# Component 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.

## Unit Tests

Builds confidence in the separate blocks/components.

* BSP Layer
  + TIM
  + NVIC
  + EXTI
  + 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
  + Proximity 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
* Flipper/Solenoid control
* Emergency stop
* Power/Reset button
* Dual-core communication
* Error Handling
* Motor Master
* Token Colour separator master
* Token picker master

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Test Conditions | Test Data | Test Set-Up |
| TIM |  |  |  |
| NVIC |  |  |  |
| EXTI |  |  |  |
| UART |  |  |  |
| I2C |  |  |  |
| LED |  |  |  |
| GPIO |  |  |  |
| Power |  |  |  |
| ADC |  |  |  |
| ETH |  |  |  |
| HSEM |  |  |  |
|  |  |  |  |
| Vacuum Pump |  |  |  |
| Vacuum Sensor |  |  |  |
| Vacuum Valve |  |  |  |
| RGB Sensor |  |  |  |
| IR Sensor |  |  |  |
| Proximity Sensor |  |  |  |
| End-Switches |  |  |  |
| Home Switches |  |  |  |
| Encoder Readout |  |  |  |
| PID calculations |  |  |  |
| PWM Signal |  |  |  |
| Motor control – X |  |  |  |
| Motor control – Z |  |  |  |
| Servo – End-effector |  |  |  |
| Servo – Board opener |  |  |  |
| Token picker controller |  |  |  |
| Token colour separator |  |  |  |
| Motor Controller |  |  |  |
| Flipper/Solenoid control |  |  |  |
| Emergency Stop |  |  |  |
| Dual-core communication |  |  |  |
| Error handling |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Module Tests

Builds confidence in the interfaces between modules

* Level 1
  + CM4
  + CM7
  + Dual-Core communication
* Level 2
  + Cortex-M4
    - Initialization
    - Task Manager
    - Motor controller
    - Token colour separator controller
    - User Detector
      * Multiple tokens at once in a single column
      * Single tokens at once in different columns
      * Multiple tokens at once in different columns
    - Board opener
    - Token picker controller
  + Cortex-M7
    - Initialization
    - Game controller
    - CM4 Task Generator
    - Game end block
    - UART controller

# Integration Test

Integration testing is an essential process to ensure that the individual components of a system can work together seamlessly. In the case of the Connect-4 robot player, the software architecture and PCB design have undergone significant restructuring with multiple changes over several years. Therefore, it is crucial to perform system testing to identify gaps in functionality or areas where the system does not perform as expected.

This includes testing the connectivity and communication between different hardware components such as motors, servos, encoders, and sensors, and verifying the robot player's ability to autonomously play Connect-4.

The tests are designed to receive their input from the level above to test the integration level by level and the inner workings 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.

# Happy-Path Test

The intended way of using the system

# 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.