Code Explanation and Report

Detailed Explanation and Report for the Provided Code

Introduction

The given code is designed for an AVR microcontroller (e.g., ATmega) and implements a control system for a process involving motor control, valve activation, and emergency stop functionality. It uses various peripherals such as UART, PWM, ADC, timers, and external interrupts.

Detailed Explanation

1. Includes and Definitions

- F_CPU: Defines the CPU clock frequency as 8 MHz for delay calculations.
- Includes:
  - avr/io.h for general I/O operations.
  - util/delay.h for delay functions.
  - avr/interrupt.h for using interrupts.

2. UART Communication

- Initialization (UART_init): Configures UART for communication by setting baud rate and enabling RX/TX.
- Data Transmission (UART_send): Transmits one character at a time.
- String Transmission (UART_send_string): Sends strings over UART for status updates.

3. PWM Configuration

- Initialization (PWM_init): Sets up Timer0 in Fast PWM mode for motor speed control.
- Duty Cycle Control: Updates PWM duty cycle using OCR0A based on ADC input.

4. ADC Configuration

- Initialization (ADC_init): Sets reference voltage and enables ADC.

- Reading (ADC_read): Reads analog data from a specified channel and converts it to a digital value.

5. Timer1 Configuration

- Initialization (Timer1_init): Configures Timer1 in CTC mode for time delays.

- Start Timer (start_timer): Starts the timer with a specified delay in milliseconds.

- Interrupt (ISR): Timer1 Compare Match A interrupt toggles a flag when the timer finishes.

6. External Interrupts

- INT0: Configured to handle an emergency stop signal on PD2.

- ISR (INT0_vect): Stops the process, disables outputs, and activates a stop flag.

7. Main Function

- Initialization: Configures UART, PWM, ADC, and Timer1.

- Stop Switch Logic:

  - Handles emergency stop by checking the stop_flag.

  - Resets outputs and displays relevant messages via UART.

- Process Control:

  - Activates valves sequentially with delays.

  - Reads ADC values to adjust motor speed using PWM.

  - Changes motor direction after a set duration.

  - Activates buzzer at the end of the process.

Code Explanation and Report

Workflow

1. Initialization: All peripherals are initialized.

2. Waiting for Start: The system waits for a button press to begin the process.

3. Process Execution:

   - Activates and deactivates valves sequentially with a 2-second delay.

   - Controls the motor speed and direction for 5 seconds each.

   - Ends with an outlet valve activation and buzzer signal.

4. Emergency Handling: Stops the process immediately if the stop switch is pressed.

Expected Questions

1. What is the purpose of F_CPU?

2. How is the ADC value used in PWM?

3. What happens during an external interrupt?

4. Why is the stop_flag necessary?

5. How are delays managed in the code?

Potential Modifications

1. Adjustable Timer Durations: Allow dynamic setting of valve and motor durations via UART.

2. Additional Emergency Features: Add logging or alarms for emergency stops.

3. Enhanced Feedback: Use an LCD display for real-time status updates.

4. Motor Control Improvements: Introduce smoother PWM transitions.

5. Energy Efficiency: Reduce power consumption during idle states.