

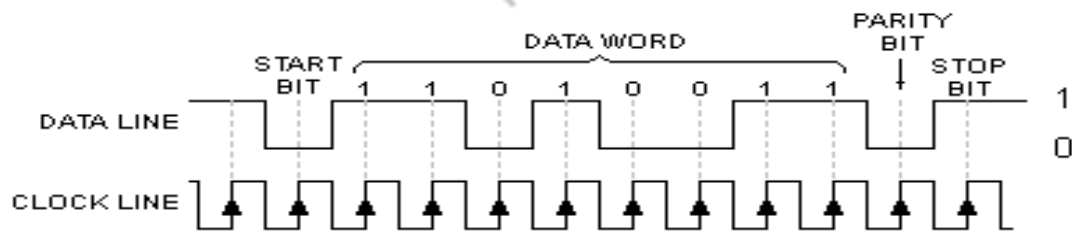
UNIT – VI Serial Port Programming Interfacing with PIC 18FXXXXQ.

Q 1. Write short note about RS232, RS 485 Protocol

RS232 Protocol

- RS232 was introduced in 1962
- The specification allows for data transmission from one transmitter to one receiver at relatively slow data rates (up to 20K bits/second) and short distances (up to 50Ft. @ the maximum data rate).
- The RS-232 supports full-duplex communications.
- IN RS232 communication, the transmitting device is the DTE such as a computer, and the receiving device is the DCE such as a printer.
- The RS-232 serial communication protocol is a standard protocol used in asynchronous serial communication.
- An interface chip known as the universal asynchronous receiver/transmitter or UART is used to implement serial data transmission.
- Every RS-232 consists of following data format:

1 start bit, 8 data bits, 1 stop bit, (optional 1 parity bit).



- In RS232 logic 0 and 1 is represented using following voltage levels,

Logic '0' = +3v to +5 v

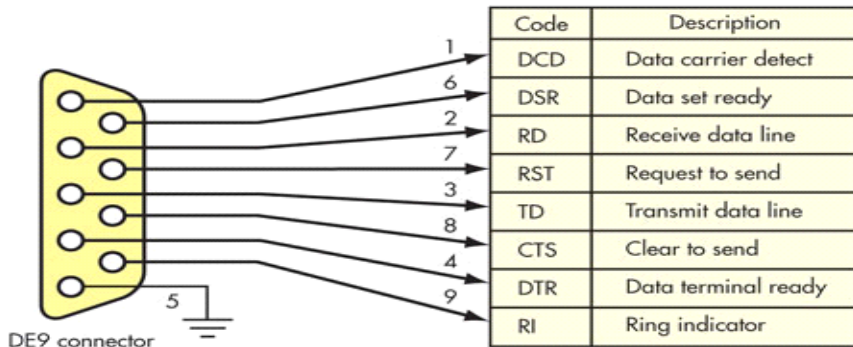
Logic '1' = -3v to -5 v

And -3 to +3 voltage range is undefined.

- The RS232 signals are represented by voltage levels with respect to a system common that is ground; this is called **single ended communication**.

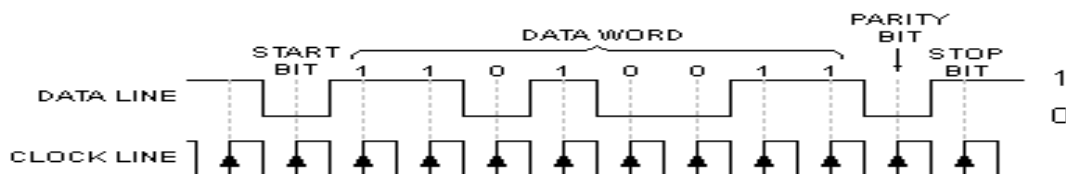
RS-232 Standard Cabling

- The cable medium can be simple parallel wires or twisted pair.
- RS-232 uses DB-25 or DB-9 pin connector to connect cable between two ends.
- DB-9 has become more popular over last several years due to compact size.



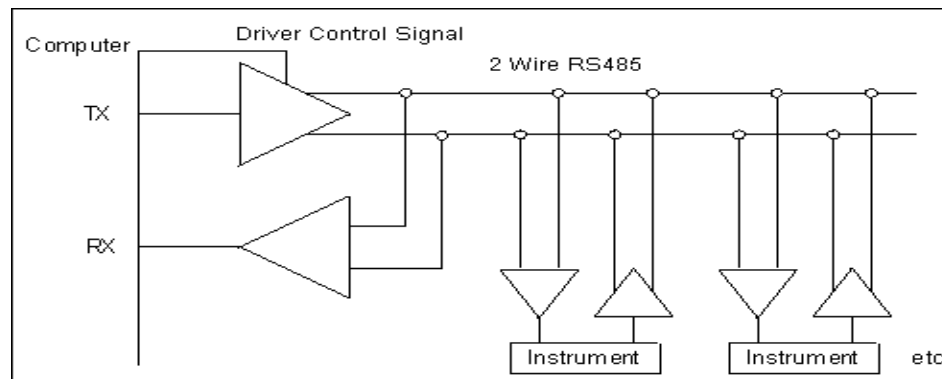
RS 485 Protocol

- The Electronics Industry Association (EIA) has produced standards RS485, which deal with data communications.
- RS-485 allows multiple devices (up to 32) to communicate at half-duplex on a single pair of wires, plus a ground wire at distances up to 1200 meters (4000 feet).
- Both the length of the network and the number of nodes can easily be extended using a variety of repeater.
- The RS-485 standard uses differential signaling on two lines rather than single-ended with a voltage referenced to ground
- In Differential data transmission The RS485receiver compares the voltage difference between both lines, instead of the absolute voltage level between on a signal line.
- Differential data transmission allows longer cables(hundreds of meters) and Allows higher speeds (in Mbps)
- Every RS-232 consists of following data format:
1 start bit, 8 data bits,1 stop bit,(optional 1 parity bit).



9. A 485 network can be configured two ways, "two-wire" or "four-wire."

- In a "two-wire" network the transmitter and receiver of each device are connected to a twisted pair.



12. The RS-485 interfaces is widely used in industrial applications where higher speeds and longer distances are needed.

Q 2. Compare RS232 and RS 485 Protocol

Parameter	RS-232	RS-485
Line configuration	Single-ended	Differential
Mode of operation	Simplex or full duplex	Simplex or half duplex
Maximum cable length	50 feet	4000 feet
Maximum data rate *	20 kbits/s	10 Mbits/s
Typical logic levels	± 5 to ± 15 V	± 1.5 to ± 6 V
Minimum receiver input impedance	3 to 7 k Ω	12 k Ω
Receiver sensitivity	± 3 V	± 200 mV

Q 3. Explain SPI Protocol

SPI (Serial Peripheral Interface) Protocol

1. SPI was first introduced by Motorola in 1979. SPI defined the external microcontroller bus, used to connect the microcontroller peripherals with 4 wires.
2. SPI Bus is a way to communicate between 2 digital devices.
3. Both SPI and I2C protocols are well-suited for communications between integrated circuits, for slow communication with on-board peripherals

Protocol

1. SPI is a protocol on 4 signal lines
 - A clock signal named SCLK, sent from the bus master to all slaves; all the SPI signals are synchronous to this clock signal;
 - A slave select signal for each slave, SS_n, or CS (chip select) used to select the slave the master communicates with;
 - A data line from the master to the slaves, named MOSI (Master Out-Slave In)
 - A data line from the slaves to the master, named MISO (Master In-Slave Out).
2. SPI is a single-master communication protocol. This means that one central device initiates all the communications with the slaves.
3. When the SPI master wishes to send data to a slave and/or request information from it, it selects slave by pulling the corresponding SS line low and it activates the clock signal at a clock frequency usable by the master and the slave.
4. SPI provides full duplex communication using these two data lines MISO and MOSI

(Draw Architecture Given below in comparison)

Q 4. Explain I2C Protocol

I2C ('Inter-Integrated Circuit') Protocol

1. The I²C bus was developed in 1982; its original purpose was to provide an easy way to connect a CPU to peripherals chips in a TV set.
2. The original specification defined a bus speed of 100 kbps (kilobits per second). The specification was reviewed several times, notably introducing the 400 kbps speed in 1995 and – since 1998, 3.4 Mbps for even faster peripherals.

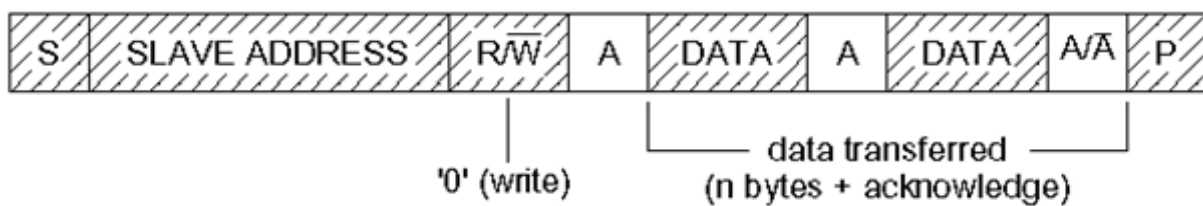
Protocol

1. I²C is a multi-master protocol that uses 2 signal lines.

2. The two I²C signals are called 'serial data' (SDA) and 'serial clock' (SCL). There is no need of chip select (slave select) or arbitration logic.
3. Virtually any number of slaves and any number of masters can be connected onto these 2 signal lines and communicate between each other using a protocol that defines:
 - 7-bits slave addresses: each device connected to the bus has got such a unique address, $2^7 = 128$ devices can communicate.
 - In extended I²C 10 bit addressing (up to $2^{10} = 1024$ addresses)
 - data divided into 8-bit bytes
4. Physically, the I²C bus consists of the 2 active wires SDA and SCL and a ground connection.
5. The active wires are both bi-directional.
6. Sequence of write operation by I²C bus protocol is as follows,

- 1) Send the start condition
- 2) Send the slave address
- 3) Check for the acknowledge
- 4) Send the sub-address to be written to
- 5) Check for the acknowledge from
- 6) Send the data to write to specified sub address
- 7) Check for the acknowledge
- 8) If No-acknowledge send the stop condition
- 9) Send a stop condition

Master writing to a Slave

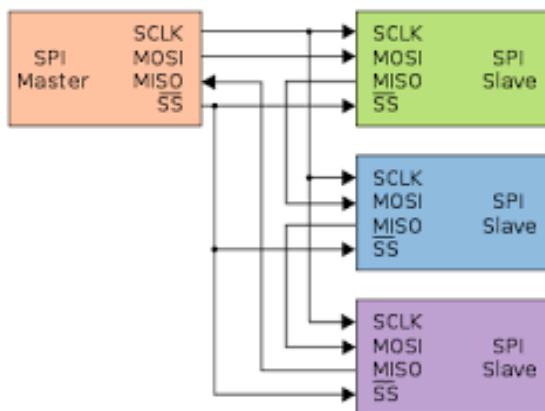


(Draw Architecture Given below in comparison)

Q 5. Compare SPI and I2C Protocol

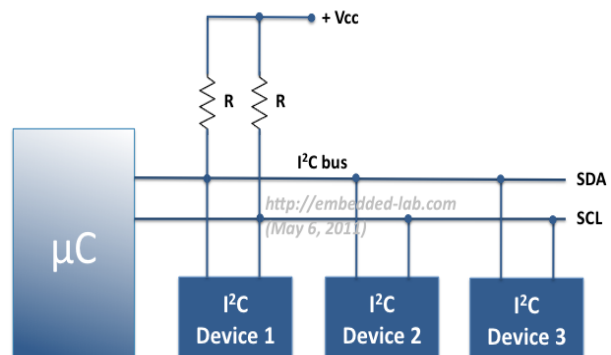
SPI

- Three bus lines are required; a data input line (SI), a data output line (SO) and a serial clock line (SCK) [plus 1 Chip Select (CS)]
- Higher data rates (up to 10 MHz or more), More speed.
- Full Duplex communication
- More efficient in point-to-point (single master, single slave) applications
- Lack of built-in device addressing, Device can be selected using CS signal
- No of devices can be increased with more no of cs/ss signals, more hardware.
- Does not have an acknowledgement mechanism to confirm receipt of data
- Architecture



I2C

- Two bus lines are required; a serial data line (SDA) and a serial clock line (SCL)
- Support transfer speeds of around 100 kHz (original standard, or 400 kHz using the most recent standard) less speed.
- Half Duplex communication
- More efficient in multi-master, multi-slave applications
- Built-in addressing scheme. Device can be selected with its unique address.
- No of devices can be increased without adding extra hardware
- Have an acknowledgement mechanism to confirm receipt of data.
- Architecture



Multiple devices on common I2C bus

Q 6.Elaborate MSSP (Master Synchronous Serial Port) Module of PIC

☐ MSSP is used for communicating the PIC microcontroller with the different peripheral devices like ADC, DAC, EEPROM, RTC, Display Drivers, SD Cards, Temperature Sensor, USB devices

☐ The MSSP module supports two operating modes. They are :

1. Serial Peripheral Interface (SPI)
2. Inter- Integrated Circuit (I2C)

☐ Both the SPI and I2C Protocols share the same signal pins.

However, both these protocols cannot be active at the same time.

The pins used by SPI and I2C MSSP module are:

Serial Data Clock (SCK) : RC3 / SCK

Serial Data In (SDI): RC4/ SDI/ SDA

Serial Data out (SDO): RC5/ SDO

SPI mode	I2C mode
<input type="checkbox"/> The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. <input type="checkbox"/> To accomplish communication, typically three pins are used: <ul style="list-style-type: none">➤ Serial Data Out (SDO) – RC5/SDO➤ Serial Data In (SDI) –RC4/SDI/SDA➤ Serial Clock (SCK) – RC3/SCK/SCL➤ Select (SS) – RA5/AN4/SS/LVDIN	<input type="checkbox"/> The MSSP module in the I2C mode fully implements all master and slave functions <input type="checkbox"/> Both 7-bit and 10-bit addressing are supported. <input type="checkbox"/> Two pins are used for data transfer: <ul style="list-style-type: none"><input type="checkbox"/> Serial clock (SCL)-RC3/SCK/SCL<input type="checkbox"/> Serial data (SDA)-RC4/SDI/SDA
<input type="checkbox"/> The MSSP module has four registers for SPI mode operation. <ol style="list-style-type: none">1. MSSP Status Register (SSPSTAT)2. Serial Receive/Transmit Buffer (SSPBUF)3. MSSP Control Register 1 (SSPCON1)4. MSSP Shift Register (SSPSR)	<input type="checkbox"/> The MSSP has six registers to support the I2C operations: <ol style="list-style-type: none">1) MSSP control register 1 (SSPCON1)2) MSSP control register 2 (SSPCON2)3) MSSP status register (SSPSTAT)4) Serial receive/transmit buffer (SSPBUF)5) MSSP shift register (SSPSR)6) MSSP address register (SSPADDD)

Q 6.Explain interfacing serial port with PC Both side communication

PIC Microcontroller performs serial communications using USART. It has two pins for serial communication TX and RX

- ❑ In PIC microcontroller there are 6 major registers associated with USART.
 - ❑ SPBRG(Serial Port Baud Rate Generator).
 - ❑ TXREG (Transfer Register)
 - ❑ RCREG (Receive Register).
 - ❑ TXSTA (Transmit Status & control register).
 - ❑ RCSTA (Receive Status & Control Register).
 - ❑ PIR1 (Peripheral Interrupt Request Register 1)

Baud rate calculation

- ❑ The PIC18 transmit & receive data serially at different baud rates.
- ❑ The baud rate in PIC18 is programmable.
- ❑ This is done with 8-bit register called as SPBRG.
- ❑ For given crystal frequency, the value loaded into the SPBRG decides the baud rate.

$$SPBRG = \frac{f_{osc}}{(64 * \text{Desired Baud rate})} - 1$$

Using Above formula if we calculate baud rate the value we are getting for SPBRG are given in below table

Assume $f_{osc} = 10 \text{ MHz}$

Baud Rate	SPBRG (Decimal Value)	SPBRG (Hex Value)
38400	3	3
19200	7	7
9600	15	F
4800	32	20
2400	64	40
1200	129	81

Steps to Programm the PIC18 to Transfer Data Serially

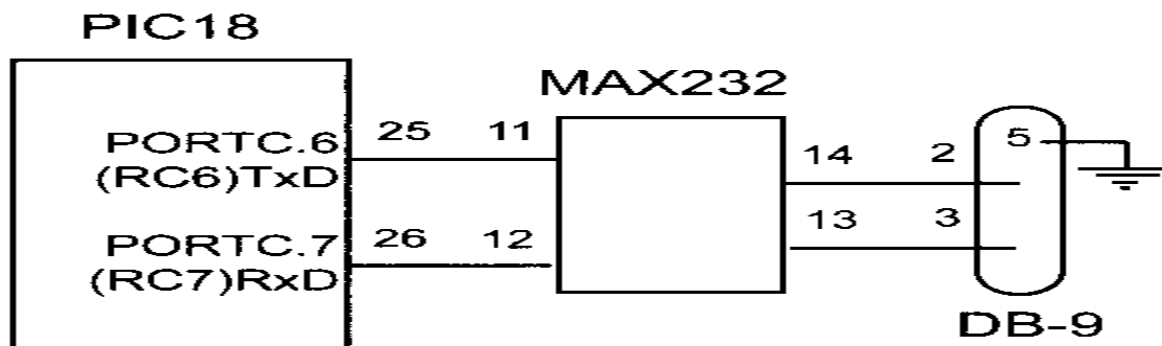
- Make TX pin of PORTC (RC6) as output pin.
- Load TXSTA register with 0x20, Indicating 8-bit data frame, low baud rate and transmit enabled
- Load SPBRG with value for required baud rate.
- To Set SPEN bit of RCSTA register HIGH to enable serial port of PIC18 initialize RCSTA=0x90.
- Write character byte to be transmitted into the TXREG.
- Monitor TXIF flag to set HIGH to indicate transmission complete.
- To transmit next character, go to previous step.

Steps to Programm the PIC18 to Receive Data Serially

- Make RX pin of PORTC(RC7) as input pin.
- Load RCSTA register with 0x90, To enable the continuous receive in addition to the 8-bit data size option.
- Load TXSTA with 0x00 to choose low baud rate option.
- Load SPBRG with value for required baud rate.
- Monitor RCIF flag to set HIGH.
- When RCIF is raised the RCREG has the data byte. Its contents are moved into safe place.

Q 7. Write C code to transfer “YES ” serially

- Interfacing Diagram of PIC18 Serial communication is as below
- Here MAX232 IC is widely used for serial communication among Microcontrollers
- The main function of this integrated circuit is to change the logic levels of TTL/CMOS to RS232 throughout the serial communication process.



```
#include<PIC18F4550.h>
```

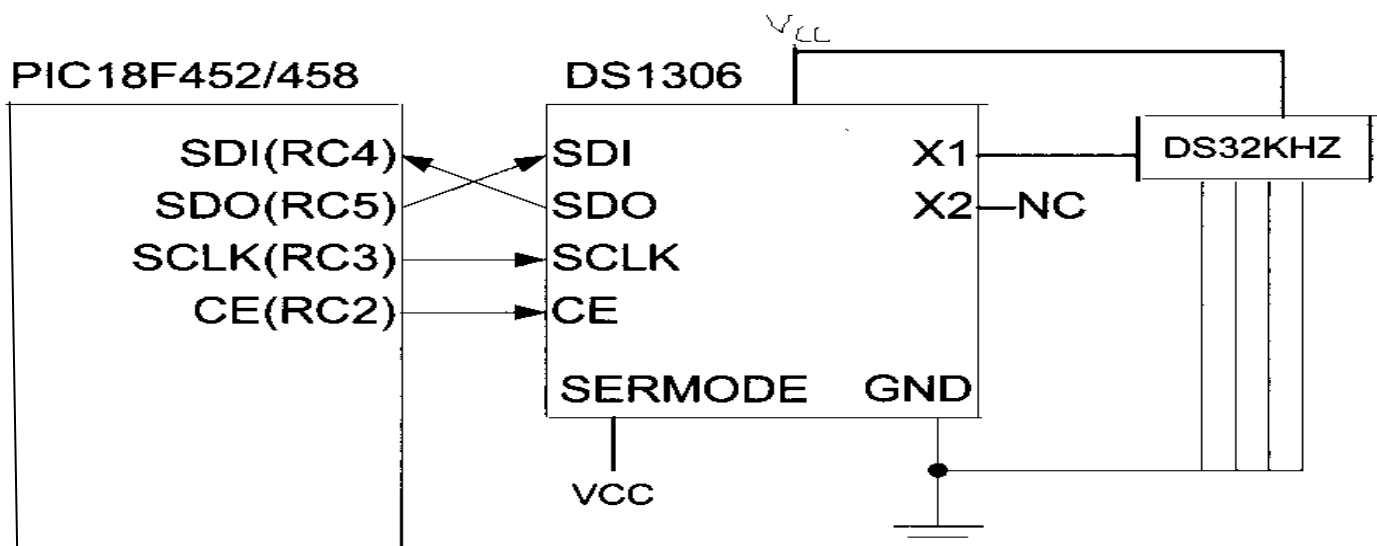
```
void SerTx(unsigned char);
void main(void)
{
    TXSTA=0x20;           //choose low baud rate,8-bit
    SPBRG=15;             //9600 baud rate/ XTAL = 10 MHz
    TXSTAbits.TXEN=1;
    RCSTAbits.SPEN=1;
    while(1)
    {
        SerTx('Y');
        SerTx('E');
        SerTx('S');
    }
}

void SerTx(unsigned char c)
{
    while(PIR1bits.TXIF==0); //wait until transmitted
    TXREG=c;                 //place character in buffer
}
```

Q 8.Explain interfacing of RTC (DS1307) with I2C

The DS1307 is Serial **Real-Time Clock** IC. It is a low-power; full binary-coded decimal (BCD) clock/calendar. Address and data are transferred serially via a 2-wire, bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator.

Interfacing Diagram:



REGISTERS For I2C Operation

The MSSP module has six registers for I2C operation

- ☐ MSSP Control Register 1 (SSPCON1)
- ☐ MSSP Control Register 2 (SSPCON2)
- ☐ MSSP Status Register (SSPSTAT)
- ☐ Serial Receive/Transmit Buffer (SSPBUF)
- ☐ MSSPSHIFT Register (SSPSR)
- ☐ MSSP Address Register (SSPADD)

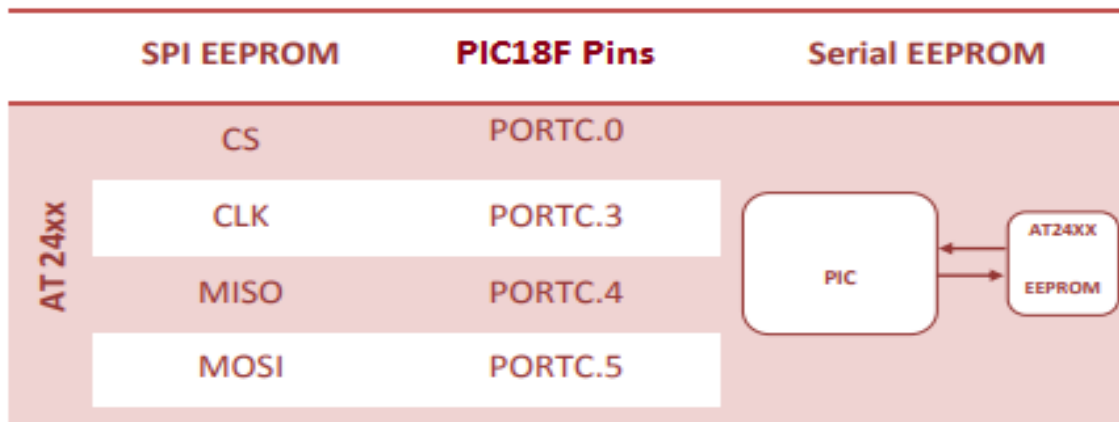
Algorithm for Program

- ☐ We would create these functions:
 1. void i2c_init(void)
 2. void i2c_start(void)
 3. void i2c_restart(void)
 4. void i2c_write(unsigned int data)
 5. void i2c_stop(void)
 6. unsigned char i2c_read(void)

Q 9. Explain interfacing of EEPROM with SPI module of PIC18.

EEPROM (electrically erasable programmable read-only memory) is user-modifiable read-only memory (ROM) that can be erased and reprogrammed (written to) repeatedly.

- The AT24xx serial Electrically Erasable PROM IC, which we use to Interface with PIC Microcontroller



- We now want to Read, write and Erase EEPROM by using SPI in PIC18F Board.
- The basic operation of the SPI based EEPROM's is to send a command, such as WRITE, followed by an address and the data.

- In WRITE operation, the EEPROM to store the data. In SPI, the clock signal is controlled by the master device PIC18F Board.
- All data is clocked in and out using this pin.
- The EEPROM read & write operations are done in PIC18F Kit by using these CS, CLK, MOSI, MISO SPI lines
- The Read & Write operations are performed in EEPROM with EEPROM address. When the EEPROM address is correct, then only you can write, read, and erase data's correctly in EEPROM.

Algorithm for Program

- ❑ We use following SPI COMMANDS

```
#define READ 0x03
#define WRITE 0x02
#define WRDI 0x04
#define WREN 0x06
#define RDSR 0x05
#define WRSR 0x01
```

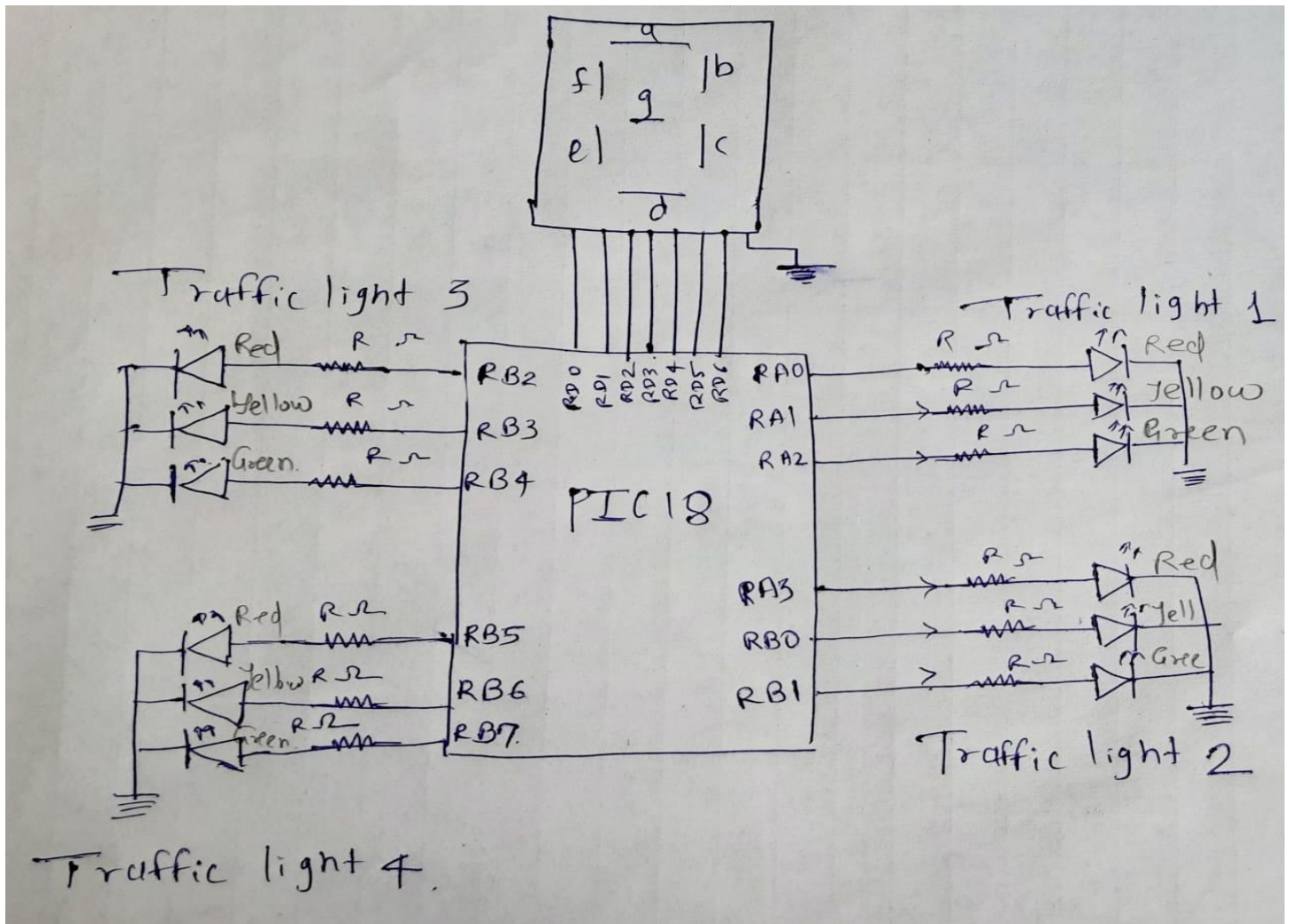
- ❑ We would create these functions:

```
void Serial_init(void);
void SPi_init(void);
void SPi_WRITE(unsigned char);
unsigned char SPi_RDSR(void);
unsigned char SPi_READ(unsigned char);
void DelayMs(unsigned int);
```

Q 10.Design of Traffic Light Controller Using PIC18

- ❑ A typical example application is controlling traffic lights. We chose to design and build a basic traffic signal for four tracks traffic intersection, three signals per track, green, yellow and red.
- ❑ Using microcontroller PIC18F 4550 to get outputs in a specific times according to the data specified in the program, which will be drive twelve LEDs that represent the different signals for each track.
- ❑ The objective is, to design and implement a traffic light control. This traffic light controller is used at the intersection that consists of main road and two side roads.
- ❑ The system is to be developed with the PIC18F4550 chip being the microcontroller that is programmed
- ❑ The drawback is that it does not take into account variables such as cars wanting to turn left or pedestrians pressing a crossing button.

Interfacing Diagram:



LED Connection

Signal target	PORT-bit	Light color
Traffic Light 1	A0	Red
	A1	Yellow
	A2	Green
Traffic Light 2	A3	Red
	B0	Yellow
	B1	Green
Traffic Light 3	B2	Red
	B3	Yellow
	B4	Green
Traffic Light 4	B5	Red
	B6	Yellow
	B7	Green

- ☐ The red LED indicates “stop driving”, the yellow LED indicates “start stopping” and the green LED indicates “drive”.
- ☐ The sequence of altering the LEDs according to their color is as below:

Green-Yellow-Red-Green.

Twelve LEDs are used; three to each traffic light.

- ☐ 7-segment LED displays are used to show the current count value.
- ☐ Since all of the traffic lights are working simultaneously, each one is to display a different digit than the other.
- ☐ When a traffic light is tuned green, its corresponding 7-segment displays start counting down from a specific value and decrements until zero is reached.
- ☐ After this the counter starts by a new count value at the moment the yellow light turns on.
- ☐ When the red light turns on after the yellow took its time, the count continues to decrement until reaching zero.
- ☐ This means that the same 7-segments, on each traffic light, are used to display the count when cars are allowed and not allowed to pass.
- ☐ In terms of counting, the yellow and red are considered one set while the green is another set.