FAT document

Project Puzzlebox

Thomas in ‘t Anker, Loek Le Blansch, Lars Faase, Elwin Hammer

Version 0.0, 2024-04-01: draft

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Compatibility matrix

Requirement \\ Test	T-01	T-02	T-03	T-04	T-05	T-06	T-07	T-08	T-09
R-01	¬								
R-02	¬								
R-03	¬								
R-04	¬								
R-05									
R-06									
R-07									
R-08									
R-09		¬							
R-10		¬							
R-11		¬							
R-12		¬							
R-13			¬						
R-14			¬						
R-15			¬						
R-16									
R-17				¬					
R-18				¬					
R-19				¬					
R-20				¬					
R-21					¬				
R-22									
R-23					¬				
R-24					¬				
R-25					¬				
R-26						¬			
R-27						¬			
R-28						¬			
R-29						¬			
R-30							¬		
R-31									
R-32							¬		
R-33							¬		
R-34							¬		
R-35								¬	
R-36								¬	

1.2. Neotrellis-puzzle

1.2.1. Basic system test setup

- ¬ Connect the neotrellis modules to an Arduino with the I²C lines only.
- ¬ Use an external powersource to provide enough power to the neotrellis modules (5V 1A should be enough).
- ¬ Make sure that both the Arduino and neotrellis modules are sharing a common ground.
- ¬ The Arduino should run the latest code from the GitHub repo.

1.2.2. Test 01

Covered requirements

- Technical requirements
  - R-34 (neotrellis function & Arduino init.)
- Functional requirements
  - **None**

Additional setup

None

Test

1. Turn both the Arduino and the neotrellis modules on

Result

- ¬ Verify that the neotrellis modules emit light in different colors (init. state).

1.2.3. Test 02

Covered requirements

- Technical requirements
  - R-166 (Arduino & RPI Pico I²C communication)
  - R-169 (Arduino & neotrellis I²C communication)
- Functional requirements
  - R-36 (game logic)
  - R-37 (neotrellis led color)
  - R-38 (game d.o.d.)

Additional setup

- ¬ Connect the Arduino to the maincontroller (RPI Pico) using the I²C interface, pull-up resistors and a common ground.

Test

1. Turn both the Arduino and the neotrellis modules on.
2. Wait for a few seconds.
3. Turn the RPI Pico on.
4. Wait for another few seconds.
5. Play the game until you have succeded.

1.2.4. Result

- ¬ Verify that the puzzle module tries to communicate with the maincontroller.
- ¬ Verify that a light should be blinking on one of the neotrellis modules.

- ¬ Press one of the emitting buttons and check of the buttons surrounding it will light
- ¬ Play the game until no buttons emit light anymore and check if the main controllers wants to load the next puzzle.

1.3. Vault-puzzle

1.3.1. Basic system test setup

- ¬ Connect the 4x 7-seg display up to the Arduino using it's serial interface and onboard power (5V).
- ¬ Connect the button matrix to the Arduino (check the latest Arduino file for the pins).
- ¬ The Arduino should run the latest code from the GitHub repo.

1.3.2. Test 03

Covered requirements

- Technical requirements
  - R-151 (4x 7-seg. type & connection)
- Functional requirements
  - R-75 (4x 7-seg. availibilty)
  - R-76 (ventilation)
  - R-81 (4x 7-seg. operation)

Additional setup

- ¬ Connect the Arduino to the maincontroller (RPI Pico) using the I²C interface, pull-up resistors and a common ground.

Test

1. Turn the Arduino on.
2. Wait for a few seconds.
3. Turn the RPI Pico on.
4. Wait until the Arduino is in it's init. state.

Result

- ¬ Verify that there is a 4x 7-seg. display and working.
- ¬ Verify that the puzzle module tries to communicate with the maincontroller
- ¬ Check if after the uniti. state (blinking zero's) there is a letter and number displayed on the 4x 7 seg. display.

1.3.3. Test 04

Covered requirements

- Technical requirements
  - **None**
- Functional requirements
  - R-85 (4x 7-seg. operation by passing the game)
  - R-87 (4x 7-seg. operation by failing the game)

Additional setup

None

Test

1. Turn the Arduino on.
2. Wait for a few seconds.
3. Turn the RPI Pico on.
4. Play the game correctly for 2 levels.
5. Play one level incorrectly.

Result

¬ Verify a new code is generated eachtime you level-up
¬ Verify that an older code is presented eachtime you level-down

1.3.4. Test 05

Covered requirements

- Technical requirements
  - R-150 (button matrix)
- Functional requirements
  - R-82 (game logic: leveling)
  - R-83 (game logic: level)
  - R-84 (game logic: succeding level)

Additional setup

- Connect the vault solenoid valve to the Arduino using a mosfet or other type of driver.

Test

1. Turn the Arduino on.
2. Wait for a few seconds.
3. Turn the RPI Pico on.
4. Play the game correctly until you reach the end of the game.

Result

¬ Verify that at the last level a code should be displayed that can be used to disarm the "bomb".
¬ Check if the vualt it self will unlock and lock again after the game is reset.

1.4. Puzzle Box Requirements

1.4.1. Puzzle Box Basic Setup and Verification

Covered requirements

- Functional requirements
  - R-1 (Puzzle box dimensions)
  - R-2 (Extension on the sides and top)
  - R-3 (Flat bottom)
  - R-4 (Key switch at the bottom of the NeoTrellis puzzle)
  - R-5 (Indicator LED at the bottom of the NeoTrellis puzzle)

Additional setup

¬ Measure the dimensions of the puzzle box to ensure compliance with R-1.

¬ Inspect the physical structure for extensions beyond specified limits.
¬ Confirm flatness of the bottom surface.
¬ Locate and identify the key switch and indicator LED as per design specifications.

Test

1. Measure and record the dimensions of the puzzle box.
2. Examine the extensions, if any, on the sides and top of the box.
3. Test the flatness of the bottom of the box.
4. Verify the functionality of the key switch.
5. Check the initial state of the indicator LED when power is supplied to the puzzle box.

Result

¬ Puzzle box dimensions are within the specified range of 30x30x30 cm ± 5%.
¬ No excess extension beyond 5 cm on any side or the top.
¬ The bottom of the puzzle box is confirmed flat.
¬ The key switch operates as intended.
¬ The indicator LED functions correctly and is visible.

1.5. Game Functional Requirements Testing

1.5.1. Puzzle Module Control Tests

Covered requirements

- Functional requirements
  - R-167 (Manual reset capability of a puzzle module)
  - R-168 (Manual solve capability of a puzzle module)

Additional setup

¬ Ensure access to the game operator controls for manual reset and solve.

Test

1. Perform a manual reset of a puzzle module during gameplay to verify it resets correctly.
2. Manually set a puzzle module as solved and observe if the system acknowledges the solve appropriately.

Result

¬ Puzzle module resets correctly when manually triggered.
¬ Puzzle module is recognized as solved when set manually.

1.5.2. Puzzle Difficulty Verification

Covered requirements

- Functional requirements
  - R-27 (Puzzle solvability without prior knowledge)

Test

1. Invite testers unfamiliar with the game to solve the puzzles.
2. Record the ease or difficulty encountered by the testers in solving the puzzles.

Result

¬ Testers without prior knowledge are able to solve the puzzles, indicating appropriate difficulty levels.

1.6. Hardware Puzzle Module Testing

1.6.1. Initial Setup Verification

Covered requirements

- Functional requirements
  - R-58 (Game operator initial switch position)
  - R-59 (Game operator initial potentiometer position)

Additional setup

¬ Access to hardware puzzle board controls for switches and potentiometers.

Test

1. Manually set all switches on the hardware puzzle board to the down position.
2. Turn all potentiometers fully to the left to their zero position.

Result

¬ All switches are confirmed to be in the down position at the start of the game.
¬ All potentiometers are set to zero at the beginning of the game, indicating correct initial setup.

1.6.2. Hardware Component Functionality

Covered requirements

- Functional requirements
  - R-48 (Presence of eight switches)
  - R-49 (Combinatorial circuit functionality)
  - R-50 (7-segment 4-digit display presence and functionality)
  - R-51 (Four potentiometers)
  - R-52 (Blue LED displaying Morse code)
  - R-53 (Green LED indicating solved state)

Additional setup

¬ Ensure all hardware components are installed as per the design specifications (Refer to Figure 8).

Test

1. Verify the presence and functionality of eight switches linked to the combinatorial circuit.
2. Confirm that the combinatorial circuit operates as depicted in the provided diagram.
3. Check the 7-segment display and potentiometers for correct operation.
4. Observe the blue LED for Morse code display and green LED for indicating the puzzle is solved.

Result

¬ Eight switches are operational and affect the combinatorial circuit as expected.
¬ The combinatorial circuit correctly processes inputs to produce the expected output.
¬ The 7-segment display and potentiometers function correctly, displaying values as manipulated.
¬ Blue LED successfully displays Morse code, and green LED lights up when the puzzle is solved.

1.7. Battery Functional Requirements Testing

1.7.1. Battery Power Source Verification

Covered requirements

- Technical requirements
  - R-89 (Rechargeable battery power source)
  - R-90 (Battery life of minimum 4 hours)

Additional setup

¬ Ensure the puzzle box is equipped with its intended rechargeable battery.

Test

1. Verify the puzzle box is powered by the rechargeable battery by observing its operation without external power.
2. Test the battery life by running the puzzle box continuously to ensure it lasts at least 4 hours under normal operating conditions.

Result

¬ The puzzle box is confirmed to be powered by a rechargeable battery.
¬ The battery sustains power for a minimum of 4 hours, confirming its capacity meets the specification.

1.7.2. Battery Replacement Capability

Covered requirements

- Conditional requirements
  - R-91 (Battery replaceability)

Test

1. Attempt to remove and replace the battery in the puzzle box to check if it can be done without damaging the device or requiring excessive effort.

Result

¬ The battery can be replaced easily and safely, indicating compliance with the design's replaceability condition.

1.8. Wireless Communication Functional Requirements Testing

1.8.1. Wireless Network Integration and Performance

Covered requirements

- Technical requirements
  - R-127 (Wireless communication via WiFi mesh or WiFi network)

Additional setup

¬ Ensure that the system controller, bomb, and puzzle box are equipped with WiFi capabilities.
¬ Setup a WiFi mesh network or a WiFi network, as applicable, for testing.

Test

1. Connect the system controller, bomb, and puzzle box to the designated WiFi network.
2. Conduct a series of communication tests to verify that messages are sent and received correctly and promptly between the devices.
3. Test the robustness of the connection by simulating various conditions, such as increased distance and interference.

Result

¬ All devices successfully connect to the WiFi network and communicate with each other without loss of data.
¬ The WiFi network remains stable and maintains communication under different tested conditions, demonstrating reliability and effectiveness. === Framework Functional Requirements Testing

1.8.2. Framework Creation and Operation

Covered requirements

- Technical requirements
  - R-128 (Framework creation for future development)
  - R-129 (Framework operation on the main puzzle box controller)

Additional setup

¬ Ensure the framework is deployed on the main puzzle box controller.

Test

1. Verify the presence of the framework on the main controller.
2. Check that the framework is accessible and functional for development purposes.

Result

¬ The framework is confirmed to be created and is operational on the main puzzle box controller, facilitating future development.

1.8.3. Puzzle Module Detection and Initialization

Covered requirements

- Technical requirements
  - R-134 (Detection of puzzle modules by the main controller)
  - R-135 (Initialization of puzzle modules by the main controller)

Additional setup

¬ Setup multiple puzzle modules connected to the main controller.

Test

1. Power on the system and observe if the main controller detects all connected puzzle modules.
2. Monitor the initialization process to ensure each puzzle module is correctly initialized by the main controller as per the framework's protocols.

Result

¬ All connected puzzle modules are detected by the main controller upon system startup.
¬ Each puzzle module is successfully initialized, confirming the effectiveness of the framework's initialization procedures.

1.8.4. Communication Testing of Uninitialized Modules

Covered requirements

- Technical requirements
  - R-165 (Uninitialized puzzle modules sending 'update' messages)

Additional setup

¬ Configure puzzle modules to start in an 'uninitialized' state.

Test

1. Monitor the communication between the uninitialized puzzle modules and the main controller.
2. Verify that 'update' messages are sent consistently by the uninitialized modules to the main controller.

Result

¬ Uninitialized puzzle modules repeatedly send 'update' messages to the main controller, confirming continuous communication while in the 'uninitialized' state.

1.9. Main Controller Technical Requirements Testing

1.9.1. I2C Connectivity Testing

Covered requirements

- Technical requirements
  - R-136 (Main controller has at least 1 I2C peripheral)
  - R-140 (Main controller I²C communication speed)

Additional setup

¬ Connect the main controller to a compatible I2C device.

Test

1. Verify the presence and functionality of the I2C peripheral on the main controller.
2. Test the communication speed to ensure it reaches up to 400kb/s.

Result

- ¬ The main controller is confirmed to have at least one operational I2C peripheral.
- ¬ I2C communication occurs at a speed of 400kb/s, meeting the specification.

1.9.2. WiFi Connectivity Testing

Covered requirements

- Technical requirements
  - R-137 (Connection to a standard 802.11b/g/n access point)

Additional setup

- ¬ Ensure the main controller is equipped with WiFi capabilities.
- ¬ Setup a standard 802.11b/g/n WiFi access point for testing.

Test

1. Connect the main controller to the WiFi access point.
2. Monitor the stability and reliability of the WiFi connection.

Result

- ¬ The main controller successfully connects to the 802.11b/g/n WiFi access point and maintains a stable connection.

1.9.3. TCP Socket Connection Testing

Covered requirements

- Technical requirements
  - R-138 (Main controller can serve TCP socket connection(s))

Additional setup

- ¬ Prepare a device to connect to the main controller via a TCP socket.

Test

1. Establish a TCP socket connection with the main controller.
2. Verify that data can be sent and received reliably through the connection.

Result

- ¬ The main controller can serve TCP socket connections effectively, handling data transmission without issues.

1.9.4. Availability as a Development Kit

Covered requirements

- Technical requirements
  - R-139 (Availability of the main controller as a development kit from Farnell)

Test

1. Verify the availability of the main controller as a development kit on the Farnell website or

through their sales channels.

Result

¬ The main controller is available for purchase as a development kit from Farnell, facilitating accessibility for developers.

1.10. Puzzle Module Controller Technical Requirements Testing

1.10.1. I2C and I/O Port Testing

Covered requirements

- Technical requirements
  - R-141 (Puzzle module controller has at least 1 I2C peripheral)
  - R-142 (Sufficient I/O ports to control a puzzle)
  - R-146 (Puzzle module I²C communication speed)

Additional setup

¬ Connect the puzzle module controller to a test setup including I2C devices and various I/O components.

Test

1. Verify the functionality of at least one I2C peripheral on the puzzle module controller.
2. Test the I/O ports by connecting various puzzle components and verifying control capabilities.
3. Measure the I2C communication speed to confirm it achieves up to 400kb/s.

Result

¬ The puzzle module controller includes a functional I2C peripheral.
¬ The controller has enough I/O ports to manage and control connected puzzle components effectively.
¬ I2C communication occurs at a speed of 400kb/s, aligning with the technical specifications.

1.10.2. Clock Speed and Development Kit Availability Testing

Covered requirements

- Technical requirements
  - R-144 (Configurable clock speed of the puzzle module)
  - R-145 (Availability as a development kit from Farnell)

Additional setup

¬ Access the configuration settings of the puzzle module controller to adjust the clock speed.

Test

1. Adjust the clock speed settings of the puzzle module controller to verify configurability.
2. Check Farnell's website or contact their sales channels to confirm the availability of the puzzle module controller as a development kit.

Result

¬ The clock speed of the puzzle module controller can be configured, demonstrating flexibility in operational settings.
¬ The puzzle module controller is available as a development kit from Farnell, ensuring accessibility for development and further customization.

1.10.3. Power Efficiency Testing

Covered requirements

- Technical requirements
  - R-143 (Power efficiency of the puzzle module)

Test

1. Measure the power consumption of the puzzle module controller during standard operation and under load.
2. Compare the observed power consumption with power efficiency standards for similar devices.

Result

¬ The puzzle module controller shows efficient power usage in comparison to standard benchmarks, confirming its efficiency in energy consumption.