Embedded System (Unit 1)

Page No.
Date

444

1) What is embedded system?

=>

- it is a combination of hardware and software designed for a specific function

- Embedded systems can be programmable or have a fixed functionality.

- industrial machines, agricultural devices, cameras, toys, vending machines are possible locations of embedded system

- embedded systems can vary between UI like having no UI to having complex graphical User Interface.

> embedded systems are a part of a complete device hence they are called embedded

- are lost cost
- are low power consuming

- embedded system has parts such as processors, power supply, memory & communication ports

- e.g. Digital camera, MP3 player, calculator, Projectors, ATM, Washing maching

These ports are used to transmit data from/to processor to/from perpherial

Page No.
Date

(2) Explan the Structure of embedded System :-

=>

Embedded system consist of three main elements

(1) Hardware :-

- The hardware of embedded system is based around microprocessors and microcontrollers.
- hardware is basically the physical components of a device
- Microprocessors refers to CPU integrated with memory and peripherial devices
- Microcontroller have those on a single chip

(2) Software & Frimware :-

- Software is a set of programs or instructions which controls the functioning of the hardware
- Frimware is a interface between the hardware and software, it provides communicating between them acting like a bridge

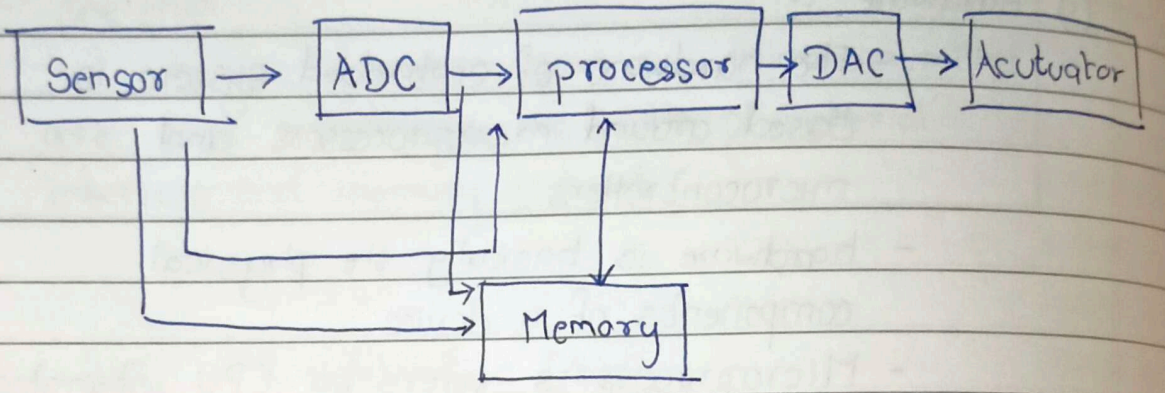
(3) Real time Operating System :-

- This is an optional part of embedded system as it is not always included
- it defines how system works by supervising the software & setting rules during program execution

Actuators - takes analog signal and gives output in the physical form

Page No.
Date

* Embedded System structure diagram :-



Sensor :- detects changes in the environment and generates electrical signals as per the change

ADC (Analog to Digital Convertor) :- it converts analog electrical signals into digital signal

Processor :- processor is used for processing the signal and data as per the set of instructions.

DAC (Digital to Analog Convertor) :- It converts digital signals back into analog signal for further processing

Actuators :- it is a component of a machine that is responsible for moving and controlling the mechasism of the system
it compares input & output signals to to provide error free processing

Important ***

Page No.
Date

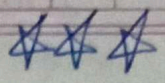
* Difference Between General purpose computer system and embedded system.

=>

General purpose Computer system	Embedded Systems
① combination of hardware & software integrated together to perform various functionalities	① for a specific functions
② needs human interaction to perform tasks	② does not need / needs comparitively less human interaction
③ 2 parts Hardware & Software	③ 3 parts Hardware, Software & Frimware
④ expensive	④ less expensive or we can say cheaper
⑤ peripheral devices such as keybard, mouse	⑤ peripheral devices such as USB, SD cards
⑥ Needs more power	⑥ needs lesser power
⑦ Difficult to used	⑦ Easier to use
⑧ not time specific	⑧ are time specific
⑨ Are complex	⑨ less complex
⑩ Bigger in size	⑩ Smaller in size

Difference between Microprocessor and Microcontroller

Page No.
Date



Microprocessor	Microcontroller
- peripherial are connected external	- peripherial are available on the single chip (RAM, ROM, memory, Timers)
- cost of system is High	- Cost of system is less
- We can connect memory externall but the speed is less	- It has an inbuilt memory which helps improve the speed
- Bulky circuit	- compact system
- Total power consumption is high	- less power consumption
- not good for devices which run of batteries	- Can be used for devices which run of Batteries
- complex & expensive	- simple & inexpensive

Difference Between RISC & CISC

RISC	CISC
- Reduced instruction set architecture	- Complex instruction set Architecture
- Focus on software	- Focus on hardware
- Uses only Hardwired control unit	- Uses both hardwired and microprogramed control unit
- Fixed size instructions	- variable size instructions
- More no. of registers	- less no. of registers
- large code	- small code
- Instructions fit in one word	- larger than size of one word

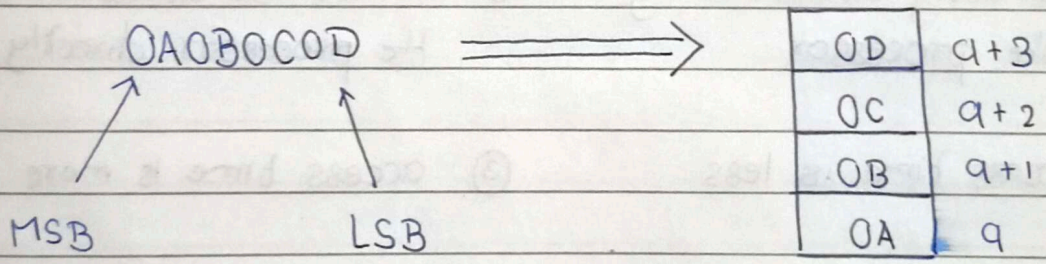
* Big endian and Little endian :-

- Endianness refers to the order in which a sequnce of bytes are stored in a memory

- Processor can have 2 types of Endianness
  ① Little Endianness
  ② Big Endianness

- Big endian :- in this the most significant bits (MSB) are stored at the lowest storage address. And the lowest significant bits (LSB) are stored at the higher storage address

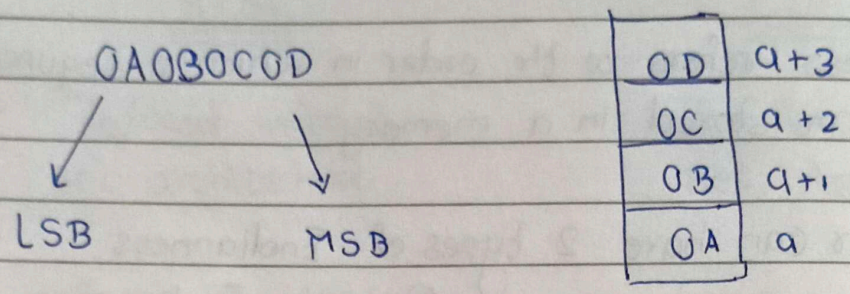
Memory representation of Big endian :-



- Little Endian :- in this the least significant bit (LSB) is stored at the lowers torage address and the ~~lowest significant~~ most significant bit is stored at the higher storage address

Page No.
Date

Memory representation of little endian :-



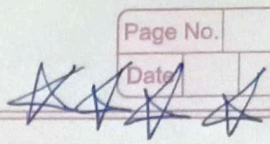
Difference between primary & secondary

Primary	Secondary
① also called as Main memory or Internal memory	① also called as Auxilary memory or external memory
② Directly accessible by the processor	② cannot be accessed by the processor directly
③ access time is less	③ access time is more
④ Less storage capacity	④ more storage capacity
⑤ expensive	⑤ less expensive
⑥ faster	⑥ slower
⑦ e.g. RAM, ROM	⑦ e.g hard disk

Difference between RAM & ROM

RAM	ROM
① Random access memory	① Read only memory
② read & write	② reading
③ Volatile memory	③ non-volatile memory
④ Temperory memory	④ Permanent memory
⑤ Higher Storage capacity	⑤ lower storage capacity
⑥ SRAM, DRAM	⑥ EEPROM

PIC microcontroller



- PIC stands for 'programmable interface controller'

- PIC microcontroller consist of RAM, ROM, CPU, timers and counters

- PIC provides low power consumption, high performance ability and easy of availability of its supporting hardware and software

- PIC is the world smallest microcontroller

- it can carry hoge range of tasks

- can be found in many devices such as alarms, embedded system

Features of PIC

- Ram
- Flash memory
- Time / counters
- EEPROM
- I/O ports
- CCP
- ADC

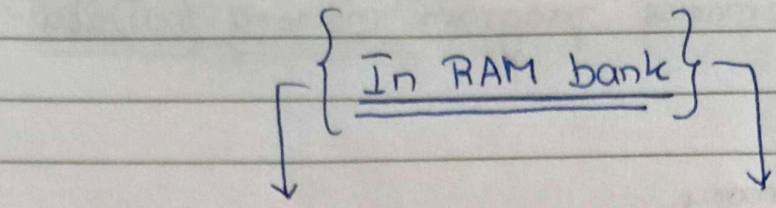
PIC Memory Structure

Page No.
Date

PIC has 2 memories program memory & Data memory

① Program memory
- 4/8 k
- stores 13 bit instruction code
- This memory is accessed by program counter register which holds the address of program memory.

② Data memory
- 368 bytes of RAM
- 256 bytes of EEPROM
- This RAM consist of multiple banks, each bank consist of general purpose registers and special function registers
- These special function registers consist control registers, ADC, I/O, Timers
- General purpose registers consit of registers that are used to store temporary data
- there are 4 general purpose registers
  1) Working register
  2) Status register
  3) File selection register
  4) Program counter register



general purpose Register

- Working
- Status
- File selection
- Program counter

Special function Registers

1. Timers
2. ADC
3. Serial ports
5. I/O

Working Register :- it stores operands for each instruction, it also stores the results of each execution

Status Register :- denotes the status of ALU (Arithmetic logic unit) after every execution

File Selection Register :- it acts as a pointer to any other general purpose register

Program counter :- 13 bit register
it acts as a pointer to the instructions stored in the program memory.

EEPROM :- - 256 bytes of memory
- permanent memory
- the contents can be changed and erased

—X—

* I/O ports :- 5 ports
Port A, port B, port C, port D & port E

Port A - 16 bit, input/output based on the status of TRISA register

Port B - 8 bit, input/output, 4 bits can be used as interrupts

Port C - 8 bit, input/output based on status of TRISC

Port D - 8 bit, acts as a slave port for connection to bus

Port E - 3 bit port, serves the additional function of the control signals to the A/D converter

—X—

* Timers :- 3 timers
T0 & T2 are 8 bit
T1 is 16 bit
↓
can be used as counter

* A/D converter :- 10 bit
8 channels
ADCON1 & ADCON2

* Oscillators :- used for timing generation
clock frequency 30KHz to 4MHz

* CCP module :-
① Capture mode - it captures the time of arrival of signal
Captures the value of TI when CCP goes high

② Compare mode - generates output when the T1 value reaches a certain reference value

③ PWM - Pulse width module control the functionality / output value by varying the width of pulses

Page No.
Date

* Serial Communication :-

↳ method of transfering one bit at a time

① USART - Universal synchronous and Asynchronous Receiver & transmitter
- used for transmitting and receiving data bit by bit over a single wire
- two pins TXD , RXD

SPI has more Data rate

② SPI Protocol - Serial peripheral interface
- Used to send data between PIC microcontroller & other pheripherals.

③ I2C protocol - Inter Integrated Circuit
- used to connect low speed devices

* ES: Unit 3 *

* What is Arduino?

Arduino is an open source and user friendly platform for creating electronic projects.
- it has a microcontroller and software for writing and running code.
  - It is easy to use
  - less expensive
  - cross platform
  - simple & clear
  - open Source

—X—

Arduino programs are saved with ".ino" file extension.

—X—

* Structure of Arduino Program :-

=>

Consist of 2 main functions

① setup() : - initialization section
- runs when arduino is powered or reset
- here we set up variables, configure pins & perform tasks required to prepare the board

② loop() :
- main program loop of our code
- continuously executes the code within the loop

—X—

* **Digital Input/Output Functions :-**

These functions allows us to read & write digital values to and from pins of Arduino.

① pinMode (pin , mode)

- pin → represents pin number
- mode → Defines I/O mode of the pin (e.g. INPUT or OUTPUT)

e.g. :- pinMode (9 , INPUT) ;

↳ set digital pin 9 as Input pin

② digitalRead (pin)

This function is used to read the digital value from a specific pin

```
int val = digitalRead (2);
```

↳ reads the value at pin 2 & stores in variable "val"

③ digitalWrite (pin, value)

↳ This function is used to write on a specific pin
The value attribute ranges from 0v to 5v

e.g. :- digitalWrite (2, LOW);

—X—

Program :- Sets pin 13 to the same value as pin 7

```
int val = 0;

void setup() {

    pinMode (13, OUTPUT);
    pinMode (7, INPUT);
}

void loop() {

    val = digitalRead (7);
    digitalWrite (13, val);
}
```

Page No.
Date

* Analog Functions *

① analogRead (pin)

↳ this function reads the analog value from a specific pin which ranges from 0 to 1023, which represents voltage from 0 to 5 volt

```
int val = analogRead (A0);
```

② analogWrite (pin, value)

↳ This function writes on a specific pin the value attributes ranges from 0 to 255

```
analogWrite (9, 127);
```

Note :-

analogRead values goes from 0 to 1023 and analogWrite values goes from 0 to 255 hence it is necessary to divide the read value by 4 while writing it

Program :- Set the output to Led proportional to the value read from potentiometer

```
int ledPin = 9 ;
int analogPin = 3 ;
int val = 0;

Void setup () {
                pinMode (ledPin, OUTPUT);
              }

Void loop() {

      val = analogRead (analogPin);
      analogWrite (ledPin , val/4);
}
```

Page No.
Date

* Timer functions *

delay (ms) : Pauses the program execution for a specified number of miliseconds

delay Microseconds (us) : Pauses the program execution for the specified number of microseconds.

program :-

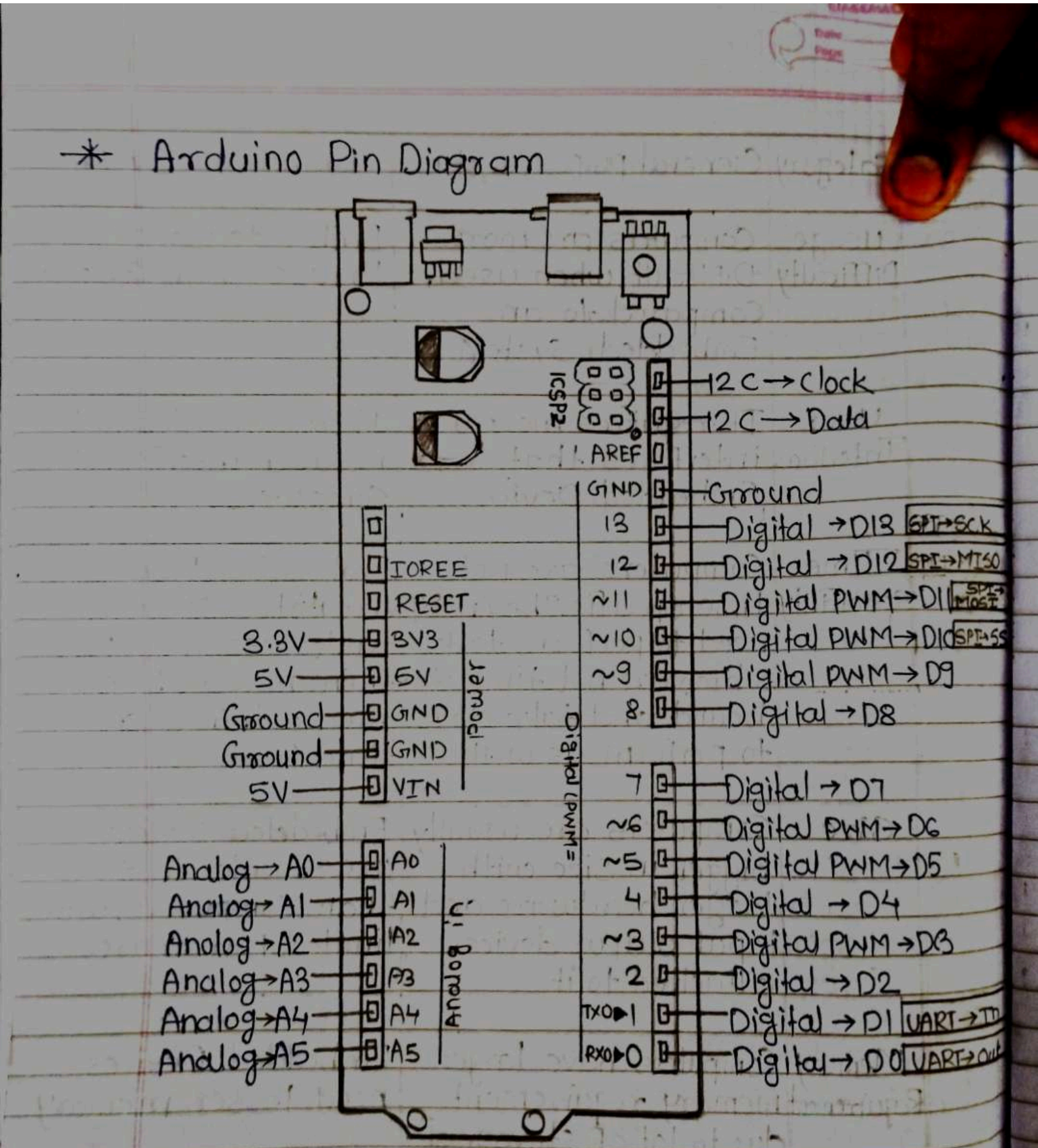
```
int ledPin = 13 ;

void setup () {
        pinMode (ledPin, OUTPUT);
      }

void loop () {
        digitalWrite (ledPin , HIGH);
        delay (1000);
        digitalWrite (ledPin , LOW);
        delay (1000);
    }
```

—X—

millis() :- Returns the number of milliseconds passed since the Arduino board started running this value will reset after approx 50 days

micros() :- Return microseconds —11— resets after 70 minutes

* Arduino Pin Diagram



VCC	Ground	Digital I/O	Analog I/O
3.3V	GND	Digital 13	Analog 0
5V	GND	Digital 12	Analog 1
5V	GND	Digital PWM 11	Analog 2
		Digital PWM 10	Analog 3
		Digital PWM 9	Analog 4
		Digital 8	Analog 5
		Digital 7	
		Digital PWM 6	
		Digital PWM 5	
		Digital 4	
		Digital PWM 3	
		Digital 2	
		Digital 1	
		Digital 0	