Embedded C through AVR-GCC

G V V Sharma*

Contents

***Abstract*—This manual shows how to control hardware using AVR-GCC. AVR-GCC is a C compiler for the Atmega328p.**

1 Components

Component	Value	Quantity
Breadboard		1
Resistor	$\geq 220\Omega$	1
Arduino	Uno	1
Seven Segment Display	Common Anode	1
LCD Display	16×2	1
Jumper Wires		20

TABLE 0

2 Blink

1. Install **subversion**

```
sudo apt update
sudo apt install subversion
```

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

2. Go to your working directory and download the folder titled **codes** using the following command.

```
svn checkout https://github.com/gadepall/
    arduino/trunk/avr-gcc/setup/codes
```

3. Connect your arduino to the computer and open a terminal.
4. Open a terminal and go to the **codes** directory. Type **make**. The built in led on the arduino should be blinking.
5. If you open **main.c** in **geany**, you can execute the code by **Shift+F9**.
6. Now open **main.c**. Explain the following lines.

```
    PORTB = ((0 << PB5));
        _delay_ms(500);
//turn led on
    PORTB = ((1 << PB5));
    _delay_ms(500);
```

Solution: ((0 << *PB*5)) writes 0 to pin 13 (PB5). _delay_ms(500) introduces a delay of 500 ms.

7. Modify the above code to keep the led on.
8. Repeat the above exercise to keep the led off.

3 Display Control

1. Complete Table 1 for all the digital pins using Fig. 1.

Port Pin	Digital Pin
PD2	2
PB5	13

TABLE 1

2. Plug the seven segment display in Fig. 2 into the breadboard.
3. Connect 5V from the Arduino to the the COM pin of the display through a resistor.
4. Make connections according to Table 4.
5. Execute the following code

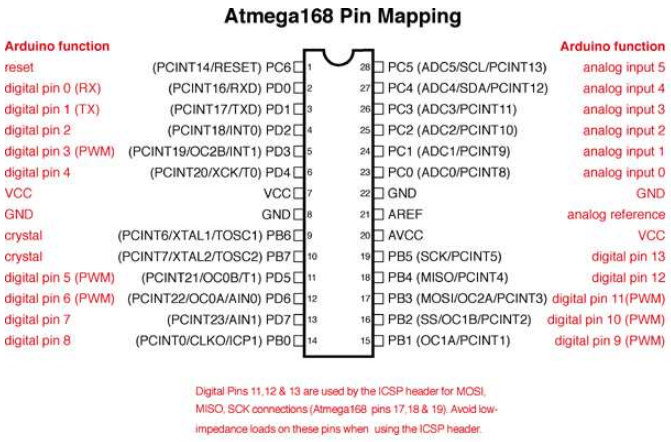

Fig. 1

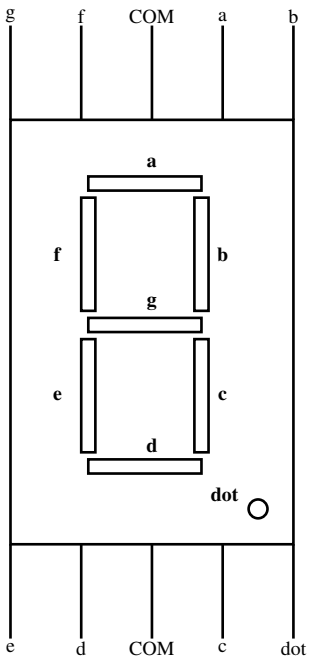

Fig. 2

Arduino	2	3	4	5	6	7	8
	PD2	PD3	PD4	PD5	PD6	PD7	PB0
Display	a	b	c	d	e	f	g
2	0	0	1	0	0	1	0

TABLE 4

```
wget https://raw.githubusercontent.com/gadepall
    /arduino/master/avr-gcc/sevenseg/codes/
    main.c
```

6. Modify the above code to generate numbers between 0-9.
7. Repeat the above by writing a function.
8. Use a for loop and include a delay to implement a decade counter

4 Control LCD

1. Plug the LCD in Fig. 2 to the breadboard.
2. Connect the Arduino pins to LCD pins as per Table 2.

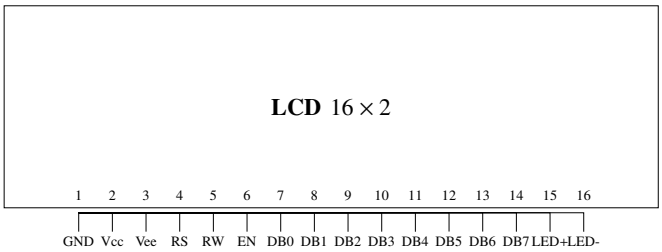

Fig. 2: LCD

TABLE 2: Arduino to LCD Pin Connection.

Arduino Pins	**LCD Pins**	**LCD Pin Label**	**LCD Pin Description**
GND	1	GND	
5V	2	Vcc	
GND	3	Vee	Contrast
D8	4	RS	Register Select
GND	5	R/W	Read/Write
D9	6	EN	Enable
D10	11	DB4	Serial Connection
D11	12	DB5	Serial Connection
D12	13	DB6	Serial Connection
D13	14	DB7	Serial Connection
5V	15	LED+	Backlight
GND	16	LED-	Backlight

3. Download the following directory from

```
svn checkout https://github.com/gadepall/
    arduino/trunk/avr-gcc/lcd/codes
```

and execute **main.c**

4. Modify the above code to display a string.

5. Modify the above code to obtain a decade counter so that the numbers from 0 to 9 are displayed on the lcd repeatedly.
6. Repeat the above exercies to display a string on the first line and a number on the second line of the lcd.
7. Write a program to implement all functions of a simple calculator and display on an LCD.

5 Standalone ATMEGA328P

1. Take the ATMEGA328P IC from the Arduino board and plug it into the breadboard.
2. Plug the seven segment display in Fig. 2 into the breadboard.
3. Plug the Quartz crystal between pins 9 and 10 of the ATMEGA328P.
4. Connect the 22pF capacitors from pin 9 and 10 to GND.
5. Connect the pin 1 through a 10 KΩ resistor to 5V.
6. Connect one end of a push button to pin 1 and the other end to GND.
7. Make connections according to Table 7.

Type	Pin No
5V	7
	20
	21
GND	8
	22
a	4
b	5
c	6
d	11
e	12
f	13
g	14

TABLE 7

8. Power up through USB.

6 Project

Design a simple calculator using the ATMEGA328P. Use push buttons for numbers and other arithmetic keys. Solder all components on a PCB and power the circuit through a micro/typeC USB port. Design and print a case for your calculator using a 3-D printer.