EXPERIMENT NO:

ROLL No:

NAME:

TITLE: Write a Program to Create Multiple Tasks and understand the Multitasking capabilities of RTOS(FreeRTOS).

Theory:

Real time operating system popularly known as RTOS provides controller with the ability to respond to input and complete tasks within a specific period of time based on priority. On the first look, an RTOS might sound like just any other embedded program or firmware, but it is built on the architecture of an Operating system. Hence, like any operating system RTOS can allow multiple programs to execute at the same time supporting **multiplexing**. As we know the core of a processor or controller can only execute a single instruction at a time, but the RTOS has something called the **scheduler** which decides which instruction to execute first and thus executes the instructions of multiple programs one after the other. Technically an RTOS only creates an illusion of multi-taking by executing paralleled instructions one at a time.

Arduino code of simulteniously running three task

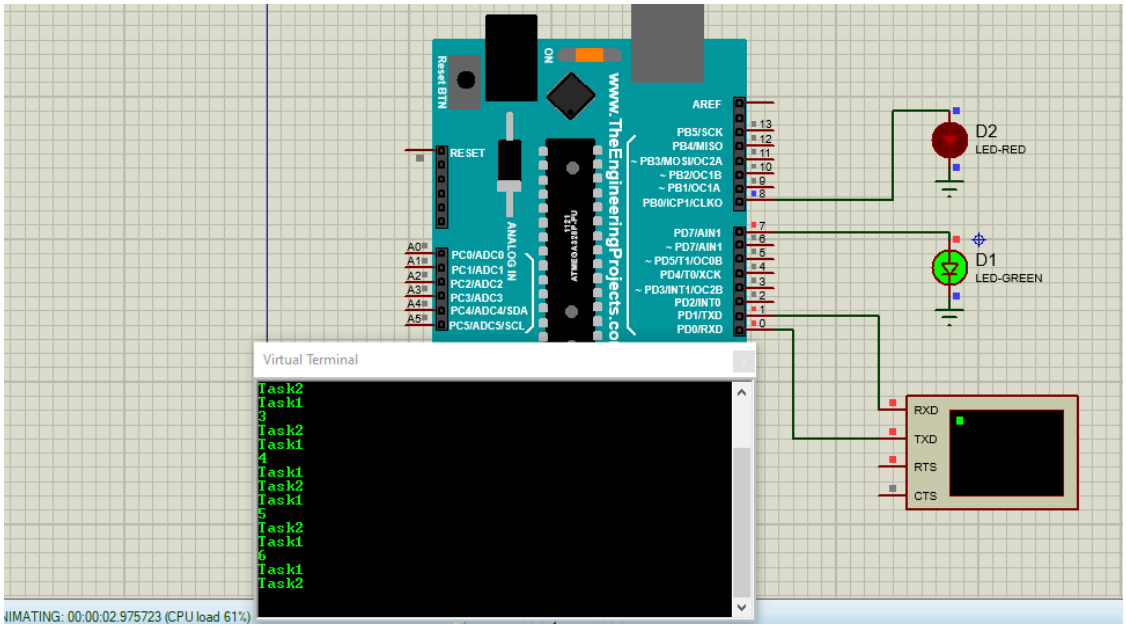
1. Blink led with 200ms delay
2. Blink led with 300ms delay
3. Print counter variable on serial monitor

Code:

```
#include <Arduino_FreeRTOS.h>

void TaskBlink1( void *pvParameters );
void TaskBlink2( void *pvParameters );
void Taskprint( void *pvParameters );

void setup() {
 // initialize serial communication at 9600 bits per second:
Serial.begin(9600);
xTaskCreate(TaskBlink1,"task1",128,NULL,1,NULL );
xTaskCreate(TaskBlink2,"task2", 128,NULL,1,NULL );
xTaskCreate(Taskprint,"task3", 128,NULL,1,NULL );
vTaskStartScheduler();
}
void loop()
{
}
void TaskBlink1(void *pvParameters)  {
```

```
pinMode(8, OUTPUT);
while(1)
 {
    Serial.println("Task1");
    digitalWrite(8, HIGH);
    vTaskDelay( 200 / portTICK_PERIOD_MS );
    digitalWrite(8, LOW);
    vTaskDelay( 200 / portTICK_PERIOD_MS );
  }

}
void TaskBlink2(void *pvParameters)
{
  pinMode(7, OUTPUT);
  while(1)
  {
    Serial.println("Task2");
    digitalWrite(7, HIGH);
    vTaskDelay( 300 / portTICK_PERIOD_MS );
    digitalWrite(7, LOW);
    vTaskDelay( 300 / portTICK_PERIOD_MS );
  }

}
void Taskprint(void *pvParameters)  {
 int counter = 0;
  while(1)
  {
    counter++;
  Serial.println(counter);
 vTaskDelay(500 / portTICK_PERIOD_MS);    }
}
```

Result:


Reset BTN
ON
RESET
ANALOG IN
A0
A1
A2
A3
A4
A5
PC0/ADC0
PC1/ADC1
PC2/ADC2
PC3/ADC3
PC4/ADC4/SDA
PC5/ADC5/SCL
AREF
PB5/SCK
PB4/MISO
~ PB3/MOSI/OC2A
~ PB2/OC1B
~ PB1/OC1A
PB0/ICP1/CLKO
PD7/AIN1
~ PD7/AIN1
~ PD5/T1/OC0B
PD4/T0/XCK
~ PD3/INT1/OC2B
PD2/INT0
PD1/TXD
PD0/RXD
www.TheEngineeringProjects.com
D2
LED-RED
D1
LED-GREEN
RXD
TXD
RTS
CTS
Virtual Terminal
Task2
Task1
3
Task2
Task1
4
Task1
Task2
Task1
5
Task2
Task1
6
Task1
Task2
IMATING: 00:00:02.975723 (CPU load 61%)