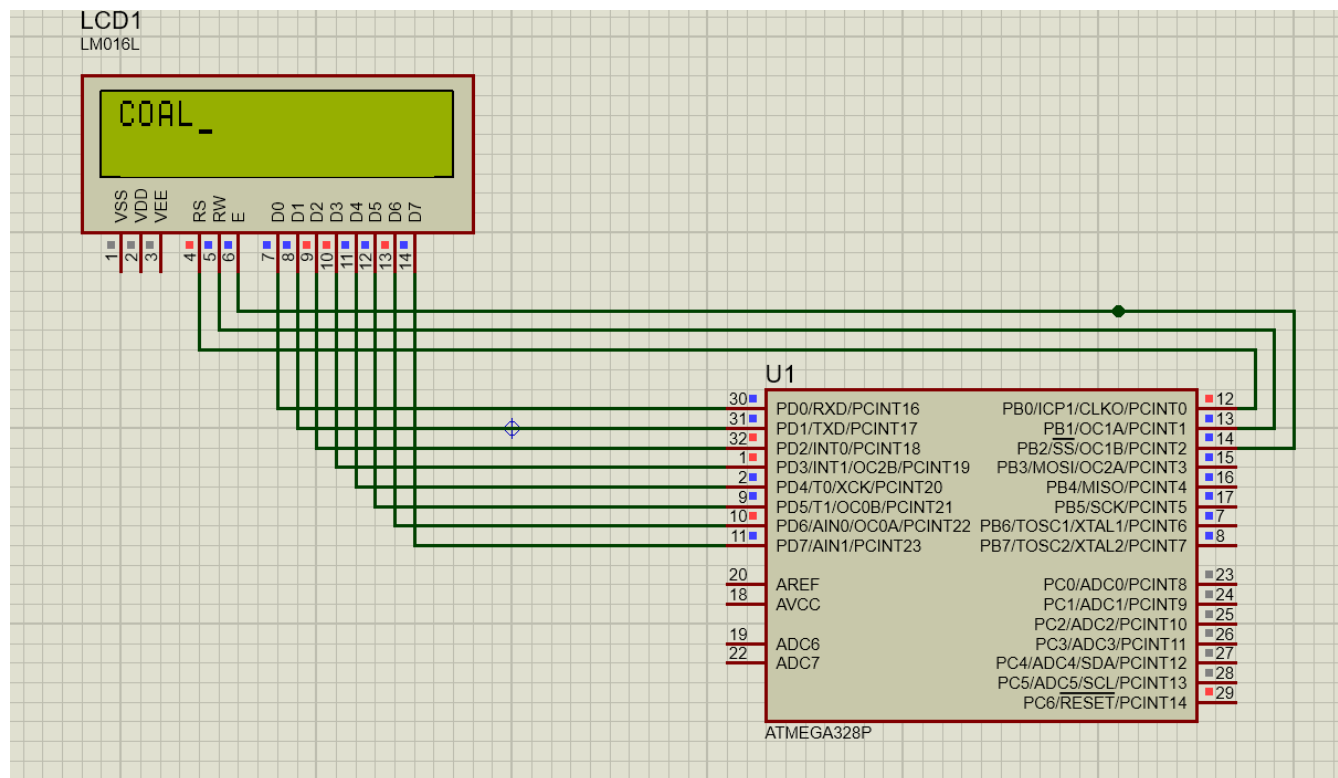


Manual for ATmega328P and LCD Interfacing

Introduction

This manual explains the concepts and corrections for interfacing an LCD with an ATmega328P microcontroller using assembly language. The goal is to display the word "COAL" on the LCD screen. We will go through the pin configurations, the connections, and the assembly code required for this operation.



Components and Pin Configurations

ATmega328P Microcontroller

The ATmega328P microcontroller has several types of pins:

- **Digital I/O Pins:** 23 pins labeled as PC0-PC6 (7 pins), PD0-PD7 (8 pins), PB0-PB7 (8 pins)
- **Power Pins:** VCC, AVCC, GND, AREF (reference voltage)
- **Oscillator Pins:** PB6, PB7 (used for external crystal oscillator)
- **Analog Pins:** ADC0-ADC5 (PC0-PC5, 6 pins)

LCD Display (LM016L)

The LCD has the following important pins:

- **VSS (Pin 1):** Ground
- **VDD (Pin 2):** +5V Power Supply
- **VEE (Pin 3):** Contrast Adjustment
- **RS (Pin 4):** Register Select (0 for command, 1 for data)
- **RW (Pin 5):** Read/Write (0 for write, 1 for read)
- **E (Pin 6):** Enable
- **D0-D7 (Pins 7-14):** Data Lines

Connections

Here's how the connections are made between the ATmega328P and the LCD:

1. Power and Ground:

- VSS to GND
- VDD to +5V
- VEE to a potentiometer for contrast adjustment (middle pin of potentiometer)

2. Control Pins:

- RS (LCD Pin 4) to PB0
- RW (LCD Pin 5) to PB1
- E (LCD Pin 6) to PB2

3. Data Lines:

- D0 to PD0
- D1 to PD1
- D2 to PD2
- D3 to PD3
- D4 to PD4
- D5 to PD5
- D6 to PD6
- D7 to PD7

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 of the Assembly Code Line by Line

Includes and Initialization

.INCLUDE "M328pDEF.INC"

- This line includes the header file for the ATmega328P microcontroller, which contains all the necessary register and bit definitions.

.ORG 0

- This directive sets the starting address of the program to 0, which is the reset vector address.

Stack Pointer Initialization

LDI R21, HIGH(RAMEND)

OUT SPH, R21

LDI R21, LOW(RAMEND)

OUT SPL, R21

- LDI R21, HIGH(RAMEND): Load the high byte of the end of RAM address into register R21.
- OUT SPH, R21: Output the high byte to the Stack Pointer High Register (SPH).
- LDI R21, LOW(RAMEND): Load the low byte of the end of RAM address into register R21.
- OUT SPL, R21: Output the low byte to the Stack Pointer Low Register (SPL).

Port Configuration

LDI R31, 0xFF

OUT DDRD, R31

OUT DDRB, R31

- LDI R31, 0xFF: Load the value 0xFF (all bits set) into register R31.
- OUT DDRD, R31: Set all pins of PORTD as output by writing 0xFF to Data Direction Register D (DDRD).
- OUT DDRB, R31: Set all pins of PORTB as output by writing 0xFF to Data Direction Register B (DDRB).

Initial Setup and Delay

CBI PORTB, 2

CALL DELAY_2ms

- CBI PORTB, 2: Clear bit 2 of PORTB (make PB2 low, which is the Enable pin for the LCD).
- CALL DELAY_2ms: Call the subroutine to generate a delay of 2 milliseconds.

LCD Initialization Commands

LDI R30, 0x38

CALL CMNDWRT

CALL DELAY_2ms

LDI R30, 0x0E

CALL CMNDWRT

CALL DELAY_2ms

LDI R30, 0x01

CALL CMNDWRT

CALL DELAY_2ms

LDI R30, 0x06

CALL CMNDWRT

CALL DELAY_2ms

- LDI R30, 0x38: Load the LCD function set command (8-bit, 2 lines, 5x7 dots) into register R30. This command sets the LCD to 8-bit mode with 2 lines and 5x8 font.
- CALL CMNDWRT: Call the command write subroutine to send the command to the LCD.
- CALL DELAY_2ms: Call the subroutine to generate a delay of 2 milliseconds after each command.
- LDI R30, 0x0E: Load the command to turn on the display and cursor into register R30.
- LDI R30, 0x01: Load the command to clear the display into register R30.
- LDI R30, 0x06: Load the command to set the entry mode (increment cursor) into register R30.

Writing Characters to the LCD

```
LDI R30, 'C'  
CALL DATAWRT
```

```
LDI R30, 'O'  
CALL DATAWRT
```

```
LDI R30, 'A'  
CALL DATAWRT
```

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

- LDI R30, 'C': Load the character 'C' into register R30.
- CALL DATAWRT: Call the data write subroutine to send the character to the LCD.
- Repeat for characters 'O', 'A', and 'L'.

Infinite Loop

```
HERE: JMP HERE
```

- HERE: JMP HERE: Create an infinite loop to keep the program running indefinitely.

Subroutines

Command Write Subroutine (CMNDWRT)

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

- OUT PORTD, R30: Output the command stored in R30 to PORTD.
- CBI PORTB, 0: Clear bit 0 of PORTB (RS = 0 for command mode).

- CBI PORTB, 1: Clear bit 1 of PORTB (RW = 0 for write mode).
- SBI PORTB, 2: Set bit 2 of PORTB (Enable high).
- CALL DELAY_100us: Call the delay subroutine for 100 microseconds.
- CBI PORTB, 2: Clear bit 2 of PORTB (Enable low).
- RET: Return from the subroutine.

Data Write Subroutine (DATAWRT)

DATAWRT:

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

- OUT PORTD, R30: Output the data stored in R30 to PORTD.
- SBI PORTB, 0: Set bit 0 of PORTB (RS = 1 for data mode).
- CBI PORTB, 1: Clear bit 1 of PORTB (RW = 0 for write mode).
- SBI PORTB, 2: Set bit 2 of PORTB (Enable high).
- CALL DELAY_100us: Call the delay subroutine for 100 microseconds.
- CBI PORTB, 2: Clear bit 2 of PORTB (Enable low).
- RET: Return from the subroutine.

Delay Subroutines

Short Delay Subroutine (SDELAY)

SDELAY:

```
NOP
NOP
RET
```

- NOP: No Operation (used for generating a short delay).
- RET: Return from the subroutine.

100 Microseconds Delay Subroutine (DELAY_100us)

DELAY_100us:

PUSH R29

LDI R29, 60

DRO: CALL SDELAY

DEC R29

BRNE DRO

POP R29

RET

- PUSH R29: Push the value of R29 onto the stack.
- LDI R29, 60: Load the value 60 into R29.
- DRO: CALL SDELAY: Call the short delay subroutine.
- DEC R29: Decrement the value in R29.
- BRNE DRO: Branch to DRO if R29 is not equal to zero.
- POP R29: Pop the value from the stack back into R29.
- RET: Return from the subroutine.

2 Milliseconds Delay Subroutine (DELAY_2ms)

DELAY_2ms:

PUSH R29

LDI R29, 20

LDRO: CALL DELAY_100us

DEC R29

BRNE LDRO

POP R29

RET

- PUSH R29: Push the value of R29 onto the stack.
- LDI R29, 20: Load the value 20 into R29.
- LDRO: CALL DELAY_100us: Call the 100 microseconds delay subroutine.
- DEC R29: Decrement the value in R29.
- BRNE LDRO: Branch to LDRO if R29 is not equal to zero.
- POP R29: Pop the value from the stack back into R29.
- RET: Return from the subroutine.

Summary of the Assembly Code for LCD Interfacing with ATmega328P

This assembly code is designed to interface an ATmega328P microcontroller with an LCD module, initializing the LCD and displaying the text "COAL" on it. Here's a brief summary of the main components and their functions:

Initialization

1. Includes and Setup:

- The code begins by including the header file M328pDEF.INC for ATmega328P definitions.
- The .ORG 0 directive sets the starting address to 0.

2. Stack Pointer Initialization:

- The stack pointer is initialized to the end of RAM using the HIGH(RAMEND) and LOW(RAMEND) macros.

3. Port Configuration:

- All pins on PORTD and PORTB are set as output by writing 0xFF to DDRD and DDRB.

LCD Initialization

4. Initial Setup and Delays:

- The Enable pin (PB2) is initially cleared.
- A 2ms delay is introduced to ensure the LCD is ready.

5. Sending Initialization Commands to LCD:

- Various commands are sent to the LCD to configure it (e.g., function set, display on, clear display, and entry mode set) using the CMNDWRT subroutine.
- Each command is followed by a 2ms delay to allow the LCD time to process the command.

Displaying Characters

6. Writing Characters to LCD:

- The characters 'C', 'O', 'A', 'L' are written to the LCD using the DATAWRT subroutine.

Infinite Loop

7. Endless Loop:

- The code enters an infinite loop to keep the program running indefinitely.

Subroutines

Command Write Subroutine (CMNDWRT)

- **CMNDWRT:**
 - Sends a command to the LCD by setting the RS and RW lines appropriately and pulsing the Enable line.
 - A short delay is used to ensure the command is properly latched.

Data Write Subroutine (DATAWRT)

- **DATAWRT:**
 - Sends data (characters) to the LCD by setting the RS line high, keeping RW low, and pulsing the Enable line.
 - A short delay ensures the data is latched.

Delay Subroutines

- **SDELAY:**
 - A very short delay subroutine using NOP instructions.
- **DELAY_100us:**
 - Generates a delay of approximately 100 microseconds by calling SDELAY in a loop.
- **DELAY_2ms:**
 - Generates a delay of approximately 2 milliseconds by calling DELAY_100us in a loop.

Summary of Key Points:

- The code initializes the ATmega328P's stack pointer and configures all PORTD and PORTB pins as outputs.
- It initializes the LCD by sending a sequence of commands.
- Characters are displayed on the LCD using appropriate subroutines.
- Delay subroutines ensure proper timing for LCD operations.
- The code enters an infinite loop to keep running indefinitely after displaying "COAL".