Practical Execution – Writing the System Code in a Portable Way

Introduction:

In this experiment, we aim to develop a system that incorporates a communication interface (UART) for efficient data exchange and processing. Traditional data communication in systems often relies on standard protocols that may not be optimized for speed and power efficiency. By leveraging the capabilities of a CPU in conjunction with a UART interface within a low power MCU system, this setup is designed to minimize energy consumption and enhance responsiveness. This approach demonstrates how communication interfaces like UART can significantly improve system performance in practical applications.

Hardware Connections Required:

- PB2 - PB0 buttons are connected to the RGB respectively.
- P2.4 is connected to the Buzzer.
- P1.3 is connected to the LDR.
- P2.5- P2.7 are connected to D4-D7 of the LCD screen.

FSM System Requirements:

Each Number of the PC keyboard, the MCU should operate with a different function as bellow:

Menu

1. Blink RGB LED, color by color with delay of X[ms]
2. Count up onto LCD screen with delay of X[ms]
3. Circular tone series via Buzzer with delay of X[ms]
4. Get delay time X[ms]:
5. LDR 3-digit value [v] onto LCD
6. Clear LCD screen
7. Show menu
8. Sleep

Further Explanation:

The default X value is 500 ms.

1. You should blink the RGB light (color by color) with delay X between each color change.
2. Counting up using int variable while keeping the counter when switching states
3. Playing sounds using the buzzer in circular way with delay of X between each frequency from tone series: [1KHz, 1.25KHz , 1.5KHz , 1.75KHz , 2KHz , 2.25KHz , 2.5 KHz].
4. Inputting new X value from the user.
5. Measuring voltage from the LDR **Dinamically** and printing the voltage to the LCD screen.
6. Clean the LCD screen + resetting the counter from ex2.
7. Menu displaying
8. Going into sleep mode.