|  |  |
| --- | --- |
|  | **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

21.01.2023

# Project overview

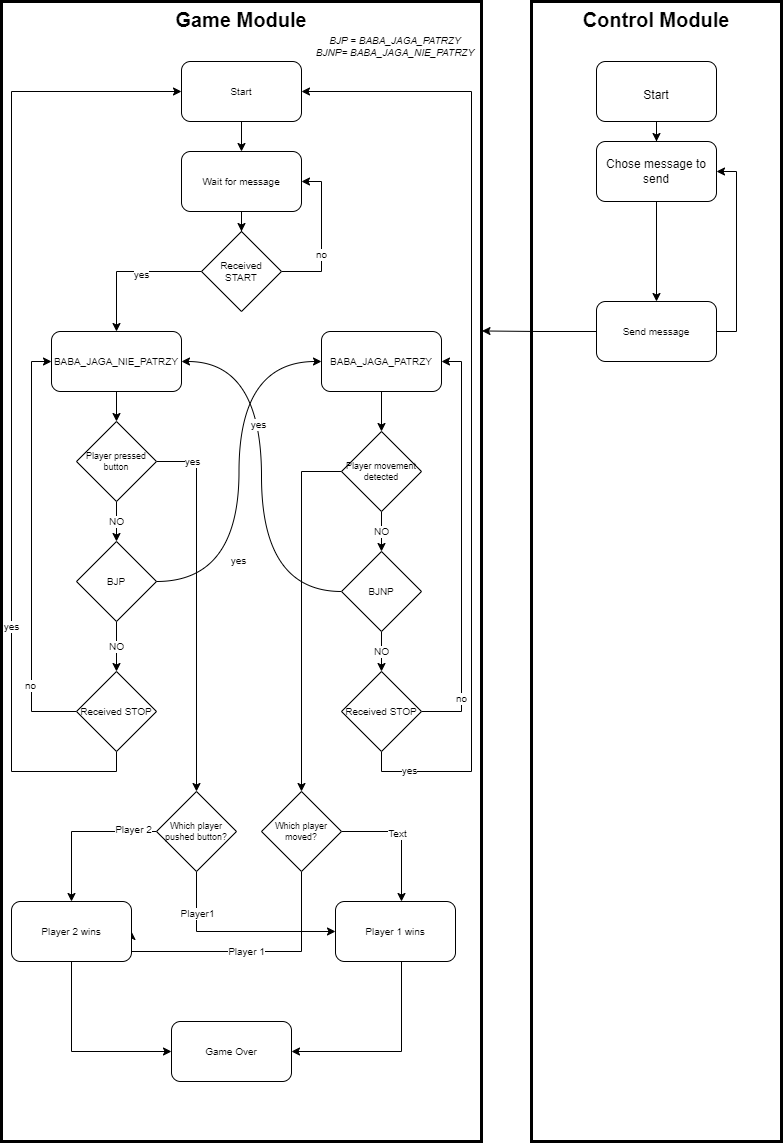
Goal of the project is to create hardware-software system which is the implementation of the real-life game *“Baba Jaga Patrzy”.*

# Design assumptions

* 1. System functionality

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

* Game Module (slave) - core of the game, if initialized can be in 2 states:
  + *BABA\_JAGA\_PATRZY –* detects players movement and reports it,
  + *BABA\_JAGA\_\_NIE\_PATRZY –* does not detect movement,
* Control Module (master) – 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 the slave module:
  + *START* – initialize the game,
  + *STOP* – finish the game,
  + *BABA\_JAGA\_PATRZY –* change the game to state: “*BABA\_JAGA\_PATRZY”,*
  + *BABA\_JAGA\_NIE\_PATRZY* - change the game to state: “*BABA\_JAGA\_NIE\_ PATRZY”,*

**

*Picture 1. Game mechanism block diagram.*

* 1. 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 (green corresponding to player no1 and yellow for player 2) lights up to inform who is the winner of the game.

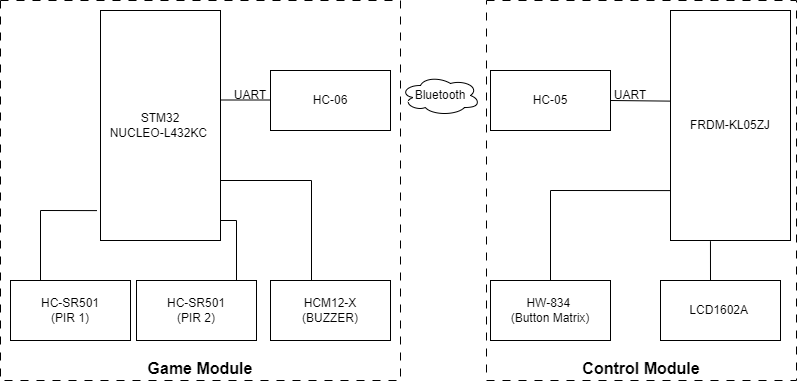
* 1. Hardware

For electronic system below mention hardware set was chosen.

*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 configuration, transmit messages from Control Module to Game Module |
| HC-06 BLUETOOTH MODULE | Game Module | Slave of Bluetooth configuration, receive messages from Control Module |
| HC-SR501 PIR MOTION DETECTOR | Game Module | Detects movement when Game Module is in *BABA\_JAGA\_PATRZY* state and reports game rules violation |
| HCM12-X BUZZER | Game module | Play 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 above-mentioned elements are to be found in project’s git repository (see resources). Apart from listed components, other such as diodes, push button and resistors were used.



*Picture 2. Project’s modules block diagram.*

* 1. Software   
     Main functionality of Control Module’ 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 form HC-06 and based on the input turn on or turn off PIRs’ interrupts.

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

# 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.

* 1. Game Module’s electronic schematic and PCB layout

(Here will be layout and description of what does what)

* 1. 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 green diode if player 1 wins |
| PA4 | PIR\_OUT2 | Output Push Pull | Lights up yellow 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,
  1. Control Module mobile application
  2. Workflow and labor division

The project’s workflow and labor division is to be found in the spreadsheet named *workflow.xslx* in git repository.

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

# Summary

* 1. Lessons learned (XD?)

(What problems were met, if they were solved, how was it done, otherwise how could we prevent it (maybe at initialization level))

* 1. Possible future project development
  2. Resources
* GIT repository: <https://github.com/Pawlicho/DesignLab-BabaJaga>
* Trello Kanban board: <https://trello.com/b/rw4OTdqo/baba-jaga-patrzy>