E



Third period- 5th semester - IRS
November 29, 2024

# Final presentation: Waypoint estimation


Team 3

*Mauricio Zavala Sánchez A00837332*
*Humberto Barrera Martínez A00836271*
*Roberto Raymundo Gómez Vargas A01285451*
*Erick Giovani Campos Aguirre A01247257*


TE2004B Design of advanced embedded systems

M

TE2004B

# Introduction

"Waypoint Estimation" challenge, proposed by John Deere. Our project focuses on designing and building an autonomous scale tractor that leverages the powerful STM32H755 microcontroller with dual-core architecture. This microcontroller enables real-time processing, which is essential for precise waypoint navigation and effective integration of various sensors, including a camera system for enhanced visual data processing.



E

TE2004B

# Our solution

## Objective:

- *Develop an autonomous scale tractor capable of following a predefined set of waypoints with high accuracy.*
- *Integrate sensor data fusion to minimize errors in movement.*
- *Utilize a dual-core STM32H755 microcontroller for efficient task handling and real-time communication.*



## Key Features:

- *Precision navigation with GPS, IMU, and encoders.*
- *Real-time camera-based position tracking.*
- *Seamless communication via CAN, SPI, I2C, and UART.*

TE2004B

# Product description

## Vision

To create an autonomous tractor that can navigate predefined waypoints accurately, improving operational efficiency and reducing human intervention in agricultural tasks.

## Needs

- *Precise navigation: Ability to follow waypoints with minimal error.*
- *Real-time decision-making: Integration of multiple sensors for accurate data processing.*
- *Robust communication: Ensuring continuous and reliable communication between components.*

## How does our product satisfy the needs of the target group?

Our workprovides an efficient, scalable solution for automated fieldwork, minimizing human error and increasing operational efficiency through precise, autonomous navigation.

## Target group

- *Farmers: Looking for efficient automation solutions to improve farming operations.*
- *Agricultural companies: Seeking reliable technology to enhance productivity and reduce labor costs.*
- *John Deere: Partnering in the development of cutting-edge autonomous solutions.*



E

TE2004B

# Software

## IDEs

- *STM32 Cube IDE*
- *Arduino IDE*

## Programming languages

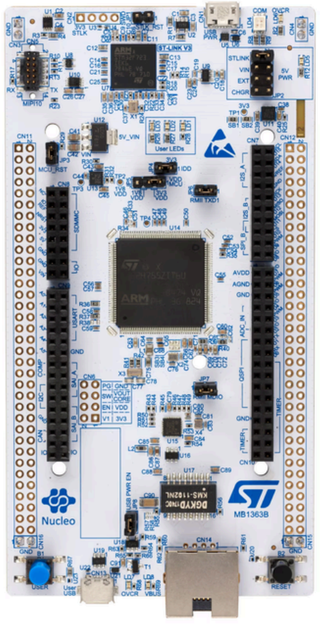
- *C*
- *C++*
- *Arduino*
- *Python*



M

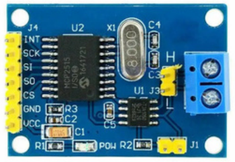
TE2004B

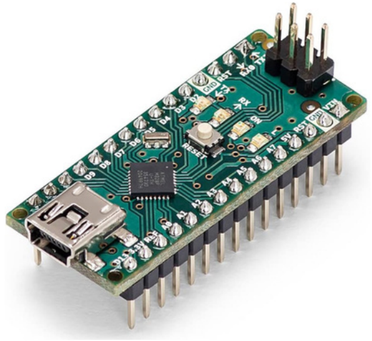
# Hardware

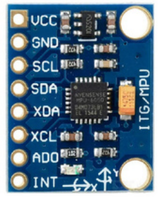




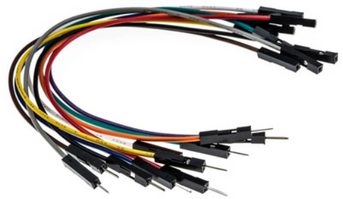




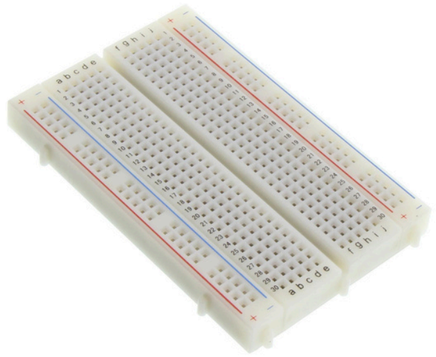








## Microcontrollers

- *STM32H755 dual core module*
- *Arduino Nano*

## Components and sensors

- ESC Module
- nRF24 Wireless module
- TJA1051 CAN transceiver
- MCP2515 CAN Shield
- DC motor with encoder (Powered by a 9 V battery)
- Servo motor
- LM2596 Voltage regulator
- LPS John Deere Camera
- 3.7 V LiPo Battery
- MPU6050 IMU Module
- 9 V Battery
- Breadboards
- Jumper wires
- 4 AA Batteries (6 V)

TE2004B

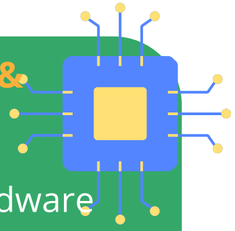
# Development stages

## Stage 1: Conceptualization & Requirements Gathering



Defining system requirements and project scope.

## Stage 2: Hardware Design & Microcontroller Setup



Selecting and integrating hardware components (STM32H755, sensors, camera).

## Stage 3: Software Development & Sensor Integration



Developing firmware for sensor fusion and communication protocols.

## Stage 4: System Testing & Optimization



Running test scenarios to ensure accuracy and system stability.

## Stage 5: Final Evaluation & Documentation

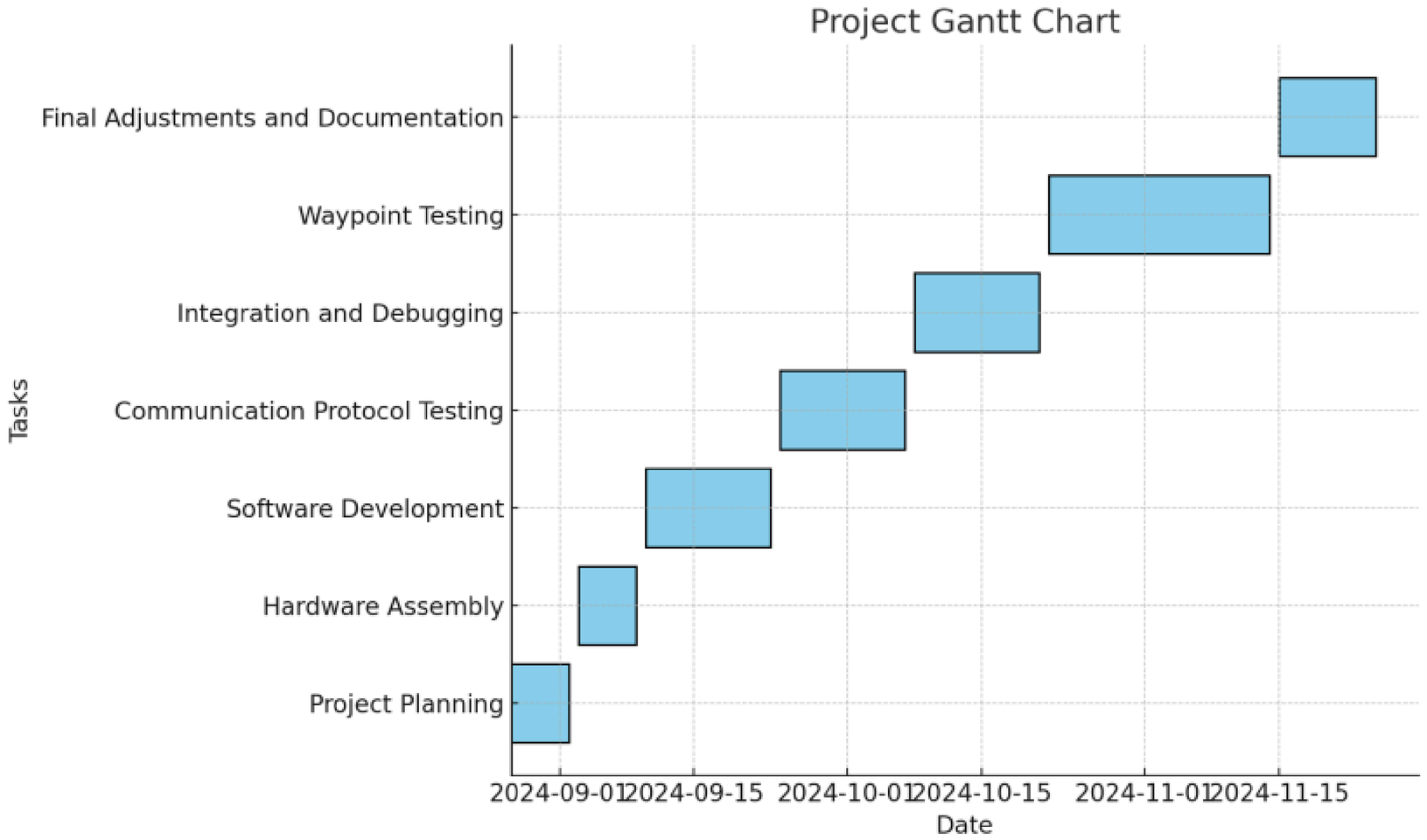


Comprehensive testing of the final system and preparing the project report.

E

TE2004B

# Gantt diagram



G

TE2004B

# Communication protocols

**CAN:** For communication between the tractor's core systems and peripheral devices.

**I2C:** I2C: For communication between low-speed devices like encoders.

**SPI:** For high-speed communication with sensors.

**UART:** For real-time communication with the camera and external devices.

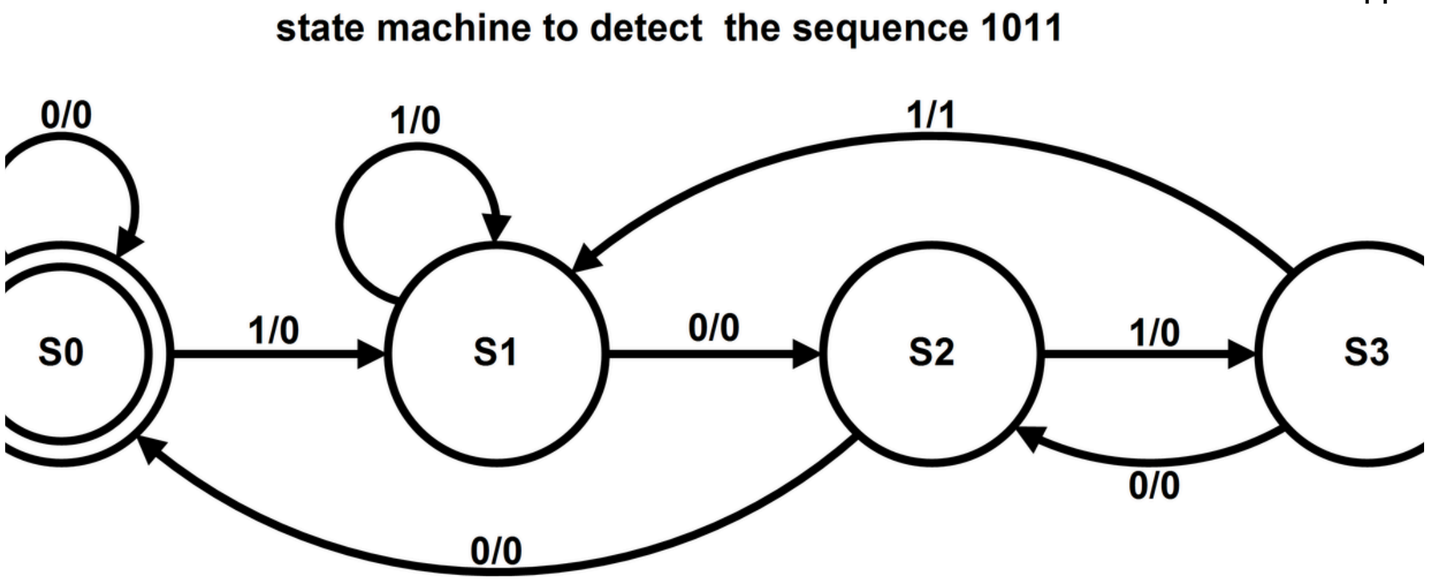
## Importance

- *Ensures reliable, low-latency data transfer across the system.*
- *Enhances synchronization between sensors, cameras, and navigation systems.*



TE2004B

# State Machines and task handling




*Why State Machines?*

- *Avoided locking the CPU during task execution.*
- *Allowed smooth multitasking by switching between states efficiently.*
- *Added robustness and scalability for handling multiple tasks.*

- *System Tick Handler Modification:*
- *Incremented a variable in the SysTick handler.*
- *Used this variable to manage periodic tasks without busy waiting.*
- *State Machines Implemented:*
- *Route Navigation State Machine:*
  - *States:*
    i. *Initial Waypoint: Starting from the first waypoint.*
    ii. *Starting the Curve: Adjusting direction for the U-like curve.*
    iii. *Finishing the Curve: Completing the curved segment.*
    iv. *Final Straight Segment: Approaching the final waypoint.*
  - *Controlled the DC motor and servo motor using PWM signals.*
- *Waypoint Notification and Update State Machine:*
  - *Notified and updated when a waypoint was reached.*
  - *Updated the system to move to the next waypoint.*

TE2004B

# State Machines and task handling

```
uint32_t PathHardCodeUpdateState(uint64_t cntMilisec) { // run periodically from main loop
switch (CurrState) {
   case START:
      stopCar();
      setServoAngle(AngleFwd);    // Set the target steering angle
      CurrState = PathFwd1;
      prevTim = cntMilisec;
      stateId = 1;
      setServoAngle(AngleFwd);    // Set the target steering angle
      setEscSpeed(SpdFwd1);       // Low forward speed
      break;

   case PathFwd1:

      if((cntMilisec - prevTim) >= intFwd) {
         prevTim = cntMilisec;
         CurrState = PathCurv;
         stateId = 2;
         setServoAngle(AngleCurv);    // Target steering angle for curv
         setEscSpeed(SpdCurv);        // Low Curv speed
      }
      else {
         CurrState = PathFwd1;
      }
      break;
```

TE2004B

# State Machines and task handling

```
case PathCurv:
   if((cntMilisec - prevTim) >= intCurv) {
      prevTim = cntMilisec;
      CurrState = PathFwd2;
      stateId = 3;
      setServoAngle(AngleFwd);    // Set the target steering angle
      setEscSpeed(SpdFwd1);       // Low forward speed
   }
   else {
      CurrState = PathCurv;
   }
   break;

case PathFwd2:


   if((cntMilisec - prevTim) >= intFwd1) {
      prevTim = cntMilisec;
      CurrState = OFF;
      stateId = 4;
      setServoAngle(AngleFwd);    // Set the target steering angle
      setEscSpeed(SpdFwd1);
      stopCar();
   }
   else {
      CurrState = PathFwd2;
   }
   break;
```

TE2004B

# State Machines and task handling

```
    case OFF:
        stateId = 4;
        CurrState = OFF;
        break;

    default: // Error: invalid state
        stateId = 5;
        CurrState = OFF;
        setServoAngle(AngleFwd);   // Set the target steering angle
        stopCar();
        break;
     }

    return(stateId);
}
```

TE2004B

# Multithreading and core efficiency



## Multithreading:

Utilizes the dual-core STM32H755 microcontroller to handle multiple tasks simultaneously (e.g., sensor data collection, communication, navigation algorithms).

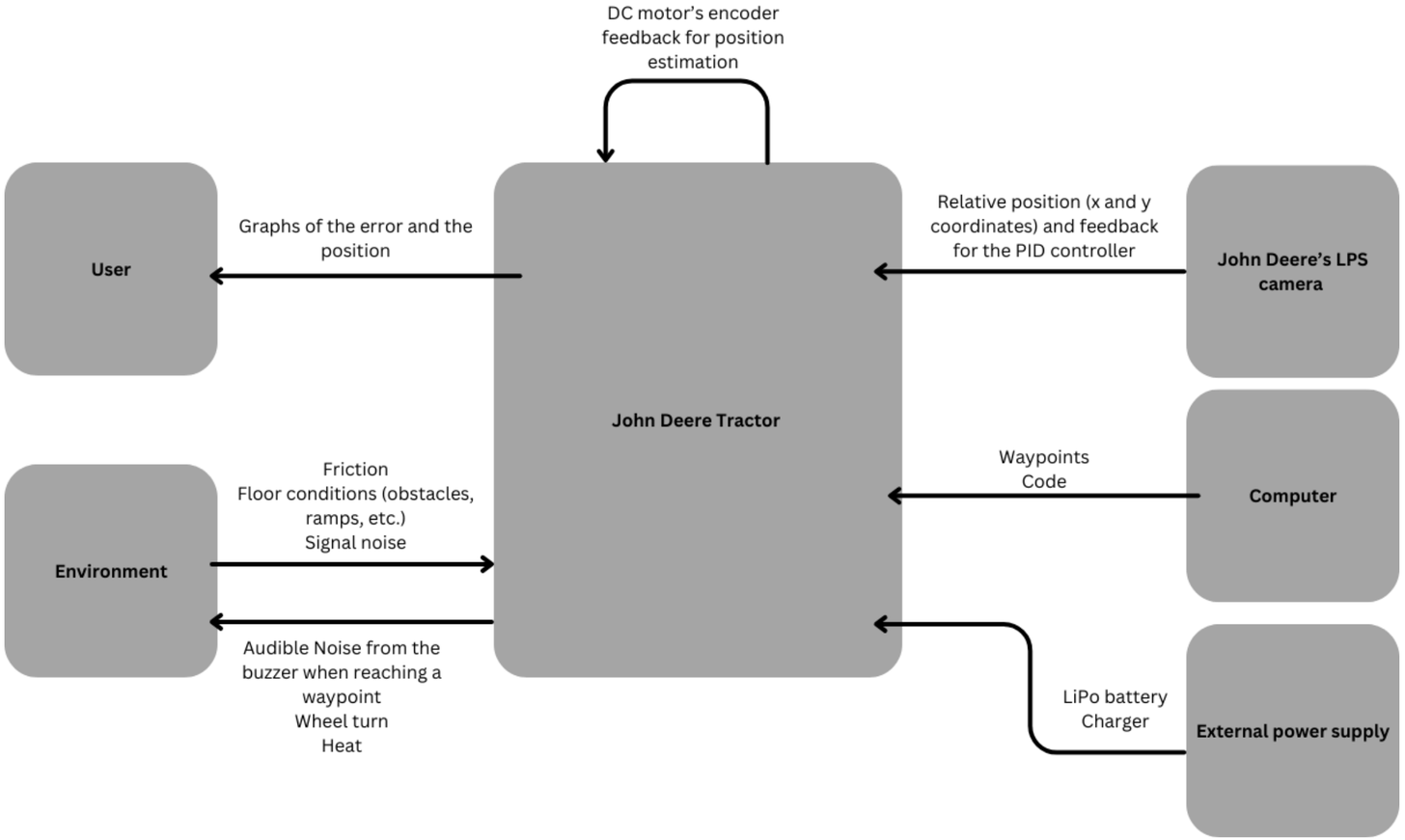
## Core Efficiency:

- Optimizing task distribution across the two cores to maintain high processing efficiency and ensure real-time performance.
- Minimizes processing delays and maximizes system responsiveness during critical operations like waypoint tracking.
- We chose to control the LED that gets toggled when crossing a waypoint inside the M4, while other tasks are handled by the M7.

TE2004B

# UML Context diagram





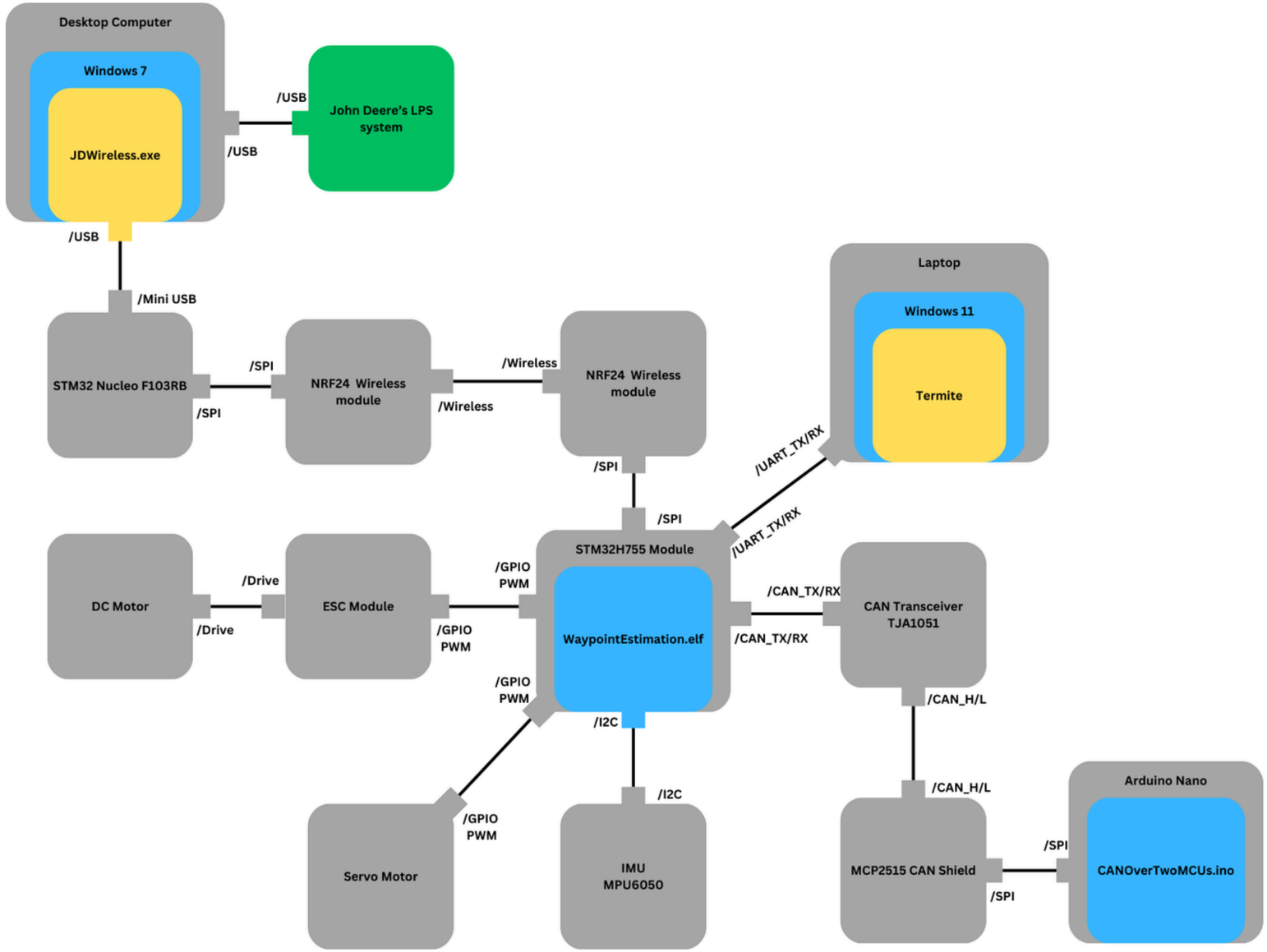
TE2004B

# Use Case diagram



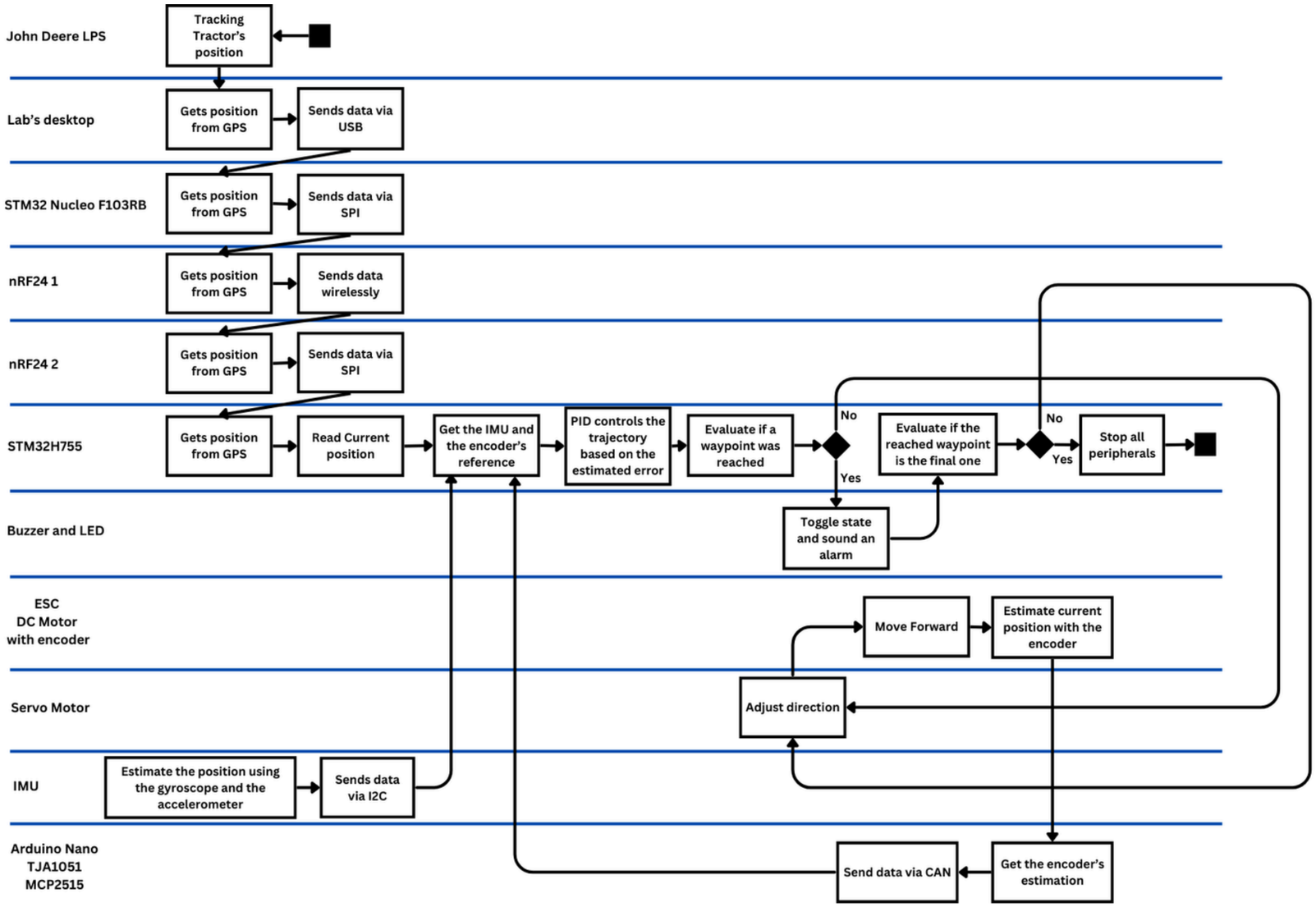
| Name | John Deere tractor with optimized waypoint following system |
|---|---|
| Purpose | To autonomously follow a pre-defined route across a field, minimizing human intervention, and ensuring precise navigation to maximize efficiency in agricultural operations. |
| Inputs | GPS data from Deere device, encoder sensor data, accelerometer and gyroscope data (IMU), CAN bus messages (J1939 protocol), power from rechargeable battery, user-defined waypoints. |
| Outputs | Real-time position tracking, deviation alerts if off-route, autonomous navigation through waypoints, wireless transmission of position, audible indicators when reaching waypoints, and sensor data, route generation and adherence. |
| Functions | • R.1 Receive data from the camera via SPI<br>  ◦ R.1.1 Interface the STM and the camera wirelessly.<br>  ◦ R.1.2 Receive the data packet<br>  ◦ R.1.3 Obtain the position data in coordinate format<br>  ◦ R.1.4 Verify if a waypoint has been reached<br>• R.2 Estimate the position using the encoder via CAN<br>  ◦ R.2.1 Interface the DC Motor to the Arduino Nano and pass the encoder's info to the STM.<br>  ◦ R.2.2 Read the packet<br>  ◦ R.2.3 Interpret the data and adjust route accordingly.<br>• R.3. Check the route's precision via I2C<br>  ◦ R.3.1 Interface the IMU with the STM<br>  ◦ R.3.2 Read the data packet and interpret it.<br>  ◦ R.3.3 Adjust the route accordingly.<br>• R.4 Interpret and follow the waypoints of the code<br>  ◦ R.4.1 Interface the DC and servo motors with the STM<br>  ◦ R.4.2 Power the motors of the tractor via the PWM pins.<br>  ◦ R.4.3 Calibrate the distance and the angles needed to navigate from the current position to the next waypoint.<br>• R.5 Manage and collect environmental data<br>  ◦ R.5.1 Estimate the relative position and, with a reference point obtained from the camera and the sensors, estimate the approximate error.<br>  ◦ R.5.2 Once verified that the tractor reached a waypoint, sound a buzzer and turn a LED on.<br>  ◦ R.5.3 Take the data and prepare to send it to the user to generate real time updating graphs about the route.<br>• R.6 Send data to the user<br>  ◦ R.6.1 Make a packet with the data and send it via UART<br>  ◦ R.6.2 Collect the packet and create graphs of the error and the position for the user.<br>• R.7 Feedback system to improve precision<br>  ◦ R.7.1 Take the error calculation and adjust the PID system to reduce the relative error of the tractor's positioning. |
| Performance | The system operates in real-time, with:<br>• Control response: <50ms latency for motors and actuators.<br>• Data processing: Sensor data (GPS, IMU, encoders) processed in <100ms for continuous position estimation.<br>• Route updates: Processed within 200ms for real-time adjustments<br>• Wireless transmission: Position updates sent with <500ms delay for accurate tracking. |
| Manufacturing cost | • STM32H755 microcontroller: $820 MXN<br>• IMU (accelerometer and gyroscope): $46 MXN<br>• 2 nRF24L01 wireless modules: $352 MXN<br>• Arduino Nano: $200 MXN<br>• TJA1051 CAN Transceiver: $305 MXN<br>• MCP2515 CAN shield: $290 MXN<br>• ESC module: $300 MXN<br>• STM32 Nucleo F103RB: $548 MXN<br>• DC motor with encoder: $250 MXN<br>• Servo motor: $170 MXN<br>• Encoder (Arduino UNO): $490 MXN<br>• Rechargeable battery: $70 MXN<br>• Deere GPS system (SPI camera): $692 MXN<br>• RC vehicle kit: $800 MXN<br>  ◦ Total: $5333 MXN |
| Power | Powered by a rechargeable LiPo battery system of 6V, sufficient to operate the tractor's motors, sensors, and communication modules during its operational hours. |
| Size/Weight | • Length: 23.6 cm<br>• Height: 5.9 cm<br>• Width: 14.8 cm<br>• 0.8 Kg (fully assembled) |

# Deployment diagram



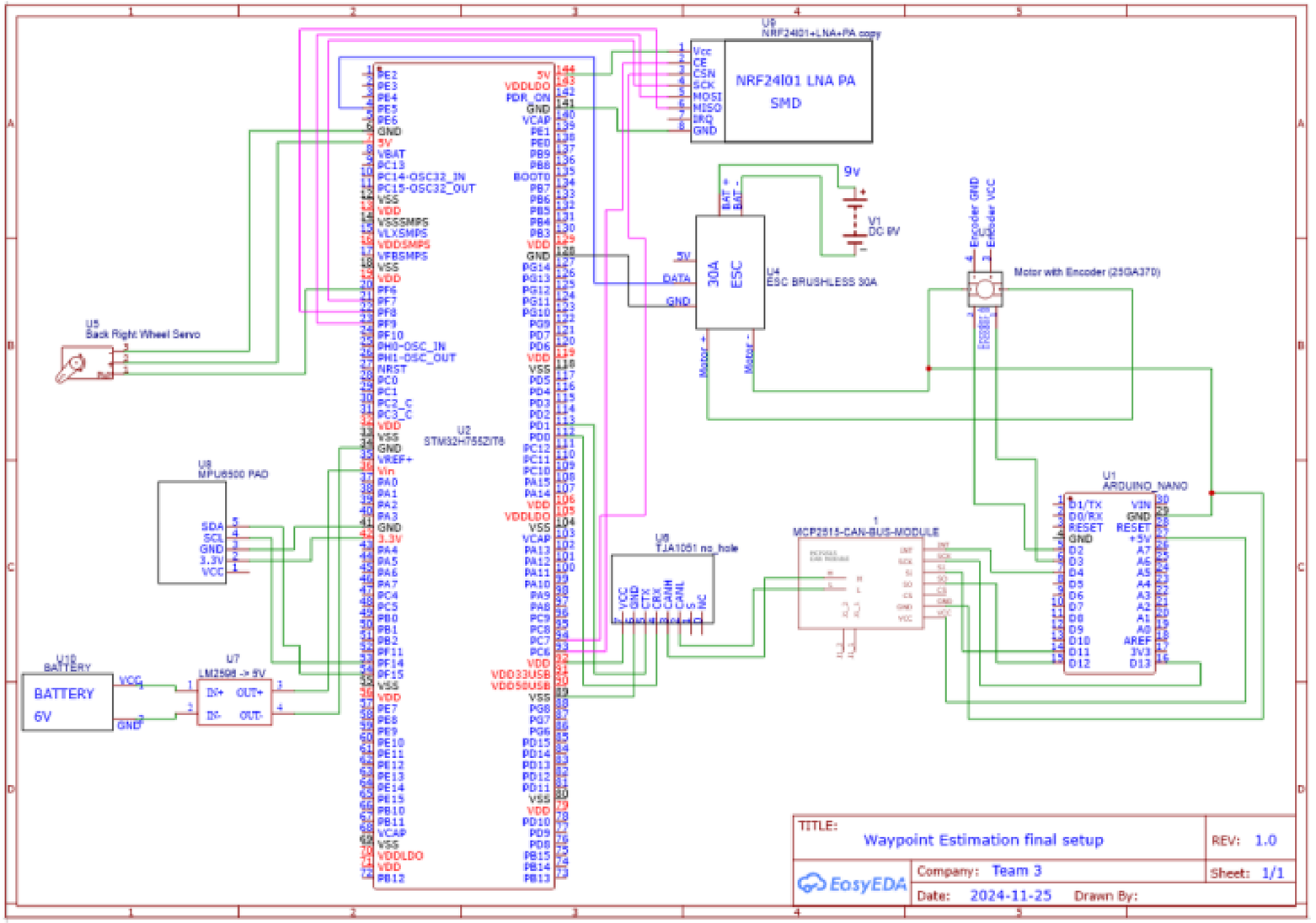
TE2004B

# Activity diagram

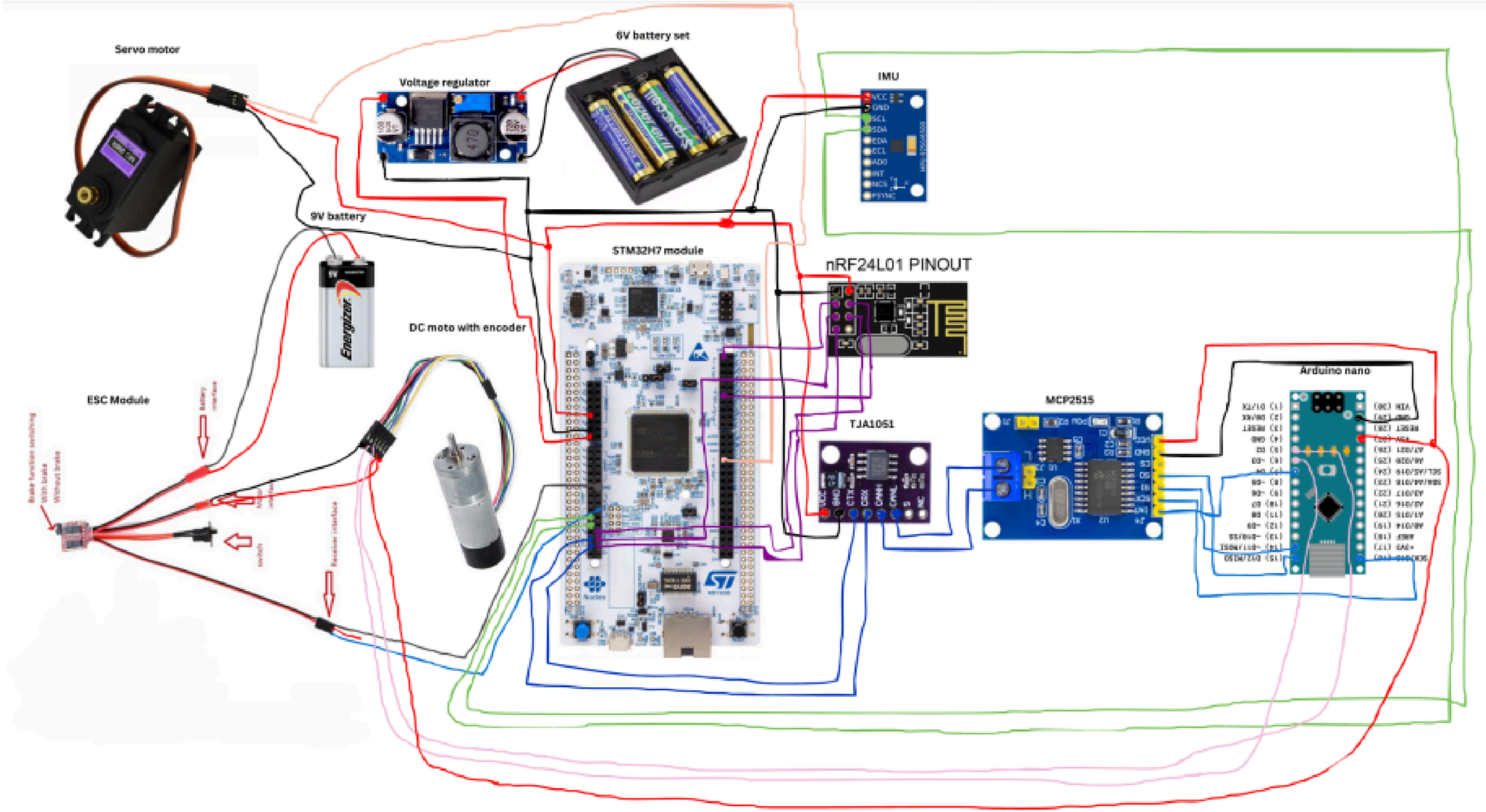


TE2004B

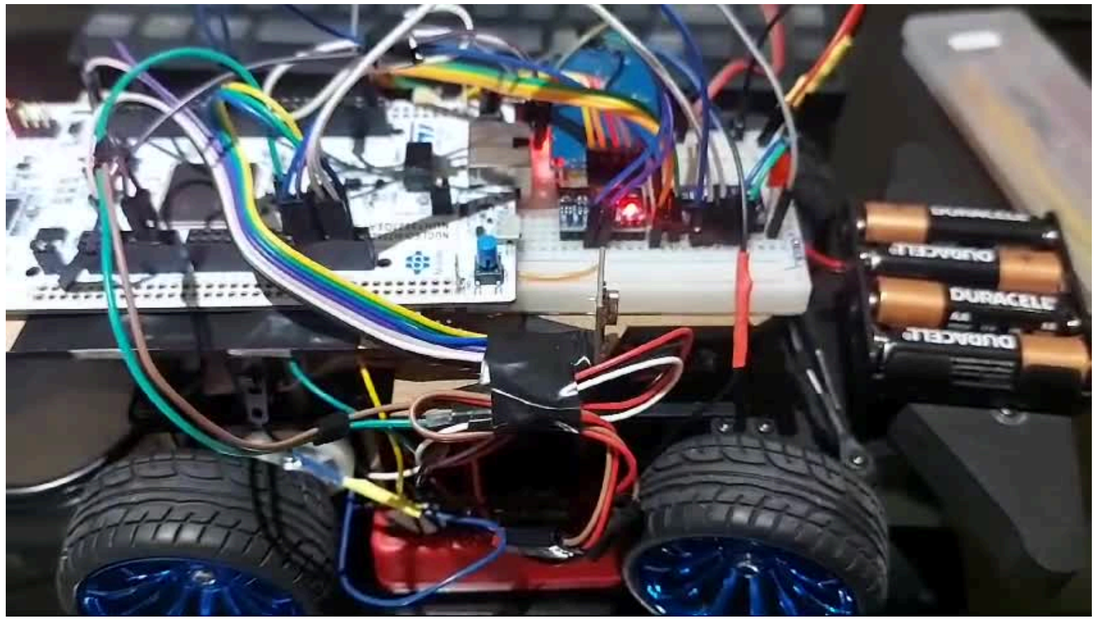
# Schematic

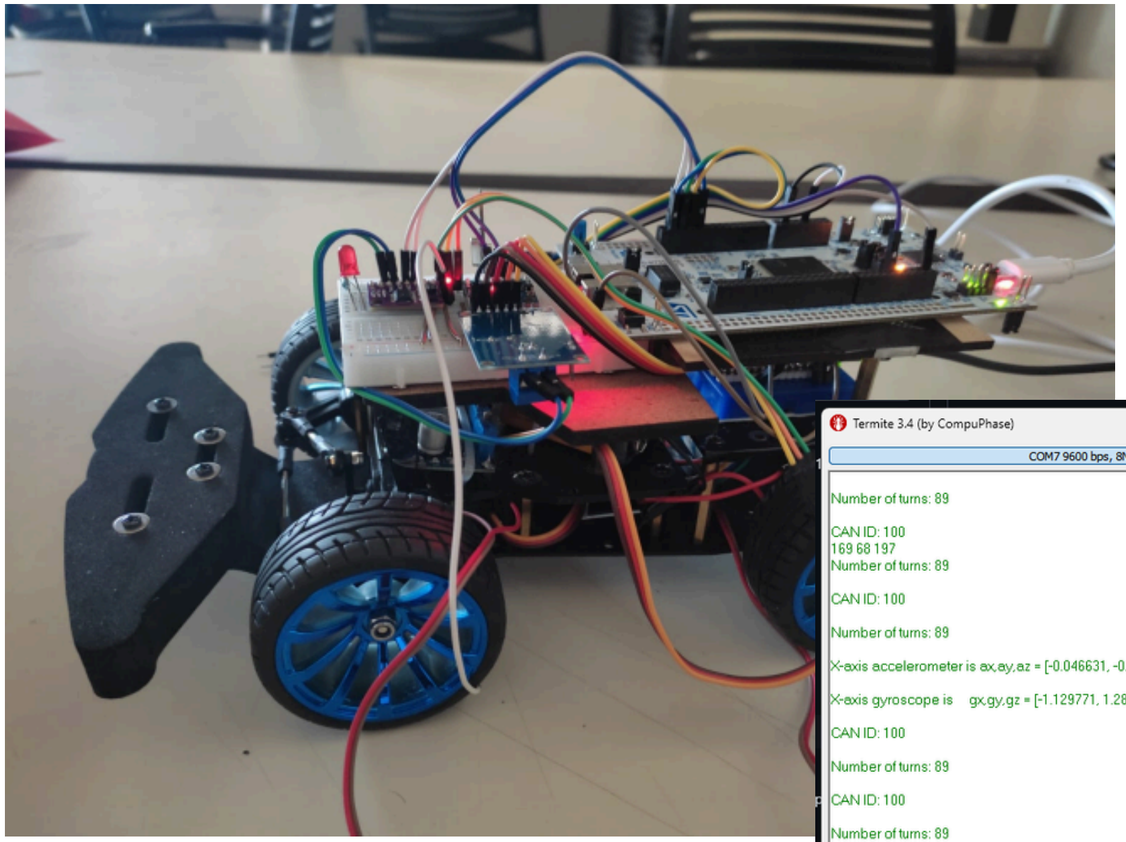


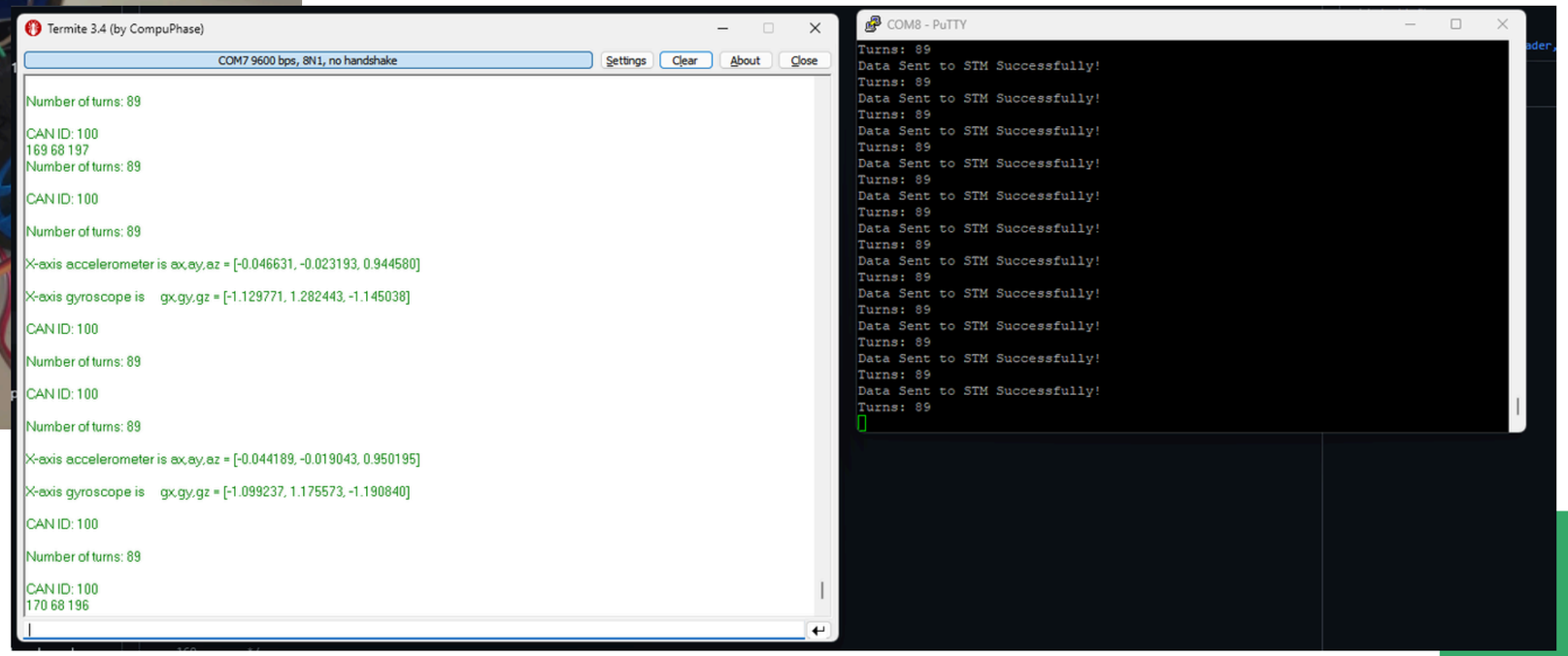
# Connections diagram



# Final product





# Testing methodology

## Videos of the development process

Videos from Lab Practices (Subsystem Testing)



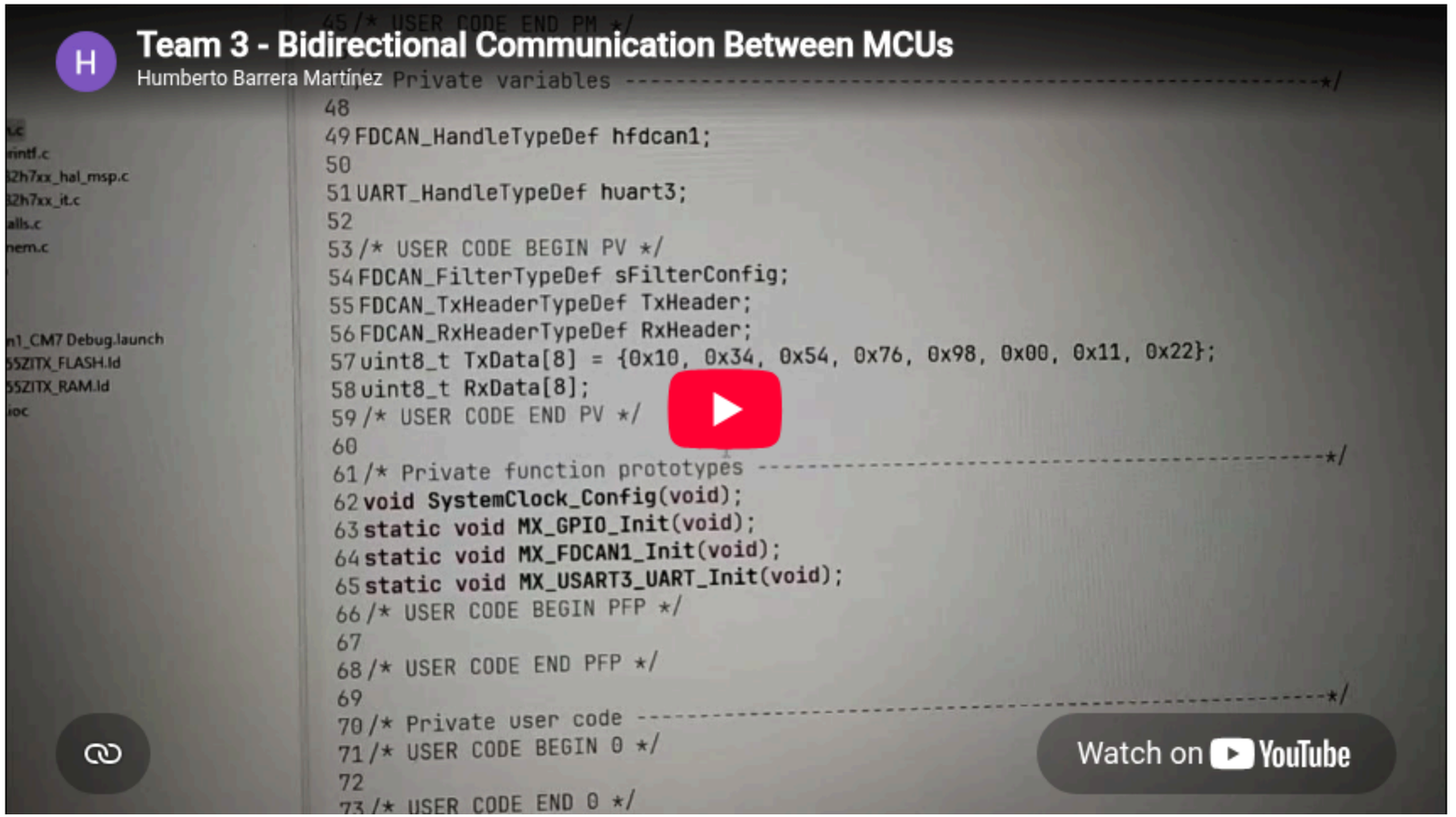

**CAN with a Toyota cluster:** https://www.youtube.com/watch?v=H7y588eh7nY&feature=youtu.be

TE2004B

# Testing methodology

## Videos of the development process

Videos from Lab Practices (Subsystem Testing)




**CAN communication between MCUs:** https://www.youtube.com/watch?v=YBVp3XSuf9o&feature=youtu.be

TE2004B

# Testing methodology

## Videos of the development process

Videos from Lab Practices (Subsystem Testing)

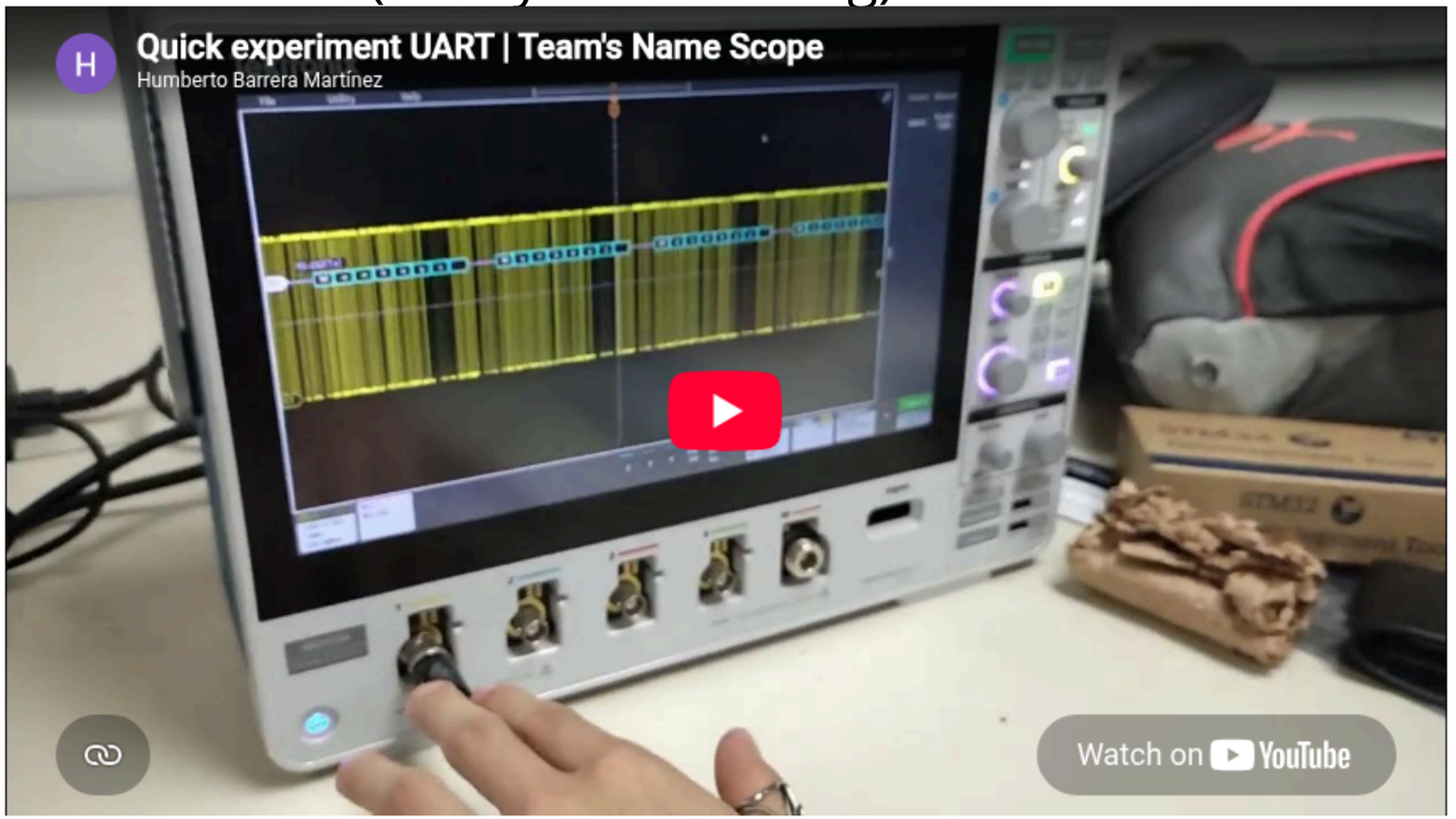



**ESC module testing:** https://www.youtube.com/watch?v=nm80LHiTjtk&feature=youtu.be

# Testing methodology

## Videos of the development process

Videos from Lab Practices (Subsystem Testing)

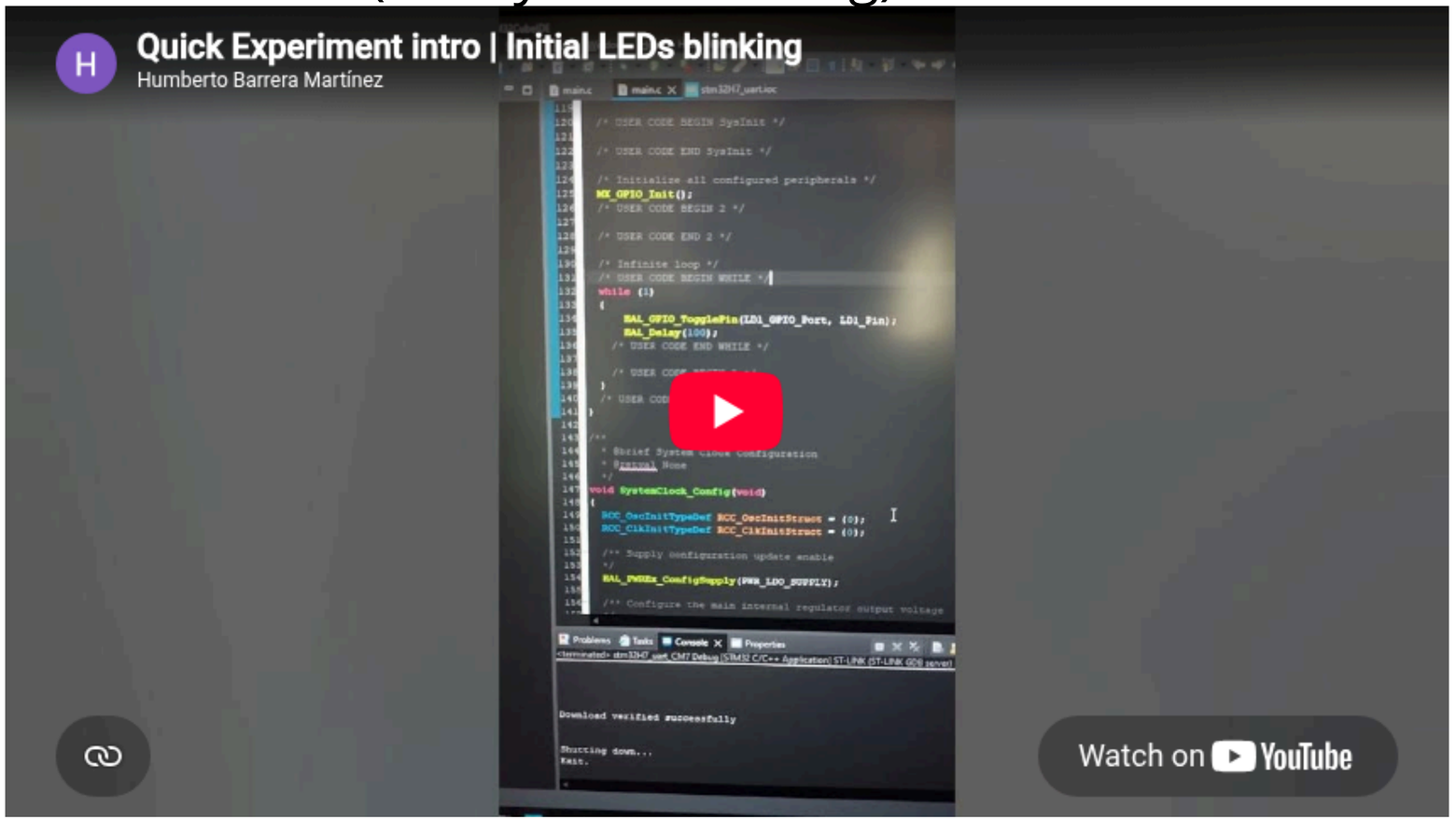



**UART communication testing:** https://www.youtube.com/watch?v=dgoM10nOsPw&feature=youtu.be

# Testing methodology

## Videos of the development process

Videos from Lab Practices (Subsystem Testing)

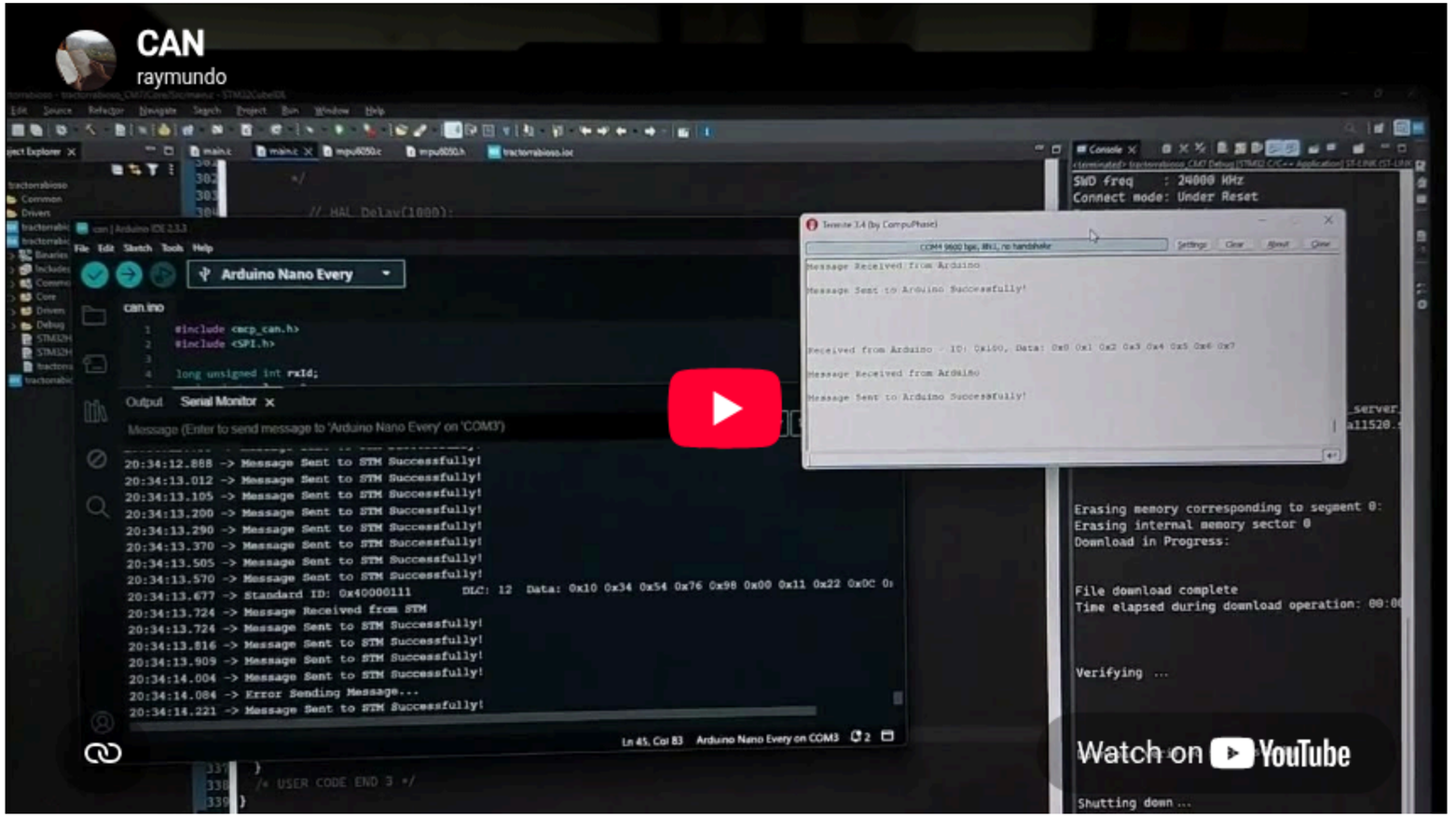



**Initial LED testing:** https://www.youtube.com/watch?v=tTxDf1LHlo8

# Testing methodology

## Videos of the development process

Videos from the Final Integrated System

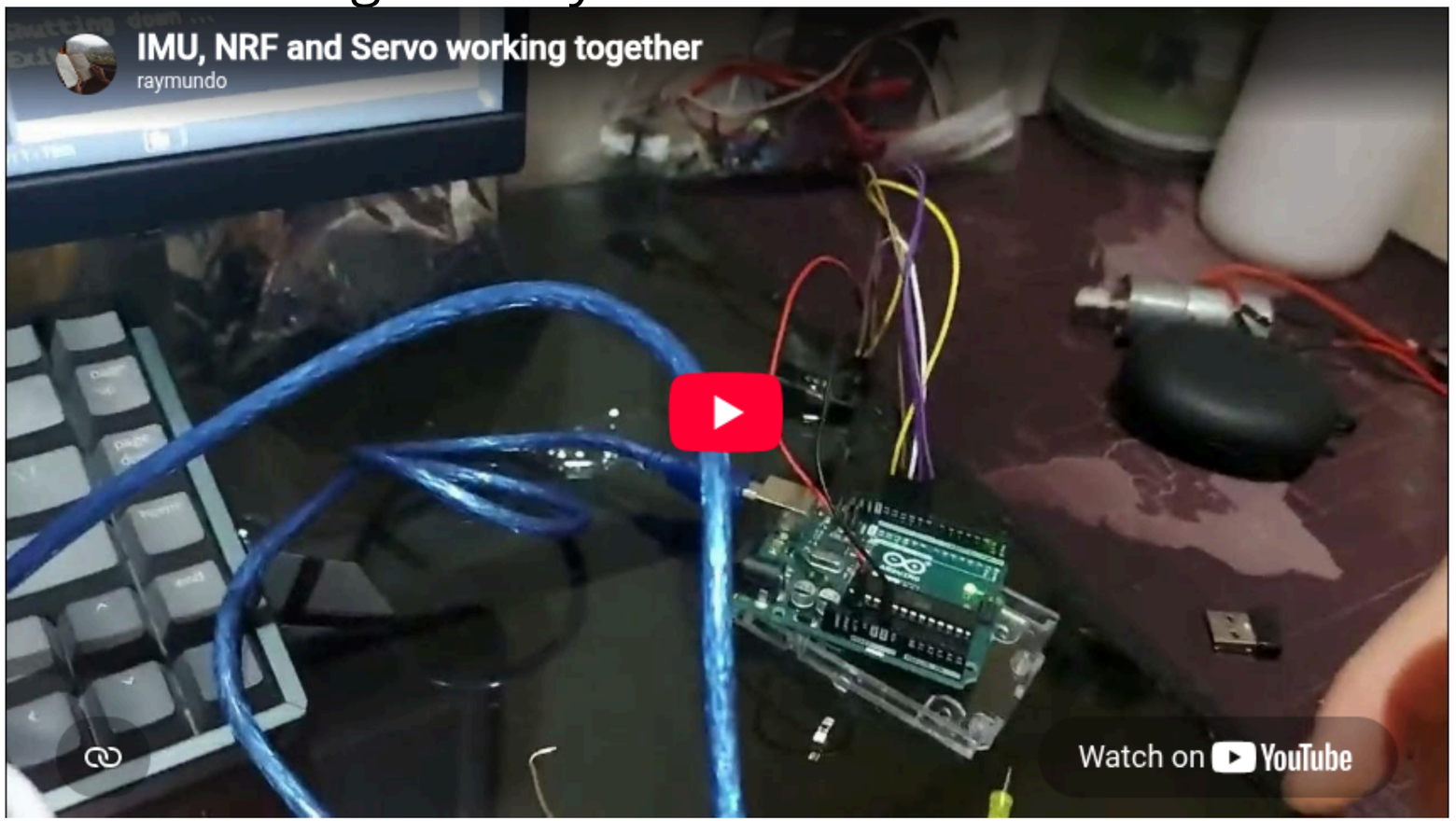



**InitiCAN communication with the integrated systemal LED testing:** https://www.youtube.com/watch?v=Ph2VxUwqdxw

# Testing methodology

## Videos of the development process

Videos from the Final Integrated System



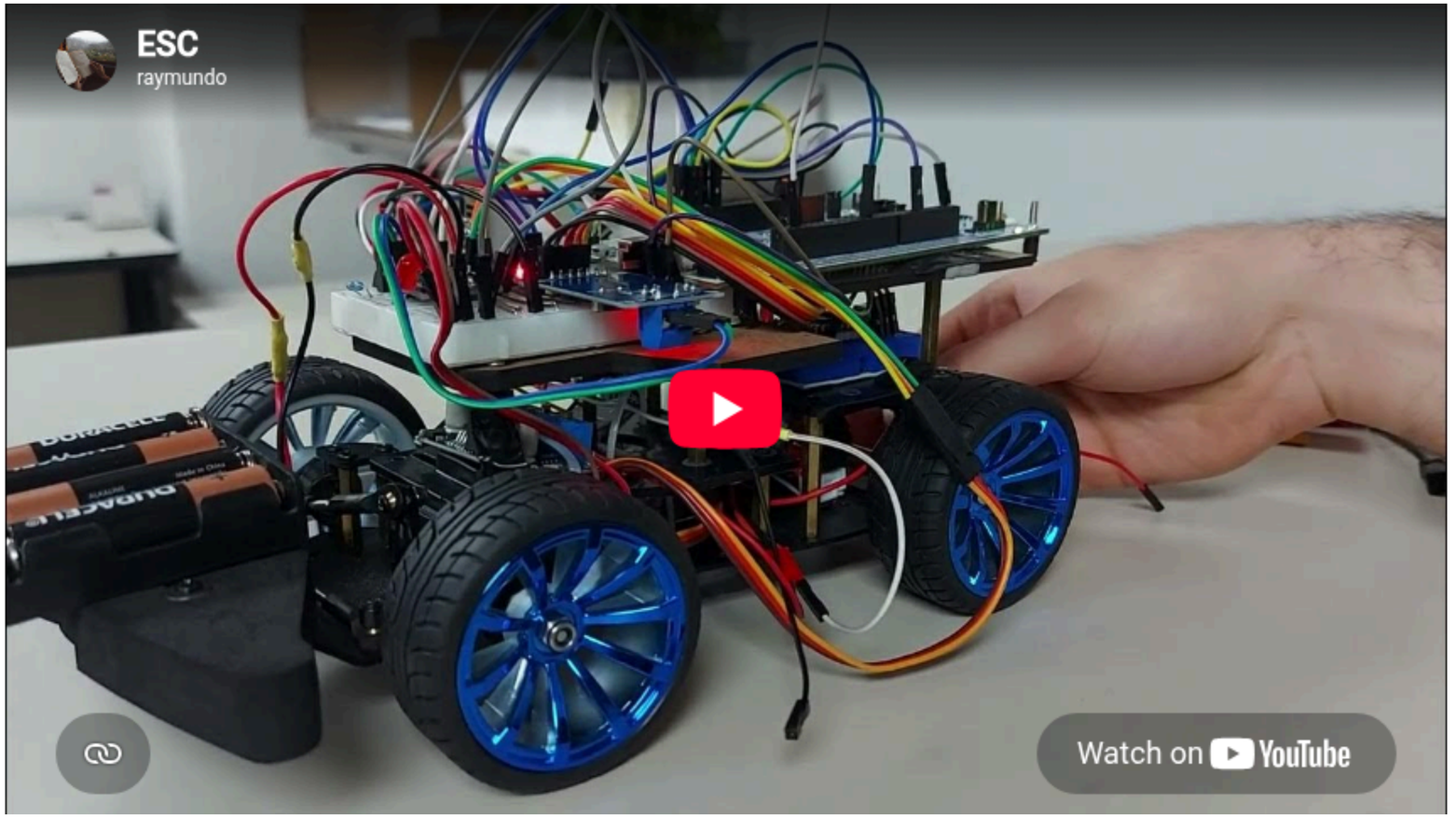

**IMU, NRF, and Servo working together:** https://www.youtube.com/watch?v=EdVwturkIk4

E

# Testing methodology

## Videos of the development process

Videos from the Final Integrated System




**ESC testing with the integrated system:** https://www.youtube.com/watch?v=u44DFOzECO4

M

# Testing methodology

## Videos of the development process

Videos from the Final Integrated System

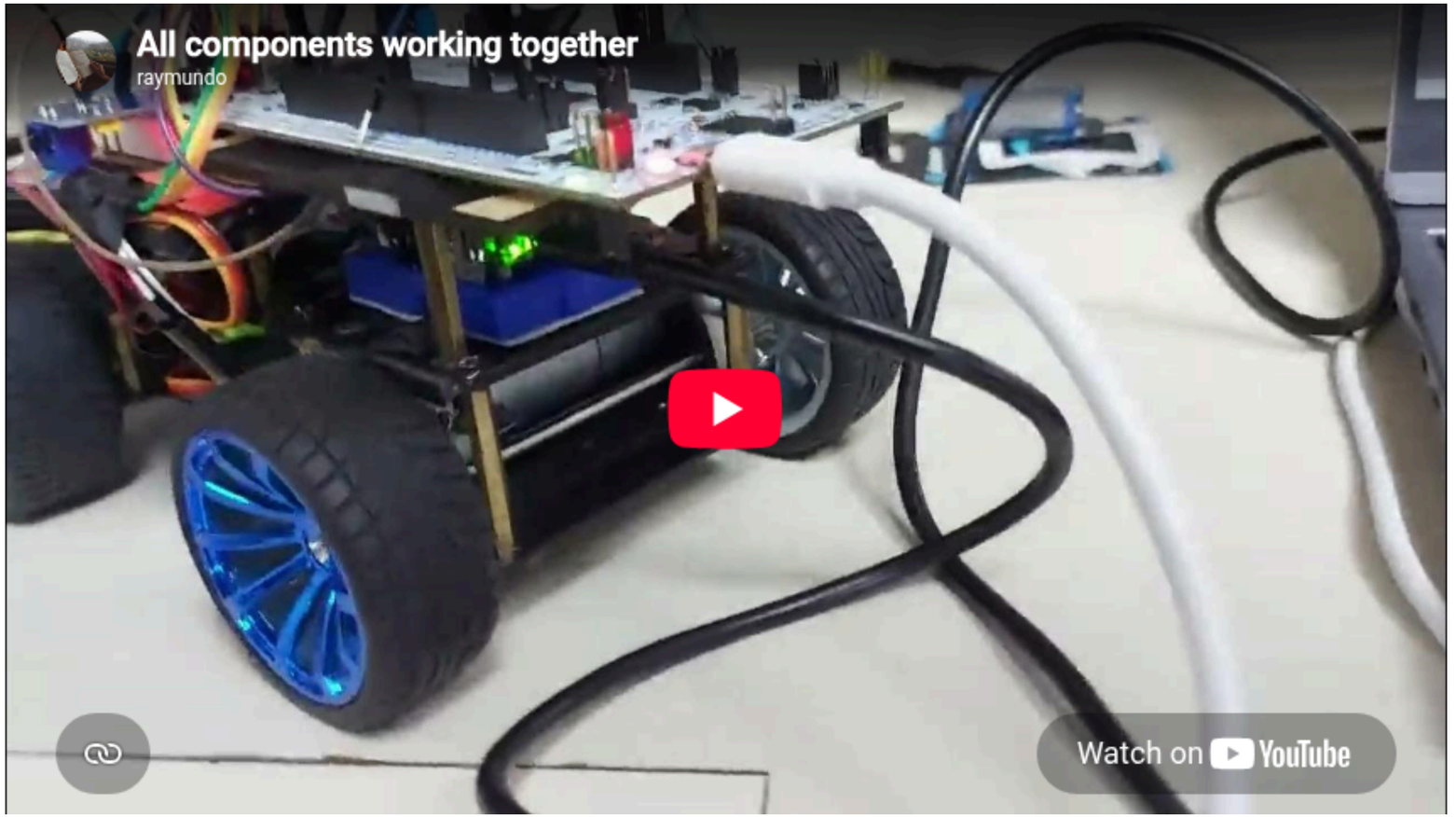



**DC motor’s encoder communicating via CAN:** https://www.youtube.com/watch?v=IM4Gi6i8Pac

# Testing methodology

## Videos of the development process

All Components Working Together



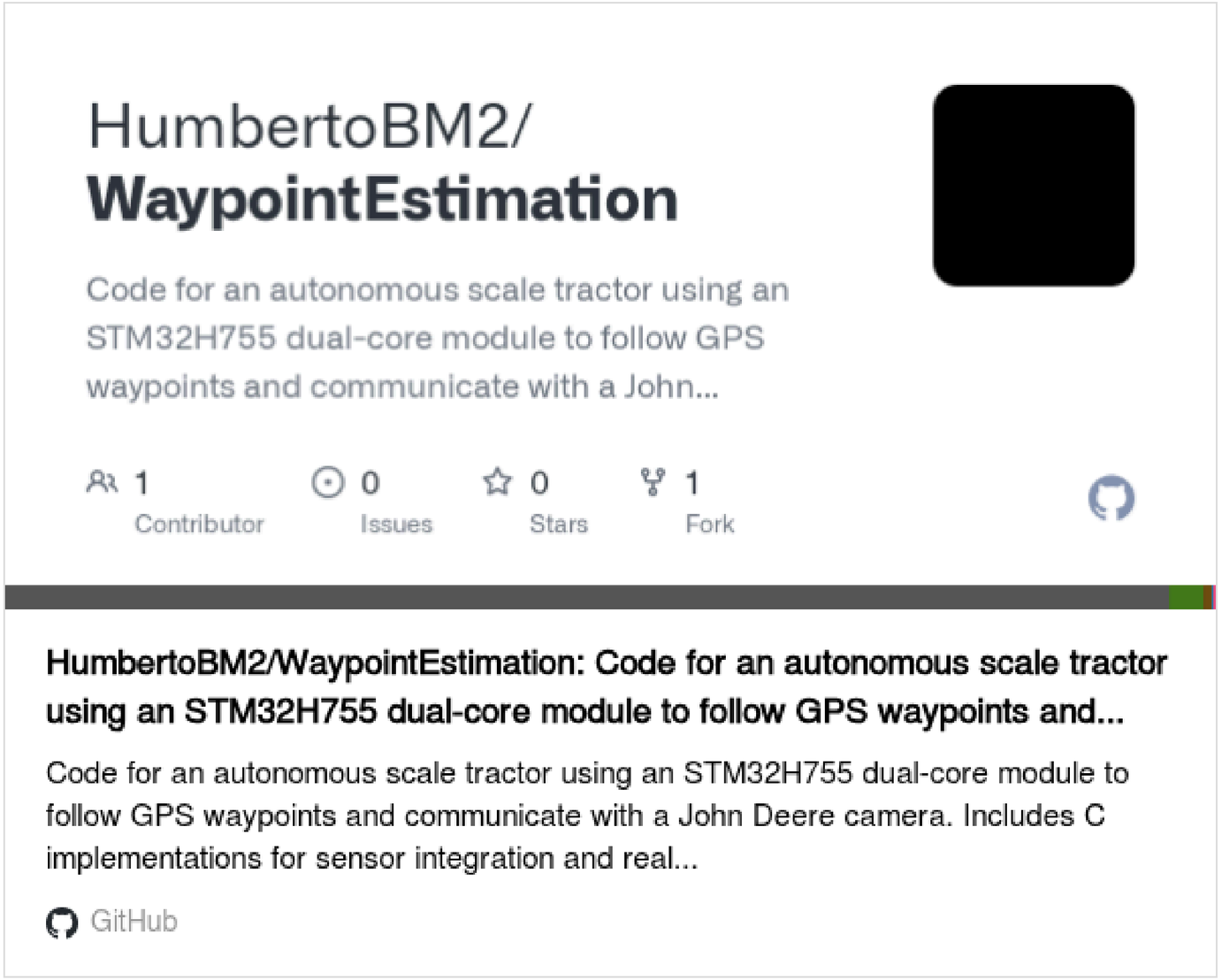

**Final Working State:** https://www.youtube.com/watch?v=7yVdGDu999Y

H

TE2004B

# GitHub repo and codes

The team prepared a GitHub repo in which we stored all the codes used for the project as well as its respective documentation.



M

TE2004B

# Waypoint Demo and videos

## Waypoints

*0,120*
*86,143*
*86,0*
*0,0*

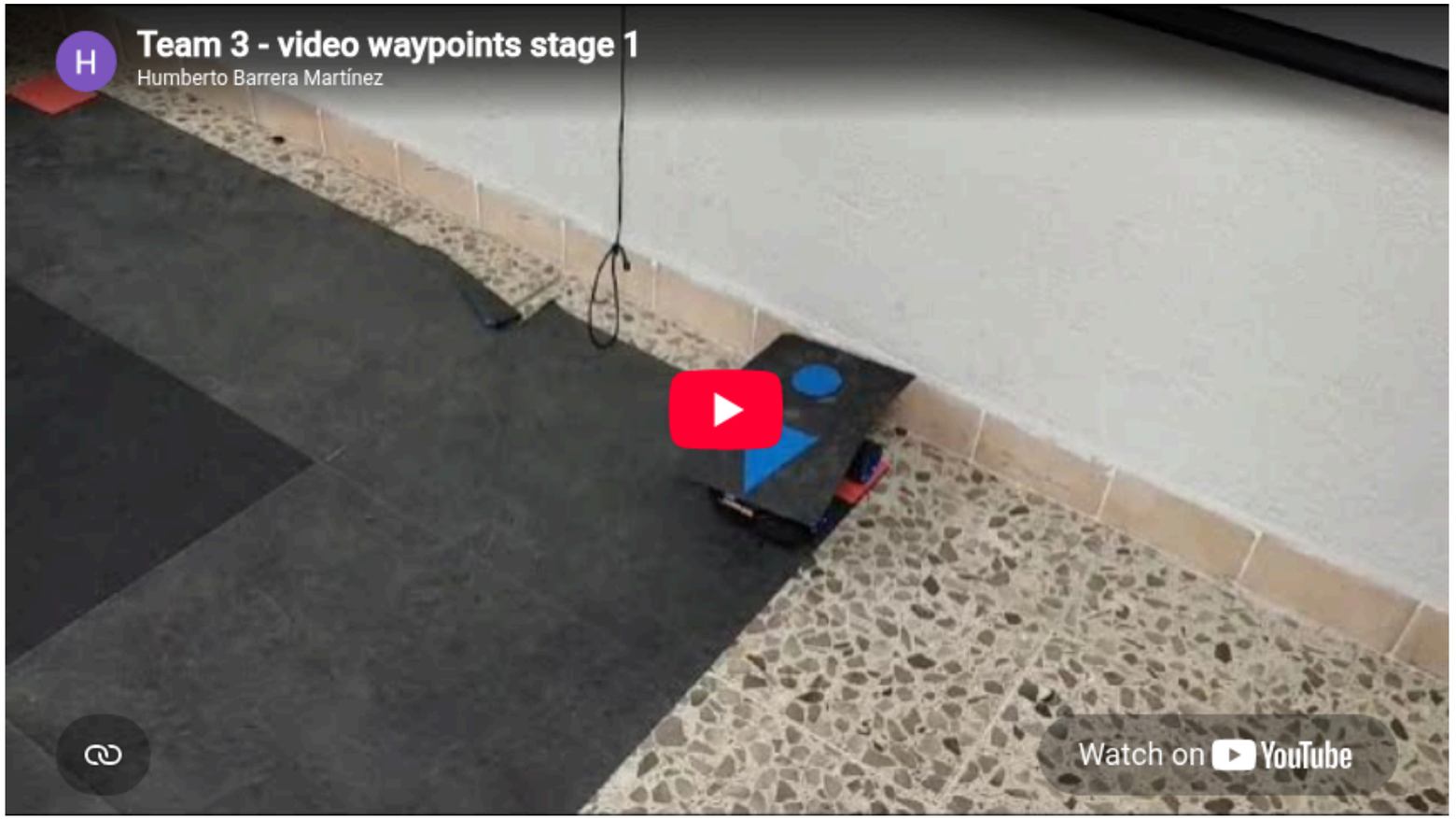
| | Testing | | | |
|---|---|---|---|---|
| **Team** | Monkeys | | | |
| **Res onsible** | Alfonso Avila | | | |
| **Description** | The tractor incorporates all the comunication protocols (I2C, CAN, UART, SPI, WIRELESS) properly for the navigation route. | | | |
| **Function** | Ensure that all systems communicate correctly for the estimation and navigation between waypoints and the tractor's route. | | | |
| **Components** | STM32H755 dual core module, Arduino Nano, ESC Module, nRF24 Wireless module, TJA1051 CAN transceiver, MCP2515 CAN Shield, DC motor with encoder, LM2596 Voltage regulator, LPS John Deere Camera, MPU6050 IMU Module | | | |
| **Hardware** | DC motor with encoder (Powered by a 9 V battery), Servo motor, 3.7 V LiPo Battery, Breadboards, Jumper wires, 4 AA Batteries (6 V) | | | |
| **Procedure** | **Initial conditions** | **Expected behavior with acceptance criteria** | **Actual behavior** | **Result passed or not** |
| **STM32H755 dual core module** | Powered on, connected to GPS and IMU, and set to receive data via SPI and I2C. Initialized with waypoint data and encoder reference values. | Receives GPS and IMU data, processes the current position, and evaluates waypoints. | Gets position form GPS, sends data via USB and Sends data via USB. Reads current position, references from IMU and the encoder, evaluates if the waypoints are reached and tests if the reached waypoint is the final one. | ok |
| **Arduino Nano** | Connected to CAN Shield and TJA1051 CAN transceiver. Initialized to communicate with peripherals and send/receive CAN data. | Sends CAN data for motor and peripheral control. | Sends data via CAN and gets the enconder's estimation. | ok |
| **ESC Module** | Connected to STM32 and DC motor. Powered by a 9V battery and initialized to adjust motor speed based on trajectory inputs from STM32. | Controls motor speed according to trajectory adjustments via STM32. | Makes the motor move forward and estimates the current position with the encoder. | ok |
| **nRF24 Wireless module** | Connected to STM32 and initialized to wirelessly transmit and receive GPS position data. | Wirelessly transmits and receives GPS and position data. | Gets position form GPS and sends data wirelessly and via SPI. | ok |
| **TJA1051 CAN transceiver** | Connected to Arduino Nano and MCP2515 CAN Shield. Initialized to facilitate CAN communication for data exchange. | Facilitates CAN communication for motor control and data exchange. | Transceives data via CAN and gets the enconder's estimation. | ok |
| **MCP2515 CAN Shield** | Powered on and connected to Arduino Nano and TJA1051 CAN transceiver. Ready to transmit/receive additional CAN data for external modules. | Provides additional CAN interface for data exchange. | Sends data via CAN and gets the enconder's estimation. | ok |
| **DC motor with encoder** | Connected to ESC module, powered by a 9V battery, and encoder initialized to estimate position. | Moves the tractor forward and provides encoder data for position estimation. | Makes the motor move forward and estimates the current position with the encoder. | ok |
| **LM2596 Voltage regulator** | Connected to STM32, Arduino, and peripherals to ensure stable power supply. | Regulates power supply for modules. | As expected | ok |
| **LPS John Deere Camera** | Connected and initialized to track the tractor's position visually. | Tracks the tractor's position visually. | As expected | ok |
| **MPU6050 IMU Module** | Connected to STM32 and initialized to send position estimation data using gyroscope and accelerometer. | Sends position estimation data using gyroscope and accelerometer. | Estimates the position using the gyroscope an the accelerometer, and Sends data via I2C. | ok |
| **Comments** | All components worked as expected, ensuring seamless communication between systems and accurate navigation through waypoints. | | | |

E

# Testing methodology

## Videos of the development process

Videos from the Waypoint navigation

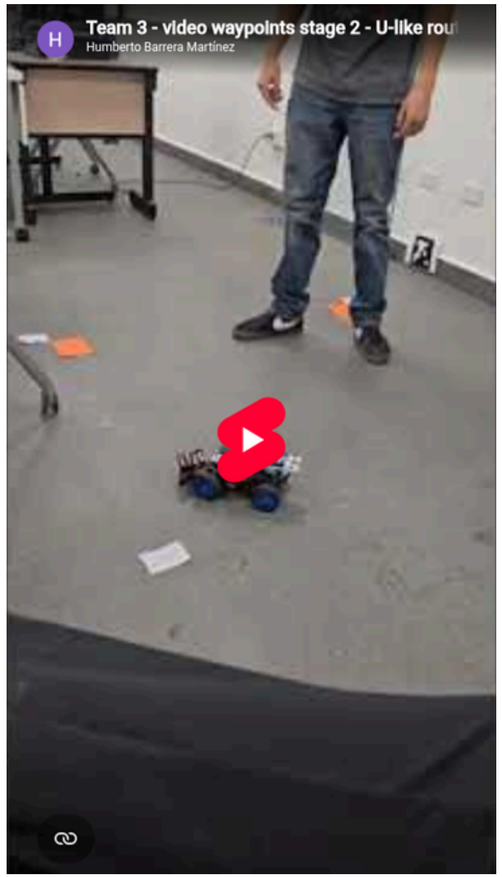



**Waypoints stage 1:** https://www.youtube.com/watch?v=dgoM10nOsPw&feature=youtu.be

# Testing methodology

## Videos of the development process

Videos from the Waypoint navigation




**Waypoints stage 2:** https://www.youtube.com/shorts/KI9Jaax0hrg?feature=share

TE2004B

# Conclusions



In conclusion, our project successfully demonstrated the integration of advanced embedded systems techniques in a practical agricultural application. By designing and building a scale tractor capable of precise waypoint navigation, we showcased the effective use of multithreading, communication protocols, and sensor fusion. This approach not only highlights the potential of autonomous systems in agriculture but also emphasizes the power of embedded technology in real-time applications.



TE2004B

# Reflections and gratitude



## Humberto

This semester was a pivotal step in my understanding of parallel computing and its practical applications. Learning how to implement FreeRTOS on a dual-core STM32H755 was both challenging and rewarding, as it allowed me to manage tasks with higher efficiency compared to single-core processors. Additionally, working with the CAN protocol expanded my understanding of automotive communication systems, which will undoubtedly be useful in future projects. Reflecting on the challenges, I realized the value of persistence and adaptability, particularly when working with unfamiliar components and debugging communication issues.

TE2004B

# Reflections and gratitude



## Roberto

This semester provided a unique opportunity to deepen my knowledge of hardware integration and communication protocols. Assembling the tractor and configuring peripherals such as the nRF24 wireless modules and servo motors required careful attention to detail and a solid understanding of PWM and SPI protocols. The experience of using HAL for the first time made it clear how abstraction can simplify programming while maintaining efficiency. I also learned the importance of collaboration, as working together on subsystem integration was essential to the project's success.

TE2004B

# Reflections and gratitude



## Erick

This semester was a great opportunity to expand my understanding of embedded systems, particularly in multithreading and signal processing. Learning to work with the STM32H755's dual-core architecture provided a new perspective on parallel task management, and implementing state machines before transitioning to FreeRTOS was a valuable exercise in system design. I enjoyed working on the I2C communication with the IMU module, as it required both theoretical knowledge and practical problem-solving. Overall, this semester strengthened my programming and debugging skills, and I feel more confident tackling complex embedded systems projects in the future.

TE2004B

# Reflections and gratitude



## Mauricio

One of the most valuable aspects of this semester was learning to program with HAL libraries and contrasting it with bare-metal programming from last semester. HAL significantly reduced the complexity of initializing and managing peripherals, allowing me to focus on higher-level tasks. Exploring multithreading and task prioritization with FreeRTOS was particularly enlightening, as it showed how parallel computing can optimize system performance. Debugging UART communication and observing the results on an oscilloscope reinforced my understanding of signal processing and communication, providing me with tools that I can carry into future projects.

TE2004B

# Participation

| Team member | Assigned presentation parts (Slide number |
|---|---|
| Mauricio | 2,4,6,7.1,19,21,22,26,30,33 |
| Roberto | 7.2,9,15,16,18,20,23,27,31,35 |
| Humberto | 7.3,10,11,12,13,14,17,24,28,32 |
| Erick | 1,3,5,7.4,7.5,8,25,29,34,36 |

TE2004B

# References


- Controller Stech (2021). PWM (Pulse Width Mod) in STM32. Controller Stech. Retrieved September 22, 2024 from, https://controllerstech.com/pwm-in-stm32/.
- John Deere (2023). Sustainability at John Deere. Retrieved March 25, 2024 from, https://about.deere.com/en-us/sustainability.
- John Deere (2024). Technology and innovation. Retrieved March 25, 2024 from, https://about.deere.com/en-us/our-company-and-purpose/technology-and-innovation.
- Karunathilake, E., Le, A. T., Heo, S., Chung, Y. S., and Mansoor, S. (2023). The path to smart farming: Innovations and opportunities in precision agriculture. Agriculture, 13(8):1593.
- Khan, S., Ahmad, K., Murad, M., and Khan, I. (2013). Waypoint navigation system implementation via a mobile robot using global positioning system (gps) and global system for mobile communications (gsm) modems. International Journal of Computational Engineering Research (IJCER), 3(7):1–7.
- Magdy, K. (2021). STM32 PWM Example – Timer PWM Mode Tutorial. Deep Blue embedded. Retrieved September 22, 2024 from, https://deepbluembedded.com/stm32-pwm-example-timer-pwm-mode-tutorial/.
- Math, M. (2024). Development of robotic vehicle with gps system for landmine detection. International Journal of Scientific Research in Mechanical and Materials Engineering, 8:50–54.
- Microchip (2019). Stand-Alone CAN Controller with SPI Interface. Microchip. Retrieved November 3, 2024 from, https://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf.
- STM32 (2021). STM32L4 - Timers. ST. Retrieved September 22, 2024 from, https://www.st.com/content/ccc/resource/training/technical/product_training/c4/1b/56/83/3a/a1/47/64/STM32L4_WDG_TIMERS_GPTIM.pdf/files/STM32L4_WDG_TIMERS_GPTIM.pdf/jcr:content/translations/en.STM32L4_WDG_TIMERS_GPTIM.pdf.
- Wang, Y., Mulvaney, D., Sillitoe, I., and Swere, E. (2008). Robot navigation by waypoints. Journal of Intelligent and Robotic Systems, 52:175–207.
- *Zhang, X., Lai, J., Xu, D., Li, H., and Fu, M. (2020). 2d lidar-based slam and path planning for indoor rescue using mobile robots. Journal of Advanced Transportation, 2020:1–14.*