28/05/2023

EMBEDDED: LINGO

INHOUD

General View 2

1. Libraries: 2

1.1 Custom Libraries: 2

3. Constant and Variable Declarations: 2

4. Helper Functions: 2

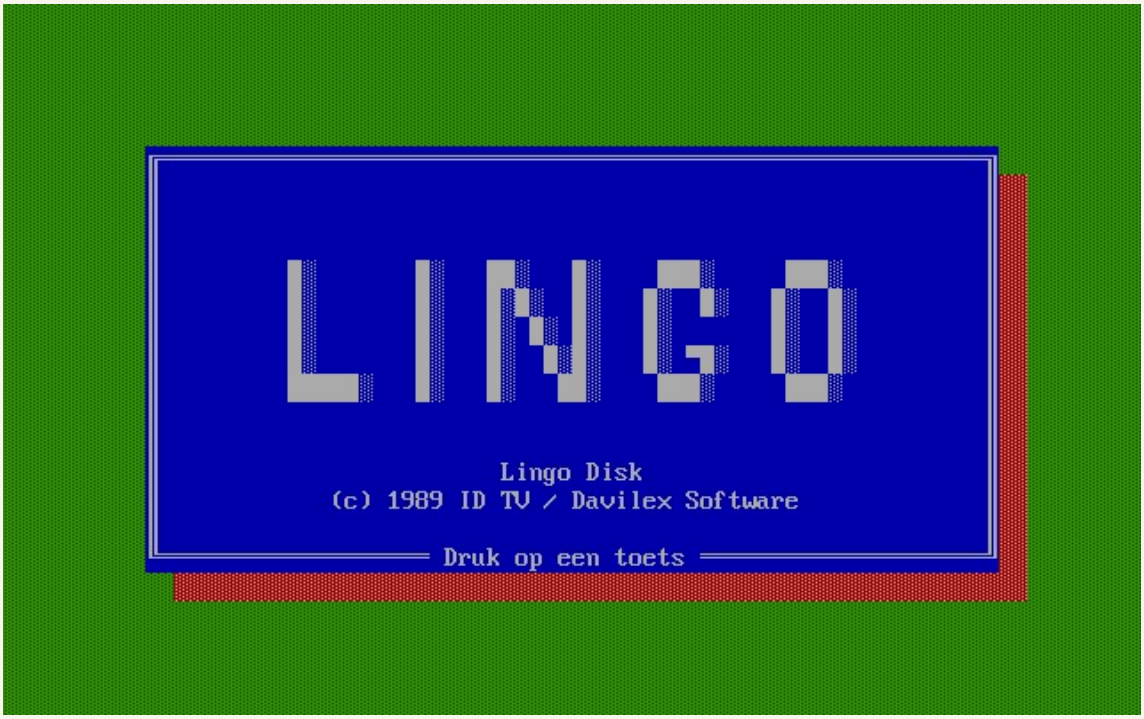
5. Button Handling Functions: 2

6. Interrupt Service Routine: 2

7. Utility Functions: 3

8. Main Function: 3

Code Checklist 4



GENERAL VIEW

1. Libraries:

- `avr/interrupt.h`: For handling interrupts.
- `util/delay.h`: For adding delay functions.
- `avr/io.h`: For AVR I/O definitions.
- `stdlib.h`: For standard library functions.
- `string.h`: For string manipulation functions.

1.1 Custom Libraries:

- `potentiometer_lib.h`: Library for interacting with the potentiometer.
- `led_library.h`: Library for controlling LEDs.
- `display_lib.h`: Library for controlling the segment display.
- `button_lib.h`: Library for handling button presses.
- `buzzer_lib.h`: Library for controlling the buzzer.
- `music_lib.h`: Library for playing music.
- `usart.h`: Library for USART communication.

3. Constant and Variable Declarations:

- Declaration of constants such as `MAX_GUESSES`, `WORD_LENGTH`, `MAX_WORDS`, and `NUM_LEDS`.
- Declaration of global variables such as `selectedWord`, `guessedWord`, `selectedIndex`, `numGuesses`, and `guesses`.
- Definition of the `words` array, storing possible words for the game.

4. Helper Functions:

- `getRandomWord()`: Generates a random word from the `words` array.
- `displaySelectedWord()`: Displays the selected word on the segment display.
- `updateLeds()`: Updates the state of the LEDs.

5. Button Handling Functions:

- `handleButtonPress()`: Handles button presses and implements the game logic.

6. Interrupt Service Routine:

- `ISR(PCINT1_vect)`: Interrupt service routine for button presses.

7. Utility Functions:

- `printGameDescription()`: Prints the game description on the serial monitor.
- `clearSerialMonitor()`: Clears the serial monitor by printing new lines.
- `printGuesses()`: Prints the previous guesses on the serial monitor.

8. Main Function:

- Initializes various components and libraries.
- Generates a random word and sets it as the selected word.
- Initializes the guessed word with the first letter of the selected word and 'A's for the remaining letters.
- Prints the game description.
- Enters an infinite loop where the guessed word is displayed on the segment display and LEDs are updated.

CODE CHECKLIST

- [x] Aansturen van minstens 1 van de LED's (flikkeren, dimmen, faden, ...) - See function `updateLeds()` (line 70).

- [x] Detecteren van het indrukken van minstens 1 van de knoppen; let daarbij op debouncing - See function `handleButtonPress()` (line 114).

- [x] Gebruik van de buzzer met verschillende toonhoogtes - See function `playWinningSound()`, `playLosingSound()`, and `playNotificationSound()`.

- [x] Aansturen van de LED display - See function `displaySelectedWord()` (line 79).

- [x] Gebruik ADC in combinatie met de potentiometer of een andere externe analoge sensor - See function `initADC()` and `readPotentiometer()`.

- [x] Gebruik van zelf geschreven libraries voor alle generieke functionaliteiten van het expansion shield/arduino - Custom libraries used: `button_lib.h`, `buzzer_lib.h`, `display_lib.h`, `embedded_lib.h`, `led_library.h`, `music_lib.h`, `potentiometer_lib.h`, `usart.h`.

- [x] Gebruik van pointers - Pointers used in various parts of the code.

- [x] Oproepen van functies met parameter "by value" en "by reference" - Function calls with both "by value" and "by reference" parameters.

- [x] Gebruik van interrupt - Interrupt service routine `ISR(PCINT1_vect)` (line 120) handles button presses.

- [ ] Gebruik van timer - No explicit use of timers in the code.

- [x] Gebruik van seriële communicatie met de verbonden computer - Serial communication used with the connected computer.

- [x] Gebruik van array - Arrays used to store previous guesses and words.

- [X] Gebruik van struct - No explicit use of structs in the code.

- [?] Gebruik van macro ('s) - No explicit use of macros in the code.

- [x] Dynamische allocatie op de heap (calloc of malloc, free) - Dynamic memory allocation used for allocating memory for guesses.