

Name: Amber Khurshid

Section: BAI-4A

Roll No: 22P-9295

COAL LAB 11

CODE:

```
.INCLUDE"M328pDEF.INC"  
.ORG 0
```

```
    LDI R21, HIGH (RAMEND)  
    OUT SPH, R21  
    LDI R21, LOW (RAMEND)  
    OUT SPL, R21
```

```
    LDI R31, 0xFF;  
    OUT DDRD, R31;  
    OUT DDRB, R31;  
    CBI PORTB, 2;  
    CALL DELAY_2ms  
    LDI R30, 0x38 ; Initialize  
    CALL CMNDWRT  
    CALL DELAY_2ms  
    LDI R30, 0x0E ; Initialize  
    CALL CMNDWRT  
    CALL DELAY_2ms  
    LDI R30, 0x01 ; Initialize  
    CALL CMNDWRT  
    CALL DELAY_2ms  
    LDI R30, 0x06 ; Initialize  
    CALL CMNDWRT  
    CALL DELAY_2ms  
    LDI R30, 'C'  
    CALL DATAWRT  
    LDI R30, 'O'  
    CALL DATAWRT  
    LDI R30, 'A'  
    CALL DATAWRT
```

```
LDI R30, 'L'  
CALL DATAWRT
```

HERE: JMP HERE

CMNDWRT:

```
OUT PORTD, R30  
CBI PORTB, 0  
CBI PORTB, 1  
SBI PORTB, 2  
CALL DELAY_100us  
CBI PORTB, 2  
CALL DELAY_100us  
RET
```

DATAWRT:

```
OUT PORTD, R30  
SBI PORTB, 0  
CBI PORTB, 1  
SBI PORTB, 2  
CALL DELAY_100us  
CBI PORTB, 2  
CALL DELAY_100us  
RET
```

// Delays -----

SDELAY:

```
NOP  
NOP  
RET
```

DELAY_100us:

```
PUSH R29  
LDI R17, 60
```

DRO: CALL SDELAY

```
DEC R29  
BRNE DRO  
POP R29  
RET
```

DELAY_2ms:

```
PUSH R29  
LDI R29, 20
```

LDRO: CALL DELAY_100us
DEC R29

```
BRNE LDRO
POP R29
RET
```

EXPLANATION:

1. Include and ORG Directive

.INCLUDE "M328pDEF.INC": This line incorporates a definition file for the ATmega328p microcontroller, which provides symbolic names for registers and bit positions, making the code more readable and understandable.

- .ORG 0: This command sets the starting point of the program to address 0 in memory, meaning the subsequent instructions will be placed beginning at address 0.

2. Stack Pointer Initialization

- LDI R21, HIGH(RAMEND): Load the high byte of the RAM end address into register R21. RAMEND is a constant defined in the included definition file, representing the highest address in RAM.
- OUT SPH, R21: Store the value in R21 to the Stack Pointer High register (SPH).
- LDI R21, LOW(RAMEND): Load the low byte of the RAM end address into register R21.
- OUT SPL, R21: Store the value in R21 to the Stack Pointer Low register (SPL).

Reason: Initializing the stack pointer is essential for any program that uses the stack for function calls, interrupts, or temporary storage.

3. Port Initialization

- LDI R31, 0xFF: Load the immediate value 0xFF into register R31, setting all bits high (1).
- OUT DDRD, R31: Store 0xFF into the Data Direction Register D (DDRD), configuring all pins of port D as outputs.

- OUT DDRB, R31: Store 0xFF into the Data Direction Register B (DDRB), configuring all pins of port B as outputs.
- CBI PORTB, 2: Clear bit 2 in PORTB, setting it to 0. This is typically used to initialize a specific pin to a known state, usually low.

Reason: Configuring the data direction registers sets the microcontroller's pins as either input or output. Here, all pins on ports D and B are set as outputs, which is necessary for controlling external devices like an LCD.

4. LCD Initialization and Command/Data Writing

- CALL DELAY 2ms: Call a subroutine that provides a 2-millisecond delay. Delays are necessary to give the LCD time to process commands.
- LDI R30, 0x38: Load the LCD function set command (0x38) into register R30, configuring the LCD for 8-bit mode, 2 lines, and 5x8 dot characters.
- CALL CMNDWRT: Call a subroutine to write the command in R30 to the LCD.
- LDI R30, 0x0E: Load the LCD display control command (0x0E) into register R30, turning on the display, enabling the cursor, and disabling blinking.
- CALL CMNDWRT: Call a subroutine to write the command in R30 to the LCD.
- LDI R30, 0x01: Load the LCD clear display command (0x01) into register R30.
- CALL CMNDWRT: Call a subroutine to write the command in R30 to the LCD.
- LDI R30, 0x06: Load the LCD entry mode set command (0x06) into register R30, setting the mode to increment the cursor position and disable display shifting.
- CALL CMNDWRT: Call a subroutine to write the command in R30 to the LCD.
- LDI R30, 'C': Load the character 'C' into register R30.
- CALL DATAWRT: Call a subroutine to write the data in R30 (character 'C') to the LCD.
- Repeat for characters 'O', 'A', 'L': Load each character into R30 and call DATAWRT to write them to the LCD.

Reason: Each step initializes the LCD and configures it for data display. The commands set up the function, control display settings, clear the display, and set entry mode. Writing 'C', 'O', 'A', 'L' displays the word "COAL" on the LCD.

5. Infinite Loop

- HERE: JMP HERE: This creates an infinite loop, causing the program to halt here indefinitely. This is often used to prevent the program from running into uninitialized memory or unwanted instructions.

6. Command Write Subroutine

- CMNDWRT: The subroutine for writing a command to the LCD.
- OUT PORTD, R30: Output the command byte stored in R30 to PORTD.
- CBI PORTB, 0: Clear bit 0 in PORTB (RS = 0 for command mode).
- CBI PORTB, 1: Clear bit 1 in PORTB (RW = 0 for write mode).
- SBI PORTB, 2: Set bit 2 in PORTB (E = 1 to enable the LCD).
- CALL DELAY 100us: Call a subroutine to introduce a short delay for timing (100 microseconds).
- CBI PORTB, 2: Clear bit 2 in PORTB (E = 0 to latch the data).
- CALL DELAY 100us: Call a subroutine to introduce another short delay for timing (100 microseconds).
- RET: Return from the subroutine.

7. Data Write Subroutine

- DATAWRT: The subroutine for writing data to the LCD.
- OUT PORTD, R30: Output the data byte stored in R30 to PORTD.
- SBI PORTB, 0: Set bit 0 in PORTB (RS = 1 for data mode).
- CBI PORTB, 1: Clear bit 1 in PORTB (RW = 0 for write mode).
- SBI PORTB, 2: Set bit 2 in PORTB (E = 1 to enable the LCD).

- CALL DELAY 100us: Call a subroutine to introduce a short delay for timing (100 microseconds).
- CBI PORTB, 2: Clear bit 2 in PORTB (E = 0 to latch the data).
- CALL DELAY 100us: Call a subroutine to introduce another short delay for timing (100 microseconds).
- RET: Return from the subroutine.

8. Delay Subroutines

- SDELAY:
 - NOP: No operation (2 cycles).
 - NOP: No operation (2 cycles).
 - RET: Return from subroutine.
- DELAY 100us:
 - PUSH R29: Save the current value of register R29.
 - LDI R17, 60: Load immediate value 60 into R17.
 - DRO:
 - * CALL SDELAY: Call the short delay subroutine.
 - * DEC R17: Decrement R17.
 - * BRNE DRO: Branch to DRO if R17 is not zero.
 - POP R29: Restore the value of register R29.
 - RET: Return from subroutine.
- DELAY 2ms:
 - PUSH R29: Save the current value of register R29.

- LDI R29, 20: Load immediate value 20 into R29.
- LDRO:
 - * CALL DELAY 100us: Call the 100 microsecond delay subroutine.
 - * DEC R29: Decrement R29.
 - * BRNE LDRO: Branch to LDRO if R29 is not zero.
- POP R29: Restore the value of register R29.
- RET: Return from subroutine.

Summary

- ****Initialization****: The program starts by initializing the stack pointer and configuring all pins on ports D and B as outputs.
- ****LCD Setup****: The LCD is set up through a series of commands, each followed by delays to ensure it is ready to receive data.
- ****Data Writing****: The program then writes the characters "COAL" to the LCD.
- ****Infinite Loop****: Finally, the program enters an infinite loop to halt further execution.

This sequence sets up the hardware and displays a string on an LCD, demonstrating a basic microcontroller program for interfacing with an external display. The subroutines DELAY 2ms, CMNDWRT, and DATAWRT handle the timing and specifics of writing to the LCD.