

→ What is IOT?

large no: of nodes which are interconnected and are talking to each other.

IOT has gained momentum because of lower cost
Smaller area.

Bandwidth → efficiency per unit time.

Latency → Round trip time

Moore's law : No: of transistors doubles every 18 months.

Varied applications in multiple fields.

Cloud computing : computing done outside that particular node.

- | |
|----------------------|
| A - application |
| R - Real time |
| M - Microcontrollers |

- Difference between microcontrollers & Microprocessors.

Microcontroller

- Slower
- Cannot compile code on itself.
- Memory limitation
- Has a general process Input/Output system.
- Can run only single code at a time.

Microprocessor

- Faster.
- Can have an OS built on it.
- Memory is large.
- Doesn't.
- Can run multiple codes

Total Registers :- 16.

R13 : SP

R14 : lr

R15 : PC

* Half word : 2 bytes.

Full word : 4 bytes.

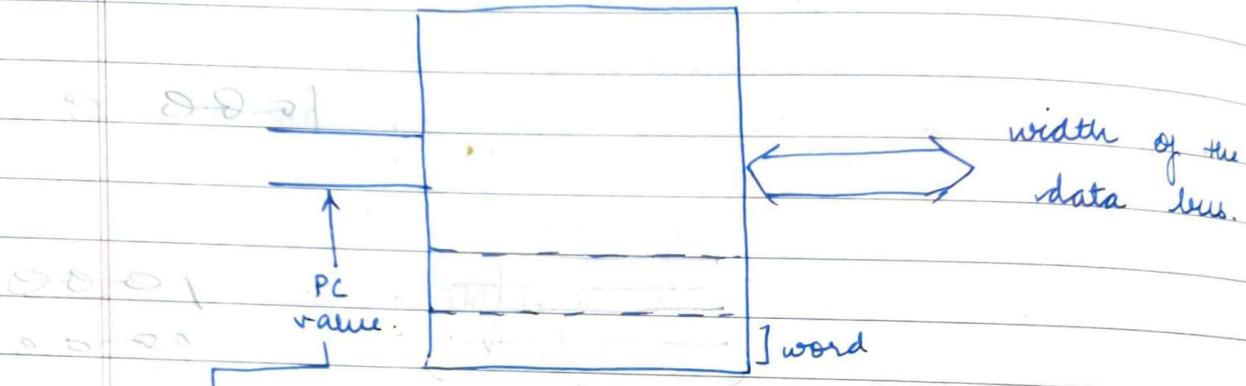
Different ISAs :- Thumb → 16 bit.

Thumb - 2 ↗ combination .

ARM. → 32 bit

→ Byte addressability.

→ Addressability: granularity of data accessible.



→ Contains ~~was~~ address of the next instruction to be executed. (precisely "fetched") .

A 4-bit address bus can access (store) 16 bytes of address locations

∴ 16-bit instructions

⇒ 8 instructions can be accessed.



The PC cannot be odd since in that case we would be accessing $\frac{1}{2}$ of instruction $n-1$ & $\frac{1}{2}$ of instruction n

• Eg: of exceptions :-

→ Segmentation fault.

→ Misaligned instruction fetch.

→ Internal interruption.

MSP

On resetting, PC jumps to 0x00000000.

→ Vector table:

It is like a look-up table that contains instructions to jump to a start-up routine.
It also contains system fault locations.

3 special registers:-

- PC: Program counter. Contains address of the next instruction to be executed.
- SP: Stack pointer. Points to the top of the stack at any instant (R13)
- Link Register (LR): It is used to return an address from a subroutine. (R14)

i.e: if a branch is encountered :-

$LR = PC$. $PC = \text{address of the branch}$.

→ STACK POINTER

MSP : Main Stack Pointer → Privileged mode

PSP : Process stack pointer. → user

Whenever an exception occurs, MSP is used
else it is the PSP is used.

At any given time, it is only in either
privilege or user mode.

Using a MUX, the mode can be toggled.

On resetting, (an exception): MSP activated
0x00000000 contains the address where
the MSP should begin from

It is only in the start reset routine that the 0x0..0 address is accessed.

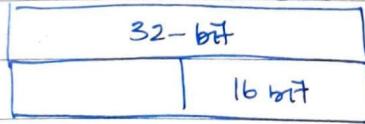
- Cortex M-4 supports both endianess.
Q: what is the endianess of Nucleo F41RE
- ISA could be:
32-bits - ARM model

Code size decreases by only 30% when 16-bit is considered.

This is because, ∵ code length is reduced, no. of instructions to do the same job increases.

Performance reduces by 20% because more decoding is req.

To efficiently decode a mixed model of 32 & 16 bit (Thumb 2) homogeneity is ensured while encoding [Eg: last bit rep. whether you're in 32-bit / 16-bit].



● Simple GCD code :-

while ($a \neq b$) {

 if ($a > b$) $a = a - b$

 if ($b > a$) $b = b - a$

}

Step 1: load values from memory to registers
 (use LDR or LDW - multiple loading
 multiple reg. from a section of memory)

Step 2: compare CMP
 (This will update & set flags.)

Order: op1

mnemonic	op 1 ↑ to where result is to be stored.	op 2, ... ↓ operands
----------	---	----------------------------

Suffixes can be attached which rep. conditions.
 so:

LDR r_0 r_1

LDR r_0 [r]

LDR r_1 [r]

loop COMP r_0 , r_1

~~SUB~~ ~~cccccceeee, etc~~

BZ ~~ccccc~~; (Branch on zero: exit).

SUB GT r_0 r_0, r_1 .

SUB LT r_1 , r_1, r_0 .

B loop

Flags can be updated during an ~~all~~ instructions

X — X



Architecture



Processor



Implementation



Processors are designed in accordance
 with a part. arch.



Implementation of certain
 processors is processor bound.

- Application PSR Stores the values of the flags
- TPSR stores interrupt no:
- Check what EPSR stand for.

- The LR increments as PC+1 ::, the instruction length could be either 2 or 4 bytes, ... it by 3 so as to decide whether after instruction execution what 'type' of inst. it is.
- Step into assembly functions to check what the required registers are filled with the assigned values.
Some C/C++ statements may not have an assembly code. Eg: char b;
- Assembly lang. codes have fast performance.

When a function call is encountered, it places the return address in the LR
Arguments passed to a function are passed as r0, r1, r2, r3

If a function returns a value it is passed through r0.

Difference b/w assemblies on different microprocessors is the mnemonics.

- Data flow inst :-
ADR, LDR, STR, MOV
ADD, CMP, SUBS.
- Branch control.
IT, BL, BX.

LSB of PC is always 0

PC points to next inst. being 'fetched' classmate

Date _____
Page _____

PC values will always increment by 4.

But because of a branch / function call, the address fetched may not be aligned

∴ last 2 bits can be either 0 or 10 / 00.

PUSH is essentially an STR ... instruction.

But PUSH address is implicit.

PUSH {r3 - r7}, LR.

POP {r3 - r7}, PC.



→ popped addresses like so.

STR r0 [r2].

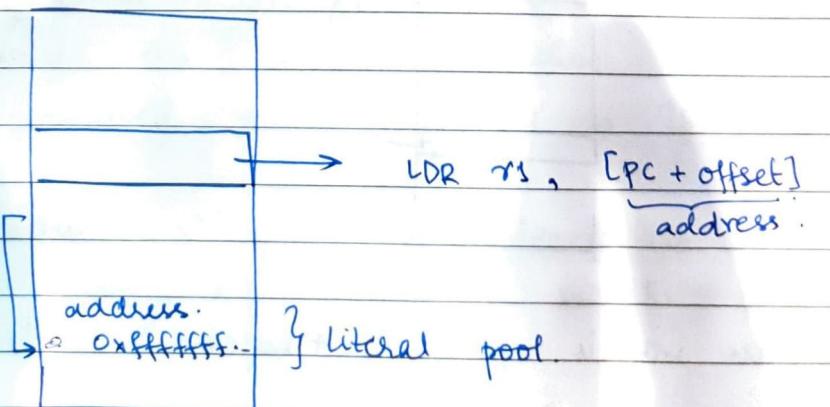
Storing the value in r0 in address of r2.

Store writes to memory

Load reads from memory.

STR r0 [sp!] ! → increment after
↓ inc before

- Storing a 32 bit no: in a register.



- IT-block.

Next 4 instructions will be conditionally executed.

ITTT

ADD $\begin{pmatrix} E0 \\ \dots \end{pmatrix}$ → truth

- BIT-BAND OP:-

To set a particular bit

LDR $r_0, [r_1]$

R

→ OR $r_0, r_0, 0x40$ W

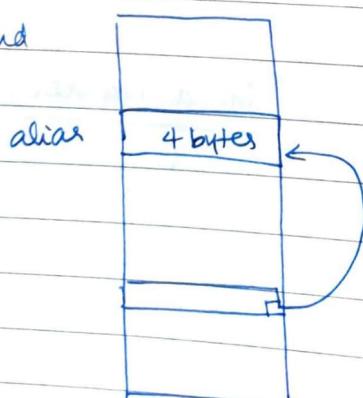
STR $r_0, [r_1]$

M.

Could create an issue for eg: if an ISR occurs at → hazard!

This can happen !! it isn't in its finest granularity.

Bitband



each bit is given 4 bytes.

Eq: $0x2000 \dots \underline{\underline{00}}$

0th bit at this add = add
 1st bit at 11 11 → add + 4

$0x2000 \dots 001$

Address :- $0x2000 \dots 20$

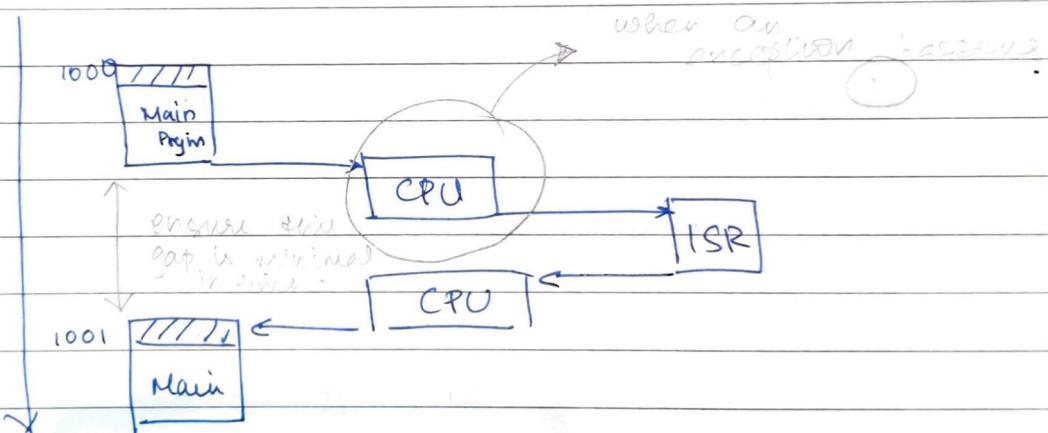
\uparrow
32 bytes. bytes

Benefits :-

- Faster bit operations
- Fewer instructions
- Avoids Hazards.



INTERRUPTS:



Variables in main prog. remain unchanged
However some global var. whose values may be required after changes in ISR, changes.

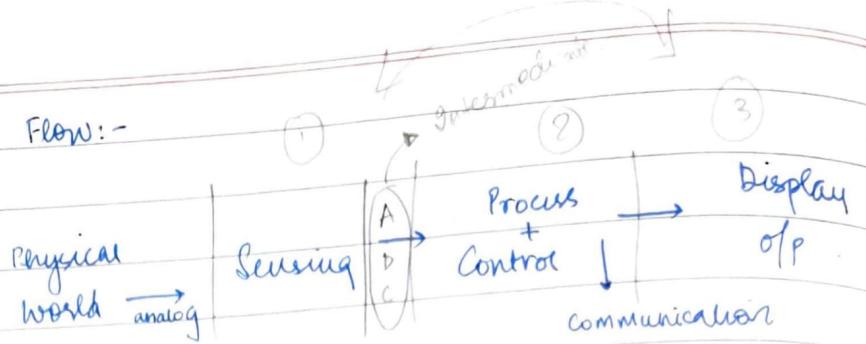
If an exception is encountered in between an instruction execution, it waits for the full insl. to be executed.

→ The LR is pushed in the ISR, hence it is always done in a MSP mode.

Address

1	handler	MSP:
9	MSP	-
13	PSP	PSP

→ Flow:-



- Assembly ... faster than C-code
∴ the assembly lang is a concise manner of what the GP compiler does to the C-code

Critical comp - Time, Area, Power

- Pull-up/down are used so that there isn't just a floating value for a particular pin

Pull-up : deterministic voltage - 5V
pull-down " " → GND

GPIO registers lie in the peripheral memory-map

gpio_set :- sets the value of the pin

gpio_set_mode : sets I/P or O/P

~~either~~ gpio_set_mode :- 5 modes.

- switch with pull-up res.

```
gpio_set_mode (PIN1, output);
```

```
gpio_set_mode (PIN2, output);
```

```
gpio_set_mode (switch, PULLUP);
```

```
write(1) ;
```

```
if (gpio_get(switch))
```

```
? gpio_set (PIN1, 0); gpio_set (PIN2, 1);
```

else

?

A waveform is sampled at a constant rate

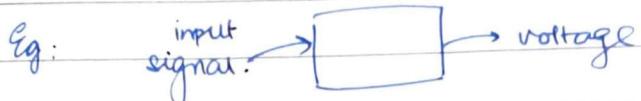
CLASSMATE

Date _____

Page _____

For an ADC:- $V_{out} = V_{ref} \frac{n}{2^N}$

Mapping input function (transfer func.)



Sampling freq:-

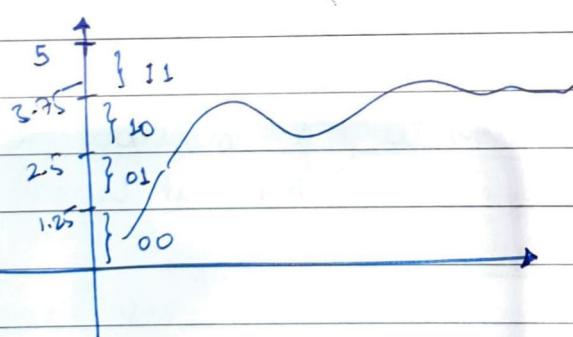
How many times we break/split the continuous wave to obtain a discrete value at each instance.

Any continuous wave can be represented as a summation of waves with different freq.

Acc. to Nyquist's sampling theorem:-

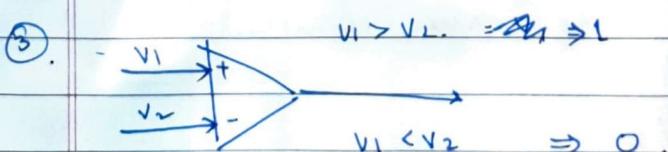
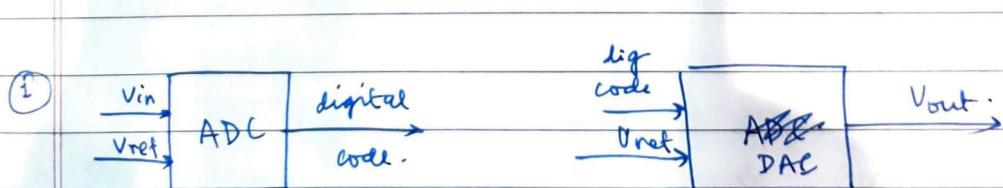
ideal sampling freq: $f_s \geq 2 f_{max}$.

In an analog to digital converter, N-bit quantization is mapped.



2 bit ADC.

$$\frac{5}{4} = 1.25$$



} use binary search at each stage.

Set each bit

$$n = \left\lfloor \frac{V_{in} \times 2^n + \frac{1}{2}}{V_{ref}} \right\rfloor \longrightarrow \text{Analog to digital code}$$

Performance Metrics:-

- Linearity of transition voltages.
- Differential linearity measured. (Equal step size)
- Conversion time + conversion rate

Comparing Flash ADC & Successive Approx.

- Flash is faster.
- ② is slower because for each clock cycle, B.S. do but for ① it is only the gate delays are taken into account.
- SAC is better for linearity & diff. linearity.
- ② takes finite time

→ Measuring transition voltage (Proximity sensor):
 Use a potentiometer & measure the pot drop & resistance across the SR sensor & vary the potentiometer value.
 Take equal intervals & note down the transition voltage from the ADC obtained

Nyquist th. \longrightarrow actual world 3/4 times

.. we sample more than required to avoid brick-roll off.

but this "cost"

The output of sampling goes to the ADC

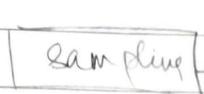
remember
which will
turn out

classmate

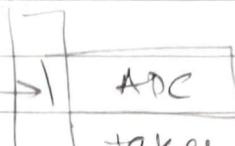
Date _____

Page _____

Problem:



continuous

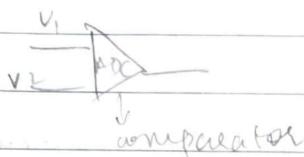


takes time

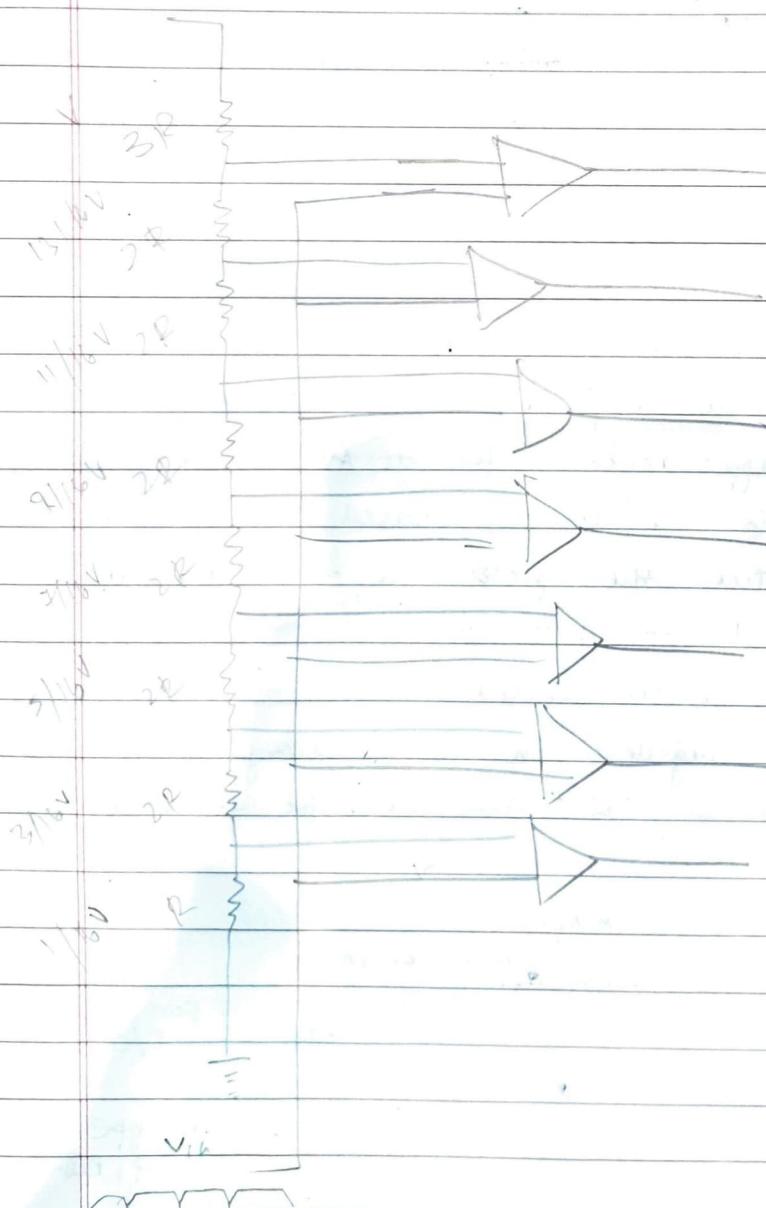
for each sample sequence

* Single ended ADC - read

Noise imm :-

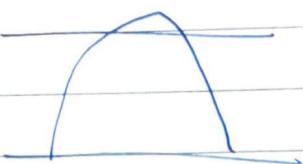


noise caged noise

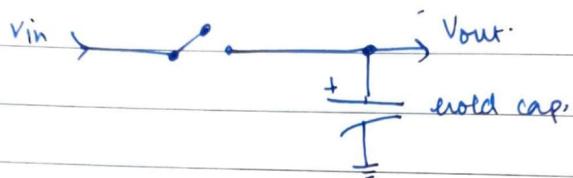


• Clamping:-

If V_{range} of sensor is $\rightarrow V_{range}$ of the ADC then it may so happen that there may be an overdrive of current. (Hence it is clamped).



- To ensure that the input analog value remains constant (doesn't change) use a sample-hold circuit.



Charging & Discharging takes time.

- In a SAC:- in the software there is a status register in the peripheral memory mapped region for the ADC & checks if the LSS is set or not. Based on this it decides whether the process is over or no.

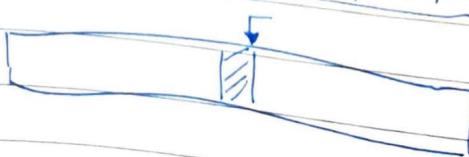
For pins to be set/cleared.

There exist registers in the memory (peripheral bit-band memory) assigned for the gpio pins.

Eg: gpios peripheral mem. map.



For mode I/O.



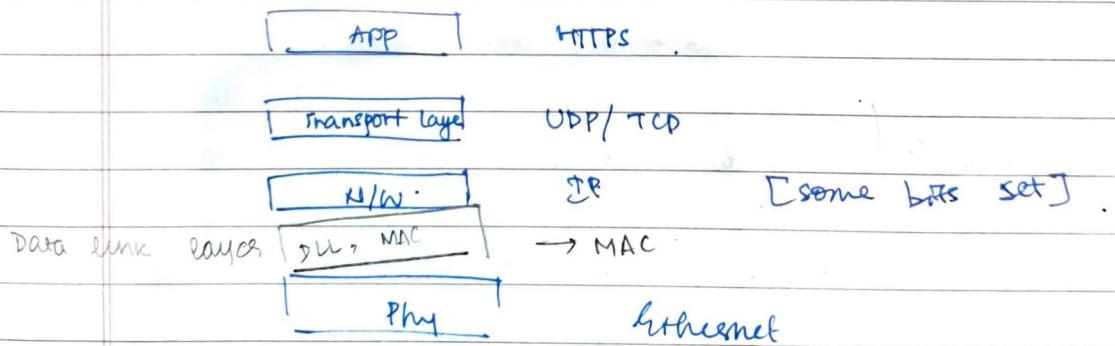
For type pull-up/pull-down

- Similarly for a counter too where the counter is set to the delay value & decrements, then the program status reg values are checked if 00000 & an interrupt happens.
- x ————— x

Digital to analog Conv:-

- square wave can be generated.
- For a smooth curve wave a built in sin function lookup table.
- A PWM can also be used to create an analog wave at a pin :: on an avg. the values generate. It also depends on the granularity of times to [decide the smallest PWM possible].

- ~~TOPIC~~ Internet Conn: (Networking Stack)



IP address varies depending on what network

Connections:- Eg:-

VPN	no proxy.	→	NOT the same IP.
VPN	Proxy	→	§ IT IP.

Creates a virtual private network. Proxy morphes the IP address to take IIT address. Proxy fakes

Zigbee \Rightarrow Protocol for nodes to communicate amongst one another

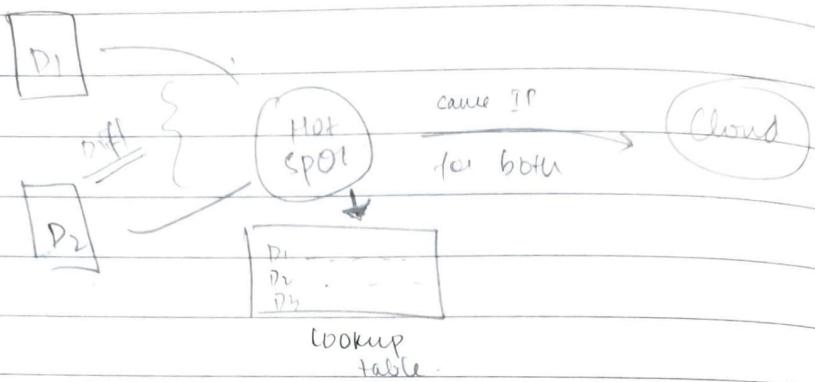
CLASSmate
Date _____
Page _____

impersonates the network head.

If config works at the shell level while IP is assigned at the browser level. [so layer dependency]

Q.

Will 2 devices connected to the same hotspot have the same IP?

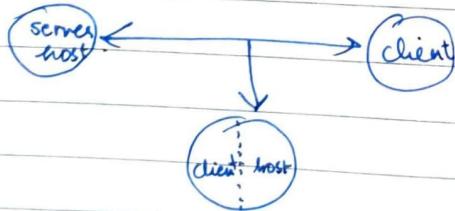


Try it & config:- should give different answers.

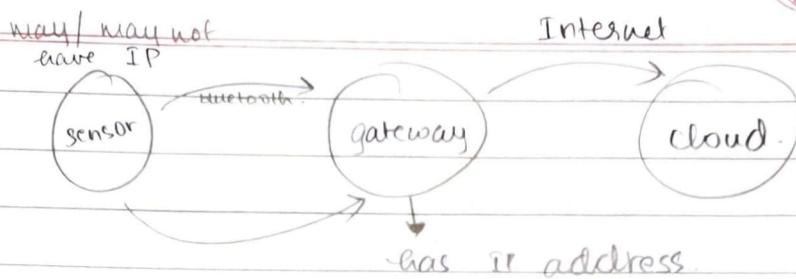
→ Fibre do not connect to the internet rather they are connected to the mobile phone via a gateway & intern the mobile connects to the internet.

Reason being:- More power, cost, area.

Zigbee Man-in-the-middle :-



- Improvement from IPv4 \rightarrow IPv6 end-to-end data encryption.
- 128 bit IP address.



NIC : Network interface card.

The tuple of the IP & the NIC must be unique.

Different topologies :

Point-to-point



sensor to

Star (Bluetooth)



gateway

Mesh

→ Bluetooth :

Radio : device that can transmit & receive signals at particular frequencies. (NFS) → radio

[1 sim ⇒ 1 radio 2 sim ⇒ 2 radio]

Radios for communication & internet usage are typically the same [depends on whether it's LTE or GSM].

→ Application data transfer rate < over the air rate
..., at each layer of the stack the layers add their own information.

* BLE : at every stage tries to go to sleep.

Minimum transfer length is determined by the protocol.

Packet length is the size of the payload you can transfer

Bluetooth low energy band is a free band allocated for ISM. There will be interferences.

CLASSMATE
Date _____
Page _____

Application transfer rate :- how much space is occupied by the application layer.
Only a small fraction is the actual data.

Over the air: uses the phy & DLL.

Everytime bluetooth, the phy layer information has a faster data transfer rate but the application layer information has a slower transfer rate.

Classic protocol

Headphones

BLE.

smaller payload



Fitbit

Mobility support both BLE & classic.

X — X

Advertising → unidirectional

- Discovery of advertising happens on the 802.15.4 gateway.
- Connect happens only b/w the 2 concerned devices only one device can decipher.
- Broadcast : send information

40 channels so that you have some guard channels.

X — X

- Standardize One N to M : problematic

Standardization occurs at the transport layer. In the networking stack, it is here that the one N to M layer is.

- One-MICOM → Global standardization.
- It is not independent of the network

'uri' used in IoT instead of 'url'

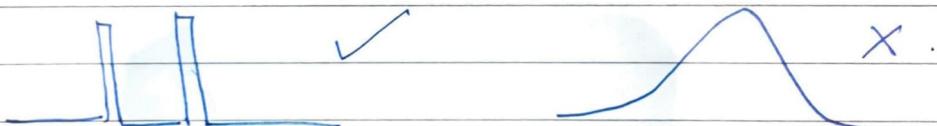
X. ————— X

→ BLE:

GFSK: Gaussian Frequency Shift Key.

Different layers of the protocol stack communicate via UART, serial interface & could be on different boards/chip

- BLE works in a race to idle condition.
- For efficiency the switching from sleep to awake time should be extremely quick.



In BLE esp

GATT :- server :- BLE + nucleo board } Step 2.
client :- phone.

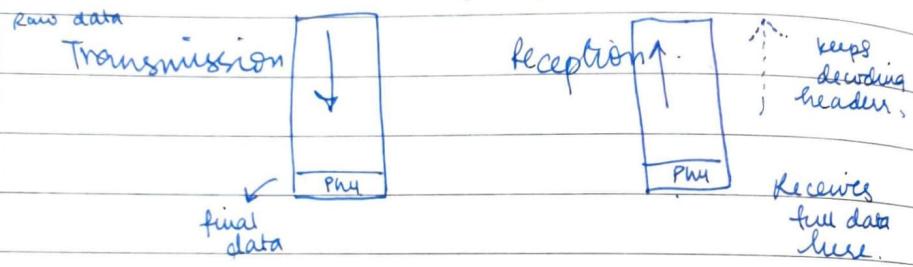
GAP :- ^{Slave} Peripherals : advertiser } Step 1.
^{Central} : phone
Master.

From Bluetooth 4.0 onwards ~~the~~ a peripheral can connect to multiple centrals & vice-versa.

- Guard frequency : margin of error, Band range
- Parity bit: detecting loss of information depending on the nos. of 1's & 0's error detection.

Error detection must happen at the data link layer. This is because, cost of failure will be more if detected in later stages.

Entire networking stack could be implemented on hardware thus giving more speed.



- A particular user application is linked to a specific QATI VID

- Both client & server need to be programmed to be compatible.

Roofing

 $V_{DD} - V_{IO}$ I_{LED}

→ Star
→ Mesh
→ P2P

Network
Topology



new
MAC
bit
banding

Communication

→ BLE
→ Zigbee

Sensors

IOT

Computer/MC
ICE

Internet + clouds

GPIO

Interrupts.

Analog interfacing

→ ADC
→ comparators
→ DAC

Serial communication

- UART
- SPI
- PWM
- I²C

