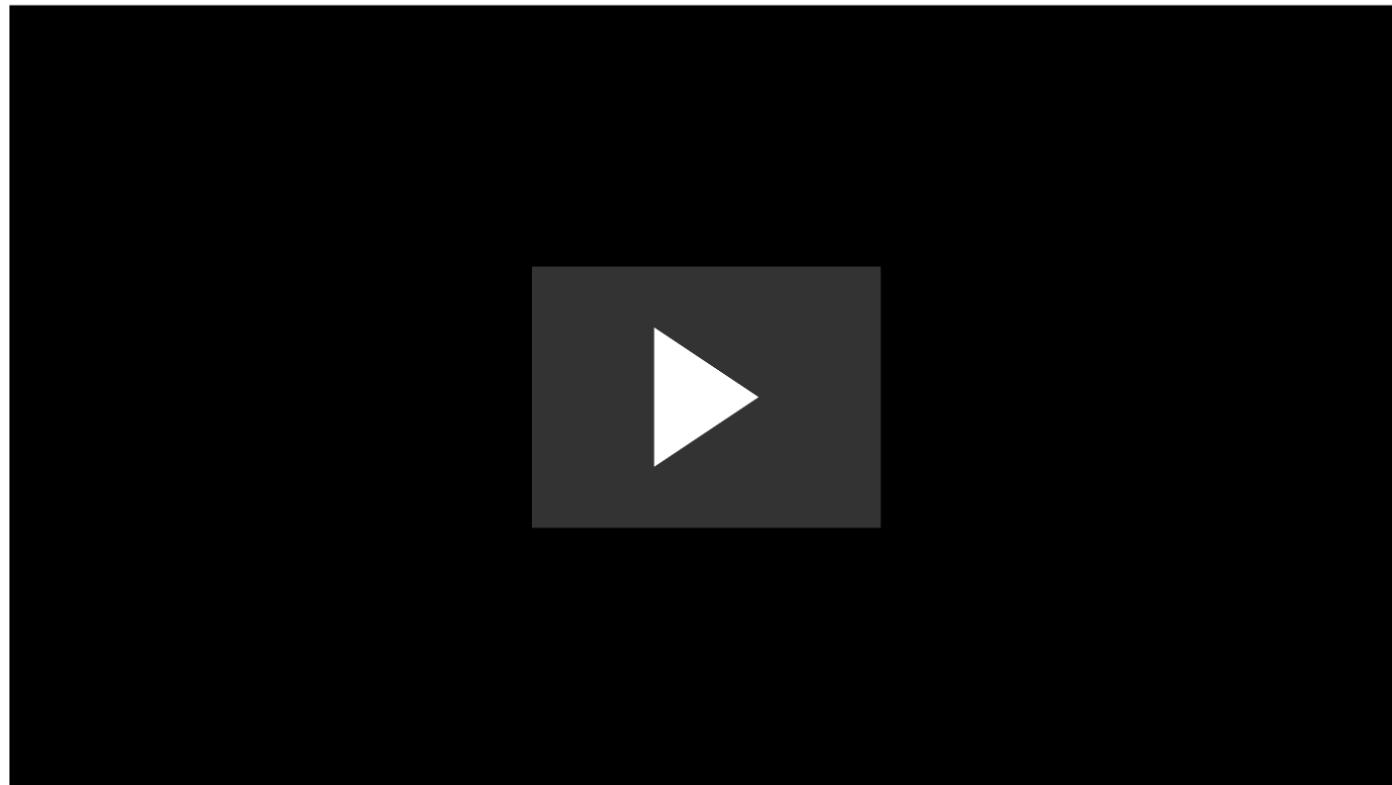
Resultados

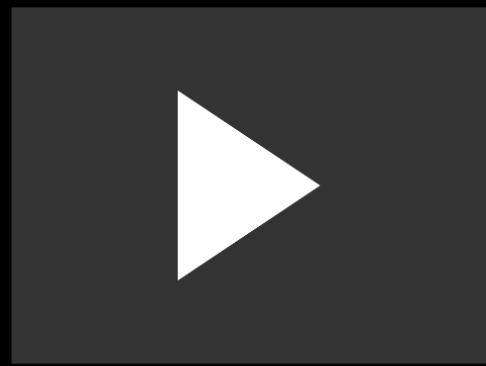


TestCase	Descripción del requerimiento	Descripción del TestCase
1.1	Incluir un sistema microcontrolado que maneje puertos I/O y comunicación	Programar el vehículo por puerto serial para ejecutar movimientos predefinidos
1.2	Comunicación entre sistema microcontrolado y app iOS	Establecer comunicación entre microcontrolador y app por medio de WiFi, enviar comando de avanzar desde la app hacia el micro. Se espera que el vehículo avance en linea recta(UT WiFi - VART-APP)
1.3	Comunicación WiFi	Establecer comunicación entre microcontrolador y app por medio de WiFi, enviar comando de avanzar desde la app hacia el micro. Se espera que el vehículo avance en linea recta(UT WiFi - VART-APP). Despues de 10 segundo enviar comando para detener el vehiculo, se espera que el vehículo se detenga.
1.4	Desarrollar una app en iOS	Ejecutar app en iOS que tenga botones para avanzar, derecha, izquierda, detener.
1.5	Evadir obstaculos	Poner una trayectoria con una caja de carton que sirve como obstaculo, enviar comandos para avanzar. Se espera que el vehículo avance rodeando hacia la derecha el obstaculo
1.6	Vehículo microcontrolado con 4 ruedas, en lugar de 3, para lograr estabilidad	Enviar comandos para avanzar, girar a la derecha, avanzar, girar a la izquierda, avanzar y detener.
1.7	Medición de temperatura	Ejecutar el comando de comando de avanzar, detener, medicion. Se espera que en la app se reciba el dato de la medición.
1.8	video del vehículo funcionando	Tomar un video de la ejecución de 1.6, 1.5 y 1.7.

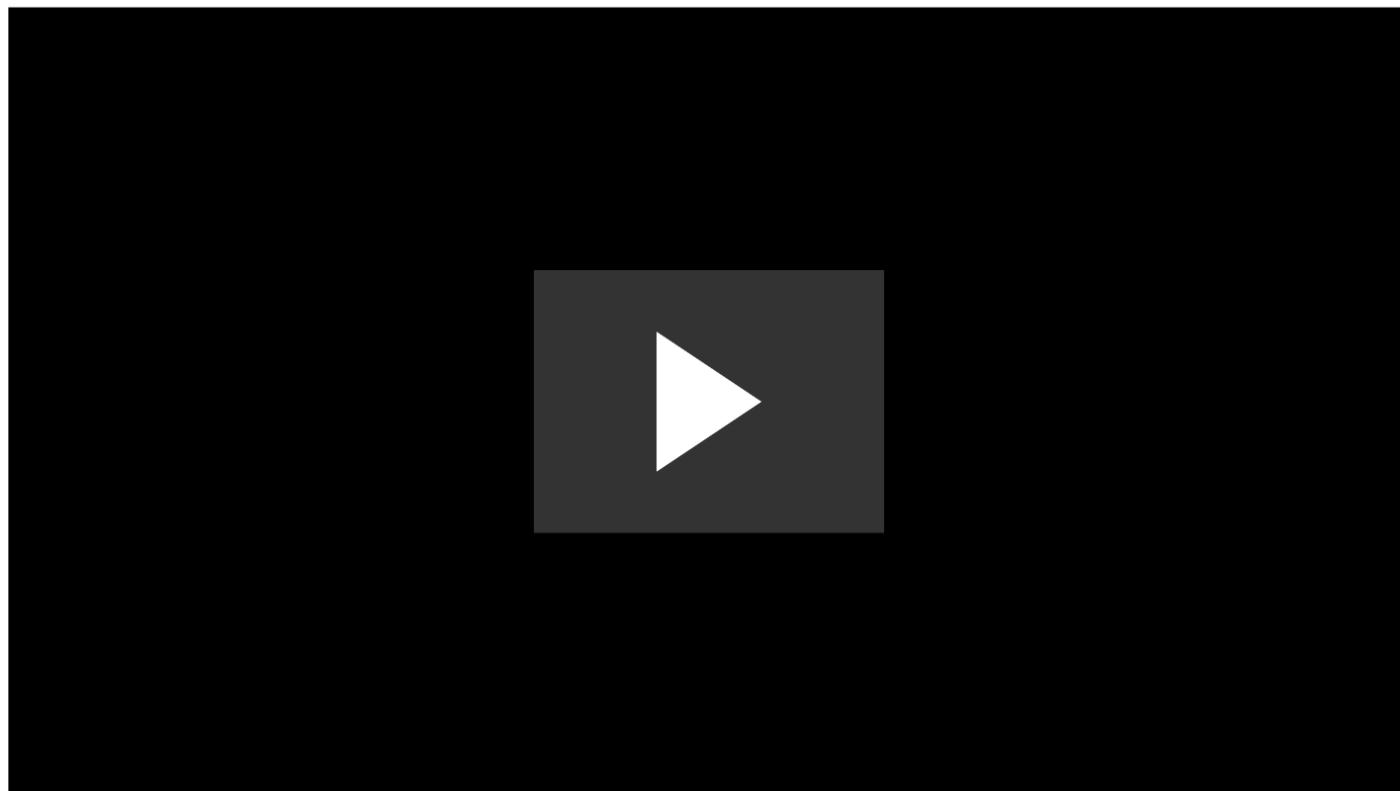
Evidencia 2

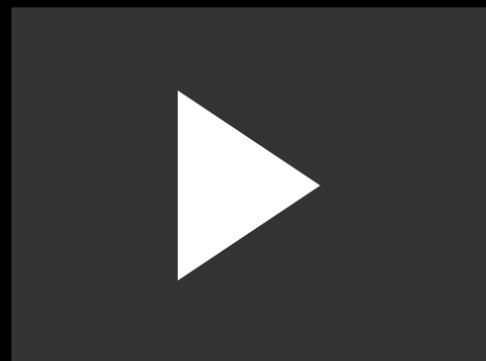
Evidencia 1





Evidencia 2





VART

Vehículo Autónomo para
Registro de Temperaturas



Conclusiones

- El microcontrolador ESP8266 con su modulo wifi incluido, su versatilidad, la amplia documentación en linea, el PWM lo convierten en el micro ideal para desarrollo de skills en arquitectura de microcontroladores.
- El uso del NodeMCU shield, nos ahorró el desarrollo de un módulo independiente de puente H y nos permitió invertir mas tiempo en desarrollo de los requerimientos del proyecto.
- La velocidad de procesamiento de los comandos está limitada por respuestas físicas del vehículo, por lo tanto el código final incluye varios delays para dar tiempo a energizar los motores adecuadamente.
- Dada las características físicas del sistema, es decir, la velocidad de respuesta de los motores y los cambios de temperatura suaves, el sistema actual es perfecto para las necesidades del proyecto.

VART

Vehículo Autónomo para
Registro de Temperaturas

