T. Y. B. Tech (ECE)

Semester: VI **Subject:** ESD & RTOS

Name: Mayank Singhania **Division: B**

Roll No: PB29 **Batch: B3**

Experiment No: 07

Name of the Experiment: Interfacing EEPROM with LPC2148 using I2C

Performed on:

Submitted on:

Marks	**Teacher's Signature with date**

Aim: Write Embedded C program for interfacing EEPROM with LPC2148 using I2C.

Part List:

- Educational practice board for ARM7 LPC2148
- All in One General Purpose Board (ASK25)
- +9V Power supply
- USB A to B type cable
- 4 Pin relimate cable
- PC
- Eclipse IDE
- Flash Magic Utility

Theory:

The I2C (Inter-IC) bus is a bi-directional two-wire serial bus that provides a communication link between integrated circuits (ICs).I2C is a synchronous protocol that allows a master device to initiate communication with a slave device. Data is exchanged between these devices. EEPROM (electrically erasable programmable read-only memory) is user-modifiable read-only memory (ROM) that can be erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage. It is a type of non-volatile memory used in computers and other electronic devices to store small amounts of data that must be saved when power is removed, e.g., calibration tables or device configuration. The LPC2148 is configured as a master and controls the EEPROM device which is configured as a slave through I2C protocol. The read-write operations are accomplished by sending a set of control signals including the address and/or data bits. The control signals must be accompanied with proper clock signals.

Hardware Connection:

- Connect 20 pin flat cable between J1 connector of ARM7 board and PL3 connector of ASK25 board.
- Connect a 4 pin relimate cable between J7 of EPBARM7 board and PL2 connector of ASK25.

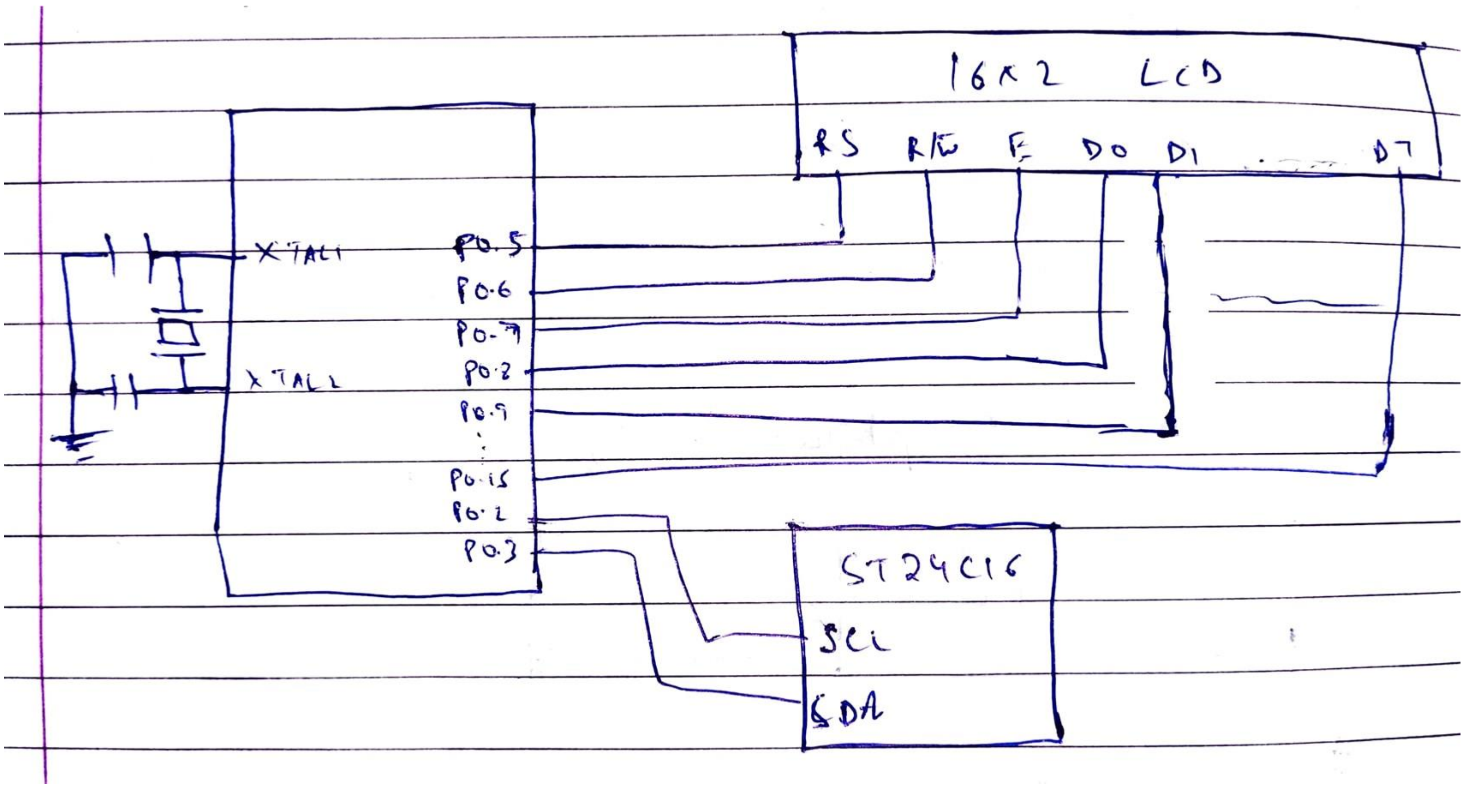
Pin Connection	PL2 Connector (ASK25)	J7 Connector (EPBARM7)
1	VCC	VCC
2	SDA	P0.3
3	SCL	P0.2
4	GND	GND

For LCD Connection:

Pin No.	PL3 Connector of Ask25	J1 Connector of EPBARM7
1		P0.0
2		P0.1
3		P0.2
4		P0.3
5		P0.4
6	RS	P0.5
7	R/W*	P0.6
8	E	P0.7
9		NC
10	D0	P0.8
11	D1	P0.9

12	D2	P0.10
13	D3	P0.11
14	D4	P0.12
15	D5	P0.13
16	D6	P0.14
17	D7	P0.15
18		3.3V
19	5V	5.0V
20	GROUND	GND

Interfacing Diagram:



Program:

```
#include "LPC214x.h"
#include "lcd.h
#define PCLK 12000000
void DelayMs(unsigned int Ms);
void ConfigI2c0(int BaudRate);
void WriteEeprom0(char SlaveAddress, unsigned char *Data,char len);
void ReadEeprom0(char SlaveAddress, unsigned char *Data, char len);
unsigned char addr= 0x00;
int main (void)
{
unsigned char EepromBufA[]={0x00,'E','C','E'};
 unsigned char EepromBufB[2]={0x00,0};
 unsigned int No;

 edutechlcdinit();
    ConfigI2c0(100);

    WriteEeprom0(0xA0,EepromBufA,4);
    edutechlcdstring("Write Success",1,0);

    for(No=0; No<4; No++)
    {
    ReadEeprom0(0xA0,EepromBufB,1);
    edutechlcdstring(&EepromBufB[0],2,No);
    DelayMs(100); // This delay is important
    addr++;
    }
    while(1);
```

```
  return 0;
}

void DelayMs(unsigned int Ms)
{
Ms = Ms * 200;
while (Ms)
--Ms;
}

void ConfigI2c0(int BaudRate)
{
PINSEL0 = 0x050;
I2C0CONCLR = 0x6c; // disable I2C
I2C0CONSET = 0x40; // Enable I2C

I2C0SCLH = (PCLK/(2*(BaudRate*1000)));
I2C0SCLL = (PCLK/(2*(BaudRate*1000)));
}

unsigned char I2C_WaitStatus0(unsigned char I2CStatus)
{
unsigned int Time,I2C_WAIT_TIME_OUT;
I2C_WAIT_TIME_OUT = 200;
Time = 0;

while (Time++ < I2C_WAIT_TIME_OUT)
{
if (I2C0CONSET & 8) // poll SI bit for Comm complete
{
```

```
if (I2C0STAT == I2CStatus)// read I2C status value
{
Time = 0;
return 1;
}
}
}
return 0;
}

char WriteDataI2c0(char SlaveAddress, char *Data,unsigned char len)
{
I2C0CONCLR = 0x2c; // disable I2C
I2C0CONSET = 0x40; // Enable I2C
I2C0CONSET |= 0x20;

if (!I2C_WaitStatus0(0x08)) // 0x08: ready for device address
return 0;

I2C0DAT = SlaveAddress;
I2C0CONCLR = 0x2C;
if (!I2C_WaitStatus0(0x18)) // 0x08: ready for device address
return 0;

I2C0DAT = *Data++;
I2C0CONCLR = 0x2C;
if (!I2C_WaitStatus0(0x28)) // 0x08: ready for device address
return 0;
```

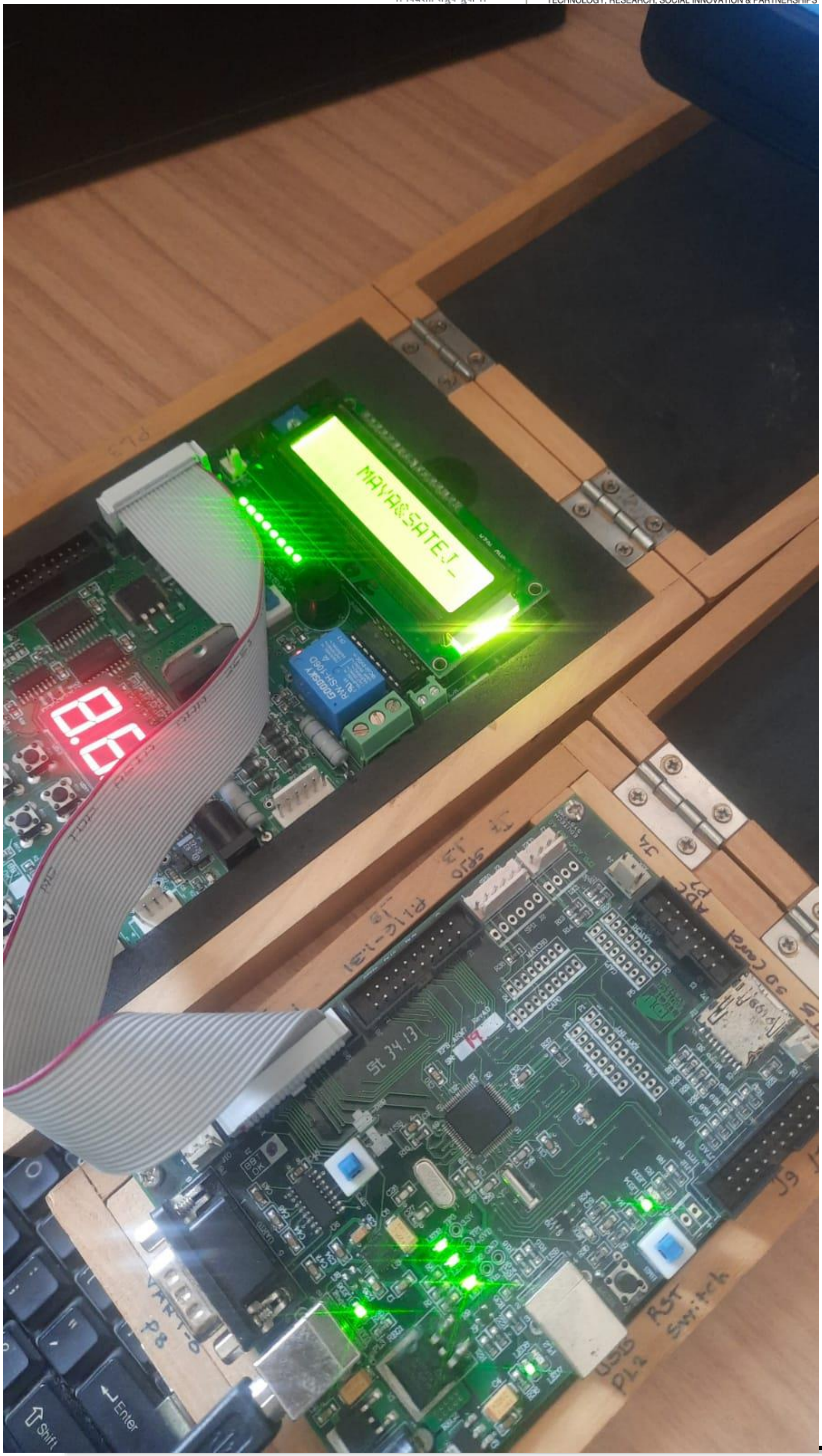
```
while(len)
{
I2C0CONCLR=0x2C;

if (!I2C_WaitStatus0(0x28)) // 0x08: ready for device address
return 0;

I2C0DAT = *Data++ ;

if(len>1)
I2C0CONSET = 0x04 | 0x40;
else
I2C0CONSET = (0x10 | 0x40);

len--;
}

I2C0CONSET = (0x10 | 0x40); // Send Stop bit.
I2C0CONCLR=0x2C;

return(1);
}

char ReadDataI2c0(char SlaveAddress, char *Data, char len)
{

I2C0CONCLR = 0x2c; // disable I2C
I2C0CONSET = 0x40; // Enable I2C
I2C0CONSET |= 0x20;
```

```
if (!I2C_WaitStatus0(0x08)) // 0x08: ready for device address
return 0;

I2C0DAT = SlaveAddress;
I2C0CONCLR = 0x28;
if (!I2C_WaitStatus0(0x40)) // 0x08: ready for device address
return 0;

while(len)
{
I2C0CONCLR=0x2C;

if (!I2C_WaitStatus0(0x58)) // 0x08: ready for device address
return 0;

*Data++ = I2C0DAT;

if(len>1)
I2C0CONSET = 0x04 | 0x40;
else
I2C0CONSET = (0x10 | 0x40);
len--;
}

I2C0CONSET = (0x10 | 0x40); // Send Stop bit.
I2C0CONCLR=0x2C;

return(1);
}
void WriteEeprom0(char SlaveAddress, unsigned char *Data,char len)
```

```
{
  WriteDataI2c0(SlaveAddress,Data,len);
}


void ReadEeprom0(char SlaveAddress, unsigned char *Data, char len)
{
WriteDataI2c0(SlaveAddress,&addr,0);
ReadDataI2c0((SlaveAddress + 0x01),Data,len);
}
```

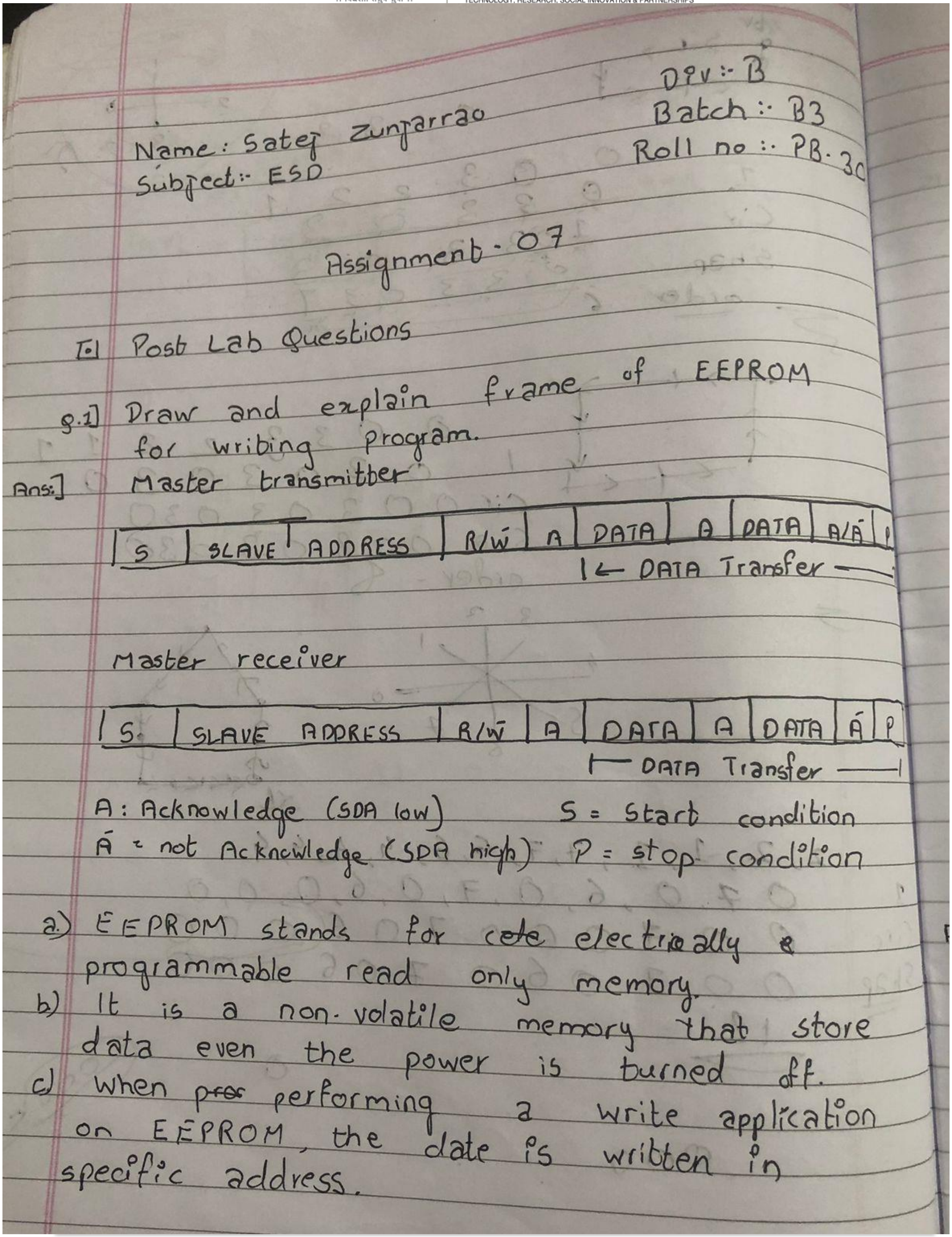
Result:

Apply reset and you will see write status of data on EEPROM and data written on EEPROM would be displayed on LCD.


MAYA&SATEJ_

Conclusion:

From this experiment, we learnt about I2C protocol and used it to configure EEPROM with microcontroller LPC2148. We also learnt about the features of EEPROM and wrote the code for the same and verified it successfully.



Div :- B
Batch :- B3
Roll no :- PB-30

Name : Satej Zunjarrao
Subject :- ESD

Assignment - 07

[•] Post Lab Questions

Q.1] Draw and explain frame of EEPROM for writing program.

Ans:] Master transmitter

S	SLAVE ADDRESS	R/W̄	A	DATA	A	DATA	A/Ā	P

|← DATA Transfer ——

Master receiver

S	SLAVE ADDRESS	R/W̄	A	DATA	A	DATA	Ā	P

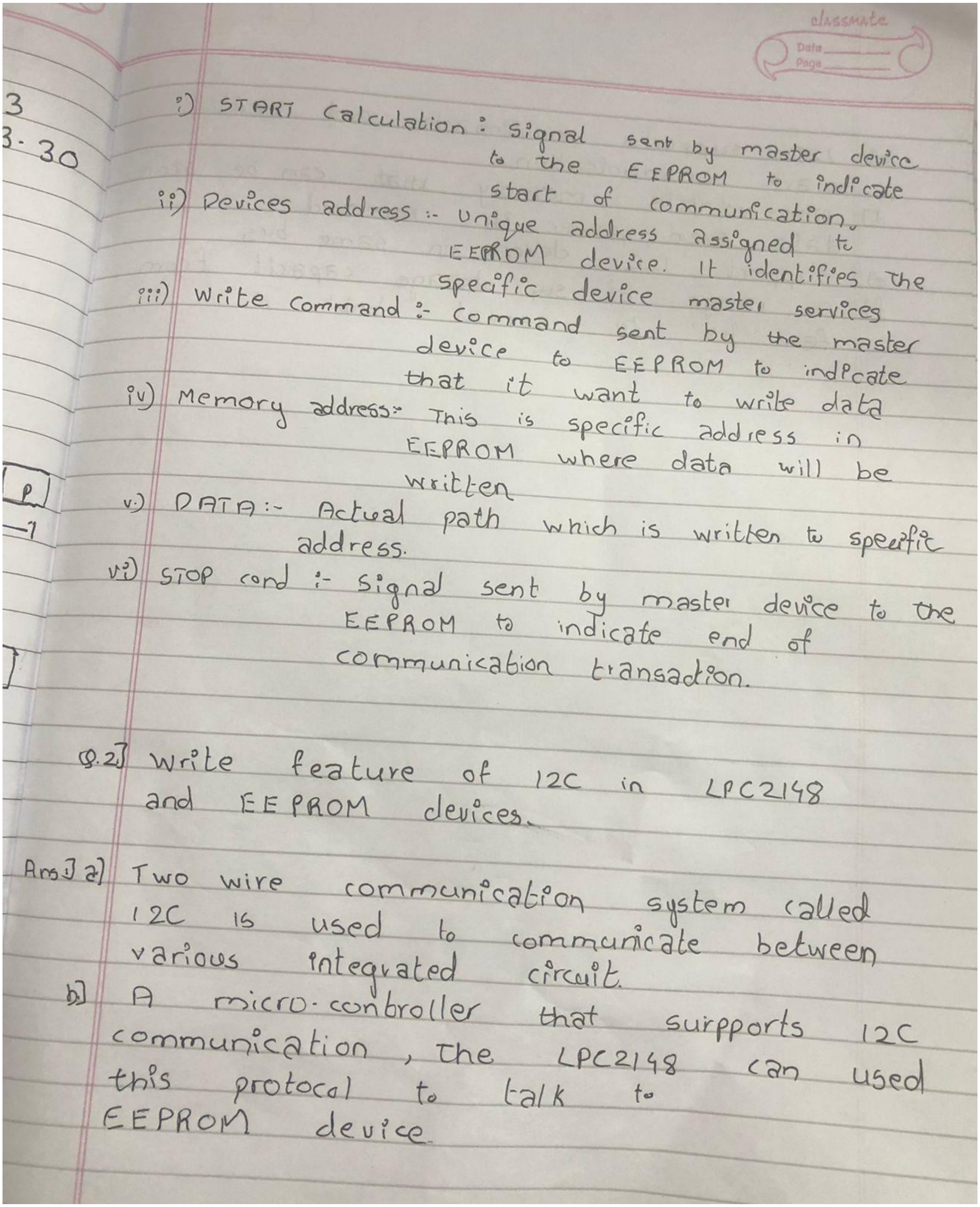
|— DATA Transfer ——|

A : Acknowledge (SDA low) S = Start condition
Ā = not Acknowledge (SDA high) P = stop condition

a) EEPROM stands for ~~cete~~ electrically e programmable read only memory.

b) It is a non-volatile memory that store data even the power is turned off.

c) When ~~pror~~ performing a write application on EEPROM, the date is written in specific address.



3
3·30

i) START Calculation : Signal sent by master device to the EEPROM to indicate start of communication.

ii) Devices address :- Unique address assigned to EEPROM device. It identifies the specific device master services

iii) Write Command :- command sent by the master device to EEPROM to indicate that it want to write data

iv) Memory address:- This is specific address in EEPROM where data will be written

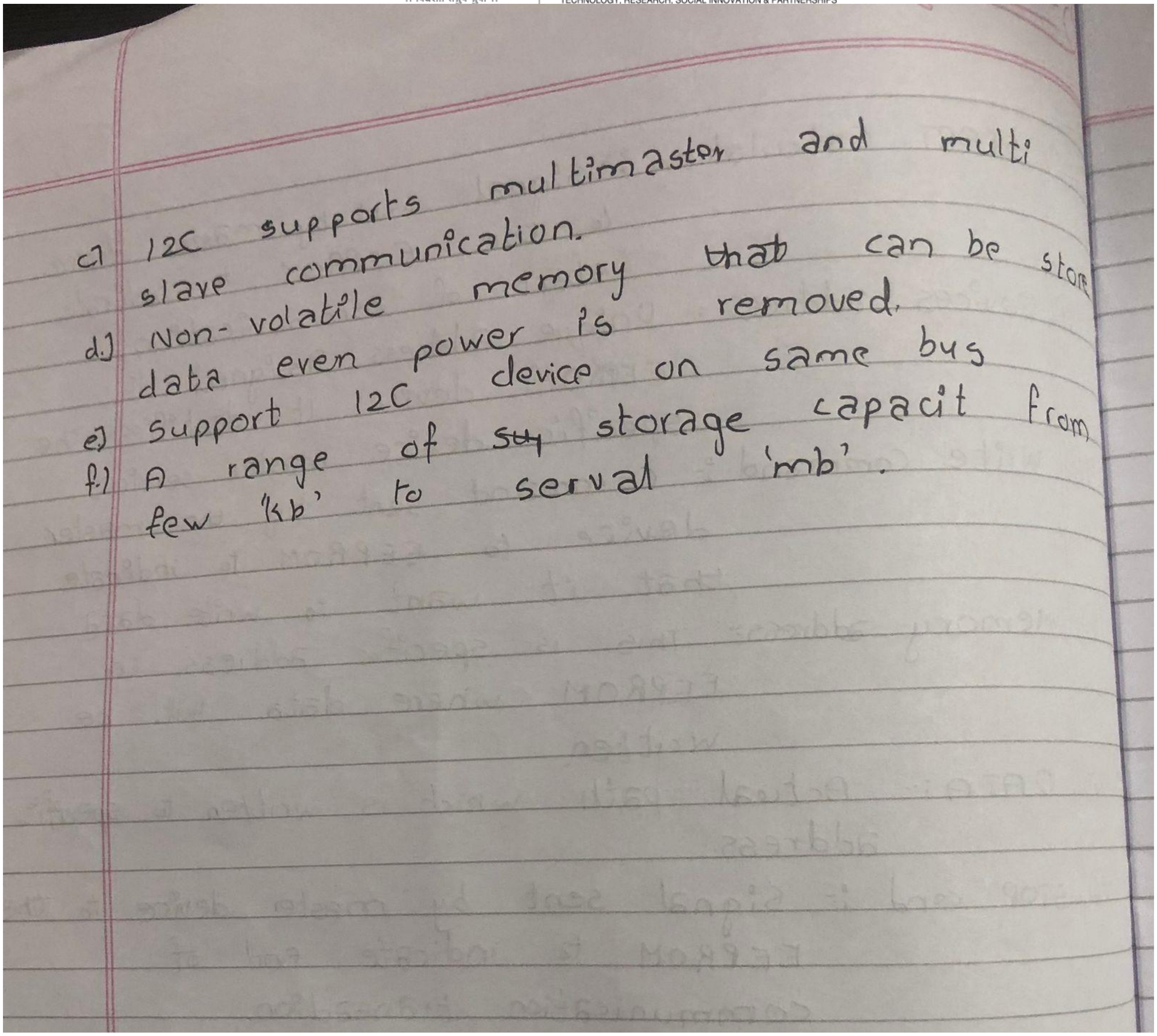
v) DATA :- Actual path which is written to specific address.

vi) STOP cond :- Signal sent by master device to the EEPROM to indicate end of communication transaction.

Q.2] Write feature of I2C in LPC2148 and EEPROM devices.

Ans] a) Two wire communication system called I2C is used to communicate between various integrated circuit.

b) A micro-controller that surpports I2C communication, the LPC2148 can used this protocol to talk to EEPROM device.



c) I2C supports multimaster and multi slave communication.

d.) Non-volatile memory that can be store data even power is removed.

e) support I2C device on same bus

f.) A range of ~~su~~ storage capacit from few 'kb' to serval 'mb'.

MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Dr. Vishwanath Karad
MIT WORLD PEACE
UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

www.mitwpu.edu.in