Project Requirements

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

In this document, the specifications are described prior to the investigation of the Puzzle Box project. These specifications are partly derived from the previously established requirements and are further supplemented and modified. The priority of specifications is indicated using the MoSCoW method, see Table 1.

Priority	Description
Must have	Represents essential system requirements. Without these, the system will not function.
Should have	Denotes desirable system features. The system can work without these, but it lacks necessary elements.
Could have	Refers to additional functionalities that can be implemented if there is extra time.
Won’t have	Specifies requirements that will not be implemented in the current version but may be considered in a future release.

Table 1. MoSCoW Method [1]

This specification document covers hardware, software, and game-specific details. The focus in this project year for the Puzzle Box is to thoroughly document the system and create a software framework for future groups.

2. Context

This chapter describes how the user will interact with the system. This is done in the form of a user story. This user story covers hardware, software, and game specifications. From this narrative, many specifications can be derived for both functional and non-functional requirements (Section 3 and Section 4).

The game administrator picks up the puzzle box and places it on a flat surface. By using the key switch, the puzzle box is turned on, and the green indicator LED lights up. Through the mesh network established by the external puzzle box hub, the corresponding web panel can be accessed. The web panel provides instructions for configuring the puzzle box, including linking it to any bomb. The instructions issue a warning if any of the start conditions are not properly set. If a criterion is incorrectly configured, it is highlighted for resolution. Additionally, a warning is given if the battery capacity is insufficient for one game duration, causing the indicator LED on the puzzle box to turn red. In such cases, the battery should be charged using the USB-C cable. While the puzzle box is charging, the indicator LED is blue. Once there are no warnings and the puzzle box is adequately charged, the game can be started in the web panel.

The puzzle box begins with the NeoTrellis game. In this game, players must turn off all LEDs on an 8x8 button LED matrix. When any button is pressed, the directly adjacent LEDs toggle. If a lit LED is toggled, it turns off; if an unlit LED is toggled, it turns on. Once all LEDs are turned off, the game is solved, and the software puzzle begins.

On the software puzzle, there are 6 banana plug connectors on both the left and right sides. The ones on the left are labeled with various logical gates, while the ones on the right are labeled from A to F. Participants in the bomb game have 6 pieces of C-code written on paper, corresponding to the logical gates on the puzzle box. The bomb participants must provide a description of the C-code to the puzzle box participants, allowing them to correctly connect the appropriate logical gate to the corresponding letter. Once the correct combination of logical gates with the correct letter is made, the game is solved. Subsequently, the automation puzzle is initiated.

Since there is no concept available for the automation puzzle yet, the hardware puzzle is started directly.

The hardware puzzle is played in two distinct phases. In Phase 1, the objective is to solve a combinatorial circuit such that its output becomes '1'. There are 8 inputs for this circuit, each controlled by an on/off switch. Once the combinatorial circuit evaluates to '1', the LED at the output lights up, indicating the completion of the first phase. In Phase 2, another LED blinks, consistently repeating a pattern. This pattern represents a randomly generated Morse code, corresponding to a number from 0 to 9999. Participants use a Morse code table to decipher the correct number. Using four potentiometers, the participants can set a number on a 7-segment display. When this number matches the randomly generated one, the hardware puzzle is solved. Subsequently, the vault puzzle is initiated.

In the vault puzzle, a 7-segment display shows a random combination of a letter and a digit. Participants have access to a list containing the correct button combination for the corresponding letter and digit. The vault puzzle consists of 5 levels, each displaying a unique button combination from the list. When participants correctly press the button on the keypad, the level advances, and a new value is shown. Pressing the wrong button restarts the game at level 1. Once all 5 levels are completed, the vault door unlocks, allowing access to the inside of the puzzle box. On the mainboard, there is a 7-segment display showing a code. This code must be relayed to the participants of the bomb game. Once the bomb team receives the code, the puzzle box is considered solved.

3. Functional Requirements

The functional requirements describe the things which are important to the client. This is mainly about the way the product is going to be used, what it is going to look like, and how the product reacts to interaction. This chapter describes all functional requirements of the puzzle box.

3.1. The puzzle box

ID	Pri.	Specification
R-001	**M**	The dimensions of the puzzle box are 30×30×30cm ± 5% (Length × Width × Height).
R-002	**M**	The puzzle box extends a maximum of 5cm on the sides and the top.
R-003	**M**	The puzzle box is flat at the bottom.
R-004	**M**	The puzzle box has a key switch at the bottom of the NeoTrellis puzzle.
R-005	**M**	The puzzle box has an indicator LED at the bottom of the NeoTrellis puzzle.
R-006	**M**	The indicator LED turns green when the system is on and not charging.
R-007	**M**	The indicator LED turns blue when the battery is charging.
R-008	**M**	The indicator LED turns red when the battery does not have enough capacity for the duration of one game and is not charging.
R-009	**M**	The puzzle box has a USB-C port at the bottom of the NeoTrellis puzzle for battery charging.
R-010	**M**	The puzzle box has a distance sensor at the bottom to detect if it is lifted.
R-011	**M**	The puzzle box main board (PCB on the bottom plate) includes a speaker.
R-012	**W**	When the puzzle box is lifted, the mainboard speaker emits an alarm sound for at least 10 seconds. It stops only when it has been on a table for another 10 seconds (detected by the distance sensor).
R-013	**W**	When the game is completed, the puzzle box produces a victory sound.
R-014	**W**	Pressing the "identify" button on the web panel causes the indicator LED to blink.
R-015	**W**	Pressing the "identify" button on the web panel triggers a sound from the speaker.
R-016	**W**	The game starts once the scheduler time is reached (refer to [2] section 3.7).

Table 2. Puzzle box specifications

3.2. The bomb

ID	Pri.	Specification
R-017	**M**	The bomb includes a 6-digit 7-segment display for showing the remaining playtime.
R-018	**M**	The bomb contains a keypad for entering the disarm code.
R-019	**W**	The 6-digit 7-segment display turns off when no game is in progress.
R-020	**W**	Once the disarm code is entered on the bomb keypad, the game is complete.
R-021	**W**	When the game is finished, the bomb emits a victory sound.
R-022	**W**	The timer on the bomb counts down from 60:00:00 to 00:00:00.
R-023	**W**	Pressing the "identify" button on the web panel causes the indicator LED to blink.
R-024	**W**	Pressing the "identify" button on the web panel triggers a sound from the speaker.

Table 3. Bomb specifications

3.3. The game

ID	Pri.	Specification
R-025	**W**	The game lasts for 1 hour.
R-026	**W**	The game should be solvable within the given playtime, without the player having prior knowledge of the game or its mechanics.
R-027	**M**	The puzzles should be easy enough to solve without any prior knowledge of the game or its mechanics.
R-167	**M**	A puzzle module can manually be reset at the discretion of the game operator
R-168	**M**	A puzzle module can manually be set as solved at the discretion of the game operator
R-028	**W**	The disarm code for the bomb consists of 4 digits.
R-029	**W**	Once all games are solved, the mainboard PCB displays the disarm code on a red 7-segment 4-digit screen.
R-030	**W**	The puzzle box records the playtime of each game.
R-031	**W**	The puzzle box features 5 playable puzzles.
R-032	**W**	Only one game is active at a time; the other games do not respond to buttons.
R-033	**W**	The game always starts with the NeoTrellis puzzle.

Table 4. General game specifications

3.3.1. NeoTrellis puzzle

ID	Pri.	Specification
R-034	**M**	There is an 8x8 LED matrix where each LED can display different colors.
R-035	**W**	At the start of the puzzle, a pattern is displayed as shown in Figure 4.
R-036	**W**	When a button is pressed, the adjacent LEDs and the pressed LED toggle (If an LED is off, it turns on. If an LED is on, it turns off).
R-037	**W**	All LEDs in the Neotrellis that are turned on are blue.
R-038	**W**	The puzzle is considered solved when all LEDs are turned off, and then the software puzzle starts.

Table 5. NeoTrellis puzzle requirements

3.3.2. Software puzzle

ID	Pri.	Specification
R-039	**S**	The software puzzle board has 6 banana plug connectors with different logic gates displayed next to them (Refer to Figure 5 for a sketch and Figure 7 for a banana plug example).
R-040	**S**	The software puzzle board has 6 banana plug connectors labeled with the letters A through F (Refer to Figure 5 for a sketch).
R-041	**S**	At the start of the puzzle box game, the preparer must connect all cables in parallel (horizontally) to the connectors.
R-042	**W**	There are C code blocks visible only to the players on the bomb side, corresponding to the letters A through F (Refer to Figure 6 for the codes).
R-043	**S**	The combinations of logic gates to letters are always the same.
R-044	**W**	The puzzle is considered solved when the cables from the logic gates match the code blocks (Refer to Figure 5 and Figure 6 for the combinations).
R-045	**W**	Once the puzzle is solved, the green indicator LED will light up (Refer to Figure 5 and Figure 6).
R-046	**W**	After the puzzle is solved, the automation puzzle begins.

Table 6. Software puzzle requirements

3.3.3. Automation puzzle

The specific details for this puzzle are not present in the previous documentation. Due to time constraints, the section will be left empty.

ID	Pri.	Specification
R-047	**W**	After the puzzle is solved, the hardware puzzle begins.

Table 7. Automation puzzle requirements

3.3.4. Hardware puzzle

ID	Pri.	Specification
R-048	**S**	There are eight switches on the hardware puzzle board.
R-049	**S**	The hardware puzzle board features a diagram of a combinatorial circuit with 8 inputs (linked to the switches) and 1 output (Refer to Figure 8 for a sketch).
R-050	**S**	The hardware puzzle board includes a red 7-segment 4-digit display (Refer to Figure 8 for a sketch).
R-051	**S**	There are 4 potentiometers on the hardware puzzle board (Refer to Figure 8 for a sketch).
R-052	**S**	A blue LED on the hardware puzzle board displays the morse code.
R-053	**S**	A green LED on the hardware puzzle board indicates whether the combinatorial circuit is solved.
R-054	**W**	At the start of the puzzle, the potentiometers are inactive.
R-055	**W**	The 7-segment display is off at the beginning of the puzzle.
R-056	**W**	The LED for the combinatorial puzzle is off initially.
R-057	**W**	The morse code LED is off at the puzzle’s outset.
R-058	**M**	The preparer must set all switches to the down position at the start of the puzzle box game.

ID	Pri.	Specification
R-059	**M**	The preparer must turn all potentiometers to the left (value '0') at the beginning of the puzzle box game.
R-060	**W**	The puzzle consists of two phases.
R-061	**W**	The puzzle begins in phase 1.
R-062	**W**	During the puzzle, the switches must be toggled to obtain a logical '1' at the output of the combinatorial circuit.
R-063	**W**	When the output of the combinatorial circuit equals '1', the green indicator LED turns on (Refer to Figure 8 for a sketch).
R-064	**W**	The puzzle proceeds to phase 2 when the output of the combinatorial circuit is a logical '1'.
R-065	**W**	The switches no longer respond once the puzzle enters phase 2.
R-066	**W**	The indicator LED from phase 1 remains green during phase 2.
R-067	**W**	In phase 2, a morse code is displayed using an LED. This morse code represents 4 numbers from 0 to 9 and repeats every second.
R-068	**W**	The morse code is randomly generated.
R-069	**W**	Each potentiometer can be rotated to display a value from 0 to 9 on the corresponding 4-digit 7-segment display. The order of the potentiometers matches the order of the segments on the display (Refer to Figure 8 for a sketch).
R-070	**W**	The puzzle is considered solved when the code displayed on the 7-segment 4-digit screen matches the 4 numbers from the morse code.
R-071	**W**	Once the puzzle is solved, the value shown on the 7-segment 4-digit display cannot be changed.
R-072	**W**	A 2-second victory sound is produced by the speaker upon solving the puzzle.
R-073	**W**	During the victory sound, the 7-segment display blinks twice per second.
R-074	**W**	After the victory sound, the puzzle has been solved and the vault puzzle begins.

Table 8. Hardware puzzle requirements

3.3.5. Vault puzzle

ID	Pri.	Specification
R-075	**S**	The vault puzzle board features a red 7-segment 4-digit display.
R-076	**S**	On the vault puzzle board, there is a 4x4 grid of holes for ventilation and sound.
R-077	**S**	The vault puzzle board includes a vault door, and the inside of the vault is transparent, allowing you to see inside the puzzle box.
R-078	**S**	A sensor is integrated with the vault to detect when the vault is closed.
R-079	**M**	At the beginning of the puzzle box game, the preparer must close the vault.
R-080	**W**	The puzzle starts at level 1.
R-081	**W**	Initially, the 7-segment display shows a code consisting of a letter and a digit. This code represents a valid key combination for level 1 (Refer to Figure 9 for all combinations).
R-082	**W**	There are a total of 5 levels. After each level, a key combination is displayed, starting with a letter followed by a digit, which is valid for that level (Refer to Figure 9).
R-083	**W**	Each level has unique key combinations for the button locations (Refer to Figure 9).
R-084	**W**	Pressing the button corresponding to the letter-digit combinations advances the puzzle to the next level.

ID	Pri.	Specification
R-085	**W**	If an incorrect button is pressed, the game resets to level 1.
R-086	**W**	An error sound is produced by the speaker when an incorrect button is pressed.
R-087	**W**	The 7-segment display blinks when an incorrect button is pressed.
R-088	**W**	After completing 5 levels, the puzzle is solved, and the vault opens.

Table 9. Vault puzzle requirements

3.4. Battery

ID	Pri.	Specification
R-089	**M**	The puzzle box is powered by a rechargeable battery.
R-090	**M**	The battery lasts for a minimum of 4 hours.
R-091	**C**	The battery in the puzzle box can be replaced.

Table 10. Battery requirements

3.5. Network Communication

ID	Pri.	Specification
R-092	**W**	The puzzle boxes, bombs, and the puzzle box hub must all be able to communicate with each other.
R-093	**M**	Communication between two devices in the network must have a range of at least 20 meters in an open field.

Table 11. Communication requirements

3.6. Framework

ID	Pri.	Specification
R-130	**M**	The main controller and its software do not need to be modified to implement a new puzzle module
R-131	**C**	Puzzle modules can be added and removed while the main controller is powered on
R-132	**M**	Puzzle modules can be added and removed while the main controller is powered off
R-133	**M**	The puzzle box provides a single external interface for accessing and controlling game state variables

3.7. Puzzle box hub

ID	Pri.	Specification
R-094	**W**	The puzzle box hub hosts a website that can be accessed by a device connected to the network.

Table 12. Puzzle box hub general requirements

4. Technical Requirements

The technical specifications describe the specifications that are important for developers. For example, this could include specific requirements related to current, voltage, or communication protocols. This chapter outlines all the technical specifications of the puzzle box.

4.1. Wireless communication

ID	Pri.	Specification
R-127	**M**	The wireless communication between the system controller, bomb, and puzzle box operates over a WiFi mesh or WiFi network.

Table 13. Wireless communication requirements

4.2. Framework

ID	Pri.	Specification
R-128	**M**	A framework has been created to assist future groups in the development of the puzzle box.
R-129	**M**	The framework runs on the main puzzle box controller.
R-134	**M**	Puzzle modules are detected by the main controller module.
R-135	**M**	Puzzle modules are initialized by the main controller module.
R-165	**W**	Puzzle modules repeatedly send 'update' messages to the main controller while their global state is 'uninitialized'
R-169	**S**	External software and libraries that are covered in the standard curriculum should be used where possible

Table 14. Development framework requirements

4.3. Main controller

ID	Pri.	Specification
R-136	**M**	The main controller has at least 1 I2C peripheral.
R-137	**M**	The main controller can connect to a standard 802.11b/g/n access point.
R-138	**M**	The main controller can serve TCP socket connection(s).
R-139	**M**	The main controller is available as a development kit from Farnell.
R-140	**S**	The main controller can communicate over I²C with a speed of 400kb/s
R-166	**S**	The main controller is power efficient.

Table 15. Main controller requirements

4.4. Puzzle module controller

ID	Pri.	Specification
R-141	**M**	The puzzle module controller has at least 1 I2C peripheral.
R-142	**S**	The puzzle module controller has enough I/O ports to control a puzzle.
R-143	**S**	The puzzle module is power efficient.
R-144	**M**	The puzzle module has a configurable clock speed.
R-145	**M**	The puzzle module controller is available as a development kit from Farnell.
R-146	**S**	The puzzle module can communicate over I²C with a speed of 400kb/s

4.5. Vault puzzle

ID	Pri.	Specification
R-147	**M**	The vault puzzle can communicate with the main controller using I²C
R-148	**W**	The vault puzzle can produce a sound signal for the buzzer
R-149	**M**	The vault puzzle can lock & unlock a solenoid lock
R-150	**M**	The vault puzzle can translate and obtain a button press from the 3x4 keypad using 5 inputs
R-151	**M**	The vault puzzle can communicate with a 4x 7 SEG. Display using 2 lines (clock & data)
R-152	**S**	The vault puzzle can read a sensor’s value to detect if the vault door is open or closed.

Table 16. Vault puzzle requirements

4.6. Bomb

ID	Pri.	Specification
R-153	**W**	The bomb can communicate with the hub using a TCP socket connection
R-154	**M**	The bomb can sync. time using the WiFi connection
R-155	**W**	The bomb can retrieve, and store a given code in order to verify it later on input
R-156	**S**	The bomb can be paired to a puzzlebox using the hub’s interface

Table 17. Bomb requirements

5. Preconditions

This section describes the aspects of the project which have been set as preconditions and cannot be changed.

ID	Precondition
R-160	The delivery of components cannot take longer than two weeks.
R-161	The price of a single puzzle box is not higher than €150.
R-162	The existing games are used in the puzzle box.
R-163	The puzzle box is not allowed to make a connection with the Avans network (Eduroam).
R-164	The bomb hardware cannot be changed.

Table 18. Preconditions

6. Documentation

This section lists requirements that apply to documentation produced during this project.

ID	Pri.	Specification
R-157	**S**	All documentation is written according to the style guide [3]
R-158	**S**	All documentation is manually checked for spelling and grammar mistakes before being published
R-159	**M**	All project documents are examined once by Jonathan Overes from Avans

Appendix A: Attachments

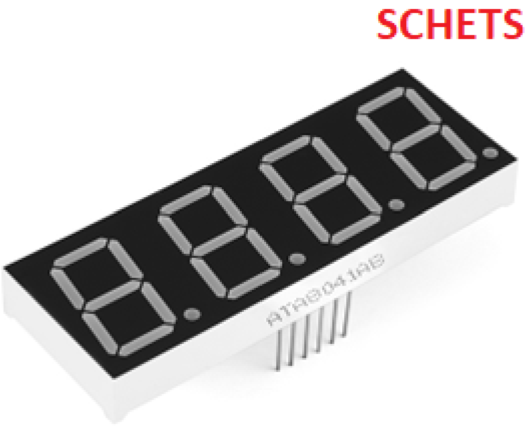


Figure 1. 7 Segment 4 digit screen (sketch)

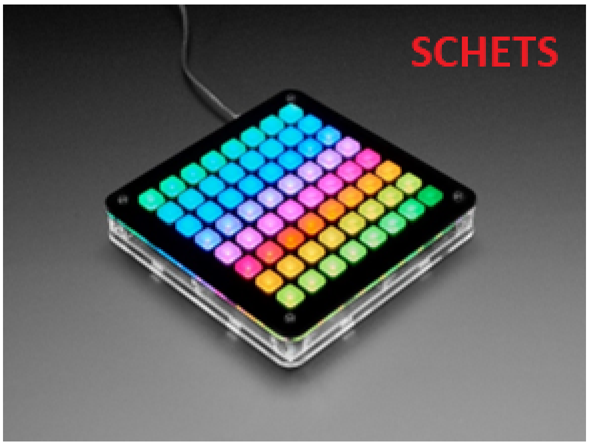


Figure 2. NeoTrellis example (sketch)

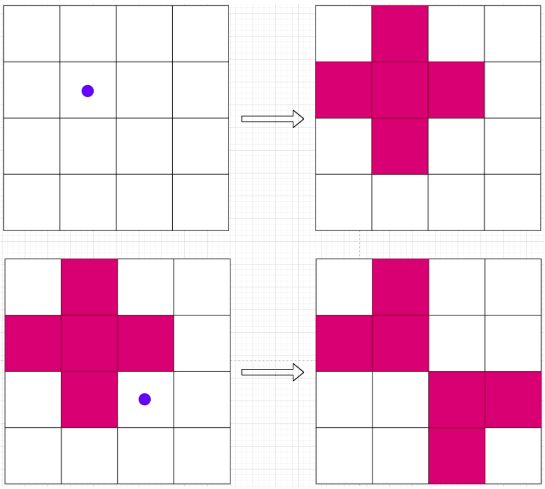


Figure 3. Toggling LEDs after the user pressed on the button (purple dot)

		🟥				🟥	🟥
	🟥	🟥					🟥
				🟥			
		🟥					
🟥				🟥	🟥		
🟥	🟥		🟥				
🟥		🟥			🟥		
			🟥	🟥			

Figure 4. Starting pattern of the NeoTrellis puzzle

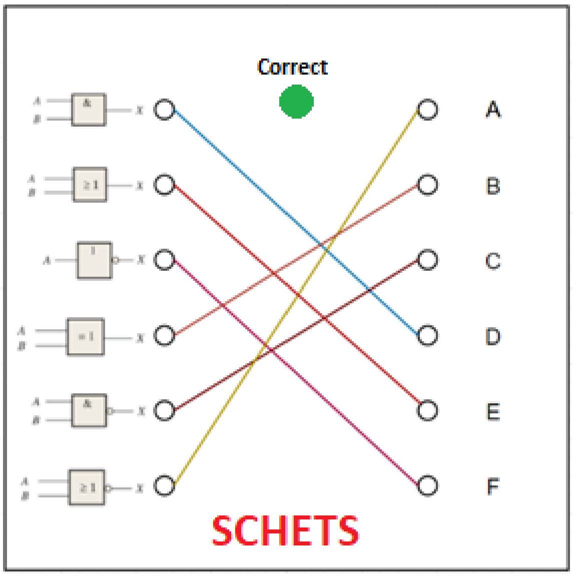
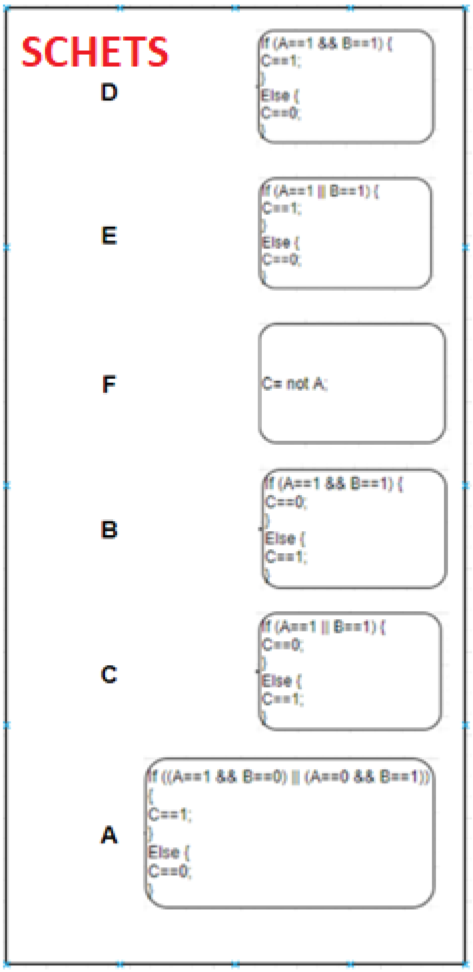


Figure 5. Software puzzle example with logical ports (left) and letters A through F (right)



D

```
If (A==1 && B==1) {
C==1;
}
Else {
C==0;
}
```

E

```
If (A==1 || B==1) {
C==1;
}
Else {
C==0;
}
```

F

```
C= not A;
```

B

```
If (A==1 && B==1) {
C==0;
}
Else {
C==1;
}
```

C

```
If (A==1 || B==1) {
C==0;
}
Else {
C==1;
}
```

A

```
If ((A==1 && B==0) || (A==0 && B==1))
{
C==1;
}
Else {
C==0;
}
```

Figure 6. The different code fragments corresponding with the letter A through F

Figure 7. Software puzzle cable example

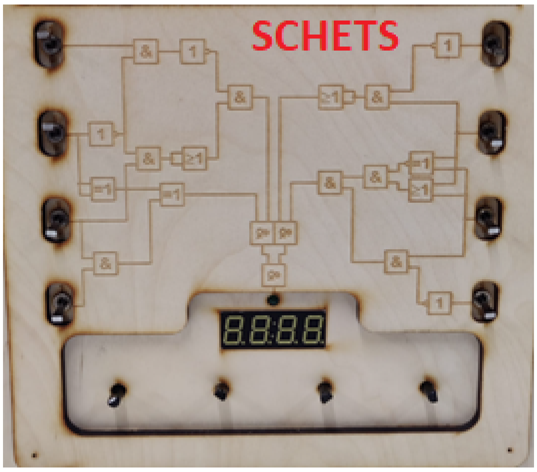


Figure 8. Hardware puzzle on the puzzle box

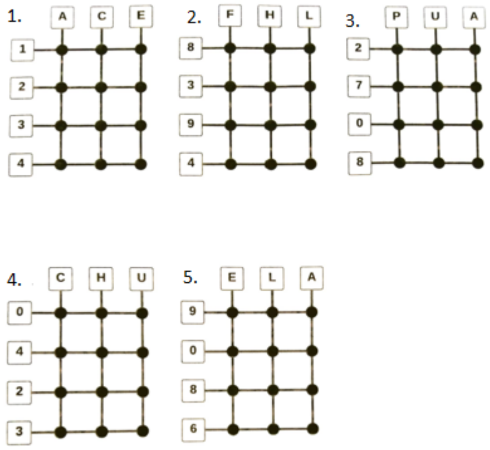


Figure 9. Buttons combinations with level numbers in the top left

Appendix B: References

[1] Wikipedia-contributors, “MOSCOW-Methode.” Apr. 2022, [Online]. Available: https://nl.wikipedia.org/wiki/MoSCoW-methode.

[2] F. Bekema and J. Gense, “Programma van Eisen Puzzel box,” 2023.

[3] L. L. Blansch, E. Hammer, L. Faase, and T. in ’t Anker, “Style guide,” Avans University of Applied Sciences, 2024.

Appendix C: Glossary

RPI
Raspberry Pi

Main board
The main board is the PCB on the bottom of the puzzle box, this communicates with the puzzles and the bomb

Puzzle box hub
The puzzle box hub communicates with the puzzle box and the bomb, as well as helps with configuring them

SID
Security identifiers

game operator
Person who organizes a puzzle box play session