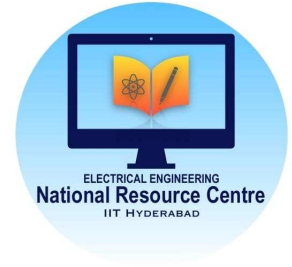




Embedded C through AVR-GCC



G V V Sharma*

CONTENTS

1	Components	1
2	Blink	1
3	Display Control	1

Abstract—This manual shows how to control an led 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		1
Jumper Wires		20

TABLE 0

2 BLINK

1. Install **subversion**

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

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

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

```
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:** $((0 \ll PB5))$ 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. Make connections according to Table 2.
3. Execute the following code

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

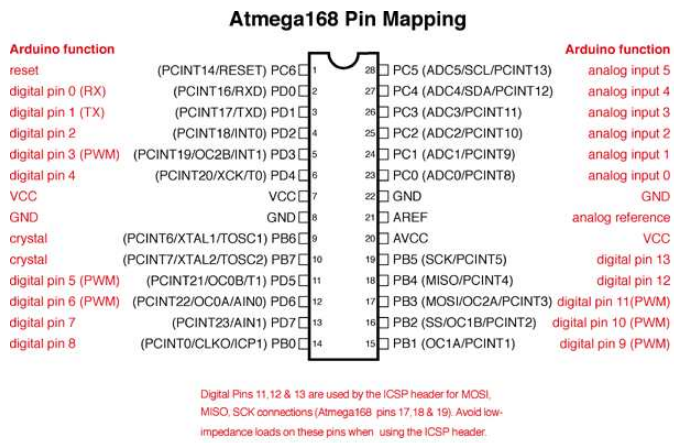


Fig. 1

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 2

- Modify the above code to generate numbers between 0-9.