



AGH University of Science and Technology

Report on the project

Baba Jaga patrzy

For classes

Design Laboratory

Elektronika i Telekomunikacja, 3rd year

Paweł, Kierzek, Paweł Tymoczko, Mariusz Furtek

Coordinator: Dr inż. Jacek Stępień

22.01.2023

1. Project's goal

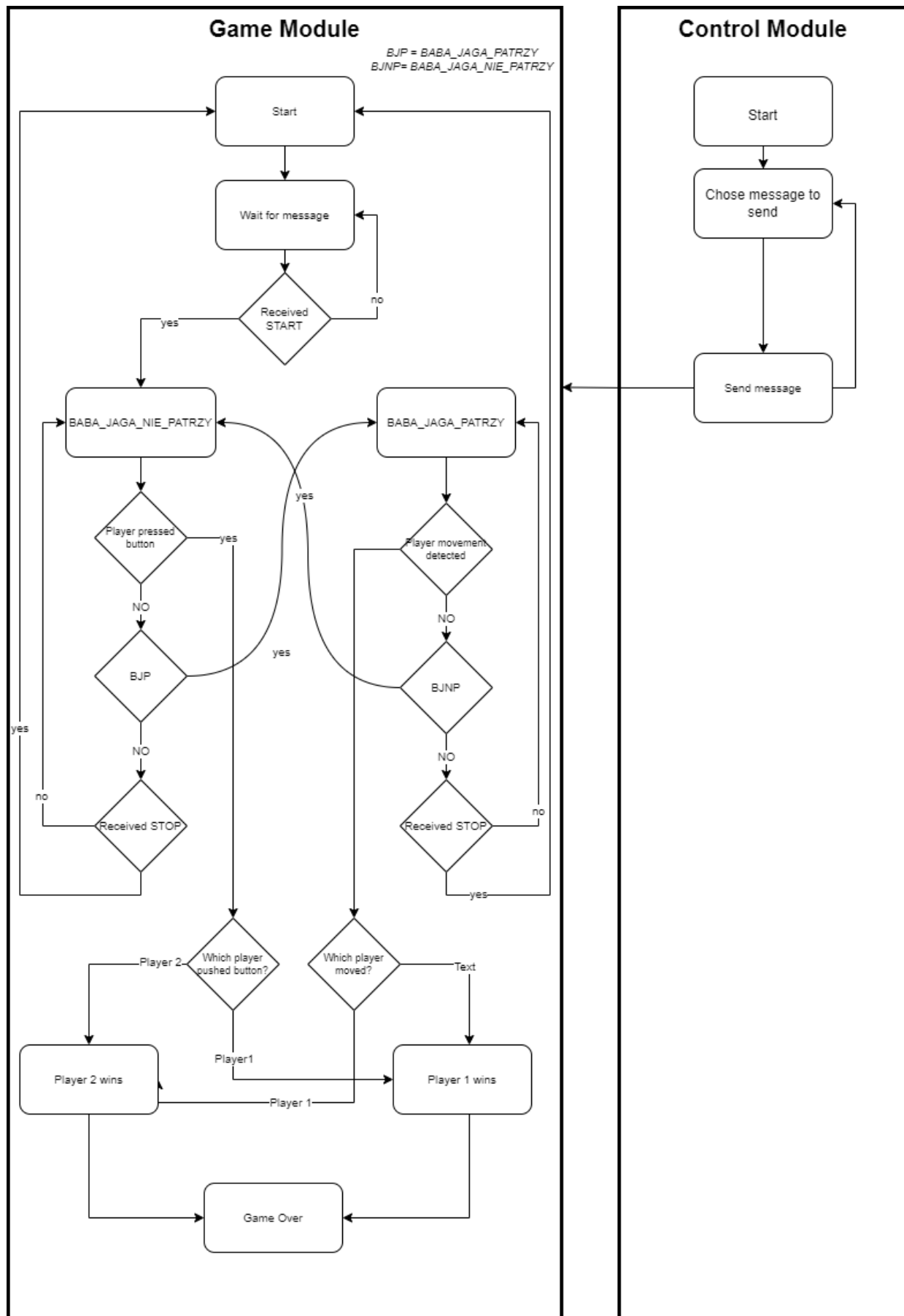
Goal of the project was to create hardware-software system which is the implementation of the real-life game "*Baba Jaga Patrzy*".

2. Design assumptions

2.1. System functionality

Game system is divided into 2 modules communicating via common Bluetooth interface:

- Game Module - core of the game, if initialized can be in 2 states:
 - *BABA_JAGA_PATRZY* – detects players movement and reports it,
 - *BABA_JAGA_NIE_PATRZY* – detects pushed buttons and reports it.
- Control Module – allows to send 4 different messages to the Game Module *START*, *STOP*, *BABA_JAGA_PATRZY*, *BABA_JAGA_NIE_PATRZY*. Depending on the message type specified action are taken by another module:
 - *START* – initialize the game,
 - *STOP* – finish the game,
 - *BABA_JAGA_PATRZY* – change the game to "*BABA_JAGA_PATRZY*" state,
 - *BABA_JAGA_NIE_PATRZY* - change the game to "*BABA_JAGA_NIE_PATRZY*" state.



Picture 1. Game mechanism block diagram.

2.2. User Interface

The game is designed for 3 players: 2 contestants and a game master.

The game master is responsible for controlling the game flow: starting/stopping the game and enabling/disabling movement detection.

If game master initializes the game, green diode lights up and keep on until the end of the game.

Contestant's goal is to arrive to the location of the button and to push it before another player – first to do so wins. Contestant must watch out for the state of red diode. If it lights up, both players must stop and do not move. Otherwise, the contestant whose movement is detected, loses.

After player pushing the button or sensors detecting movement, high pitch sound is to be heard. One of two diodes (yellow corresponding to player no1 and green for player 2) lights up to inform who is the game winner.

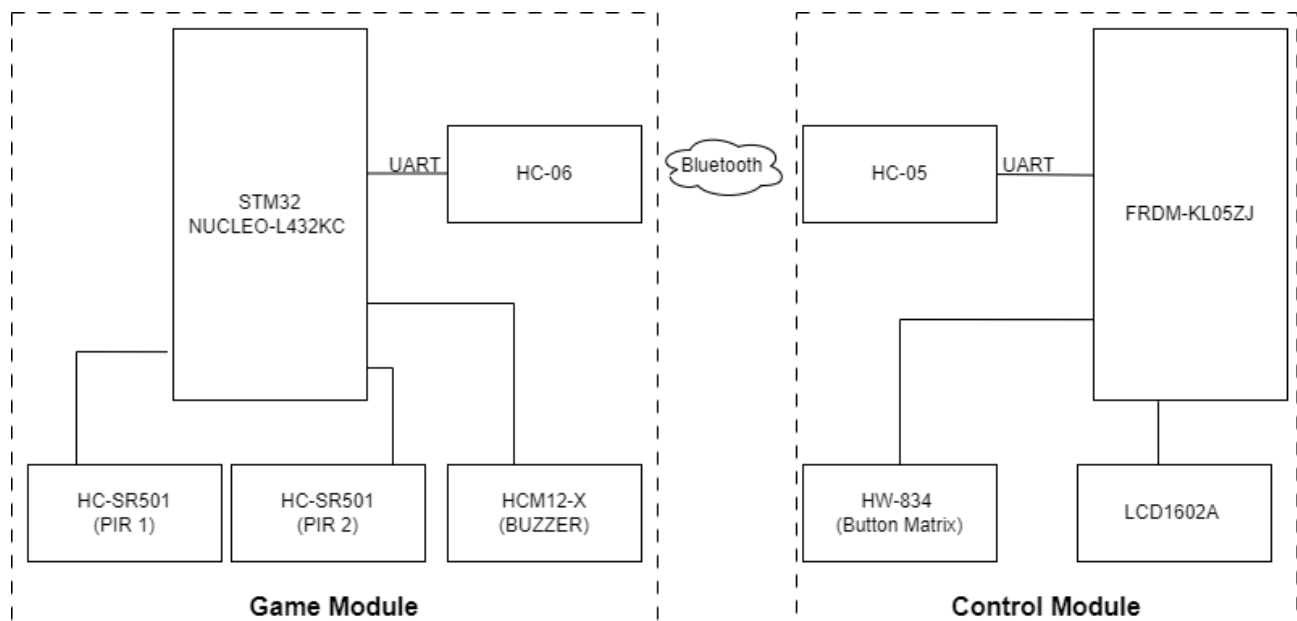
2.3. Hardware

Table 1. Electronic elements chosen for project implementation.

Element name	Module	Application
STM32 NUCLEO-L432KC	Game Module	Controls game flow, PIR movement detectors and all other game peripherals (buzzer, diodes), receives data from HC06 Bluetooth module
FRDM-KL05ZJ	Control Module	Sends messages to Bluetooth module HC05, based on pressed buttons, displays data on lcd
HC-05 BLUETOOTH MODULE	Control Module	Master of Bluetooth communication, transmit messages from Control Module to Game Module
HC-06 BLUETOOTH MODULE	Game Module	Slave of Bluetooth communication, receive messages from Control Module
HC-SR501 PIR MOTION DETECTOR	Game Module	Detects movement when Game Module is in <i>BABA_JAGA_PATRZY</i>

		state and reports game rules violation
HCM12-X BUZZER	Game module	Plays sound when game ends
LCD1602A (with I2C expander)	Control Module	Displays messages, currently transmitted to Game Module
HW-834 4x4 BUTTON MATRIX	Control Module	Pressing button corresponds to sending predefined messages

Datasheets of elements are to be found in project's Git repository. Apart from listed components, other such as diodes, push button and resistors were used.



Picture 2. Project's modules block diagram.

2.4. Software

Main functionality of Control Module's program is to handle button matrix interrupts and transmit messages over UART to HC-05 Bluetooth module.

Game module's program:

- handles interrupts from PIRs which detects players' movement,
- lights up appropriate diode to inform players about game status,
- receives messages over UART from HC-06 and based on the input turns on or off PIRs' interrupts.

Both Game Module and Control Module application were chosen to be written in C, using STM32CubeIDE.

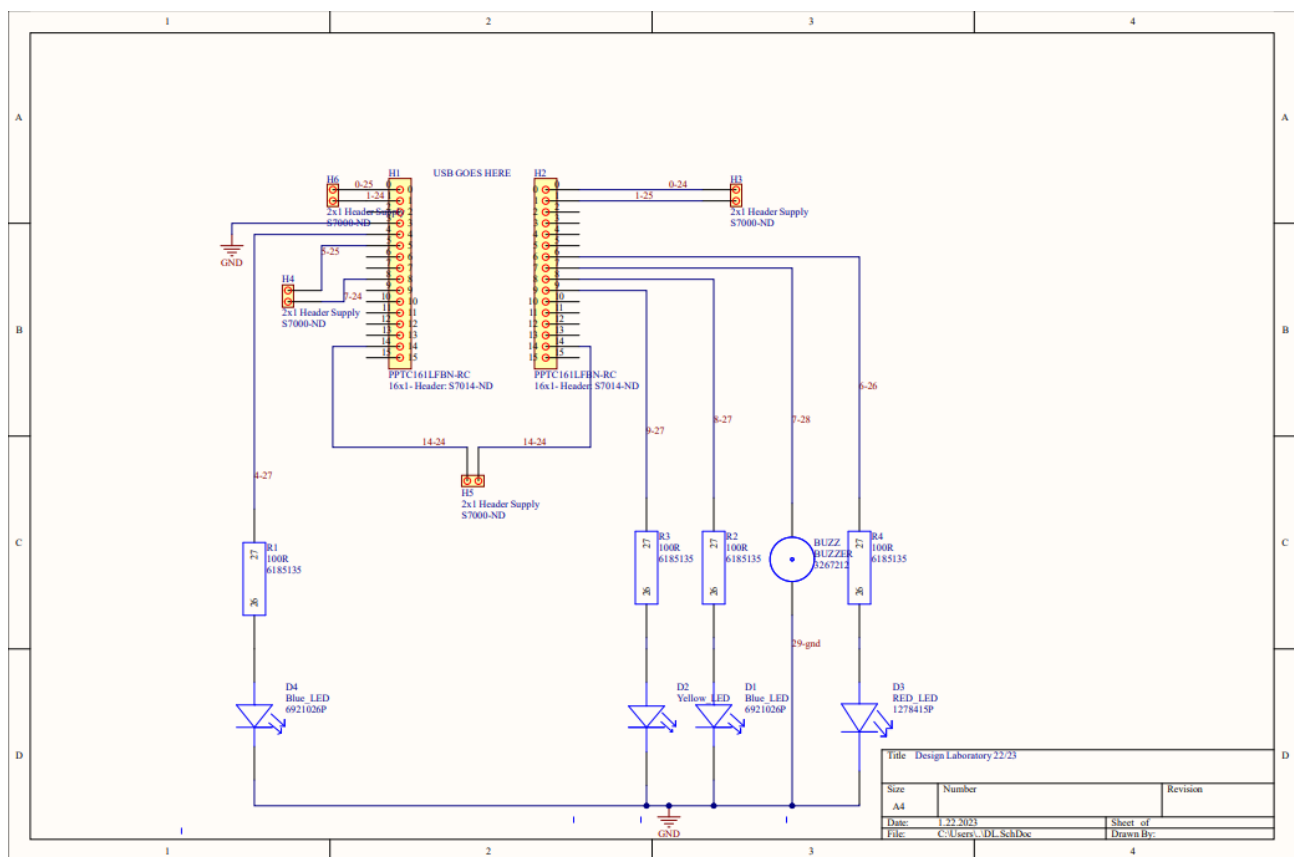
3. Project Realization

Both modules have been designed according to presumptions described in previous part 2. *Design assumptions*. However, during integration process issue concerning HC05 and HC06 modules has occurred. There has been problem with entering command mode on Bluetooth modules, thus with pairing device for wireless communication. To mitigate impact on whole project and to ensure feature of wireless game controlling, new solution has been proposed: to substitute intended Control Module with simple mobile android application.

Created app has all assumed functionality and from user perspective is easier to use.

3.1. Game Module's electronic schematic and PCB layout

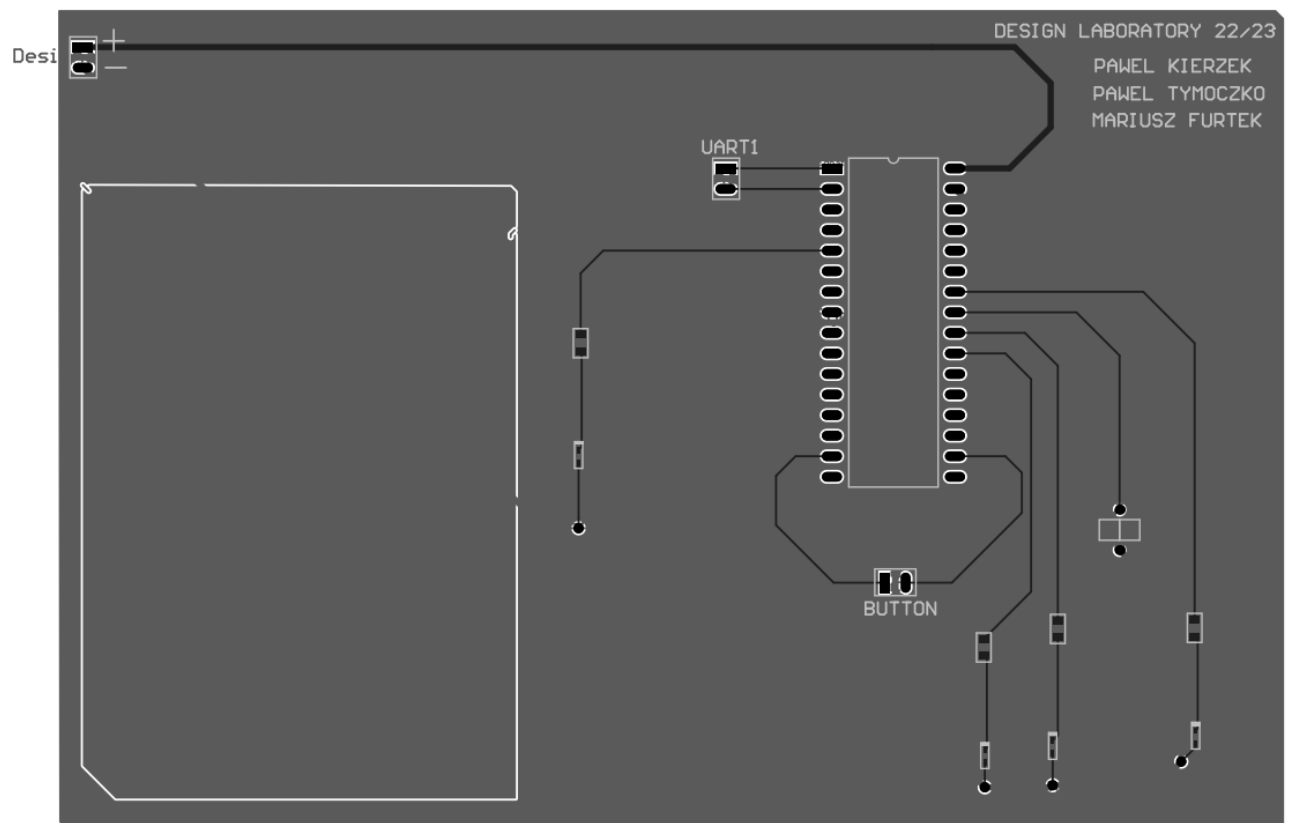
Game Module's elements were connected according to the schematic in the *Picture 3*.



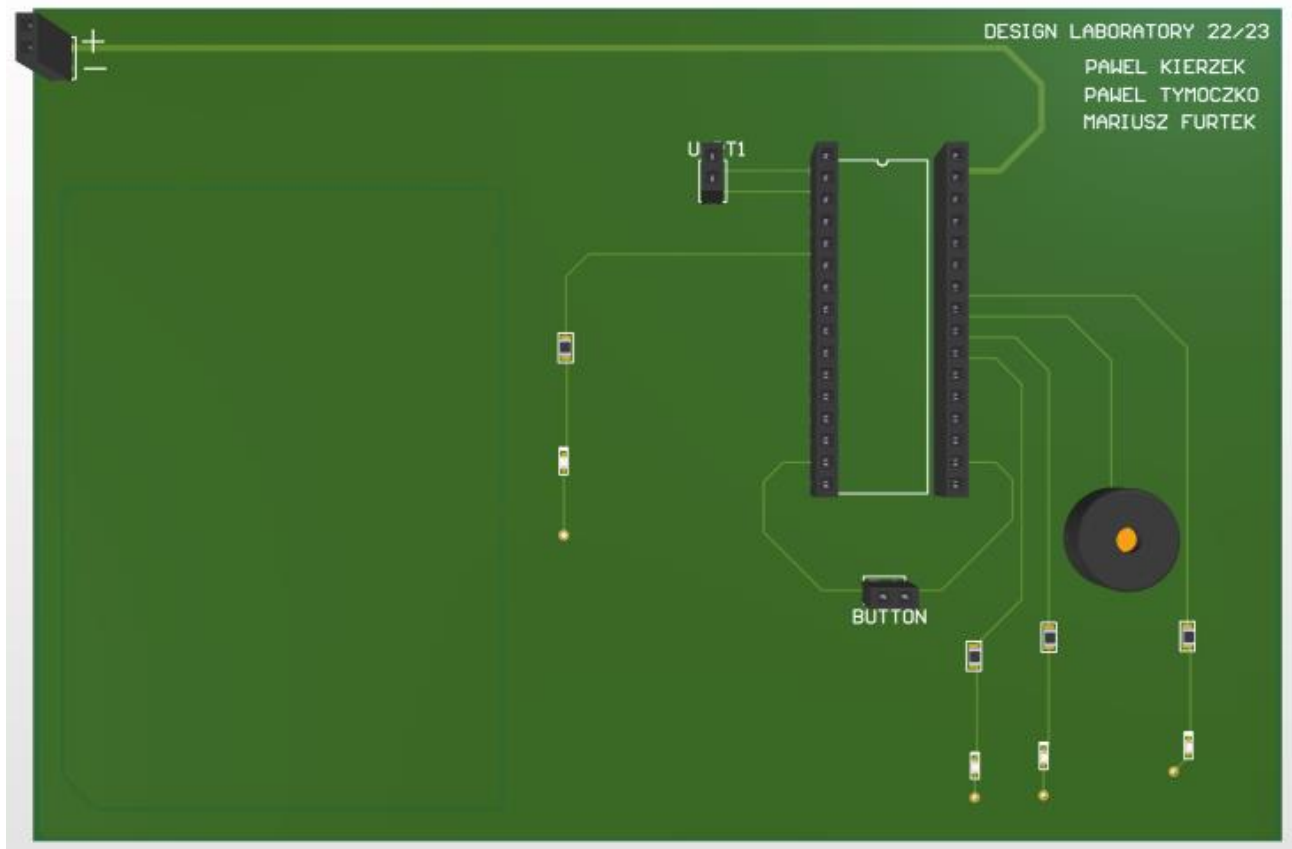
Picture 3. Game Module electronic schematic

On the schematic both PIRs and push buttons power on have not been put. PIRs are powered on according to the datasheet specification to +5V and GND and push buttons are connected to the power supply +5V.

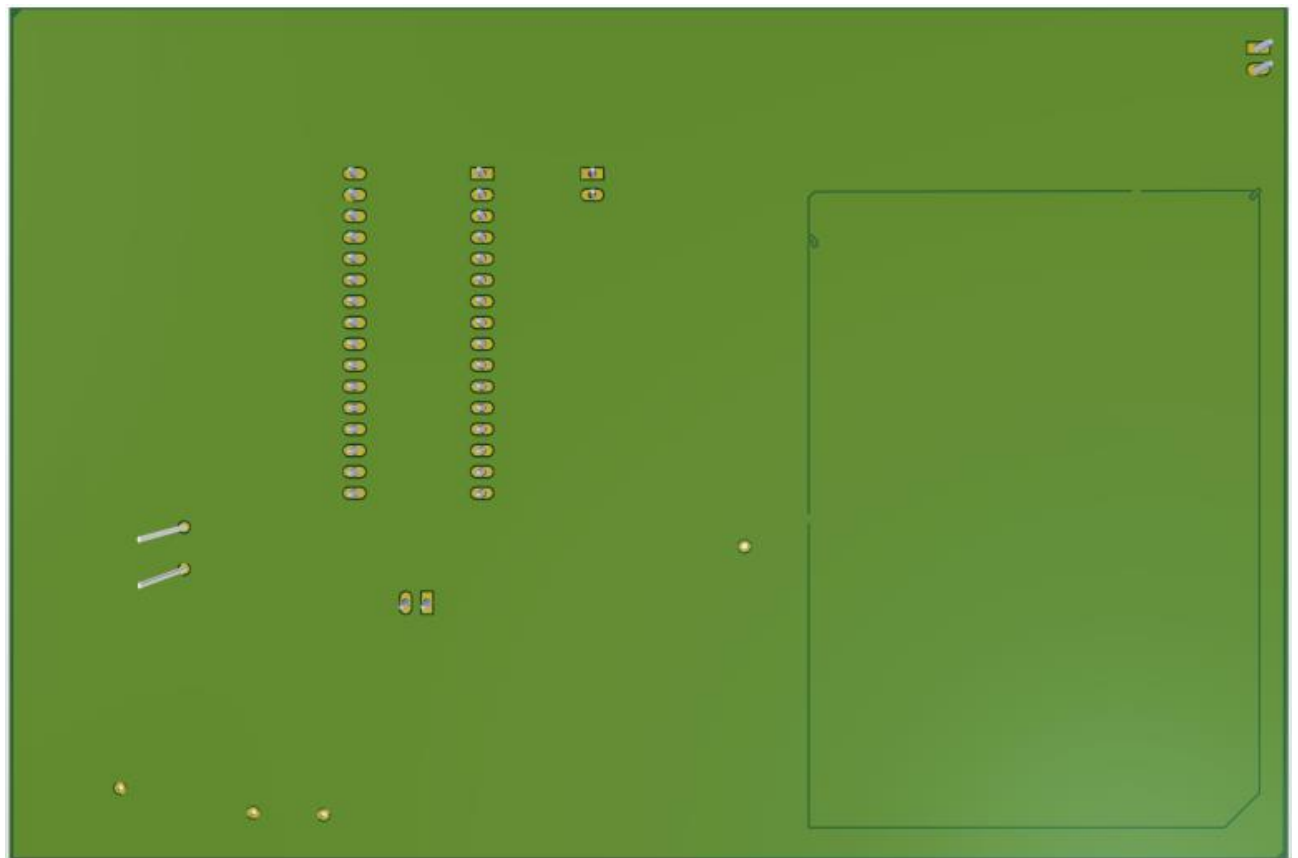
Based on the electronic schematic attempt at creating PCB layout has been made. Unfortunately, created layout correctness has not been confirmed with necessary analysis.



Picture 4. PCB Layout attempt



Picture 5. Expected frontside PCB 3D visualization.



Picture 6. Expected backside PCB 3D visualization.

3.2. Game module software

Table 2. Used GPIO pins, their configuration mode and functionality.

Pin name	User labels	Mode	Function
PB0	PIR_SIG1	External Interrupt Mode with rising edge trigger detection	Receives information about detected movement by PIR1
PB1	PIR_SIG2	External Interrupt Mode with rising edge trigger detection	Receives information about detected movement by PIR2
PA3	PIR_OUT1	Output Push Pull	Lights up yellow diode if player 1 wins
PA4	PIR_OUT2	Output Push Pull	Lights up green diode if player 2 wins
PA6	BABA_JAGA_PATRZY	Output Push Pull	Lights up red diode if Game Module is in BABA_JAGA_PATRZY mode
PA5	BUZZER	Output Push Pull	Generate sound if game finishes
PA12	GAME_ON	Output Push Pull	Lights up green diode when game is initialized
PA10	USART1_RX	UART	Enables Bluetooth communication with Control Module
PA9	USART1_TX	UART	
PB3	BUTTON_1	External Interrupt Mode with rising edge trigger detection	Receives information if player no. 1 pressed button
PB4	BUTTON_2	External Interrupt Mode with rising edge trigger detection	Receives information if player no. 2 pressed button

All pins above are configured to the appropriate mode and initialized in Game Module program's main function.

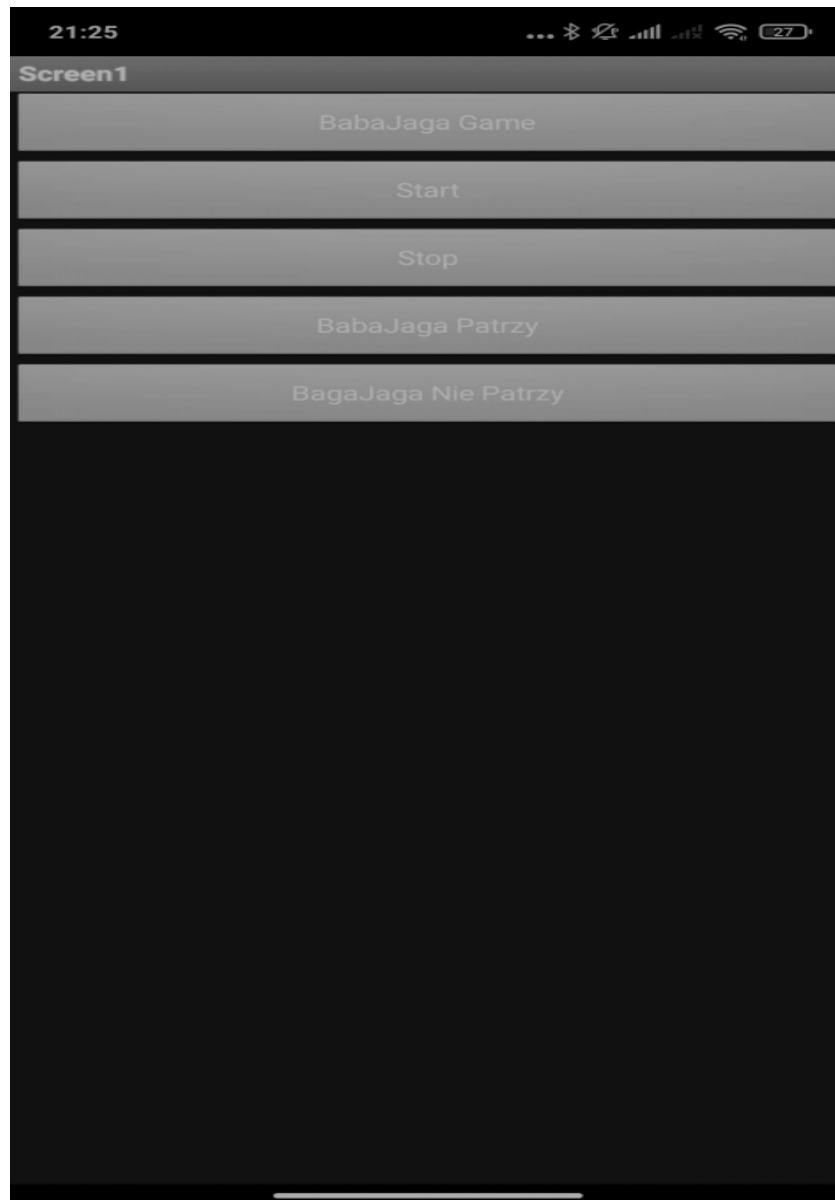
In the program game loop, program is waiting for new messages from UART1. If new message is the same as one of defined below, it takes corresponding action:

- START – 'A': initializes the game, reset all diode pins and buzzer pin, clear players' score, light up GAME_ON diode, set state to BABA_JAGA_NIE_PATRZY,
- STOP – 'B': finishes game, reset all diode pins and buzzer pin, clear players' score,

- BABA_JAGA_PATRZY – 'C': enables PIRs interrupts to allow movement detection,
- BABA_JAGA_NIE_PATRZY – 'D': disable PIRs interrupts,

3.3. Control Module mobile application

The application has been created using MIT App Inventor online tool in Scratch programming language. It allows to communicate with HC-05 module and send defined Game Module messages.



Picture 7. Screen shot of Control Module App GUI

3.4. Workflow and labor division

The project's workflow and labor division are to be found in the spreadsheet named *workflow.xlsx* in git repository.

For project management and team organization Trello Kanban board was used.

4. Resources

4.1. Project resources

- GIT repository: <https://github.com/Pawlich0/DesignLab-BabaJaga>
- Trello Kanban board: <https://trello.com/b/rw4OTdgo/baba-jaga-patrzy>

4.2. External sources

- MIT App Inventor: <https://appinventor.mit.edu/>
- CAD footprints: <https://www.digikey.pl/>