

# \* Process Architecture [ Unit 3,4,5,6 ]

Syllabus :-

- 3.1] **PIC Interrupts :** Interrupt v Polling , IVT , Steps in execution interrupt  
 Sources of interrupts; Enabling and disabling interrupts, Interrupt registers,  
 Priority of interrupts ,  
Programming of : Timer using interrupts, external hardware, interrupt  
 Serial communication interrupt; External hardware interrupts, Serial  
 communication interrupt; Interface of LED, Interfacing 16x2 LCD(9 bits)  
 and keys board( 4x4 matrix), Interfacing Relay & Buzzer.

## Easy Solutions

- **Steps in executing an Interrupts in PIC18F458**

→ The Following steps take place when an interrupt occurs:

- step 1 : The current instruction execution is completed.
- step 2 : The return address is pushed onto the stack.
- step 3 : The control transfers to → interrupt vector Table (IVT)  
That has Interrupt service Routine (ISR) address.
- step 4 : The ISR address is copied into the Program Counter (PC)  
and hence the control is transferred to the new location
- step 5 : The last instruction of ISR i.e. RETFIE when encountered  
the microcontroller gets back (POPs) the return address  
from the stack and hence returns to the location from  
where branching had occurred.

- Explain the interrupt structure of PIC18 micro / write a short Note

→ There are 13 registers that are used to control interrupt operation  
 Register are :-

• RCON	• INTCON	• INTCON2	• INTCON3
• PIR1, PIR2, PIR3	• PIE1, PIE2, PIE3	• IPR1, IPR2, IPR3	

- (A) **[PIR Register]** - The Peripheral Interrupt Request (PIR) registers contain the individual Flag bits for the peripheral Interrupts. Due to the number of peripheral source

there are three Peripheral Interrupt Request [Flag] registers.  
 [PIR1, PIR2, PIR3]

## (B) Interrupt control register

- There are 3 interrupt control register their structure and significance of each bit is shown in (6.1) a)

R/W-0	R/W=0	R/W=0	R/W=0	R/W-0	R/W-0	R/W-0	R/W-x
GIE/GIEH	PEIE/GIEL	TMROIE	INTOIE	RBIE	TMROIF	INTOIF	RBIF

bit7

R/W-0	R/W-0	R/W-0	R/W-0	R/W*	R/W-0	R/W	R/W
GIE/GIEH	PEIE/GIEL	TMROIE	INTOIE	RBIE	TMROIF	INTOIF	RBIF

bit 7

IPEN - Interrupt Priority  
Enable bit.

bit 7 [GIE/GIEH] : Global Interrupt Enable bit  
When IPEN (RCON<7>) = 0;

- 1 = Enable all unmasked interrupts
- 0 = Disables all Interrupts.

When IPEN (RCON<7>) = 1;

- 1 = Enables all high priority interrupt.
- 0 = Disables all priority interrupt.

bit 6 [PEIE/GIEL] : Peripheral Interrupt Enable bit

When IPEN (RCON<7>) = 0

- 1 = Enables all unmasked peripheral interrupt.
- 0 = Disables all peripheral interrupt.

When IPEN (RCON<7>) = 1

- 1 = Enables allow priority peripheral interrupt.
- 0 = Disables allow priority peripheral interrupt.

bit 5 [TMROIE] : TMRO overflow Interrupt Enable bit

- 1 = Enables the TMRO overflow interrupt
- 0 = Disables the TMRO overflow interrupt.

bit 4 [INTOIE] : INTO External Interrupt Enable bit

- 1 = Enables the INTO external interrupt
- 0 = Disables the INTO external interrupt.

bit 3 [RBIE] : RB Port change interrupt Enable bit

- 1 = Enables the RB port change interrupt
- 0 = Disables the RB port change interrupt.

bit 2 [TMROIF]: TMRO Overflow Interrupt Flag bit

1 =

bit 2 [TMROIF] : TMRO Overflow Interrupt Flag bit.

- 1) TMRO register has overflowed

- 0 = TMRO register did not overflow.

R/W-1	R/W-1	U-0	R/W-0	R/W-0	H-0	R/W-0	R/W-0
INT2IP	INT1IP	---	INT2IE	INT1IE	INT2IF	INT1IF	
bit7					bit0		

R/W-1	R/W-1	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0
INT2IP	INT1IP	---	INT2IE	INT1IE	---	INT2IF	INT1IF
bit7			INTCON3; Interrupt Control registers.	bit0			

bit 7 [INT2IP] : INT2 External Interrupt Priority bit  
 1 = High Priority  
 0 = Low Priority

bit 6 [INT1IP] : INT1 External Interrupt Priority bit  
 1 = High Priority  
 0 = Low Priority

bits 5 [Unimplemented] : Read as '0'

bit 4 [INT2IE] : INT2 External Interrupt Enable bit  
 1 = Enables the INT2 external Interrupt.  
 0 = Disables the INT2 external Interrupt

bit 3 [INT1IE] : INT1 External Interrupt Enable bit  
 1 = Enables the INT1 external Interrupt  
 0 = Disables the INT1 external Interrupt

bit 2 [unimplemented] : Read as '0'

bit 1 [INT2IF] : INT2 External Interrupt Flag bit  
 1 = The INT2 external interrupt occurred  
 0 = The INT2 external interrupt did not occur.

bit 0 [INT1IF] : INT1 External Interrupt Flag bit  
 1 = The INT1 external interrupt occurred.  
 0 = The INT1 external interrupt did not occur.

\* Differentiate between ARM & RISC.

Advanced RISC Machine

Reduced Instruction set computer

### ARM

- 1) Based on proprietary IP, and the companies license their products.
- 2) Relatively small number of Processor vendors.
- 3) ARM, being proprietary, can be export blocked by government.
- 4) ARM has teams of engineers developing hw sys. that make it easy for designers to incorporate ARM CPU's.
- 5) OP-TEE is an open source trusted execution environment (TEE) implementing the ARM Trustzone technology.
- 6) ARM effectively implementing Predicated execution without the instruction sets help.

### RISC

- 1) Royalty-free specification architecture.
  - 2) Open source RISC.
  - 3) It is open source and available for anyone with an internet connection to look up the standard and implement their own design.
  - 4) While designers can be experiment and develop RISC system for free, there is little to no support for hw design.
  - 5) RISC Multizone security sw model is small & therefore faster.
  - 6) The basic RISC-V instruction set does not support Predicated execution.
- | Parameter.        | ARM                                       | RISC                                       |
|-------------------|---|--|
| 1) Based on       | Proprietary IP                            | Royalty free & Open source                 |
| 2) Dev. hw system | team of engineer present                  | little to no support for hw system/design. |
| 3) OP-TEE         | Implementing the ARM trustzone technology | Multizone security sw                      |
| 4)                |   |  |

## Explain in brief Interrupt Vs Polling.

Interrupt	Polling
1] In interrupt, the device notices the CPU that is requires its attention.	1] whereas, in polling, CPU steadily checks whether the device needs.
2] An interrupt is not a protocol, it's a h/w mechanism ✓	2] whereas it isn't a h/w its a protocol
3] In interrupt, the device is serviced by interrupt handler	3] whereas CPU steadily polls that device at regular or proper interval
4] Interrupt can take place at any time.	4] whereas occurs at regular interval
5] In interrupt, IRL is used as indication for indicating that device requires servicing	5] while in polling command ready bit is used as indicating indication for indicating that device requires servicing.
6] In interrupt, processor is simply disturbed once any device interrupts it	6] On the opposite hand, in polling, processor waste countless processor cycles by repeatedly checking the command-ready little bit of each device. wastes lot of CPU cycles.
7] Does not waste CPU cycles	In polling, the CPU provides the service.
8] The interrupt handler executes	
9] I'	

## \* State and explain external h/w interrupt.

- It's a computer system interrupt
- 2 external interrupts.
  - [ INT0
  - [ INT1 & INT3
- Pin no. 10, 12 and 13 in PORT 9 are for the EH/WINT
- Both are Active low
- Ex. INT → inform → MICRO → that → Ext. devices need Routine service
- Memory location 0003H and 0013H in the interrupt vector table belongs to INT0 & INT1 respectively.

## \* Difference Between PIC and ARM

	PIC	ARM
1]	PIC microcontroller refers to Peripheral Interface Controller.	ARM microcontroller refers to Advanced RISC Machine
2]	It is available in 8, 16, 32 bit	It is available in 32 & 64 bit
3]	It has an effective instruction rate of 4 clock cycle per instruction	It has an effective instruction rate of clock cycle per instruction
4]	Family includes PIC16, PIC18, PIC=24	Family includes ARM17, ARM9, ARM11
5]	It has a very good community support	It has a vast community support
6]	Its manufacturer is microchip	It manufacturers are Apple, Nvidia, Samsung, Electronic an T.I.
7]	It is available with an average cost as compared to features	It is available with a low cost as compared to features.

## \* LED

→ Light Emitting diodes - basic display units in electronic world.

→ P-N Junction Diode

→ It has found its way to the low power display systems recently

→ LED's have been ruling the electronic area as no other similar technology has ever replaced them.



→ infra red wavelength, Remote control

→ 7 segment display system → popular.

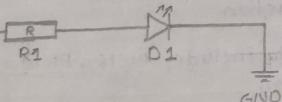
Faster

→ Small in size, easy to integrate, LEC, switching

## \* Interfacing LED.

### PIC18F4550

RB7/PDN	40
PB6/P4C	39
PB5/PGM	38
PB4/AN11	37
PB3/AN3/CCPL/VPO	36
PB2/INT2	35
PB1/AB10/INT1	34
PB0/INT0/SDA	33
VDD	32
VSS	31



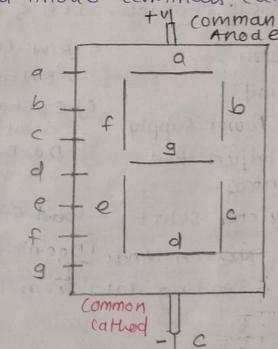
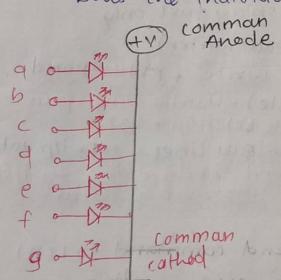
## Interfacing 7 segment display

- The 7-SD consist of 7 LED's in square fashion.
- Each of 7 LEDs called as Segment 'cause when illuminated it forms part of a numerical digit to be displayed
- An additional 8th LED is sometimes used within the same package thus allowing the indication of Dec. Pt.
- 2 or more 7-seg-dis are ~~connected~~ together Display no.

1  
1  
1

### \* The Common Cathode (CC) & Common Anode

- In CC, all the cathode connectn of the LED segments are joined together to logic "0" or ground.
- The individual segments are illuminated by application of a "HIGH", or logic "1"
- Send signal via a current limiting resistor to forward bias the individual Anode terminals (a-g)



- Send signal via a current limiting resistor to forward bias the individual Anode terminal (a-g)
- The individual segments are illuminated by application of a HIGH or logic "1"
- Initially all LED's are off
- In CC All LED segments are joined together to logic 1
- The individual segment are illuminated by applying a ground logic 0 or low
- Send signal via a suitable current limiting resistor to the cathode of the particular segment (a-g)

## \* LCD interfacing in PIC

- LCD → Liquid crystal Display, type of flat panel display
- Displays characters, numbers.
- Consumes low and inexpensive price

### Applications

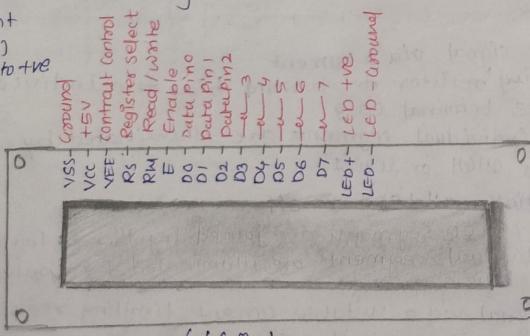
- Electronic machines
- Calculator
- Router
- Fax machine
- Laser printer

### \* Hitachi HD44780 LCD.

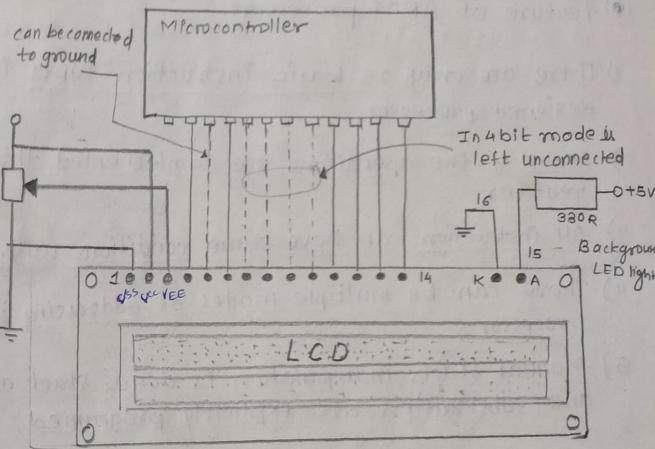
- It shows 2,8 character lines on display screen, It has 16 pins
- It is a ASCII character LCD & display text only.

### ► Pin description.

- 1) VSS : Ground
- 2) VCC : +5V Power Supply
- 3) VEE : To adjust the contrast
- 4) RS (Register Select) - serial sig  
RS = 0 : No commands (Doesn't send command to LCD)  
RS = 1 : Reading data from LCD Register & Do it
- 5) Backlight (+/LED) connected to +ve
- 6) Backlight Connect to ground.
- 7) R/W (Read/Writer)  
RW = 0 Write , RW = 1 Read.
- 8) E (Enable) - Handle time span  
From (E=0 to E=1) (High to low)
- 9) D0-D7 - Bus Lines - take i/p data & command to displaying desired message on screen.



## \* LCD Interfacing



- VCC pin connected to +5V
- VEE used to control the brightness of display (2K to 61c mod)
- The 3 control pins E, R/W & RS allows the user to let the display know what kind of data is sent
- MC send 8 bit data through D0 to D7 pin
- R/W = 0, string of data is written to the LCD &
- R/W = 1, Reading data from LCD.
- When 'E' provide time delay for few ms (10ms) before outputting next command or data
- When 'E' is high, display characters one at a time

Q Explain the features & advantages of ARM processor.

### ● Feature of ARM processor.

- 1) There are only 25 basic instruction types in this designed processor.
- 2) Most of the operations are implemented through register
- 3) All instruction can have same conditions with them.
- 4) There can be multiple modes of addressing in this processor.
- 5) Manual stacks manipulation is done. stack addressing and subroutines are explicitly programmed.
- 6) The 32 bits micro-processor made it possible to have access to memory & data manipulation saved in these 32 bits.
- 7) The address range of this processor is 26 bits wide. It allows 64 megabytes of memory to be accessed directly.
- 8) Usually there is a single cycle execution.
- 9) There is no support in this processor for misaligned memory access.

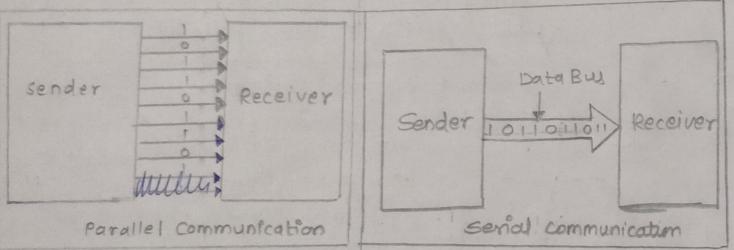
### Advantages of ARM Processor.

- Affordable to create
- Low power consumption
- Work faster
- Multiprocessing feature.
- Better battery life
- Load store architecture.
- Simple circuits.

### Advantages of ADC

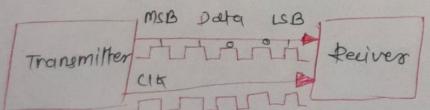
## ★ Parallel vs Serial Data transfer

Parallel Data Transfer	Serial Data Transfer
1) Group of bits (B-bits) of data is transferred at a time	1) Only one bit of data transferred at a time
2) Data transfer rate is high	2) Data transfer rate is low
3) Preferred for short distance	3) Preferred for long distance
4) Cable cost is more	4) Cable cost is less
5) Less reliable	5) More reliable
6) Ex: GPIB, PCI, ISA etc	6) Ex: RS232C, USB, HDLC etc.



## What is RS232

- RS232C "Recommended Standard 232C"
- D-type connector which of 9 pins.
- RS232 is standard protocol used for serial commn.
- It is used for connecting computer and its peripheral devices to allow serial data exchange between them.
- It is used in serial communication with the rate of 1.492 kbps



1) CD → Incoming signal from DCE

2) RD → Receive data

3) TD → Transmit data

4) DTR → Data Terminal Ready

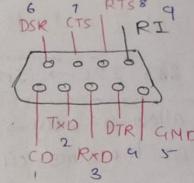
5) GND → Ground Signal

6) DSR → Data Set Ready

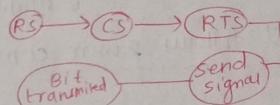
7) RTS → Request to send

8) CTS → Clear to send

9) RI → Ring Indicator



DB9 connector

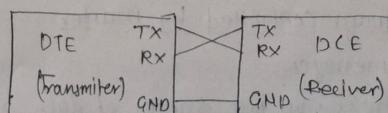
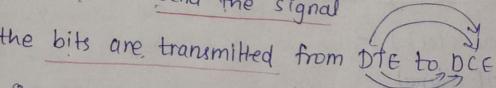


## How RS232 works?

→ RS232 works on the 2-way communication that exchanges data to one another

→ There are 2 devices connected to each other (DTE) Data Transmission Equipment & Data Communication Equipment which has the pins like TXD, RXD, RTS & CTS.

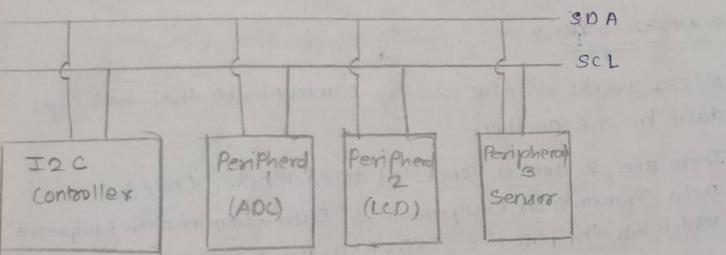
- 1) Now, from DTE source, the RTS generates the request to send Request to send
- 2) Then from the other side DCE, the CTS, clears the path for receiving the data
- 3) After clearing a path it will give some signal to (RTS) of the DTE source to send the signal
- 4) Then the bits are transmitted from DTE to DCE
- 5) This is the whole process through which data transmission takes place.



RS232 Protocol

## About I<sub>2</sub>C Protocol

- I<sub>2</sub>C → Inter Integrated Circuit → by Philips
- Allow → easy → between components which reside on the same circuit board
- You can connect multiple slave to single master  
multiple master & multiple slave
- It follows 2-wire communication & serial communication protocol



## \* I<sub>2</sub>C Terminologies

- I<sub>2</sub>C only uses two wires to transmit data between devices.

### \* SDA (Serial Data)

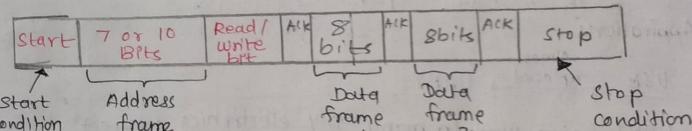
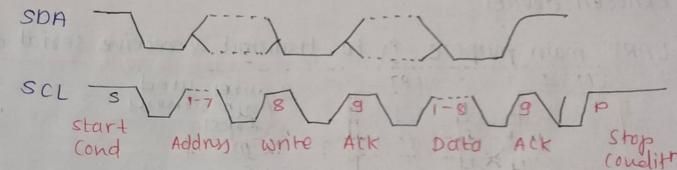
- The line for the master & slave to send & receive data
- Bit by Bit along a single wire

### \* SCL (Serial Clock)

- Carries clk signal
- Sampling of bits by a clk signal, shared between master & slave
- clock signal is always controlled by master.
- Data transfer in messages.
- messages are broken up into frames of data

## Working steps of I<sub>2</sub>C

- 1) Start condition (SDA → SCL) (High to Low)
- 2) Slave address + R/W bit
- 3) ACK bit by slave
- 4) Master → data
- 5) For every Data frame ← ACK bit by slave → or
- 6) Stop condition (SDA → SCL) (Low to High)



## Advantages :-

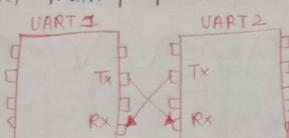
- 1) More than 1 master device can be used in the system
- 2) I<sub>2</sub>C requires only 2 wires for communication
- 3) Small less expensive PCBs, Addressing is very simple, addition of any extra devices is very easy.
- 4) diagnostic & debugging easy to perform

## Disadvantages :-

- 1) H/W complexity increases when No. of master & slave
- 2) The protocol is managed by SW stack
- 3) It is 1/2 duplex mode of communication.

VARTUniversal Asynchronous Receiver / Transmitter

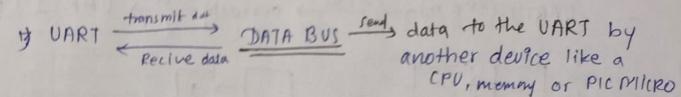
- It's not a communication protocol like SPI & I2C but physical circuit in a microcontroller or a stand-alone IC
- VART transmit data asynchronously which means there is no clock signal to synchronize the seq. of bits from the transmitting
- VART has only uses two wires to transmit data between devices
- VART main purpose is to transmit & receive serial data

Applications

- USB almost replaced by VART
- VART being used in many DIY electronic projects
- To connect GPS modules, Bluetooth modules and REID card Reader modules to your Raspberry Pi, Arduino or other microcontroller.

Working of VART

- Step 1: Data bus send data to the VART

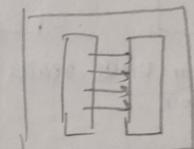


Data is transferred from the data bus to the transmitting VART in parallel form.

Micro



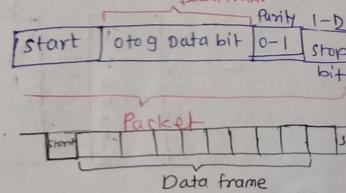
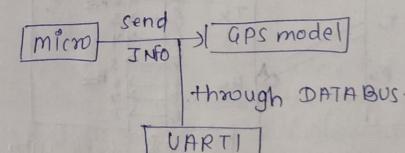
form of bit &  
Parallel format



Data bus = →

VART1 — transmitter (T) —> Receiver ←

Send info to VART1 transmitter through the data bus.



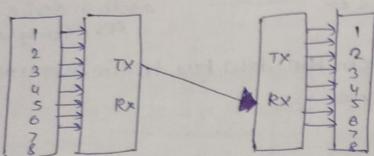
- 1) INFO → Info through Data bus in bit & Parallel format
- 2) VART1 → Info sent by micro & setup into form of packet.

- microcontroller send data to VART1 through the data bus
- VART1 receive those information & send all info into packet

- > start bit → logic 0
- > data frame → Actual info bits upto 8 bits long.
- > parity bit → even not → set 0  
odd not → set 1.
- > stop bit → Stop - Always logic 1 bit  
signal the end of data packet.

### ③ UART Transmitter send data to UART Receiver.

UART1 (tx pin) → Parallel → serially → to UART2 (rx pin)

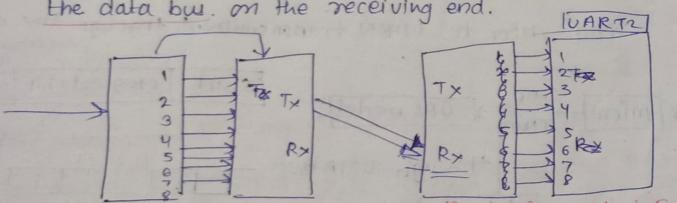


→ The receiving UART reads the data packet bit by bit at its RX pin.

### ④ UART Receiver (UART) received data & sent it parallel.

→ converts the data back into parallel form & removes the start bit, parity bit & stop bit

→ UART Receiver transfers data packet parallel to form the data bus. on the receiving end.



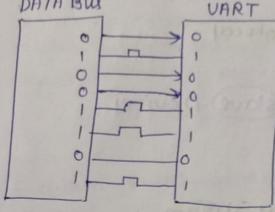
### ⑤ UART Receiver received data & send it parallel.

### ⑥ correct info send in parallel mode

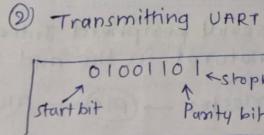
Databus send info to GPS model

## Working UART

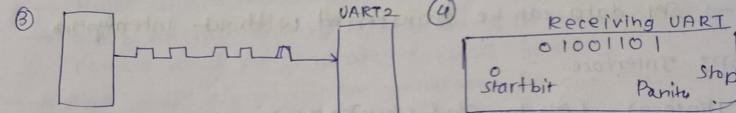
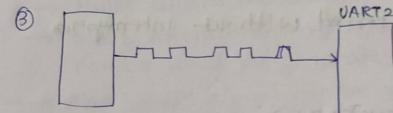
### ① DATA BUS



### ② Transmitting UART



### ③



steps → Receiving UART converts the serial data back into Parallel & transfers it to data bus. on receiving e

## Advantages

1) Only 2 wires

2) No clk

3) Has a parity bit to allow for error checking

4) The structure of data packet can be changed

5) Well documented & widely used

## Disadvantages

1) size of the data frame limited

2) Doesn't support multiple slave or multiple master system.

## SPI Protocol

→ serial peripheral Interface protocol

→ motorot in the mid 1980's

Master → (PIC) — multiple (slave) Devices

→ full duplex synchronous serial

→ send or received in a continuous stream

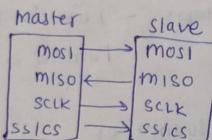
→ SPI data can be transferred without interruption SPI Interface.

1) (MOSI) - (Master O/p & slave I/p)

2) MISO (Master I/p / slave O/p)

3) SCLK (Clk)

4) SS/CS (Slave select / chip select)



## Working SPI

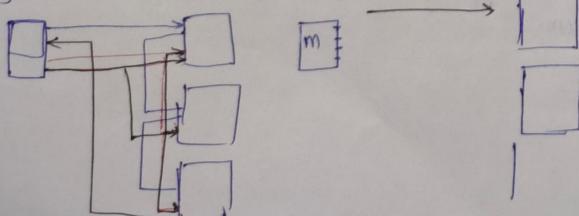
Step 1 : Provide Clk signal

Step 2 : select slave devices

Step 3 : MOSI / MISO — send data to slave , slaves give response  
step 4 :

Independent SPI method 1 master — many slaves

## Daisy chained SPI method.



## Adv

- No start & stop
- No complicated slave
- Higher data transfer
- Separate MISO & MOSI

## Disadv

- 1) Four wire
- 2) No Acknlt.
- 3) No error checking
- 4) Only allow single master

## \* Explain PWM generation.

→ Pulse width modulation

→ Most microcontrollers have a built in timer that can be used to generate a PWM signal

→ The PWM signal can be generated in our PIC microcontroller by using the CCP (Compare Capture PWM mode) module

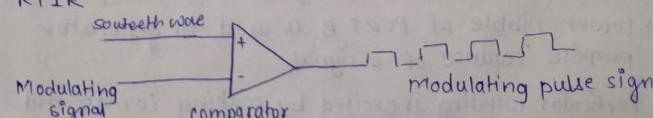
→ The resolution of our PWM signal is 10 bit that is for a value of 0 there will be a duty cycle of 0%. for value of 1024 ( $2^{10}$ ) there be a duty cycle of 100%

→ The PWM model works by setting the voltage level of the digital pin, incrementing the timer preset pulse width. & toggling the pin for the remaining cycle

→ PWM is a technique that alters a continuous digital signal into series of pulses.

→ commonly used as a way to generate an analog signal that correlates to a digital value.

GPTI RTIR

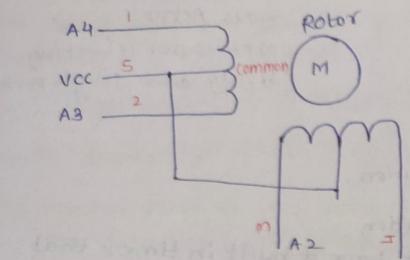


Used in TCM for encoding

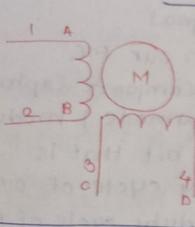
Computer motherboard requires PWM signals that controls the head generated in the board.

Also used in audio/video amplifier.

# Stepper motor interfacing with PIC

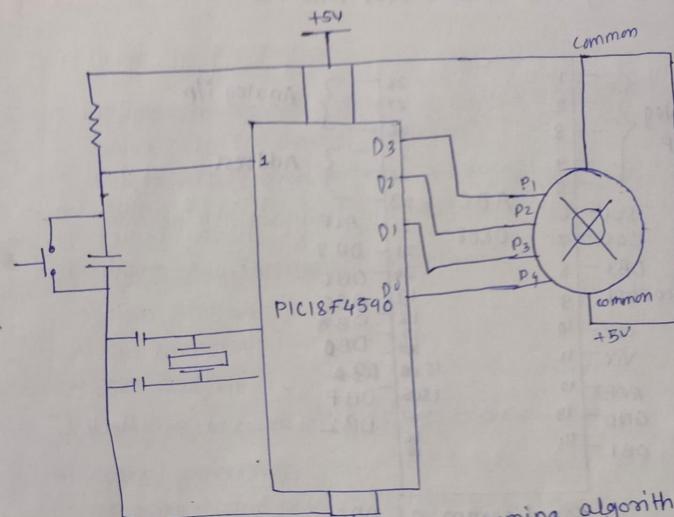


A1 = Phase 1  
A2 = Phase 2  
A3 = Phase 3  
A4 = Phase 4

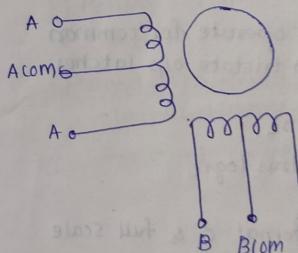


→ Stepper motor are working on the same principle of DC motor.

- It is a digital motor.
- Driven by digital signal.
- Motor has 2 phases & center tap winding.
- Center tap with winding are connected to 12V supply.
- By this motor can be excited by grounding the terminal of 2 winding.
- Motor can be rotated by excitation sequence.
- Lower nibble of PORT B is used to generate purpose sequence of  $\ominus$  signal.
- Particular winding is excited by making corresponding PORT Pin Low.

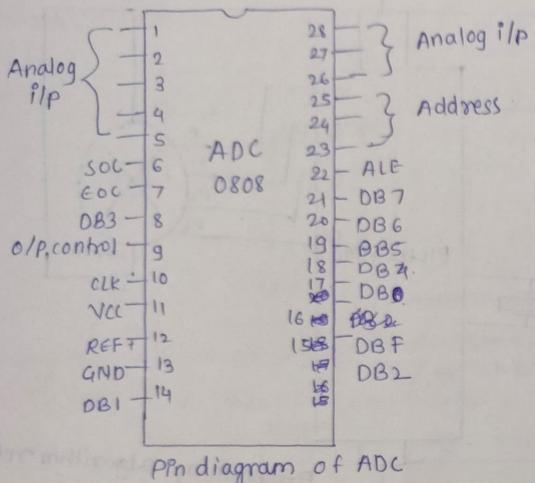


AND Programming algorithm rotation clockwise



- 1) Initialize the port pins used for the motor as O/p.
- 2) Write a common delay program of say 500 ms.
- 3) Output first sequence 0x05 of 8 on the pins.
- 4) Call delay function.
- 5) O/p 3rd seqn 0x06 on the pins.
- 6) Call delay function.
- 7) O/p 4th seqn 0x09 on the pins.
- 8) Call delay function.
- 9) O/p 1st seqn 0x05 on the pins.
- 10) Call delay function.
- 11) Goto step 3

# Interfacing ADC & DAC 0808 with PIC



- ADC 0808 & ADC0809 are monolithic CMOS device with an 8-channel Multiplexer.
- These devices are also designed to operate for common microprocessor / microcontroller with tristate o/p latches.
- Features are
  - 8-bit successive approximation ADC
  - 8-channel multiplexer with address logic
  - Conversion time 10.0  $\mu$ s.
  - It eliminates the needs for external o & full scale adjustment
  - Easy to interface to all micro controller
  - It operates on single 5V power supply
  - O/p meet TTL voltage level specification

## Steps to Programming ADC.

### 1) Configure the A/D module.

- Configure analog pins , voltage & digital I/o [ADCON1]
- Select A/D P/p channel [ADCON0]
- Select A/D conversion clk [ADCON0]
- Turn on A/D module [ADCON0]

### 2) Configure A/D interrupt [if desired].

- clear ADIF bit
- set ADIE bit
- Set GIE bit.

### 3) Wait the required acquisition time.

### 4) Start conversion.

Set Go / Done bit [ADCON0]

### 5) Wait for A/D conversion to complete, by either

- Polling for the GO/DONE bit to be cleared
- Waiting for A/D interrupt

### 6) Read A/D Result register [ADRESH/ADRESL]

clear bit ADIF if required

For next conversation goto step 1

CPSR	SPSR
- current Program status Register	saved Program status register
- This stores additional information about the state of the processor	This stores a copy of the previous CPSR value when an exception occurs.
-	

- 1) Explain in brief centralized and client server architecture.
- 2) Describe 2 Tier and 3 Tier architecture
- 3) State and explain properties of transaction management.
- 4) Write short note on mobile database & cloud
- 5) Explain Time stamping method
- 6) Explain shadow paging
- 7) Write short note on log-based recovery
- 8) Write short note on purpose of normalization
- 9) Explain the process of normalization
- 10) Write short note on selection and join operation