KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Department of Electronics and Communication Engineering

Course Activity Report on

Stack Implementation using eeprom

In

Data Structures(16ECL68)

Submitted by

Sanskruti Samant (2GI17EC113)

Sanah Hassan (2GI17EC109)

Prateek Mahajan (2GI17EC080)

Omkar Sulakhe(2GI17EC065)

Guide

Prof. Maya Choughale

Prof. Pratijnya Ajawan

KARNATAK LAW SOCIETY'S GOGTE INSTITUTE OF
TECHNOLOGY UDYAMBAG, BELAGAVI-590008 (An Autonomous
Institution under Visvesvaraya Technological University, Belagavi)
(APPROVED BY AICTE, NEW DELHI)

Department of Electronics And Communication Engineering

CERTIFICATE

Certified that the project entitled Stack Implementation Using EEPROM done at Electronics and Communication dept. in Data Structures is a bonafide work carried out by

Ms. ....Sanskruti Samant.........……USN....2GI17EC113.......
Ms …………Sanah Hassan………...... ….USN.....2GI17EC109…...
Mr ....Prateek Mahajan........... ……USN......2GI17EC080.......
Mr. ....Omkar Sulakhe............... …...USN.....2GI17EC065…….

In partial fulfillment for the award of **Bachelor of Engineering** in .....KLS GOGTE INSTITUTE OF TECHNOLOGY............... of the Visvesvaraya Technological University, Belagavi during the year 2020..- 2021... It is certified that all corrections/suggestions indicated have been incorporated in the report. The project report has been approved as it satisfies the academic requirements prescribed for the said Degree.

SL.NO	NAME	USN	MARKS	SIGNATURE
1	SANSKRUTI SAMANT	2GI17EC113		
2	SANAH HASSAN	2GI17EC109		
3	PRATEEK MAHAJAN	2GI17EC080		
4	OMKAR SULAKHE	2GI17EC065		

Signature of the staff in charge:

Date:

Table Of Contents:-

INTRODUCTION:-

A data structure is a data organization, management, and storage format that enables efficient access and modification. Data structures serve as the basis for abstract data types (ADT). The ADT defines the logical form of the data type. The data structure implements the physical form of the data type. Different types of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks. Data structures provide a means to manage large amounts of data efficiently for uses such as large databases and internet indexing services. Usually, efficient data structures are key to designing efficient algorithms. Some formal design methods and programming languages emphasize data structures, rather than algorithms, as the key organizing factor in software design. Data structures can be used to organize the storage and retrieval of information stored in both main memory and secondary memory.

DATA STRUCTURE USED:-

Stack is an abstract data type with a bounded (predefined) capacity. It is a simple data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the top of the stack and the only element that can be removed is the element that is at the top of the stack, just like a pile of objects. Stacks hold objects, usually all of the same type. Most stacks support just the simple set of operations we introduced thus, the main property of a stack is that objects go on and come off of the top of the stack. One of the distinguishing characteristics of a stack, and the thing that makes it useful, is the order in which elements come out of a stack.

A stack is an abstract data structure (ADT) that allows the following operations:

1) Push: Add an element to the top of a stack.
2) Pop: Remove an element from the top of a stack.
3) isEmpty: Check if the stack is empty.
4) isFull: Check if the stack is full.
5) Peek: Get the value of the top element without removing it.
6) Display: The display() function displays all the elements in the stack. It uses a for loop to do so. If there are no elements in the stack, then Stack is empty is printed.
7) loadSP / storeSP : For of serving the purpose of being a non-volatile memory stack it is necessary that the stack pointer SP needs to be stored in the stack as well, SP can be stored anywhere in the EEPROM, for simplicity we store it at address 0, these functions are executes in the initialization of the stack and when there is a change done to the stack by push or pop operations.
8) Clear stack: The clear operation removes all the elements from a stack simply by making the stack pointer value to 0.

HARDWARE USED:-

AT24C32 I2C (EEPROM WITH 32KB STORAGE)
ESP8266-12E WITH TENSILICA CPU AND SERIAL INTERFACE VIA UART AT 115200bits/sec BAUD RATE
BATTERY AND BMS WITH 3.3V LINEAR REGULATOR
CP2102 INTERFACE PROGRAMMER BOARD TO CONVERT USB TO UART SIGNALS AND VICE-VERSA
IDE: ARDUINO IDE WITH EMBEDDED C++

USAGE OF EEPROM:-

EEPROM stands for electrically erasable programmable read-only memory and is a type of non-volatile memory. EEPROMs are organized as arrays of floating-gate transistors. EEPROMs can be programmed and erased in-circuit, by applying special programming signals. EEPROM devices use a serial or parallel interface for data input/output. The client/user can change the quality of certain units without expecting to erase the programming on different cells. Consequently, areas of data might be erased and replaced without expecting to adjust whatever rest of the chip programming. Data saved in an EEPROM chip is permanent, until the client chooses to delete and replace the data that it contains. The information saved in an EEPROM chip is not lost even when power is turned off. The EEPROM uses the principle same as that of the UV-EPROM. The electrons which are trapped in a floating gate will modify the characteristics of the cell, so instead of that logic “0” or logic “1” will be stored.

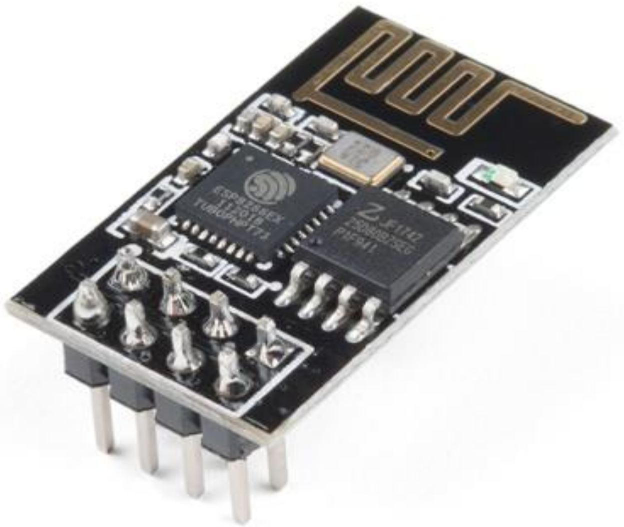
ESP8266 MICROCONTROLLER:-

ESP8266 is a wifi SOC (system on a chip). ESP8266 can be used as an external Wifi module, using the standard AT Command set Firmware by connecting it to any microcontroller using the serial UART, or directly serve as a Wifi-enabled micro controller, by programming a new firmware using the provided SDK.

It has been used mostly in IoT contexts, where we want to add connectivity for example to an Arduino project which makes it a lot more user friendly. Supports serial communication hence compatible with many development platform like Arduino. Can be programmed using **Arduino IDE** or AT-commands or Lua Script. I/O Voltage: 3.6V (max). The maximum current that can be drawn from a single GPIO pin is 12mA**.**

The ESP8266 has 17 GPIO pins (0-16), however, you can only use 11 of them, because 6 pins (GPIO 6 - 11) are used to connect the flash memory chip. This is the small 8-legged chip right next to the ESP8266. GPIO 1 and 3 are used as TX and RX of the hardware Serial port (UART), so in most cases, you can't use them as normal I/O while sending/receiving serial data. The default PWM range is 10-bits @ 1kHz, but this can be changed (up to >14-bit@1kHz). ESP8266 Arduino module comes with PCB trace antenna which seems to be have a good average.

ESP8266-01 development board:



APPLICATIONS:-

1. DFS Algo - Depth First Search

2. Graph Connectivity

3. Newer Technologies such as FRAM and MRAM are replacing by EEPROMs in some applications.

CODE:-

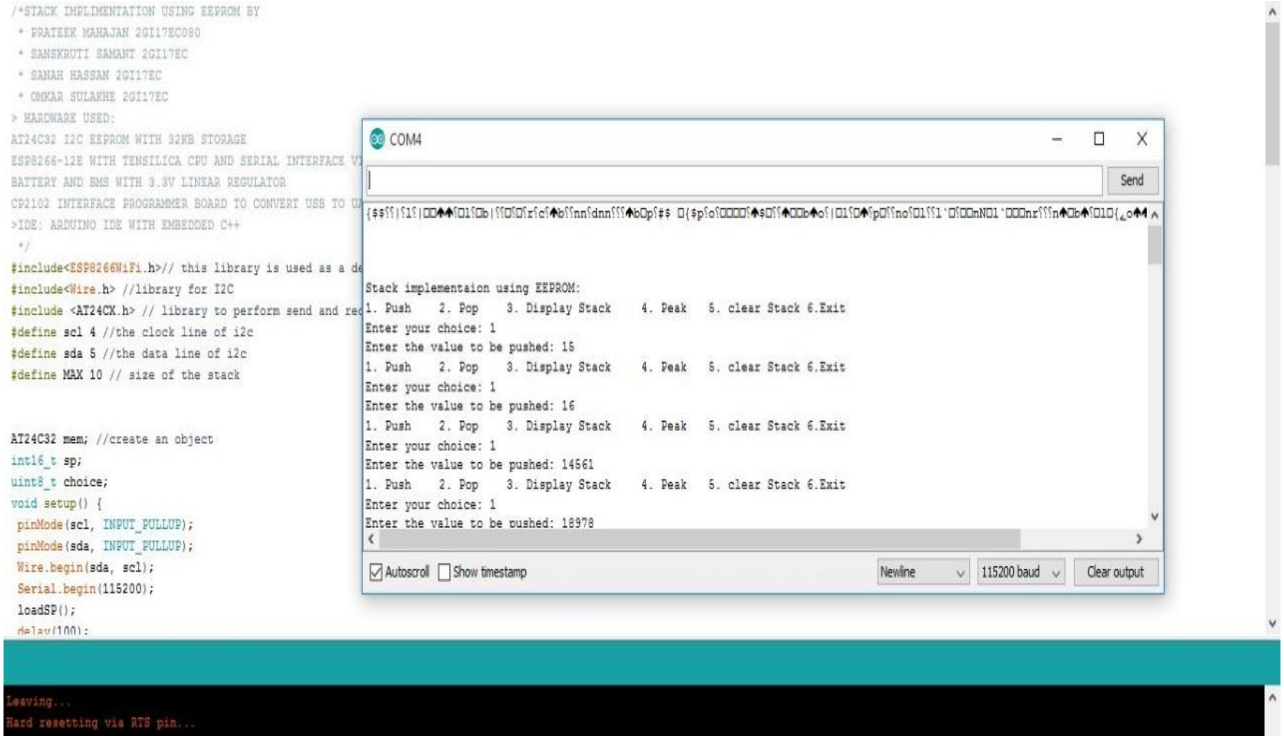
```
#include<ESP8266WiFi.h> // this library is used as a default library which contains register definitions
#include<Wire.h>  //library for I2C
#include <AT24CX.h> // library to perform send and receive operations to the EEPROM
#define scl 4 //the clock line of i2c
#define sda 5 //the data line of i2c
#define MAX 10 // size of the stack


AT24C32 mem; //create an object
int16_t sp;
uint8_t choice;
void setup() {
pinMode(scl, INPUT_PULLUP);
pinMode(sda, INPUT_PULLUP);
Wire.begin(sda, scl);
Serial.begin(115200);
loadSP();
delay(100);

Serial.println("\n\n\n\nStack implementaion using EEPROM: ");
}

void loop() {

Serial.printf("1. Push\t 2. Pop\t 3. Display Stack\t 4. Peak\t 5. clear Stack 6.Exit\n");
Serial.print("Enter your choice: ");
while(!Serial.available());
```

```
choice=(Serial.readString()).toInt();
Serial.println(choice);
switch(choice){
case 1:{
Serial.print("Enter the value to be pushed: ");
while(!Serial.available());
int16_t val=Serial.readString().toInt();
Serial.println(val);
push(val);
break;
}
case 2:
Serial.printf("The value popped is %d\n", pop());
break;
case 3:
displayStack();
break;
case 4:
Serial.printf("Peak value is %d\n", peak());
break;
case 5:
clearStack();
break;
case 6:
Serial.println("Thankyou");
ESP.deepSleep(0);
break;
default: Serial.println("Enter a valid choice\n");
}
}

void push(int16_t val){
if(sp==2*MAX){
Serial.println("Stack Overflow");
return;
}
sp+=2;
storeSP();
```

```
mem.writeInt(sp, val);
}


int16_t pop(){
if(sp==0){
Serial.println("Stack underflow");
return 0;
    }
   sp-=2;
   storeSP();
  return mem.readInt(sp+2);
  }


void displayStack(){
  if(sp==0){
    Serial.println("Stack empty");
    return;
  }
  Serial.println("Displaying Stack:\n");
  for(int i=sp; i>0; i-=2)
    Serial.printf("%d\n",(int16_t)
mem.readInt(i));
  }


int16_t peak(){
  return mem.readInt(sp);
  }


void loadSP(){
  sp=mem.readInt(0);
  }


 void storeSP(){
```

```
 mem.writeInt(0,sp);
 }


void clearStack(){
 mem.writeInt(0,0);
 loadSP();
 }
```

SNAPSHOTS (OUTPUT DIALOGUE AND CIRCUIT REPRESNTATION):-



```
/*STACK IMPLIMENTATION USING EEPROM BY
 * PRATEEK MAHAJAN 2GI17EC080
 * SANSKRUTI SAMANT 2GI17EC
 * SANAH HASSAN 2GI17EC
 * OMKAR SULAKHE 2GI17EC
> HARDWARE USED:
AT24C32 I2C EEPROM WITH 32KB STORAGE
ESP8266-12E WITH TENSILICA CPU AND SERIAL INTERFACE VI
BATTERY AND BMS WITH 3.3V LINEAR REGULATOR
CP2102 INTERFACE PROGRAMMER BOARD TO CONVERT USB TO UA
>IDE: ARDUINO IDE WITH EMBEDDED C++
 */
#include<ESP8266WiFi.h>// this library is used as a de
#include<Wire.h> //library for I2C
#include <AT24CX.h> // library to perform send and rec
#define scl 4 //the clock line of i2c
#define sda 5 //the data line of i2c
#define MAX 10 // size of the stack


AT24C32 mem; //create an object
int16_t sp;
uint8_t choice;
void setup() {
 pinMode(scl, INPUT_PULLUP);
 pinMode(sda, INPUT_PULLUP);
 Wire.begin(sda, scl);
 Serial.begin(115200);
 loadSP();
 delay(100);
```

```
Stack implementaion using EEPROM:
1. Push    2. Pop    3. Display Stack    4. Peak    5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 15
1. Push    2. Pop    3. Display Stack    4. Peak    5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 16
1. Push    2. Pop    3. Display Stack    4. Peak    5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 14561
1. Push    2. Pop    3. Display Stack    4. Peak    5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 18978
```

☑ Autoscroll ☐ Show timestamp   Newline   115200 baud   Clear output

```
Leaving...
Hard resetting via RTS pin...
```

OUTPUT:

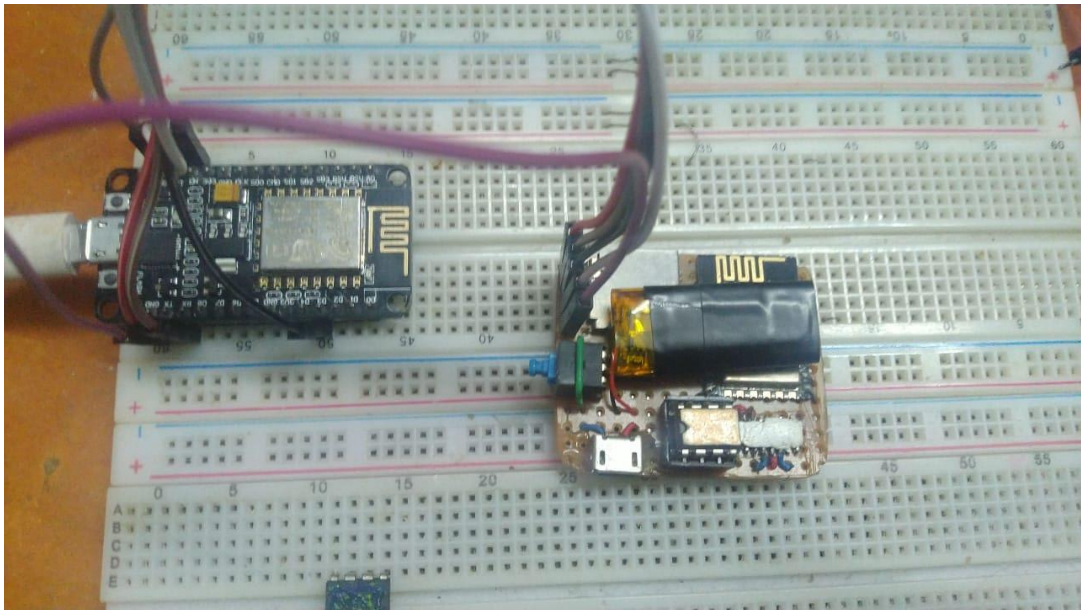
```
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 15
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 16
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 14561
```

```
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 18978
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 15642
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 15642
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 18978
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 12
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 123
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 256
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 789
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 1789
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 317
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 11
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 16
Stack Overflow
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
```

```
Enter your choice: 3
Displaying Stack:

11
317
1789
789
256
123
12
14561
16
15
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 4
Peak value is 11
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 11
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 317
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 1789
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 789
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 256
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 123
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 12
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
```

```
Enter your choice: 2
The value popped is 14561
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 16
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
The value popped is 15
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 2
Stack underflow
The value popped is 0
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 1
Enter the value to be pushed: 12
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 3
Displaying Stack:

12
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 5
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 3
Stack empty
1. Push 2. Pop  3. Display Stack        4. Peak         5. clear Stack 6.Exit
Enter your choice: 6
Thankyou
```

HARDWARE IMPLEMENTATION:-



Analysis:

ADVANTAGES:-

1**.** Using stack helps in very fast accessing of data

2. Space is managed efficiently by CPU, memory will not become fragmented

3. It is very easy to program and erase the contents of EEPROM without removing it from board or test jig.

4. It is possible to re-program EEPROM infinite number of times.

DISADVANTAGES:-

1**.** The size of the stack must be defined in prior and cannot be changed.

2. Space Complexity for n push operations.

3. EEPROM requires different voltages for erasing, reading and writing the data.

4. EEPROM has limited data retention time period.