LED Blinking:

```
ORG 0000H

MAIN:

    SETB P2.0     ;

    ACALL DELAY   ;

    CLR P2.0      ;

    ACALL DELAY   ;

    SJMP MAIN     ;

DELAY:

    MOV R4, #50   ;

DELAY_OUTER:

    MOV R3, #255  ;

DELAY_INNER:

    DJNZ R3, DELAY_INNER  ;

    DJNZ R4, DELAY_OUTER  ;

    RET         ;

END
```

LED Toggle:

```
ORG 0000H

UP: MOV P2,#55H

ACALL DELAY

MOV P2,#0AAH

ACALL DELAY

SJMP UP

DELAY:MOV R4,#10

H1:MOV R3,#255

H2:DJNZ R3,H2

DJNZ R4,H1

RET

END
```

LED Chaser

```
ORG 0000H

UP: MOV P2,#01H
```

```
ACALL DELAY
MOV P2,#02H
ACALL DELAY
MOV P2,#04H
ACALL DELAY
MOV P2,#08H
ACALL DELAY
MOV P2,#10H
ACALL DELAY
MOV P2,#20H
ACALL DELAY
MOV P2,#40H
ACALL DELAY
MOV P2,#80H
ACALL DELAY
SJMP UP
DELAY: MOV R4,#255
H1: DJNZ R4,H1
RET
END
```

Relay and Bulb:

```
ORG 0000H
    UP: SETB P2.0
          ACALL DELAY
          CLR P2.0
          ACALL DELAY
          SJMP UP
DELAY: MOV R4,#18
     H1: MOV R3,#255
     H2: DJNZ R3,H2
            DJNZ R4,H1
           RET
            END
```

Relay and LED:

```
ORG 0000H
```

```
MOV P1, #00H      ;
MAIN_LOOP:
   SETB P1.0     ;
   ACALL DELAY   ;
   CLR P1.0      ;
   ACALL DELAY   ;
   SJMP MAIN_LOOP;
DELAY:
   MOV R1, #255  ;
DELAY1:
   MOV R2, #255  ;
DELAY2:
   DJNZ R2, DELAY2 ;
   DJNZ R1, DELAY1 ;
   RET          ;
END              ;
```

Square Wave:

```
ORG 0000H
       UP: SETB P2.0
                 ACALL DELAY
                 CLR P2.0
                 ACALL DELAY
                 SJMP UP
DELAY: MOV R4,#35
       H1: MOV R3,#255
       H2: DJNZ R3,H2
                 DJNZ R4,H1
                RET
                END
```

Triangular Wave:

```
ORG 00H             ;
MOV P2, #00H         ;
MOV A, #00H          ;
MOV R0, #00H         ;
UPWARD:
  INC A            ;
  MOV P2, A        ;
  ACALL DELAY      ;
  CJNE A, #0FFH, UPWARD ;
DOWNWARD:
  DEC A            ;
  MOV P2, A        ;
  ACALL DELAY      ;
  CJNE A, #00H, DOWNWARD ;
SJMP UPWARD        ;
DELAY:
  MOV R1, #255     ;
DELAY_LOOP1:
  MOV R2, #255     ;
DELAY_LOOP2:
  DJNZ R2, DELAY_LOOP2 ;
  DJNZ R1, DELAY_LOOP1 ;
  RET              ;
END
```

Anticlockwise Motor:

```
ORG 00H          ; Start program at address 0x00

MAIN:
    MOV P2, #0F0H  ; Initialize Port 2 as output (upper nibble)
    ACALL COUNTERCLOCKWISE  ; Rotate stepper motor in counterclockwise direction
    ACALL DELAY    ; Call delay
    SJMP MAIN      ; Repeat forever

; Subroutine to rotate stepper motor counterclockwise
ANTICLOCKWISE:
    MOV A, #01H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #02H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #04H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #08H    ;
    MOV P2, A
    ACALL DELAY
    RET            ;
DELAY:
    MOV R1, #0FFH  ;
DELAY_LOOP1:
    MOV R2, #0FFH  ;
DELAY_LOOP2:
    DJNZ R2, DELAY_LOOP2 ;
    DJNZ R1, DELAY_LOOP1 ;
    RET            ;
END
```

Clockwise Rotation:

```
ORG 00H             ; Start program at address 0x00
MAIN:  MOV P2, #0F0H  ; Initialize Port 2 as output (upper nibble)
    ACALL CLOCKWISE ; Rotate stepper motor in clockwise direction
    ACALL DELAY     ; Call delay
    SJMP MAIN       ; Repeat forever

; Subroutine to rotate stepper motor clockwise
CLOCKWISE:
    MOV A, #08H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #04H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #02H    ;
    MOV P2, A
    ACALL DELAY
    MOV A, #01H    ;
    MOV P2, A
    ACALL DELAY
    RET          ;
DELAY:
    MOV R1, #0FFH  ;
DELAY_LOOP1:
    MOV R2, #0FFH  ;
DELAY_LOOP2:
    DJNZ R2, DELAY_LOOP2 ;
    DJNZ R1, DELAY_LOOP1 ;
    RET          ;
END
```

Digital Clock:

```
ORG 0000H          ;
MOV R7, #00H       ;
MOV R6, #00H       ;
MOV R5, #00H       ;
ACALL INIT_LCD     ;
MAIN_LOOP:
  ACALL UPDATE_LCD       ;
  ACALL DELAY_1_SEC      ;
  ACALL INCREMENT_TIME   ;
  SJMP MAIN_LOOP         ;
INIT_LCD:
  MOV A, #38H
  ACALL CMD_WRITE        ;
  ACALL DELAY_SHORT
  MOV A, #0CH
  ACALL CMD_WRITE        ;
  ACALL DELAY_SHORT
  MOV A, #06H
  ACALL CMD_WRITE        ;
  ACALL DELAY_SHORT
  MOV A, #01H
```

```
   ACALL CMD_WRITE       ;
   ACALL DELAY_SHORT
   RET
INCREMENT_TIME:
   INC R5              ;
   CJNE R5, #60, DONE_SEC ;
   MOV R5, #00H        ;
   INC R6              ;
   CJNE R6, #60, DONE_SEC ;
   MOV R6, #00H        ;
   INC R7              ;
   CJNE R7, #24, DONE_SEC ;
   MOV R7, #00H        ;
DONE_SEC:
   RET
UPDATE_LCD:
   MOV A, #80H
   ACALL CMD_WRITE      ;
   MOV A, R7           ;
   ACALL DISPLAY_TWO_DIGIT ;
   ACALL DISPLAY_COLON    ; Display ':'
   MOV A, R6           ;
   ACALL DISPLAY_TWO_DIGIT ;
   ACALL DISPLAY_COLON    ; Display ':'
   MOV A, R5           ;
   ACALL DISPLAY_TWO_DIGIT ;
   RET
DISPLAY_TWO_DIGIT:
   MOV B, #10          ;
   DIV AB              ;
   ADD A, #30H         ;
   ACALL DISPLAY_CHAR    ;
   MOV A, B            ;
   ADD A, #30H         ;
   ACALL DISPLAY_CHAR    ;
```

```
    RET
DISPLAY_COLON:
    MOV A, #3AH          ;
    ACALL DISPLAY_CHAR    ;
    RET
DISPLAY_CHAR:
    MOV P2, A            ;
    SETB P3.2            ;
    CLR P3.3             ;
    SETB P3.4            ;
    NOP                  ;
    CLR P3.4             ;
    ACALL DELAY_SHORT     ;
    RET
CMD_WRITE:
    MOV P2, A            ;
    CLR P3.2             ;
    CLR P3.3             ;
    SETB P3.4            ;
    NOP                  ;
    CLR P3.4             ;
    ACALL DELAY_SHORT     ;
    RET
DELAY_SHORT:
    MOV R0, #250         ;
DELAY_SHORT_LOOP:
    DJNZ R0, DELAY_SHORT_LOOP
    RET
DELAY_1_SEC:
    MOV R3, #100         ;
DELAY_LOOP:
    MOV R4, #255         ;
DELAY_LOOP_INNER:
    DJNZ R4, DELAY_LOOP_INNER
    DJNZ R3, DELAY_LOOP
    RET
END
```

Digital Thermometer:

```
ORG 0000H
    JMP START              ;
LCD_INIT  EQU 38H          ;
LCD_ON    EQU 0CH          ;
LCD_CLEAR EQU 01H          ;
TEMP1     EQU 30H          ;
TEMP2     EQU 31H          ;
START:
    ; Initialize LCD
    MOV DPTR, #LCD_CMD     ;
    MOV A, #LCD_INIT       ;
    ACALL LCD_WRITE_CMD
    MOV A, #LCD_ON         ;
    ACALL LCD_WRITE_CMD
    MOV A, #LCD_CLEAR      ;
    ACALL LCD_WRITE_CMD
    ; Main loop
MAIN_LOOP:
    ; Start ADC conversion
    SETB P3.3              ;
    CLR P3.4               ;
    NOP                    ;
    SETB P3.4              ;
WAIT_ADC:
    JB P3.5, WAIT_ADC      ;
    CLR P3.3               ;
    MOV A, P1              ;
    MOV TEMP1, A           ;
    MOV TEMP2, #0AH        ;
    MOV B, TEMP2
    DIV AB                 ;
```

```
  ACAL
  DISPLAY_TEMP
  SJMP MAIN_LOOP       ;
LCD_WRITE_CMD:
  MOV P2, A            ;
  CLR P3.0             ;
  CLR P3.1             ;
  SETB P3.2            ;
  NOP
  CLR P3.2             ;
  RET
LCD_WRITE_DATA:
  MOV P2, A            ;
  SETB P3.0            ;
  CLR P3.1             ;
  SETB P3.2            ;
  NOP
  CLR P3.2             ;
  RET
DISPLAY_TEMP:
  MOV A, TEMP1         ;
  ADD A, #30H          ;
  ACALL LCD_WRITE_DATA    ;
  MOV A, #0DFH         ;
  ACALL LCD_WRITE_DATA
  MOV A, #'C'          ;
  ACALL LCD_WRITE_DATA
  RET
END
```