Handheld Game Console Using STM32F4 Discovery Board with USART Communication

1. Introduction and Objective

The goal of this project is to design and build a handheld game console using the STM32F4 Discovery Board which is based on the ARM Cortex-M4 processor which microcontroller is used to handle game logic, joystick input, button-based game switching, audio output, and rendering of game visuals via USART communication to a PC terminal. Simple embedded games like Snake, Tetris, and Breakout can be played on the system, which will also communicate with users through a graphical user interface, joystick controllers, and audio output. In contrast to traditional SPI-based displays, this approach introduces a new output and debugging interface by utilizing USART communication to interact with a terminal-based display/monitor or to communicate with a serial-enabled display module.

Key embedded systems concepts are highlighted in this project, such as serial connection, real-time programming, microcontroller-based graphics control, and designing user interfaces with the Universal Synchronous/Asynchronous Receiver-Transmitter (USART) protocol.

2. Project Scope

This embedded console setup includes the following:

- **Microcontroller**: STM32F407VG on the STM32F4 Discovery board
- **Joystick module** (Analog): ADC1 channels are connected for reading X and Y input
- **Push button**: Connected to GPIOA Pin 0 that is used to switch between games.
- **Audio**: Speaker connected via DAC Channel 1which is triggered using TIM6
- **Display**: USART2 output over PA2 (TX) to a serial terminal
- **LEDs**: PD10, pins are used for debug or game-over indication
- **Power Supply**: USB or battery-powered for portability
- **PA2 & PA6**: USART

From a list of built-in games, the user can choose from a menu interface driven by the USART in real-time. To consider the resource constraints of the embedded platform, each game has a limited amount of visual and logical complexity.

3. Technical Implementation

- **Peripherals and GPIOs Used**

ADC1: Reads analog X/Y position of the joystick**.** It is Configured with 12-bit resolution and software triggering.

USART2: Text data transmission to PC via UART. It has a baud rate of 115200 and used for ANSI-based terminal graphics and debugging output.

DAC1 + TIM6: DAC Channel 1 generates analog PWM audio for sound effects and TIM6 triggers the DAC output using interrupts.

DMA1 Stream 5: Output of DAC audio using memory-to-peripheral streaming.

GPIOA Pin 0: Tactile push button for game switching, read in polling mode.

GPIOD Pin 10: LEDs is used to indicate game over or success events.

TIM6: 1ms base tick used for audio and delay logic.

PA2 & PA6: USART

- **Game Structure**

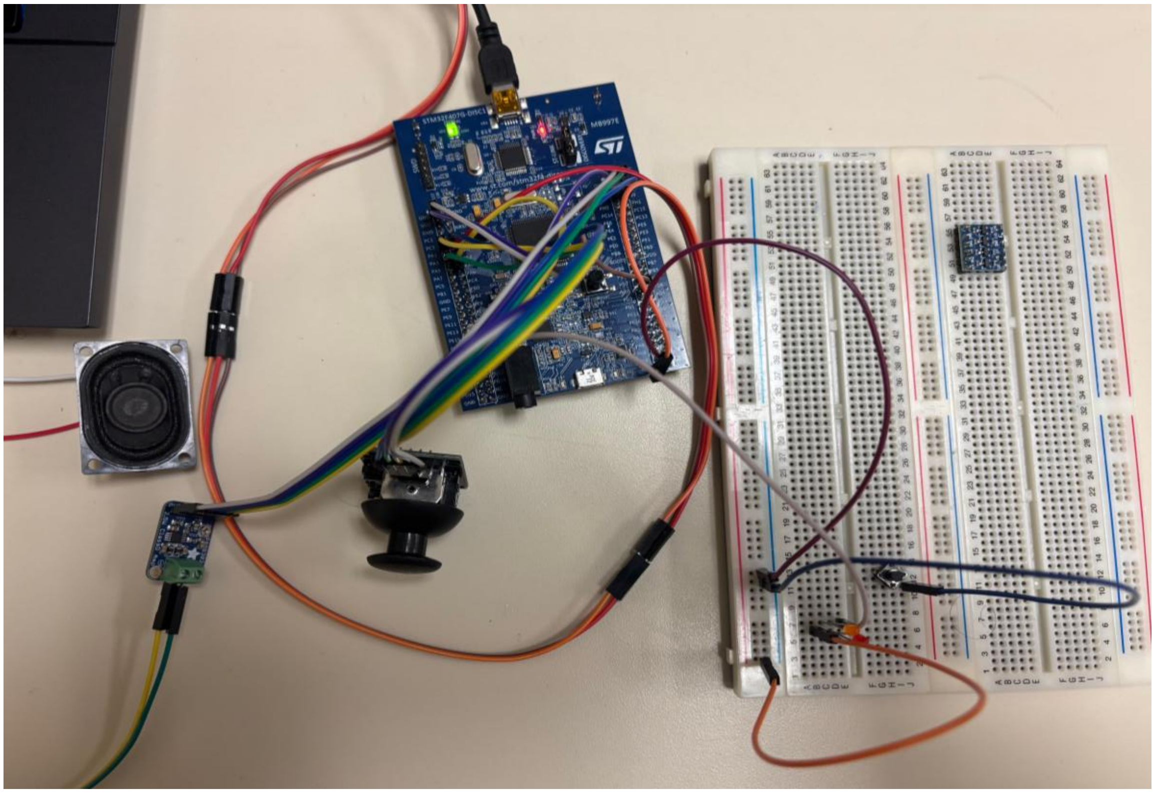
Game Menu: **The menu is** displayed via UART, the user toggles between Snake and Tetris using the button.

Joystick Input: Mapped to movements.

Tetris Game: Tetromino logic based on 4x4 matrix rotation where collision is detected via piece placement, score updating, and game-over handled internally.

Snake Game: It is launched from main loop when game mode toggled and implemented in separate snake_run_game() function.

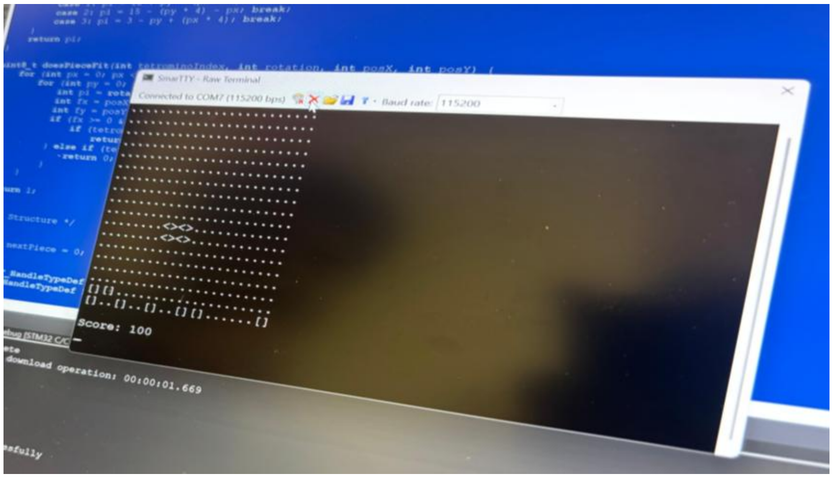
The project's technical implementation revolves around the STM32F407VG microcontroller, which uses the ADC1 peripheral to record joystick input and a push button on GPIOA pin 0 to switch between Tetris and Snake. ANSI escape sequences are used by USART2 to create game images, enabling real-time updates on a serial terminal without a graphical display. Timer 6 (TIM6) activates DAC Channel 1 to provide audio feedback, and DMA manages the smooth waveform playback to a speaker or buzzer that is connected. On-board LEDs are controlled by additional GPIO pins to show game events such as "game over." All gameplay logic is controlled in a loop that modifies display output and user interaction based on joystick and button input. The system is initialized using STM32 HAL drivers via STM32CubeMX.



4. Outcomes

- A completely functional handheld console prototype that communicates display output via USART.
- Three or more playable games, each responding to joystick/button input and transmitting game images or statuses via serial connection.
- Modular source code that showcases USART-driven UI creation and may be modified to support new games.
- The use of lab concepts including as ADC, GPIO interrupts, PWM, and serial communication protocols.
- Complete technical documentation, including schematic diagrams, source code, setup instructions, and USART-based debugging methods.

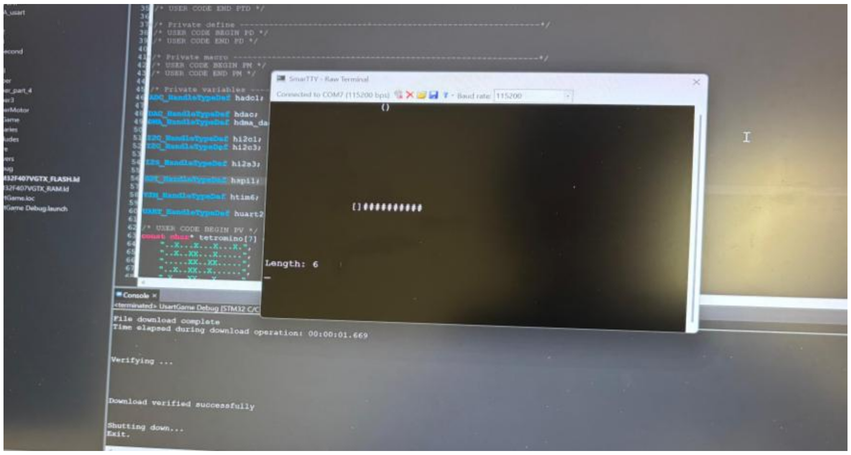
Tetris Game



```
Score: 100
```

```
download operation: 00:00:01.669
```

Snake Game



SmarTTY - Raw Terminal
Connected to COM7 (115200 bps)
Baud rate: 115200
Console
<terminated> UsartGame Debug [STM32 C/C

```
Length: 6
```

```
File download complete
Time elapsed during download operation: 00:00:01.669

Verifying ...

Download verified successfully

Shutting down...
Exit.
```

5. Tools and Resources

Hardware

- STM32F4 Discovery Board (STM32F407VG)
- Analog joystick
- Push button (GPIO input)
- Piezo buzzer/speaker
- USB cable or battery pack
- Jumper wires, breadboard (for prototyping)

Software

- STM32CubeIDE
- STM32CubeMX
- Smart TTY for serial interface

Libraries and Modules

- STM32 HAL drivers (for UART, ADC, DAC, TIM, GPIO)
- Custom modules:

  joystick.h: ADC-based joystick interface

  audio.h: DAC + PWM-based audio playback

  snake.h: Logic and rendering for Snake game

6. Conclusion

This project investigates an alternate approach to embedded game console development that replaces SPI with USART for display or output communications. It shows how to address real-time embedded design difficulties such as human interaction, display control, and auditory feedback using minimum hardware interfaces.

The USART-based technology also facilitates debugging and flexible prototyping across serial terminals, giving it an adaptable foundation for future embedded system development.

Team

Ahsan Shadab Patel
Vedanth Gowda Suresh
Monisha Venkatesh