# Assembly Programming through Arduino

G V V Sharma*

Contents



***Abstract*—This manual is a beginner's guide to assembly programming using the arduino. The instruction set of the ATMEGA328P IC, which is the arduino microcontroller, is used for building a decade counter by programming in assembly language. This manual can also be used for a first course on microprocessor architecture.**

## 1 Components

The components required for this manual are listed in Table 1.0.1.

## 2 Seven Segment Display

### *2.1 Hardware Setup*

The breadboard can be divided into 5 segments. In each of the green segements, the pins are internally connected so as to have the same voltage. Similarly, in the central segments, the pins in each column are internally connected in the same fashion as the blue columns.

**Problem 2.1.** Plug the display to the breadboard in Fig. 2.1

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

| Component | Value | Quantity |
|---|---|---|
| Breadboard | | 1 |
| Resistor | $\geq 220\Omega$ | 1 |
| Arduino | Uno | 1 |
| Seven Segment Display | Common Anode | 1 |
| Decoder | 7447 | 1 |
| Flip Flop | 7474 | 2 |
| Jumper Wires | | 20 |

TABLE 1.0.1

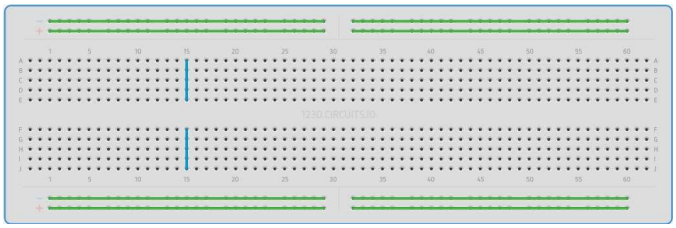

Fig. 2.1

The seven segment display in Fig. 2.2 has eight pins, $a, b, c, d, e, f, g$ and *dot* that take an active LOW input, i.e. the LED will glow only if the input is connected to ground. Each of these pins is connected to an LED segment. The *dot* pin is reserved for the · LED.

**Problem 2.2.** Connect one end of the resistor to the COM pin of the display and the other end to an extreme pin of the breadboard.

The Arduino Uno has some ground pins, analog input pins A0-A3 and digital pins D1-D13 that can be used for both input as well as output. It also has two power pins that can generate $3.3V$ and $5V$. In the following exercises, only the GND, $5V$ and digital pins will be used.

**Problem 2.3.** Connect the 5V pin of the arduino to an extreme pin that is in the same segment as the 1K resistor pin.

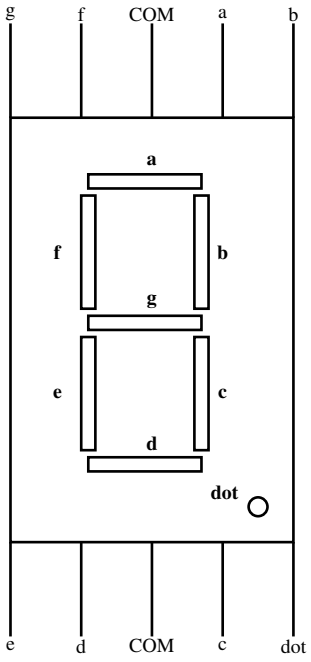


Fig. 2.2

**Problem 2.4.** Connect the GND pin of the arduino to the opposite extreme pin of the breadboard

**Problem 2.5.** Connect the Arduino to the computer.

**Problem 2.6.** Connect the *dot pin of the display to a pin in the same segment as the GND pin. What do you observe?*

*2.2 Software Setup*

**Problem 2.7.** Install AVRA and AVRDUDE on your Linux system through the following commands

```
sudo apt-get install avra avrdude
%Finding the port

sudo dmesg | grep tty
%The output will be something like
[    6.153362] cdc_acm 1-1.2:1.0:
   ttyACM0: USB ACM device
%and your port number is ttyACM0
```

**Problem 2.8.** Download the m328Pdef.inc file from

```
http://tlc.iith.ac.in/img/m328Pdef
   .inc
```

and copy into the directory home (/home/user/) directory

**Problem 2.9.** Open **geany** and type the following code. Save the file as **hello.asm**

```
;hello
;using assembly language for
   turning LED on

.include "/home/alarm/m328Pdef.inc
   "


  ldi r16,0b00100000
  out DDRB,r16
  ldi r17,0b00000000
  out PortB,r17
Start:
  rjmp Start
```

**Problem 2.10.** With **hello.asm** open in **geany**, go to Build→Set Build Commands→Compile and enter

```
avra "%f"
```

Then go to Build→Set Build Commands→Execute and enter

```
avrdude -p atmega328p -c arduino -
   P /dev/ttyACM0 -b 115200 -U
   flash:w:%e.hex
```

**Problem 2.11.** Change **/home/linaro/m328Pdef.inc** in **hello.asm** to **/home/username/m328Pdef.inc**, where *username* is your linux login name.

*2.3 Controlling the Display*

**Problem 2.12.** Now connect the dot pin of the display to pin 13 of the arduino. Use **F8** to compile **hello.asm** and **F5** to execute.

**Problem 2.13.** Turn the LED off by modifying **hello.asm.**

**Problem 2.14.** Complete Table 2.14.1 for all the digital pins using Fig. 2.14

| Port Pin | Digital Pin |
|---|---|
| PD2 | 2 |
| PB5 | 13 |

TABLE 2.14.1

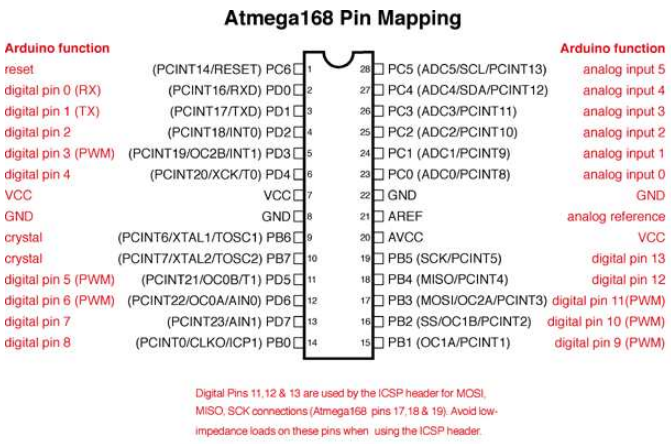


Fig. 2.14

**Problem 2.15.** Connect the $a-g$ pins of the display to the 2-8 pins of the arduino. Run the following code.

```
;using assembly language for
;displaying number on
;seven segment display

.include "/home/linaro/m328Pdef.
    inc"


;Configuring pins 2-7 of Arduino
;as output
  ldi r16,0b11111100
  out DDRD,r16
;Configuring pin 8 of Arduino
;as output
  ldi r16,0b00000001
  out DDRB,r16
;Writing the number 2 on the
;seven segment display
  ldi r17,0b10010000
  out PortD,r17

  ldi r17,0b00000000
  out PortB,r17
Start:
  rjmp Start
```

**Problem 2.16.** The output of Problem 2.15 can be explained by Table 2.16.1 below. Complete Table 2.16.1 for all numbers between 0-9. Use this information to display the numbers from 0-9 .

| PD2 | PD3 | PD4 | PD5 | PD6 | PD7 | PB0 | decimal |
|---|---|---|---|---|---|---|---|
| a | b | c | d | e | f | g | |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 2 |

TABLE 2.16.1

## 3 Display Decoder

The 7447 IC helps in displaying decimal numbers on the seven segment display. The $\bar{a}-\bar{f}$, pins of the 7447 IC are connected to the $a-f$ pins of the display. $V_{cc}$ should be connected to a 5V power source. The input pins of the decoder are A,B,C and D, with A being the lowest significant bit (LSB) and D being the most significant bit (MSB). For example, the number 5 is visible on the display when the A,B,C and D inputs are the following.

| D | C | B | A | Decimal |
|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 5 |

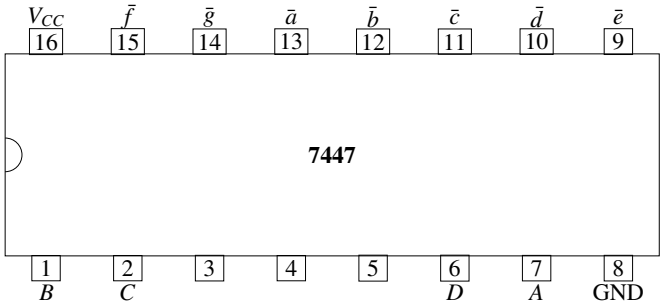


Fig. 3.0

**Problem 3.1.** Connect the 7447 IC decoder $\bar{a}-\bar{g}$ pins to the $a-g$ pins of the display respectively.

**Problem 3.2.** Connect the $V_{cc}$ and GND pins of the decoder to the 5V supply and GND pins of the breadboard.

### *3.1 Driving the Segments*

**Problem 3.3.** Connect the A-D pins of the 7447 IC to the pins D2-D5 of the Arduino.

**Problem 3.4.** Type the following code and execute. What do you observe?

```
;driving the 7447 decoder
```

```
.include "/home/alarm/m328Pdef.inc"

        ldi r16, 0b00111100 ; identifying output pins 2,3,4,5
        out DDRD,r16        ; declaring pins as output
        ldi r16,0b00010100  ; loading the number 5 in binary
        out PORTD,r16       ; writing output to pins 2,3,4,5

Start:
        rjmp Start
```

**Problem 3.5.** Now generate the numbers 0-9 by modifying the above program.

## 4 Boolean Operations

Test all the following using the 7447 IC

**Problem 4.1.** Verify the AND,OR and XOR operations in assembly.

**Solution:**

```
;logical AND, OR and XOR operations
;output displayed using 7447 IC

.include "/home/alarm/m328Pdef.inc"

ldi r16, 0b00111100 ;identifying output pins 2,3,4,5
out DDRD,r16        ;declaring pins as output



ldi r16,0b00000000  ; initializing W
ldi r17,0b00000001  ; initializing X
;logical AND
;and r16,r17        ;W AND X
;logical OR
;or r16,r17         ;W OR X
;logical XOR
eor r16,r17         ;X XOR X

;following code is for displaying output
;shifting LSB in r16 to 2nd position
ldi r20, 0b00000010 ;counter = 2

rcall loopw         ;calling the loopw routine

out PORTD,r16       ;writing output to pins 2,3,4,5


Start:
rjmp Start

;loop for bit shifting
loopw:  lsl r16             ; left shift
        dec r20             ; counter --
        brne    loopw       ;if counter != 0
        ret
```

**Problem 4.2.** Write a routine for finding the complement of a number.

**Solution:**

```
;program to complement a boolean


.include "/home/alarm/m328Pdef.inc"

ldi r16, 0b00111100 ;identifying output pins 2,3,4,5
out DDRD,r16        ;declaring pins as output

ldi r16,0b00000000  ;A=1
```

```
rcall comp        ;jumpting to comp
   routine below

;following code is for displaying
   output
;shifting LSB in r16 to 2nd
   position
ldi r20, 0b00000010      ;counter =
    2

rcall loopw              ;calling
   the loopw routine

out PORTD,r16            ;writing
   output to pins 2,3,4,5


Start:
rjmp Start

;loop for bit shifting
loopw:  lsl r16                  ;
   left shift
        dec r20                  ;
           counter --
        brne    loopw    ;if
           counter != 0
        ret

;comp routine begins
comp:
        mov r0,r16
                  ;using r0 for
           computations
        ldi r16,0b00000001       ;
           loading 1

        eor r16,r0
                  ;A'=A XOR 1
        ret
   ;returning from comp
```

## 5 Combinational Logic

**Problem 5.1.** In the truth table in Table 5.2.1, $W, X, Y, Z$ are the inputs and $A, B, C, D$ are the outputs. This table represents the system that increments the numbers 0-8 by 1 and resets the number 9 to 0 Using K-maps, write the boolean logic functions for $A, B, C$ in terms of $W, X, Y, Z$.

| Z | Y | X | W | **D** | **C** | **B** | **A** |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | **0** | **0** | **0** | **1** |
| 0 | 0 | 0 | 1 | **0** | **0** | **1** | **0** |
| 0 | 0 | 1 | 0 | **0** | **0** | **1** | **1** |
| 0 | 0 | 1 | 1 | **0** | **1** | **0** | **0** |
| 0 | 1 | 0 | 0 | **0** | **1** | **0** | **1** |
| 0 | 1 | 0 | 1 | **0** | **1** | **1** | **0** |
| 0 | 1 | 1 | 0 | **0** | **1** | **1** | **1** |
| 0 | 1 | 1 | 1 | **1** | **0** | **0** | **0** |
| 1 | 0 | 0 | 0 | **1** | **0** | **0** | **1** |
| 1 | 0 | 0 | 1 | **0** | **0** | **0** | **0** |

TABLE 5.2.1

**Solution:** The desired equations are

$$A = W' \tag{5.1.1}$$
$$B = WX'Z' + W'X \tag{5.1.2}$$
$$C = WXY' + X'Y + W'Y \tag{5.1.3}$$
$$D = WXY + W'Z \tag{5.1.4}$$

**Problem 5.2.** Write an assembly program for implementing Table 5.2.1 and verify if your logic is correct by observing the output on the seven segment display.

## 6 Decade Counter through Arduino

**Problem 6.1.** Use the following code for LED blinking. You will have to connect pin 13 to the LED on the seven segment display. The blinking delay that you obtain is 1 second.

```
.include "/home/alarm/m328Pdef.inc
   "

;works as 1 second delay on
   arduino: 16 MHz (16x10^6)

sbi DDRB, 5 ;set pin 13 as output
   pin (DDRB pin 5)
ldi r16, 0b00000101 ;the last 3
   bits define the prescaler, 101
   => division by 1024
out TCCR0B, r16
;prescalar used = 1024. So new
   freq. of clock cycle = (16 MHz /
    1024) = 16 kHz
```

```
clr r18 ;output bits. we are only
   interested in bit 6 from the
   right.

ldi r20,0b00100000      ;
   initializing

loop:
        eor r18, r20            ;
           change the output of LED
        out PORTB, r18
        ldi r19, 0b01000000 ;times
            to run the loop = 64
           for 1 second delay
        rcall PAUSE             ;
           call the PAUSE label
        rjmp loop

PAUSE:  ;this is delay (function)
lp2:    ;loop runs 64 times
                IN r16, TIFR0 ;
                   tifr is timer
                   interupt flag (8
                    bit timer runs
                   256 times)
                ldi r17, 0
                   b00000010
                AND r16, r17 ;need
                    second bit
                BREQ PAUSE
                OUT TIFR0, r17  ;
                   set tifr flag
                   high
        dec r19
        brne lp2
        ret

;prescalar * loop_iterations *
   timer_duration =~ 16 million
   cycles
;16 million cycles at 16 MHz = 1
   second
```

The 7474 IC in Fig. 6.2 has two D flip flops. The D pins denote the input and the Q pins denote the output. CLK denotes the clock input.

**Problem 6.2.** Connect the D2-D5 pins of the arduino to the Q pins of the two 7474 ICs. Use the D2-D5 pins as Arduino input.

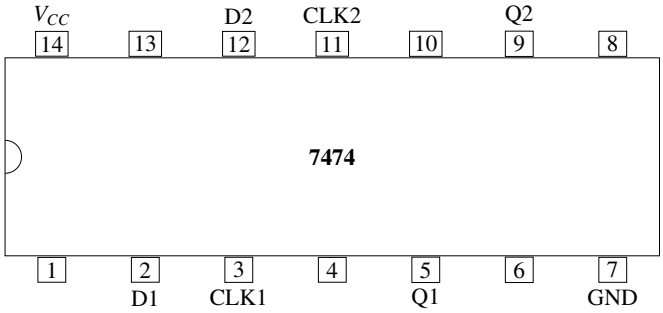


Fig. 6.2

**Problem 6.3.** Connect the Q pins to IC 7447 Decoder as input pins. Connect the 7447 IC to the seven segment display.

**Problem 6.4.** Connect the D6-D9 pins of the arduino to the D input pins of two 7474 ICs. Use the D6-D9 pins as Arduino output.

**Problem 6.5.** Connect pin 13 of the Arduino to the CLK inputs of both the 7474 ICs.

**Problem 6.6.** Using the logic for the counting decoder in assembly in Section 5, implement the decade counter using flip-flops.