EXPERIMENT NO:

ROLL No:

NAME:

TITLE: Write a Program to illustrate the Queue Management Features of FreeRTOS

Theory:

A queue in a real-time operating system (RTOS) is **a kernel object that is capable of passing information between tasks without incurring overwrites from other tasks or entering into a race condition**. A queue is a first in, first out (FIFO) system where items are removed from the queue once read.

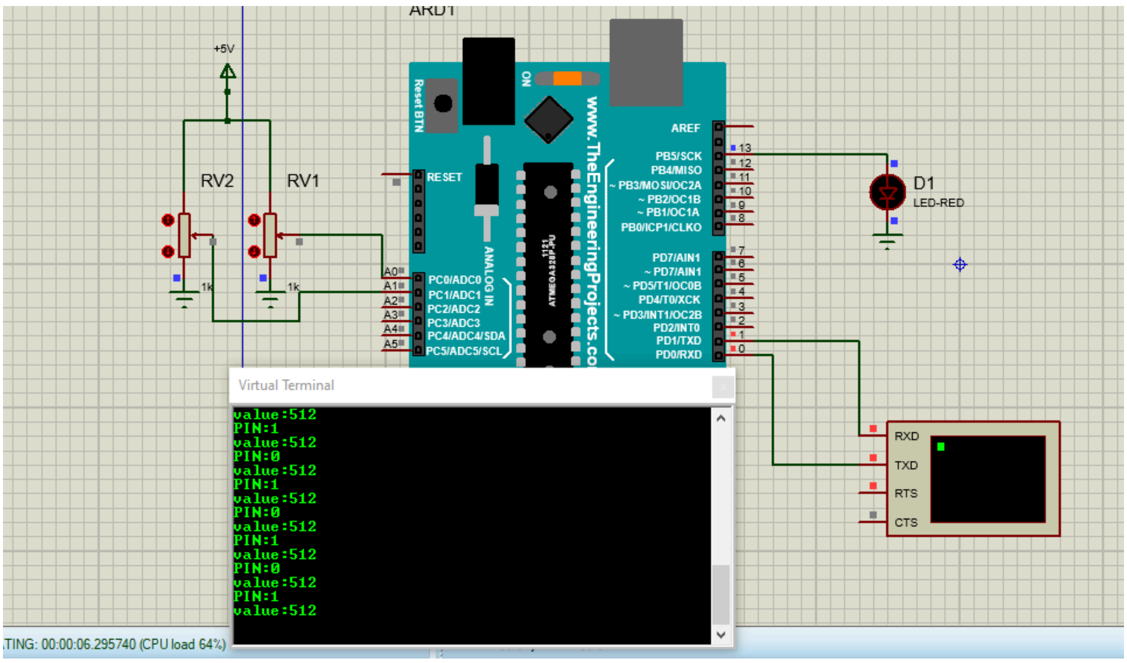
Arduino code for creation of queue and sending and receiving data from queue in freeRTOS

```
/*
 * Example of a basic FreeRTOS queue
 * https://www.freertos.org/Embedded-RTOS-Queues.html
 */

// Include Arduino FreeRTOS library
#include <Arduino_FreeRTOS.h>

// Include queue support
#include <queue.h>

// Define a Array
int pinReadArray[4]={0,0,0,0};

//Function Declaration
void TaskBlink(void *pvParameters);
void TaskAnalogReadPin0(void *pvParameters);
void TaskAnalogReadPin1(void *pvParameters);
void TaskSerial(void *pvParameters);

/*
 * Declaring a global variable of type QueueHandle_t
 *
 */
QueueHandle_t arrayQueue;

void setup() {

 /**
```

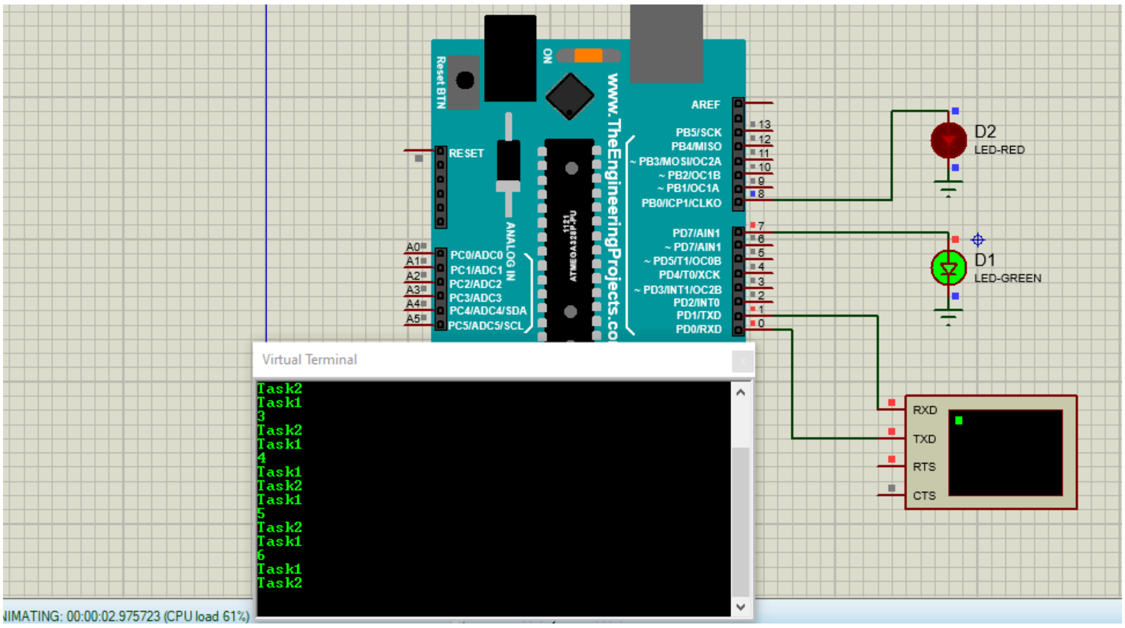
```
   * Create a queue.
   * https://www.freertos.org/a00116.html
   */
arrayQueue=xQueueCreate(10, //Queue length
                        sizeof(int)); //Queue item size
if(arrayQueue!=NULL){

  // Create task that consumes the queue if it was created.
  xTaskCreate(TaskSerial,// Task function
              "PrintSerial",// Task name
              128,// Stack size
              NULL,
              2,// Priority
              NULL);

  // Create task that publish data in the queue if it was created.
  xTaskCreate(TaskAnalogReadPin0, // Task function
              "AnalogRead1",// Task name
              128,// Stack size
              NULL,
              1,// Priority
              NULL);

   // Create other task that publish data in the queue if it was created.
   xTaskCreate(TaskAnalogReadPin1,// Task function
              "AnalogRead2",// Task name
              128,// Stack size
              NULL,
              1,// Priority
              NULL);


   xTaskCreate(TaskBlink,// Task function
              "Blink", // Task name
              128,// Stack size
              NULL,
              0,// Priority
              NULL);

}
}

void loop() {}

/**
```

```
 * Analog read task for Pin A0
 * Reads an analog input on pin 0 and send the readed value through the
queue.
 * See Blink_AnalogRead example.
 */
void TaskAnalogReadPin0(void *pvParameters){
  (void) pvParameters;
  for (;;){
  pinReadArray[0]=0;
  pinReadArray[1]=analogRead(A0);
  /**
     * Post an item on a queue.
     * https://www.freertos.org/a00117.html
     */
  xQueueSend(arrayQueue,&pinReadArray,portMAX_DELAY);
  // One tick delay (15ms) in between reads for stability
  vTaskDelay(1);
  }
}

/**
 * Analog read task for Pin A1
 * Reads an analog input on pin 1 and send the readed value through the
queue.
 * See Blink_AnalogRead example.
 */
void TaskAnalogReadPin1(void *pvParameters){
  (void) pvParameters;
  for (;;){
  pinReadArray[2]=1;
  pinReadArray[3]=analogRead(A1);
  /**
     * Post an item on a queue.
     * https://www.freertos.org/a00117.html
     */
  xQueueSend(arrayQueue,&pinReadArray,portMAX_DELAY);
  // One tick delay (15ms) in between reads for stability
  vTaskDelay(1);
  }
}



/**
 * Serial task.
 * Prints the received items from the queue to the serial monitor.
 */
void TaskSerial(void *pvParameters){
  (void) pvParameters;

  // Init Arduino serial
  Serial.begin(9600);

  // Wait for serial port to connect. Needed for native USB, on LEONARDO,
MICRO, YUN, and other 32u4 based boards.
  while (!Serial) {
    vTaskDelay(1);
  }

  for (;;){
    if(xQueueReceive(arrayQueue,&pinReadArray,portMAX_DELAY) == pdPASS ){
      Serial.print("PIN:");
      Serial.println(pinReadArray[0]);
      Serial.print("value:");
      Serial.println(pinReadArray[1]);
      Serial.print("PIN:");
      Serial.println(pinReadArray[2]);
      Serial.print("value:");
      Serial.println(pinReadArray[3]);
      vTaskDelay(500/portTICK_PERIOD_MS);
    }
  }
}

/*
 * Blink task.
 * See Blink_AnalogRead example.
 */
void TaskBlink(void *pvParameters){
  (void) pvParameters;
  pinMode(LED_BUILTIN,OUTPUT);
  digitalWrite(LED_BUILTIN,LOW);
  for (;;){
    digitalWrite(LED_BUILTIN,HIGH);
    vTaskDelay(250/portTICK_PERIOD_MS);
    digitalWrite(LED_BUILTIN,LOW);
    vTaskDelay(250/portTICK_PERIOD_MS);
  }
}
```

Result:



```
value:512
PIN:1
value:512
PIN:0
value:512
PIN:1
value:512
PIN:0
value:512
PIN:1
value:512
PIN:0
value:512
PIN:1
value:512
```



```
Task2
Task1
3
Task2
Task1
4
Task1
Task2
Task1
5
Task2
Task1
6
Task1
Task2
```