# Unit Test Plan Unit testing is essential for verifying the correct functioning of each individual component in the software architecture of the Connect-4 robot player, including the initialization procedure, low-level code about different peripherals, and communication between the two cores.

* BSP Layer
  + TIMER block
  + NVIC block
  + EXTI block
  + UART
    - RX
    - TX
  + I2C
    - SCL
    - SDA
  + LED
  + GPIO
  + Power
  + ADC
  + ETH
  + HSEM
* Vacuum components
  + Vacuum Pump
  + Vacuum Sensor
  + Vacuum Valve
* Sensors/Library Testing
  + RGB Sensor
  + IR sensor
* End-switches
* Home-switches
* Encoder readout
* PID calculations
* PWM signal accuracy
* Motor control and **accuracy**
  + X-axis
  + Z-axis
* Servo control
  + End-effector rotation
  + Board clean-up piece
* Software-driven movements
* Token detector – entry point
  + Multiple tokens at once in a single column
  + Single tokens at once in different columns
  + Multiple tokens at once in different columns
* Flipper/Solenoid control
* Emergency stop
* Power/Reset button
* Dual-core communication
* Error Handling

# Module Test

* Level 1
  + CM4
  + CM7
  + Dual-Core communication
* Level 2
  + Cortex-M4
    - Initialization
    - Task Manager
    - Motor controller
    - Token colour separator
    - User Detector
    - Board opener
    - Token picker controller
  + Cortex-M7
    - Initialization
    - Game controller
    - CM4 Task Generator
    - Game end block
    - UART controller
* Level 3
  + Motor Controller
    - PID X
    - PID Z
    - Motor X
    - Motor Z
    - Motor Master
  + Token colour separator
    - Colour Sensor
    - Proxy Sensor
    - Solenoid
    - Token colour separator master
  + Token picker controller
    - Servo Controller
    - Vacuum pump
    - Token picker master

# Integration Test

Integration testing is needed to ensure that these individual components can work together seamlessly as a system. Given that the system has undergone a major restructure of its software architecture and PCB design with numerous changes over several years, it is important to perform system testing to identify gaps in functionality or areas where the system does not perform as expected.

The tests describe here are intended to receive their input as if from the level above them, in order to test the integration level by level. And the inner working of the blocks.

[Think how to verify/validate]

1. Test the integration between the RGB sensor and the robot's flipper, by checking if the robot can correctly detect and sort tokens based on their color.
2. Test the integration between the IR sensors and the robot's movement, by checking if the robot can detect when and where a token is dropped on the board.
3. Test the integration between the vacuum pump and the vacuum sensor, by checking if the robot can pick up a token and verify that the vacuum sensor detects the presence of a token.
4. Test the integration between the servos and the robot's end-effector, by checking if the robot can correctly rotate the end-effector and open the board for resetting the game state.
5. Test the integration between the motors, encoders, and home/end switches, by checking if the robot can move its end-effector to the correct position to pick up or drop a token, and if it can return to its home position after each move.
6. Test the integration between the two cores of the STM32H7 processor, by checking if they can communicate with each other correctly and if they can delegate tasks and take care of the higher-level logic like the game decisions, displaying results, etc.
7. Test the integration between the Connect-4 game logic and the robot's subsystems, by playing a game of Connect-4 against the robot and verifying that it makes valid moves and follows the rules of the game.

* Level 1
* Level 2
* Level 3
  + Motor Controller
  + Token Colour Separator
  + Token Picker Controller
  + User Detect
  + Board Opener

# Happy-Path Test

# System Test

System testing will ensure that the various sub-systems of the robot player work together seamlessly as a complete system, and that the system meets its overall performance and functional requirements. System testing may include testing the performance of the dual-core STM32H755 processor, testing the connectivity and communication between different hardware components such as motors, servos, encoders, and sensors, and verifying that the robot player can autonomously play the Connect-4 game. Additionally, system testing may involve testing the system's ability to handle unexpected events such as power outages or hardware failures, as well as testing the system's compatibility with future upgrades such as Ethernet communication with the internet.

# Acceptance Test

Acceptance testing should be performed from the client's perspective to ensure that the system meets the requirements outlined in the assignment, is ready for deployment, and supports future upgrades such as Ethernet communication with the internet for transferring and receiving data to keep high scores, current player’s turn, and a human-machine interface.