TECHNIQUES FOR WRITING THE EMBEDDED CODE.

MEMORY MANAGEMENT

FAQ

1. Discuss the limitations of memory in embedded devices. How is it managed? Explain.

- The rapid development of the Internet of Things has created a number of exciting new opportunities and challenges for designers.
- Yet whether an IoT device is brand-new technology specifically designed to be connected, or an upgrade to an existing device to create more capability, there is one consideration that cannot be overlooked: The need for the device to have memory.
- And while the number of memory options hasn't quite reached the number of IoT devices on the market, when you are trying to make a decision about what to include, it can certainly feel that way.
- From traditional RAM-based and flash memory to more advanced, chip-based memory solutions, there are plenty of options to choose from.
- Limitation of memory in embedded devices are:
  - **Cost:** Cost is a concern in any project; the more expensive the memory selection, the more expensive the final device. Depending on the market, you need to weigh the cost vs. performance options.
  - **Size:** Most IoT devices are small, and thus the embedded technology must also be small. The amount of space required for memory processing must also be kept to a minimum, as the more silicon wafer space required, the more costs go up.
  - **Power Consumption:** Most IoT devices either run on small batteries or rely on energy harvesting for recharging. For this reason, it's important to consider the power consumption of the memory selection, and choose an option that uses the least amount of power and voltage, both in use and during standby.
  - **Startup time:** Users want excellent device performance, so memory needs to be sufficient to allow for a quick startup.
- Implementing a code-in-place option, which allows the device to execute code directly without needing to copy operating code from a separate EEPROM chip reduces the time required to boot up, as well as the cost of the chip since there is less need for RAM with substantial on-chip storage.
- One of the major aspects in any computing device is memory, whenever the system runs out of memory OS is responsible to active special reserved memory known as virtual memory.
- The memory management is simpler in a computer system, getting a warning message of low memory space, OS will active Virtual Memory.
- But on embedded environment with no screen and OS similar situation become difficult to handle.

TYPES OF MEMORY

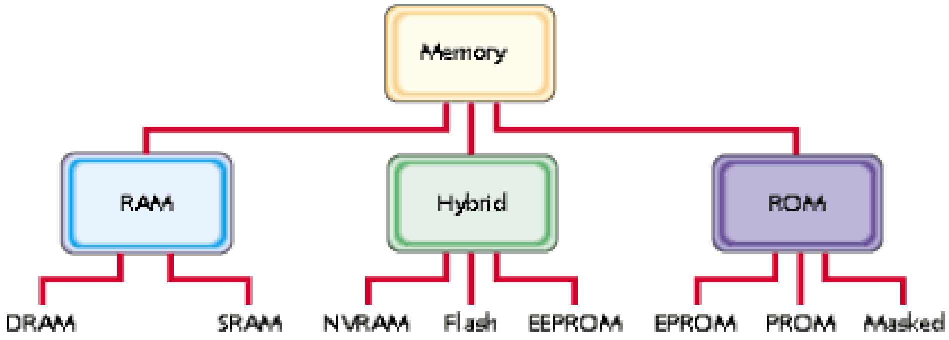
FAQ

1. Explain different types of memory.

MEMORY

- Memory is the electronic holding place for the instructions and data that a computer needs to reach quickly.
- It is where information is stored for immediate use.

- Memory is one of the basic functions of an embedded system because, without it, an embedded system would not be able to function properly.
- Memory is also used by a computer's operating system, hardware, and software.
- Memory system in embedded systems can be divided into 3 categories:
  - RAM
  - ROM
  - Hybrid



Random Access Memory (RAM)

- Random Access Memory (RAM) is one of the faster types of main memory accessed directly by the CPU.
- It is the hardware in a computer device to temporarily store data, programs, or program results.
- It is a read/write memory.
- It is volatile, which means if a power failure occurs or the computer is turned off, the information stored in RAM will be lost.
- All data stored in computer memory can be read or accessed randomly at any time.
- There are two types of RAM:
  - SRAM
  - DRAM

SRAM

- It stands for Static Random-Access Memory.
- It is made up of flip-flops.
- Flip-flop is a storage device capable of storing a bit.
- It is faster than DRAM
- It is used as a cache memory due to its high speed.

DRAM

- It stands for Dynamic Random-Access Memory.
- In DRAM, each cell carries one-bit information.
- The cell is made up of two parts: a capacitor and a transistor.
- The size of the capacitor and the transistor is so small, requiring millions of them to store on a single chip.
- Hence, a DRAM chip can hold more data than an SRAM chip of the same size.
- However, the capacitor needs to be continuously refreshed to retain information.
- It requires continuous refreshment to retain the data.
- It is slower than SRAM

READ ONLY MEMORY

- ROM is a memory device or storage medium that is used to permanently store information inside a chip.
- It is a read-only memory that can only read stored information, data, or programs, but we cannot write or modify anything.

- A ROM contains some important instructions or program data that are required to start or boot a computer.
- It is a non-volatile memory; it means that the stored information cannot be lost even when the power is turned off or the system is shut down.
- The types of ROM are:
  - PROM
  - EPROM
  - Masked ROM

PROM

- It stands for Programmable Read Only Memory.
- It is a type of digital read-only memory, in which the user can write any type of information or program only once.
- It means it is an empty PROM chip in which the user can write the desired content or program only once using the special PROM programmer or PROM burner device.
- After that, the data or instruction cannot be changed or erased.

EPROM

- It stands for Erasable and Programmable Read Only Memory.
- It is the type of read only memory in which stored data can be erased and reprogrammed.
- In EPROM, if we want to erase any stored data and reprogram it, first we need to pass the ultraviolet light for 40 minutes to erase the data.
- Then the new data can be stored in EPROM
- It is a non-volatile memory chip that holds data when there is no power supply and can store data for a minimum of 10 to 20 years.

MASKED ROM

- The very first ROMs were hardwired devices that contained a pre-programmed set of data or instructions.
- The contents of the ROM had to be specified before chip production, so the actual data could be used to arrange the transistors inside the chip.
- Hardwired memories are still used, though they are now called "masked ROMs" to distinguish them from other types of ROM.
- The primary advantage of a masked ROM is its low production cost.
- Unfortunately, the cost is low only when large quantities of the same ROM are required.

Hybrid

- Hybrid memories can be read and written as desired, like RAM, but maintain their contents without electrical power, just like ROM.
- These are typically used to store code.

NVRAM

- An NVRAM is usually just an SRAM with a battery backup.
- When the power is turned on, the NVRAM operates just like any other SRAM.
- When the power is turned off, the NVRAM draws just enough power from the battery to retain its data. NVRAM is fairly common in embedded systems.
- However, it is expensive--even more expensive than SRAM, because of the battery--so its applications are typically limited to the storage of a few hundred bytes of system-critical information that can't be stored in any better way.

FLASH

- Flash memory devices are high density, low cost, non-volatile, fast (to read, but not to write), and electrically reprogrammable.
- These advantages are overwhelming and, as a direct result, the use of flash memory has increased dramatically in embedded systems.
- From a software viewpoint, flash and EEPROM technologies are very similar.
- The major difference is that flash devices can only be erased one sector at a time, not byte-by-byte.
- Typical sector sizes are in the range 256 bytes to 16KB. Despite this disadvantage, flash is much more popular than EEPROM and is rapidly displacing many of the ROM devices as well.

EEPROM

- It stands for Electrically Erasable and Programmable Read Only Memory.
- The stored data can be erased using a high voltage electrical charge and can be re-programmed.
- It takes around 10 milliseconds to erase the data.
- It is a non-volatile memory whose data cannot be erased or lost; even the power is turned off.

MAKING THE MOST OF YOUR RAM

FAQ

How can we make optimum use of RAM while writing code for embedded devices?

- RAM provides the working memory for the system.
- If you have more RAM, you may be able to do more things or have more flexibility over your choice of coding algorithm.
- With respect to RAM every embedded system is unique, so the optimization techniques you apply will depend on the specific requirements and constraints.
- Some points while writing code on embedded devices are as follows :-
    - Choose the right data types as it will consume a bit less amount of memory and shows your data and presents it much accurately.
    - Analyze your tools in order to monitor your RAM's usage and identify the areas for improvement.
    - Reducing the code's size which directly helps in RAM's optimization, i.e. smaller the code the lesser resources it requires to run.
    - Avoid recursion as it uses RAM and uses it without any reason and becomes a reason for slow running of the RAM being used.
    - Methods like memory pooling, and efficient data structures are employed to make the best use of the limited resources.
    - Code for embedded devices should be written with portability in mind, allowing it to be easily adapted to different hardware platforms or configurations.

RAM ORGANIZATION: STACK AND HEAP

FAQ

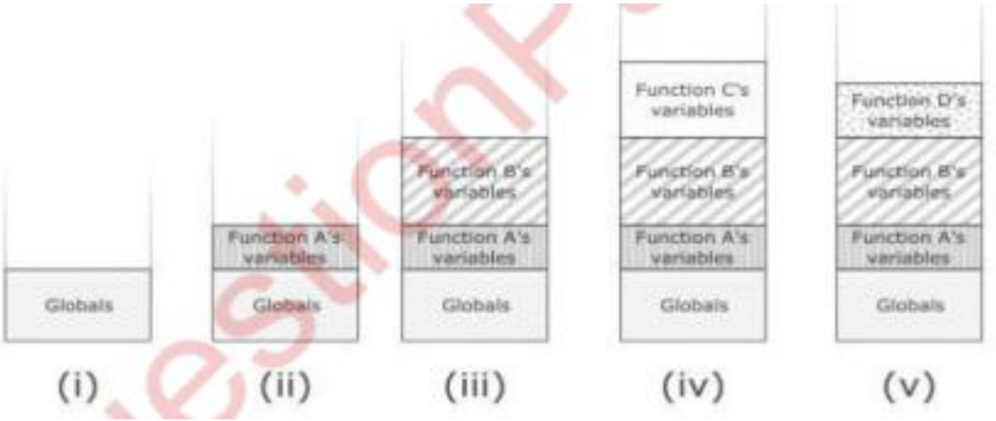
1. With the help of examples, compare stack and heap.

Stack:

- New items which are added to the stack go to the top, and the items can be removed only in strict reverse order, so the first thing to be removed is the last item that was placed to the stack
- This arrangement makes it easy for the processor to keep track of where things are and how much space is being used because it has to track only the top of the stack.

- The stack is only useful for Items that aren't going to survive for long periods of time and Items that remain in constant use, from the beginning to the end of the program.
- As all the variables within a function are available only to code inside it, when you reach the end of that function, all those parameters and variables are ready to be discarded.
- So, the stack gets unwound back to the same size it was just before control passed to the function.
- Example

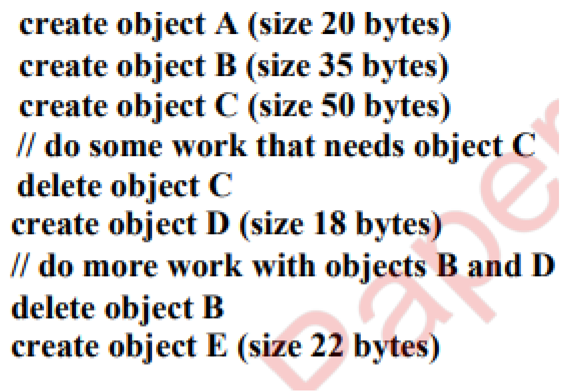
```
//Global Variables
function A
{
variable A1;
variable A2;
call B();
}
function B
{
variable B1;
variable B2;
variable B3;
call C();
call D();
}
function C
{
variable C1;
variable C2;

}
function D
{
variable D1;
}
call A();
```



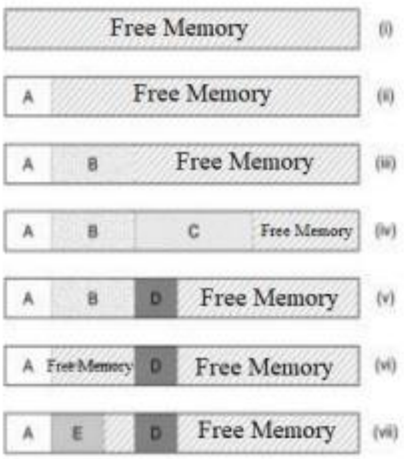
Heap:

- Heap allows the allocation of chunks of memory at any time.
- The heap is a bit like the seating area of a train where you fill up the seats strictly from the front and have to keep everyone who is travelling as a group in consecutive seats.
- To begin, all the seats are empty.
- As groups of people arrive, you direct them to the next available block of seats.

- Example:



```
 create object A (size 20 bytes)
 create object B (size 35 bytes)
 create object C (size 50 bytes)
 // do some work that needs object C
 delete object C
create object D (size 18 bytes)
// do more work with objects B and D
delete object B
create object E (size 22 bytes)
```



- At the start of execution, the heap will be empty (i).
- Object A is added to the heap (ii), taking up 20 bytes of space.
- Object B is added to the heap (iii), consuming a further 35 bytes straight after the space for object A.
- Object C is added to the heap (iv), adding 50 bytes to the heap right after object B.
- Object C is no longer needed and is deleted, releasing the space it consumed on the heap and taking us back to heap (iii).
- Object D is created and takes up 18 bytes of the space just vacated by object C (v).
- Now object B is finished with and deleted. As other code might be relying on the position of object D, we can't move it, so there's now a free space between objects A and D (vi).
- Object E is created. It requires 22 bytes of space, which means it will fit in the hole left by object B (vii)

Stack	**Heap**
Stack provides static memory allocation, i.e., it is used to store the temporary variables.	Heap provides dynamic memory allocation. By default, all the global variables are stored in the heap.
It is a linear data structure means that elements are stored in the linear manner, i.e., one data after another.	It is hierarchical data structure means that the elements are stored in the form of tree.
It is used to access the local variables.	It is used to access the global variables by default.
The size of the stack memory is limited which is dependent on the OS.	The size of the memory is not limited.
As it is a linear data structure, so data is stored in the contiguous blocks.	As it is hierarchical data structure, so elements are stored in the random manner.
In stack, the allocation and deallocation are automatically managed.	In heap, the memory is manually managed.

The implementation of stack can be done in three forms using array, linked list and dynamic memory.	The implementation of heap can be done in two forms using arrays and trees.
The main issue that occurs with a stack is the shortage of memory because the memory size cannot be changed at the runtime. The size of the stack is determined at the compile time.	The main issue that occurs with a heap is the memory fragmentation. Here, memory fragmentation means that the memory gets wasted.
It is of fixed size.	It is flexible to use as the size of the heap can vary as per our needs.
The access time in stack is faster.	The access time in heap is slower.
The size of the stack memory is decided by the operating system.	The size of the heap memory is decided by the programmers.
The scope of the variable cannot be changed.	The scope of the variable can be changed.

PERFORMANCE AND BATTERY LIFE
FAQ

1. What are the concerns regarding performance and battery life while writing code for embedded systems?

- When it comes to writing code, performance and battery life tend to go hand in hand— what is good for one is usually good for the other.
- A device which is tethered to one place and powered by an AC adaptor plugged into the wall isn't as reliant on energy conservation.
- Similarly, if you're building something which doesn't have to react instantly— maybe an ambient notifier for a weather forecast, which doesn't have any ill effect if it updates a few seconds later—or if it doesn't have an interactive user interface which needs to respond promptly to the user's actions, maximizing performance might not be of much concern.
- For items which run from a battery or which are powered by a solar cell, and those which need to react instantaneously when the user pushes a button, it makes sense to pay some attention to performance or power consumption.
- A lot of the biggest power-consumption gains come from the hardware design.
- In particular, if your device can turn off modules of the system when they're not in use or put the entire processor into a low-power sleep mode when the code is finished or waiting for something to happen, you have already made a quick win.

Code Efficiency

- One of the easiest ways to make your code more efficient is to move to an event-driven model rather than polling for changes.
- The reason for this is to allow your device to sit in a low power state for longer and leap into action when required, instead of having to regularly do busywork to check whether things have changed and it has real work to do.

Hardware Design Efficiency

- On the hardware side, use processor features such as comparators or hardware interrupts to wake up the processor and invoke the processing code only when the relevant sensor conditions are met.
- If your code needs to pause for a given amount of time to allow some effect to occur before continuing, use calls which allow the processor to sleep rather than wait in a busy-loop.

Reduce amount of Data being processed

- If you can reduce the amount of data that you're processing, that helps too.
- The service API that you're talking to might have options which reduce how much information it sends you.
- When you're downloading tweets from a certain account on Twitter, for example, you can ask for only tweets after a specified ID.
- The first time you call the API, you have to deal with all the tweets it sends, but on subsequent calls, you can ask just for tweets since the ID of the most recent one that you already processed.
- The shim service has all the processing power and storage available to a web server and so can do most of the heavy lifting.
- After this, it just sends the minimum amount of data across to be processed in your embedded system.

LIBRARIES
FAQ

1. Write a short note on libraries for embedded systems.
2. What are Libraries? Explain with examples.

- When developing software for a server or desktop machine, you are accustomed to having a huge array of possible libraries and frameworks to make your life easier.
- To make software development easier we can make use of built in libraries given to us by the programming languages.
- Here are few libraries that are available:
  - **lwIP**: LightWeight IP, is a full TCP/IP stack which runs in low-resource conditions. It requires only tens of kilobytes of RAM and around 40KB of ROM/flash.
  - **uIP:** It can run on systems with only a couple of kilobytes of RAM.
  - **UClibc:** It is a similar version of the standard GNU C library (glibc) targeted at embedded Linux systems.
  - **Atomthreads:** It is a lightweight real-time scheduler for embedded systems, used for multitasking.
  - **BusyBox:** It is not really a library, it is only a collection of a host of UNIX utilities into a single, small executable and a common and useful package to provide a simple shell environment and commands on your system.

DEBUGGING
FAQ

1. Explain in detail the process of debugging the code for embedded devices.
2. What is debugging an Internet of Things Device? Explain.

- Debugging code for embedded devices is a crucial process to identify and resolve software issues and errors in the embedded system.
- Debugging in the embedded context can be more challenging compared to traditional software due to limited resources, real-time constraints, and the lack of a standard user interface.

- Here's a step-by-step explanation of the process of debugging code for embedded devices:
- Step 1: Set Up a Debugging Environment :
  - To begin debugging, you need a development environment specifically tailored for embedded systems.
  - This environment typically includes an Integrated Development Environment (IDE) that supports debugging features

- Step 2: Use Logging and Print Statements :
  - Embedded devices might not have a display screen or a standard user interface, making traditional debugging challenging.
  - A common practice in embedded systems is to use logging and print statements to output diagnostic messages to a serial port or a debugging interface.
  - These messages provide insight into the program flow, variable values, and important events, helping developers identify potential issues.

- Step 3: Use Debugging Tools :
  - Modern debugging tools designed for embedded systems offer valuable features to aid the debugging process.
  - These tools include Hardware Debuggers/Emulators Real - Time Debugging/Emulators and Code Profilers

- Step 4: Reproduce the Issue :
  - To effectively debug, it's crucial to reproduce the issue or error consistently.
  - This might involve setting specific conditions or using test scenarios that trigger the problem reliably.
  - Reproducibility is essential for isolating the cause of the issue.

- Step 5: Analyze the Code :
  - Once the issue is reproduced, developers use the debugging tools to step through the code, inspect variable values, and examine program flow.
  - By analyzing the code execution in the context of the specific issue, developers can identify where the problem occurs and understand why it happens.

- Step 6: Fix the Issue :
  - With a clear understanding of the problem, developers can now fix the issue.
  - This might involve modifying the code, adjusting configurations, or changing hardware settings.

- Step 7: Test the Fix
  - After making the changes, it's essential to thoroughly test the fix to ensure it resolves the issue without introducing new problems.
  - Debugging embedded code requires a combination of skills, tools, and attention to detail.
  - It involves a systematic approach to identify and fix issues to ensure the embedded device operates reliably and as intended.

BUSINESS MODELS

FAQ

1. Define Business Model. Explain different factors in the definition.
2. What is a business model? Who is the business for? Explain.

- Business model is defined as a “hypothesis about what customers want, how they want it, and how an enterprise can organize to best meet those needs, get paid for doing so, and make a profit”.

- This definition brings together a number of factors:
  - A group of people (customers)
  - The needs of those customers
  - A thing that your business can do to meet those needs
  - Organizational practices that help to achieve this goal—and to be able to carry on doing so, sustainably
  - A success criterion, such as making a profit
- The model is also useful if you want to get other people involved.
- This could be an employee or a business partner or an investor.
- In each of these cases, the other parties will want to know that the business has potential, has been thought out, and is likely to survive and perhaps even go places.
- With a new business startup, you have no track record of success to point to.

HISTORY OF BUSINESS MODELS

SPACE AND TIME

- While neighbouring tribes might have discovered variants in the local area's resources it is when trade develops with others from far-off lands that it becomes really interesting.
- A merchant might sell silks made in his village to a region where these cloths are rare and in demand in exchange for aromatic spices which will be highly prized back home.
- But long-distance trade brings with it a whole set of problems:
  - merchants have to carry larger quantities of goods for sale
  - Their goods and food carried will have to last far longer, so they will need to be protected and preserved.
  - They need to have a reliable means of transport for themselves and their merchandise
  - Preservation is also a way of transporting goods through *time*.
- A farmer or trader who can afford to not sell all his produce during the harvest can fetch a better price months later at a higher price.
  - Money, then, abstracted trade further, setting an easy-to-calculate exchange rate between a fixed currency (a certain size disc of gold or weight of grain).
  - Ease of calculation which this development brought with it made it easier to develop new business models, such as the development of interest on loans.

FROM CRAFT TO MASS PRODUCTION

- When Gutenberg demonstrated his printing press circa 1450, books changed from being priceless treasures, hand-crafted by monks and artisans, to a commodity that could be produced.
- The invention laid the foundations for an information culture.
- Information is no longer so rare and valuable that it must be preserved by gatekeepers but can be so widely spread that everyone can have access to it.
- As the printing press spread to the New World and India via the sea routes that would be discovered by the end of the century.
- The cost of printing would become ever smaller as the technology spread, leading to new business models with the rise of newspapers and pamphlets.
- In other areas, the ethic of mass production resulted in new business models such as supermarkets, which pioneered both "self-service shopping" and the sale of a whole range of products under one roof.
- Fast-food franchising began in the 1930s and exploded with McDonald's and Burger King in the 1950s.

- Standardized menus, pre-prepared ingredients, and standard practices for each franchisee to follow meant that you could now eat
- Exactly the same meal in any of a chain restaurant's stores in your country.

LONG TAIL OF THE INTERNET

FAQ

1. Write note on Long tail of internet

- From Tim Berners-Lee's first demonstration of the World Wide Web in 1990, it took only five years for eBay and Amazon to open up shop and emerge another five years later as not only survivors but victors of the dot-com bubble. Both companies changed the way we buy and sell things.
- A physical bricks shop has to pay rent and maintain inventory, all of which takes valuable space in the shop; therefore, it concentrates on providing what will sell to the customers who frequent it: the most popular goods.
- In comparison, an Internet storefront exposes only bits, which are effectively free. Of course, Amazon has to maintain warehouses and stock, but these can be much more efficiently managed than a public-facing shop.
- Long tail Internet giants help this process by aggregating products from smaller providers, as with Amazon Marketplace or eBay's sellers.
- This helps thousands of small third-party traders exist, but also makes money for the aggregator, who don't have to handle the inventory or delivery at all, having outsourced it to the long tail.
- Yet although Google's stated goal is "to organize the world's information and make it universally accessible and useful", it makes money primarily through exploiting the long tail of advertising, making it easy for small producers to advertise effectively alongside giant corporations.

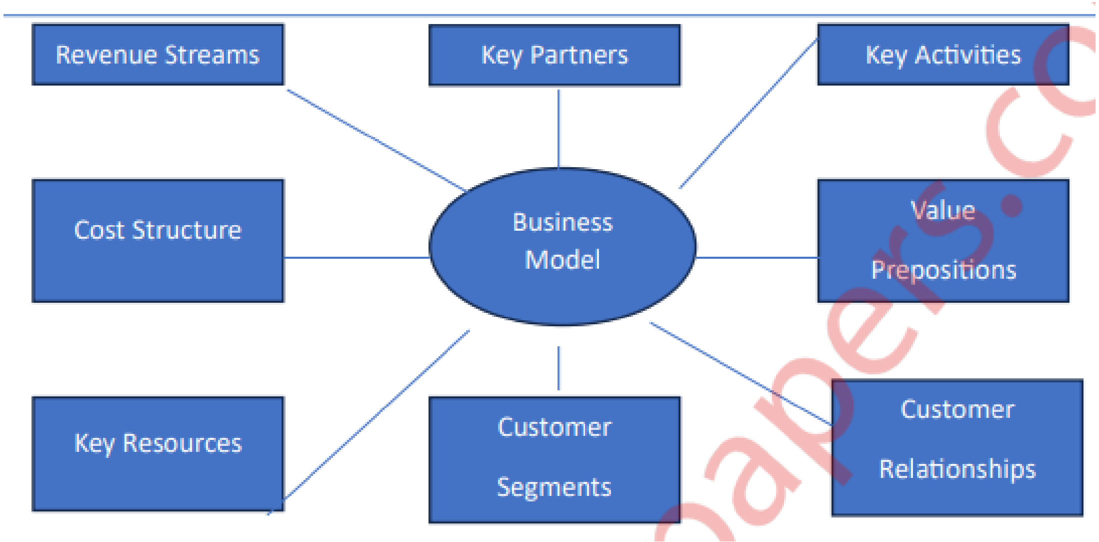
LEARNING FROM HISTORY

- We have learnt the following to apply to an Internet of Things project that we want to turn into a viable and profitable business
  - First, we've seen that some models are ancient, such as Make Thing Then Sell It.
  - The way you make it or the way you sell it may change, but the basic principle has held for millennia.
  - Second, we've seen how new technologies have inspired new business models.
  - Third, although there are recurring patterns and common models, there are countless variations.
- Subtle changes to a single factor, such as the manufacturing process or the way you pay for a product or resource, can have a knock-on effect on your whole business.
- Finally, new business models have the power to change the world like mass production changed the notion of work itself.

THE BUSINESS MODEL CANVAS

FAQ

1. Discuss the business model canvas for IOT.
2. With the help of a diagram, explain business model canvas.



- At first sight, it looks as though each box is simply an element in a form and the whole thing could be replaced by a nine-point checklist.
- However, the boxes are designed to be a good size for sticky notes, emphasizing that you can play with the ideas you have and move them around.
- Also the layout gives a meaning and context to each item.
- Let's look at the model, starting with the most obvious elements and then drilling down into details that we might neglect without this kind of template.
- **Key Partners :**
  - Identify the company's key partners.
  - This can consist of important supplies in your supply chain.
  - How does the company works with these key partners and the motivation behind them.
- **Key Activities:**
  - What specific key activities are necessary to deliver you value proportion?
  - Its revenue streams, distribution channels or customer relationships
- **Value Propositions :**
  - Identify the core value the company provides the customers.
  - What exactly is the company trying to give to its customers?
  - What problem is your company trying to solve?
  - Is it satisfying for the customers or not
- **Customer relationships :**
  - What type of relationship do we have with the customers
  - How does it differ from the customer segments
  - Do you communicate frequently with your customers
- **Customer segments:**
  - Identify who is your value proposition targets.
  - Who are you creating value for?
  - Who are your most important customers?

- **Key resources :**
  - What specific key resources or assets are necessary to deliver your value or position?
  - Consider what resources your distribution channels and revenue system do need to preserve.
  - Are they well integrated and cost efficient?
  - Are they utilized efficiently?
- **Cost structure :**
  - Are they properly utilizing economies of scale?
  - What proportion of costs are fixed and variable?
  - Is your company focused on cost optimization value?
- **Revenue Streams** :
  - Identify the ways your value proposition generates money from your business?
  - Does your company have multiple methods of generating revenue?
  - What is pricing strategy for the products offered by your company?
  - Does your company offers multiple forms of payment?

WHO IS THE BUSINESS MODEL FOR?

- Primarily, the reason to model your business is to have some kind of educated hypothesis about whether it might deliver what you want from it.
  - The canvas help you think about the business and give you ways to brainstorm different ideas:
  - What if we target the product at students instead of businesses?
  - What if we outsource our design to an agency?
  - What if we sell at low volume/high value instead?
- The model is also useful if you want to get other people involved.
- This could be an employee or a business partner or an investor.
- Your customers will also be considering whether to invest their time and money in your product.
- They will ask themselves certain questions about it.
  - Why should I waste time trying out yet Another Social Network? I think I'll wait and see whether all my friends join it first.
  - Your online document collaboration looks great, but is it worth my moving my whole business to it? If you stop trading or change the platform, we may have to redo all the work again.
  - This free service is fantastic, but why don't you let me pay for it, so I can get consistency, receive support, and avoid adverts?

MODELS

FAQ

1. Explain the following business models: Make things, Sell things, Subscriptions and customization.

Make things, sell things:

- It is the simplest business model of the Internet of Things.
- However, there are many small projects that choose the option of selling their product in kit form which requires some additional assembly work.

- As these kinds of kit products are aimed for specific customers and not for general people, the administrative burden reduces.
- Thus, limiting the sale of specific products also limits the revenue generation.

Subscription:

- Now-a-days maximum number of things are available on subscription form.
- It helps in maintaining the rapidly growing demand of the customers efficiently.
- As the price in this model is fixed, the customers are known what services they are availing and how does that fit into their budget.
- Subscription helps in recurring sales, thus, it is easy to determine the revenue generation which in turn helps to maintain the resources and inventories and manage the growth of the business.
- For example, Netflix, Hot Star, etc.

Customization:

- Mass Production helped in development of business and generation revenue.
- But there are many premium customers who demand for customization of the product they buy.
- Thus, options for customization may also lead to the rise of new business models.
- When it comes to achieving both mass production and customization, it becomes difficult as it is time consuming and more resources are utilized in completing the customer demand.
- Thus, it can be limited by allowing customization in a fixed boundary.
- For eg. Let us consider a company called makieMe that produces customized dolls.
- This company has its own website wherein, the customer end customize their doll by changing the wigs, scarves, dresses and make your doll unique.
- All these changes are done in a fixed boundary.

Achieving Customization

FAQ

1. Explain how to achieve customization in an Internet of Things Devices.

- The provider can get more close to the customer by providing customization in its product.
- It means that the customer can choose according to its requirement and choice.
- The customization can also be implemented in mass production.
- The provider has a menu of customization so that the provider is prepared with any form of customization.
- For example, in an automotive setup, customizing or selecting the color of a car, or what type of seat can be provided that does not manipulate the working of the car.
- Similarly, customization should be provided in such a way that it doesn’t affect the making & working of a product.
- Many Internet of Things products have some possibility of customization:
- Every Bubblino has a name (given to it by Adrian), but the user can also change which phrases he listens to on Twitter.

- BERG’s Little Printer offers a selection of content to be printed but also an option of which smiley face it will print for display while waiting for a new delivery.
- (Of course, only a limited number of choices are available currently.)
- Although printer output is as flexible as the software that feeds it, BERG has limited the options to fit into its product aesthetic.
- The new manufacturing techniques, such as laser cutting and 3D printing, should allow great possibilities for customizing even the physical devices.
- MakieLab (http://makie.me) make dolls that can be designed online. Built to your specification, they are therefore unique and entirely yours in a way that a mass-produced doll

Be a Key Source:

- Not all the business would sell a product with mass production.
- There would be business who would specialize in manufacturing some of the parts of a device such as PCBs.
- Such companies would result to be a key source to those companies which require PCBs for making their products.

Provide Infrastructure: Sensor Networks

- Internet of Things provides many such systems where the system requires sensor data as input.
- This sensor data must be accurately calibrated and is very expensive to create.
- Such large but finite data is created by government agencies and organizations.
- These organizations provide infrastructure to the companies that produce such systems which require sensor data.

Take a Percentage:

- In case, if the value of the gathered data is more than that of the physical device, then it might happen that you might sell the physical product for free.
- For example, the energy providing companies follow this strategy with the smart meters.

FUNDING AN INTERNET OF THINGS

- If you have enough money with you to invest in your IOT start up as full time without working extra then you can fund your business yourself.
- If you don’t have enough funding, it does not mean that you cannot have a startup build, but you need to try some other ways.
- If your project doesn’t need huge funding initially, you can have limiting factors where you can work part time to fund your startup.
- Many people club their startup with consulting business where they take up small project to have experience about the same ad also gain profit out of it.
- Make sure you don’t have to spend huge amount on startup.
- Investments on things like office can be cut down as you would not be requiring an office at initial stage of your startup as you can actually work from your home.

HOBBY PROJECTS AND OPEN SOURCE

- If your project is also your hobby, you may not have to spend a lot on your free-time activity.
- One way to make a project grow faster might be to release all the details as open source and try to foster a community around it.
- This approach can be hard work and can benefit from a natural talent, experience, or luck in attracting and maintaining good collaborators.
- After you have open-sourced a project, you can't close-source it again.
- The alternative is you fork the project and continue to work on it in secret, but the existing project may carry on if your collaborators are enthusiastic enough about it.
- Indeed, your idea, code, and schematics could be used by others in their own commercial offering.
- Careful consideration of the license used may be critical here: A more restrictive license such as the GPL requires those who build on your work to share their source code also under the same terms.
- Hence, using the GPL may help restrict commercial exploitation to only those groups that are happy for you to, in turn, reap the benefits of their work.
- Also, when thinking about open source, remember that as the project initiator and owner, you would be the best placed in forming a company around the project and are more likely to reap benefits from the relationship with the community:
  - Many pairs of eyes and hands testing, reporting problems, fixing them, and building new features
  - Many passionate users with real use cases and opinions about the product—better than any focus group
- The goodwill of that community, with its ready-made network of personal recommendations and social-media marketing, running an open source project takes work, and the risk of losing control of your project may not be for everyone, but it is certainly an option to consider.

VENTURE CAPITAL

FAQ

1. Explain the term Venture Capital.
2. What is venture capital? How can one exit?
3. Write a short note on venture capital.

- Venture capital is finance that investors provide to startup companies and small businesses that are believed to have long term growth potential.
- However, it does not always take just a monetary form, it can be provided in the form of technical or managerial expertise.
- Venture capital means risk capital.
- The risk visualized may be very high, may be so high as to result in loss or very loss so as to result in high gains.
- The main objective of venture capital is not to gain interest but capital gain.
- The venture capital investment is made when a venture capitalist buys shares of such a company and becomes a financial partner in the business.

- The Venture capital typically comes from institutional investors and high net worth individuals and is pooled together by dedicated investment firms.
- It is the money provided by an external investor to finance a new, growing, or troubled business.

How can one exit?

- An exit does not mean that you have to leave the company.
- In fact, in many cases, the exit terms may require that you stay for a maximum of one to three years after acquisition in order to ensure continuity and value for the buyer.
- An exit strategy is a "method by which a venture capitalist or business owner intends to get out of an investment that he or she made".
- Because your investors will want a return, your long-term goal can't just be to make your company successful but to do it in such a way as to pay back the investment.
- Typically, there are only two exits:
  - You get bought by a bigger a company
  - You do an IPO(Initial Public Offering)-that is, float on the stock market:

You get bought by a bigger a company

- In this case, the buyer buys out the investors that is, the buyer pays the investors value of their percentage equity of their perceived valuation of the worth of the company.
- Founding members of the company often transfer to the purchasing company, as they constitute one of the company's principal resources.

You do an IPO (Initial Public Offering)-that is, float on the stock market:

- This involves new shares being issued and sold to the stock market.
- Although this option "dilutes" the value of the shares already issued, the existing holders are able to the sell their shares on the market too, to get back their investment, or to retain the shares if they believe that the shares will grow in value.

GOVERNMENT FUNDING

FAQ

1. Explain Government funding for IOT Projects.

- Governments typically want to promote industry and technological development in their country, and they may provide funds to help achieve particular aims.
- Although governments can and do set up their own venture capital funds or collaborate with existing funds in various ways, they generally manage the majority of their funds differently.
- For one thing, they also want to fund existing companies to do new research and innovation, which might sit uncomfortably with the concept of equity.
- The money provided still has "strings attached", but they are likely to be handled differently:

Outputs:

- Deliverables (aka outputs) are the metrics that an awarding body may use to tell if you are doing the kind of thing that the body wants to fund.
- This metric may simply be a test that you are managing the money well or may be related to the goals that the body itself wishes to promote.
- You might be required to write regular reports or pass certain defined milestones on schedule.
- If your funding is given in stages, the later payments may be conditional on successful delivery of previous outputs.
- You should be very clear on what needs to be done and how onerous the task is.
- Even a large sum of money is worth less if you are going to spend significant amounts of time on secondary activities.

Spending constraints:

- You need to think twice before spending on the resources.
- It means that, just because you are been funded by the government, you blindly cannot spend on the resources, you are expected to analyze the cost and then spend accordingly.
- You also must consider the time and management overhead in making use of the (possibly unwanted) service.
- It is perfectly normal for companies to work through multiple sources of funding.
- Having successfully worked through a government innovation award may stand you in good stead for approaching VCs for more money later.

CROWD FUNDING

- We can consider crowd funding as the long tail of funding projects.
- Getting many people to contribute to a project isn’t exactly a new phenomenon.
- Over millennia many civic and religious monuments and constructions have been funded at least partly by the public.
- However, such projects have been mostly sponsored and given focus by some influential person or body. Different types of appealing text, slick videos, audios and great design makes the difference between yours and a competing project.
- Because even some successfully funded projects may fail, older and wiser crowd funders may be more likely to fund projects in which they see some attention to the business model or a track record for successful completion by the project team.
- In short, your funders are real people and will have all the variety of concerns and foibles that any group of real people have.
- This interaction with a large and diverse group is a key part of the interest of this method of funding: It is far more than just the money.
- Crowdsourcing allows consumers with all manner of niche interests to find the producers who are interested in satisfying those niches before even investing your time and money in the product.
- If there is no interest, perhaps the product is not a winner as currently specified and advertised.

- If the project goes viral, as happens occasionally, and gains far more than the targeted amount, you know you have a potential hit on your hands.
- Just as receiving funding from VCs might be valuable as much for the mentoring and networking opportunities, so crowd funding may be as valuable for the market research and viral marketing as for the money raised.

LEAN STARTUPS

- Lean startup is a methodology for developing businesses and products that aims to shorten product development cycles and rapidly discover if a proposed business model is viable; this is achieved by adopting a combination of business-hypothesis-driven experimentation, iterative product releases, and validated learning.
- Lean startup emphasizes customer feedback over intuition and flexibility over planning.
- This methodology enables recovery from failures more often than traditional ways of product development.
- Some of the core lean startup principles are as follows:
- **Minimum viable product:** A minimum viable product (MVP) is the "version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort" (similar to a pilot experiment)
- **Continuous deployment (only for software development):** Continuous deployment, similar to continuous delivery, is a process "whereby all code that is written for an application is immediately deployed into production," which results in a reduction of cycle times.
- **Split testing:** A split or A/B test is an experiment in which "different versions of a product are offered to customers at the same time." The goal of a split test is to observe differences in behavior between the two groups and to measure the impact of each version on an actionable metric.
- **Actionable metrics:** Actionable metrics can lead to informed business decisions and subsequent action.[
- **Pivot:** A pivot is a "structured course correction designed to test a new fundamental hypothesis about the product, strategy, and engine of growth
- **Innovation accounting:** This topic focuses on how entrepreneurs can maintain accountability and maximize outcomes by measuring progress, planning milestones, and prioritizing.[
- **Build-Measure-Learn:** The Build–Measure–Learn loop emphasizes speed as a critical ingredient to customer development. A team or company's effectiveness is determined by its ability to ideate, quickly build a minimum viable product of that idea, measure its effectiveness in the market, and learn from that experiment.