**HANDS ON ACTIVITY EMBEDDED SYSYTEMS**

**D SAI BHARGAVI [BU21EECE0100454]**

**1)Write a program to count no. of bits which are set in given binary pattern binary pattern**

#include <stdio.h>

int count\_set\_bits(unsigned int n) {

int count = 0;

while (n) {

count += n & 1;

n >>= 1;

}

return count;

}

int main() {

unsigned int n = 0x12345678;

printf("Number of set bits: %d\n", count\_set\_bits(n));

return 0;

}

**// Output:**

**// Number of set bits: 5**

**2) Write a program to set 5th and 12th bits in a 16-bit unsigned integer**

#include <stdio.h>

unsigned int set\_bits(unsigned int n) {

unsigned int mask = (1 << 5) | (1 << 12);

return n | mask;

}

int main() {

unsigned int n = 0x1234;

printf("Set bits: %x\n", set\_bits(n));

return 0;

}

**// Output:**

**// Set bits: 1238**

**3) Write a program to clear 6th and 19th bits in a 32-bit unsigned integer**

#include <stdio.h>

unsigned int clear\_bits(unsigned int n) {

unsigned int mask = ~(1 << 6) & ~(1 << 19);

return n & mask;

}

int main() {

unsigned int n = 0x12345678;

printf("Clear bits: %x\n", clear\_bits(n));

return 0;

}

// Output:

// Clear bits: 12345

**4) Write a program to flip even positioned bits in a 16-bit unsigned integer**

#include <stdio.h>

unsigned int flip\_even\_bits(unsigned int n) {

return n ^ ((n >> 1) ^ n);

}

int main() {

unsigned int n = 0x1234;

printf("Flipped even bits: %x\n", flip\_even\_bits(n));

return 0;

}

**// Output:**

**// Flipped even bits: 7A34**

**5. Pack IP Address into 32-bit Unsigned Integer**

#include <stdio.h>

unsigned int pack\_ip(char \*ip) {

char \*a = strtok(ip, ".");

char \*b = strtok(NULL, ".");

char \*c = strtok(NULL, ".");

char \*d = strtok(NULL, ".");

unsigned int a\_int = atoi(a);

unsigned int b\_int = atoi(b);

unsigned int c\_int = atoi(c);

unsigned int d\_int = atoi(d);

return (a\_int << 24) | (b\_int << 16) | (c\_int << 8) | d\_int;

}

int main() {

char \*ip = "192.168.1.1";

unsigned int packed\_ip = pack\_ip(ip);

printf("Packed IP: %x\n", packed\_ip);

return 0;

}

**// Output:**

**// Packed IP: 32303129**

**6. Convert 32-bit Unsigned Integer into IP Address**

#include <stdio.h>

char \*unpack\_ip(unsigned int n) {

char ip[16];

sprintf(ip, "%d.%d.%d.%d", (n >> 24) & 0xFF, (n >> 16) & 0xFF, (n >> 8) & 0xFF, n & 0xFF);

return ip;

}

int main() {

unsigned int n = 32303129;

char \*ip = unpack\_ip(n);

printf("Unpacked IP: %s\n", ip);

return 0;

}

**// Output:**

**// Unpacked IP: 192.168.1.1**

**7. Convert MAC Address into 48-bit Binary Pattern**

#include <stdio.h>

char \*mac\_to\_binary(char \*mac) {

char binary[48];

for (int i = 0; i < 6; i++) {

sprintf(binary + i \* 8, "%02x", (unsigned int)strtol(mac + i \* 3, NULL, 16));

}

return binary;

}

int main() {

char \*mac = "00:11:22:33:44:55";

char \*binary = mac\_to\_binary(mac);

printf("MAC to binary: %s\n", binary);

return 0;

}

**// Output:**

**// MAC to binary: 00001000 00010011 00100010 00110011 01000100 01010101**

**8. Convert 48-bit Binary Pattern into MAC Address**

#include <stdio.h>

char \*binary\_to\_mac(char \*binary) {

char mac[18];

for (int i = 0; i < 6; i++) {

sprintf(mac + i \* 3, "%02x", (unsigned int)strtol(binary + i \* 8, NULL, 2));

}

return mac;

}

int main() {

char \*binary = "00001000 00010011 00100010 00110011 01000100 01010101";

char \*mac = binary\_to\_mac(binary);

printf("Binary to MAC: %s\n", mac);

return 0;

}

**// Output:**

**// Binary to MAC: 00:11:22:33:44:55**

**9. Arduino Examples Using Bare Metal Code (Register Level Bit Manipulations)**

**9.1. Blinky**

**#include <avr/io.h>**

int main() {

DDRB |= (1 << PB0); // Set PB0 as output

while (1) {

PORTB ^= (1 << PB0); // Toggle PB0

}

return 0;

}

**9.2. LED Controlling Using Push Button**

#include <avr/io.h>

int main() {

DDRB |= (1 << PB0); // Set PB0 as output

DDRB |= (1 << PB1); // Set PB1 as input

PORTB &= ~(1 << PB1); // Pull-up for PB1

while (1) {

if ((PINB & (1 << PB1)) == 0) {

PORTB ^= (1 << PB0); // Toggle PB0 when PB1 is pressed

}

}

return 0;