Embedded Devices

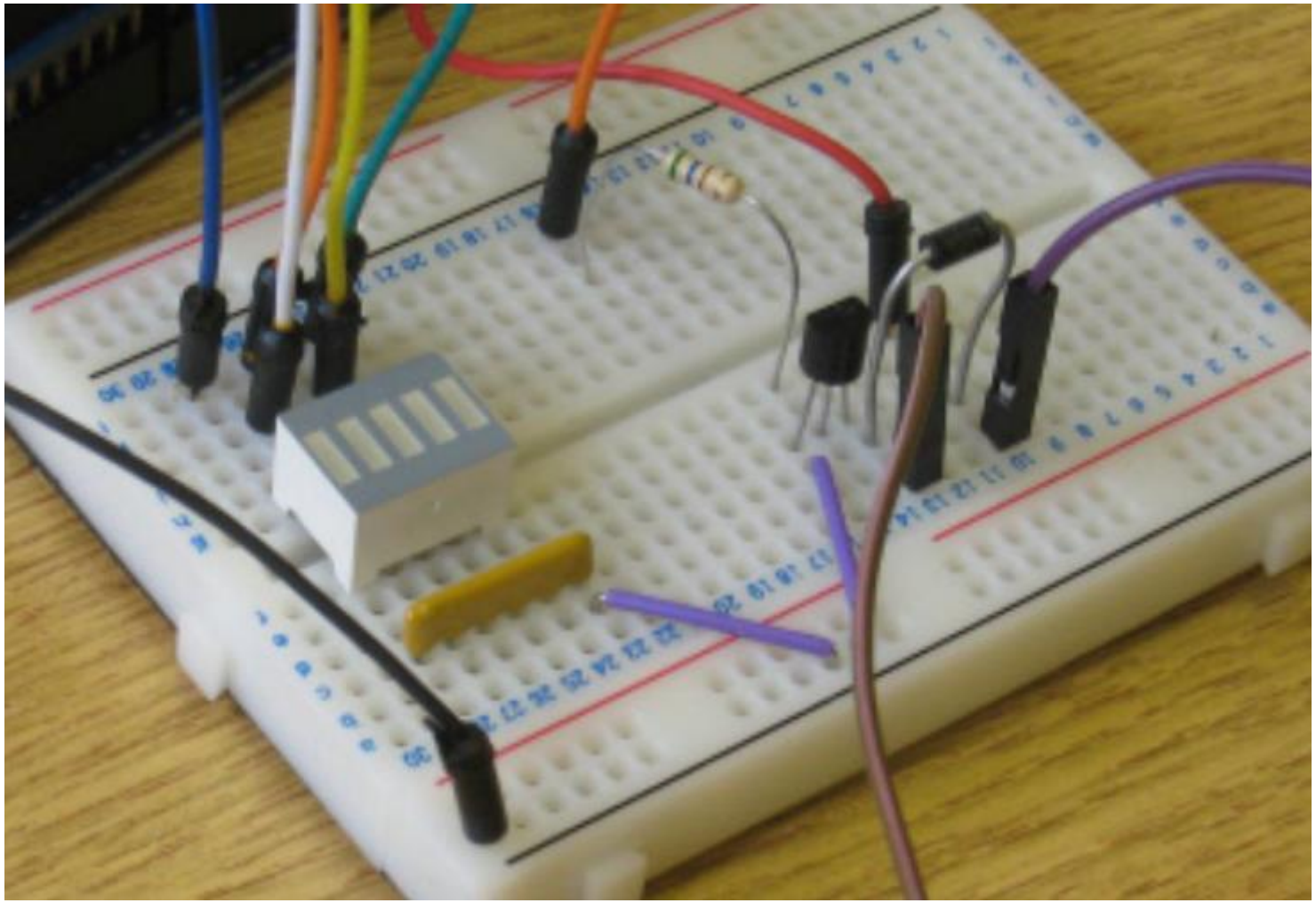
Unit 2

ELECTRONICS

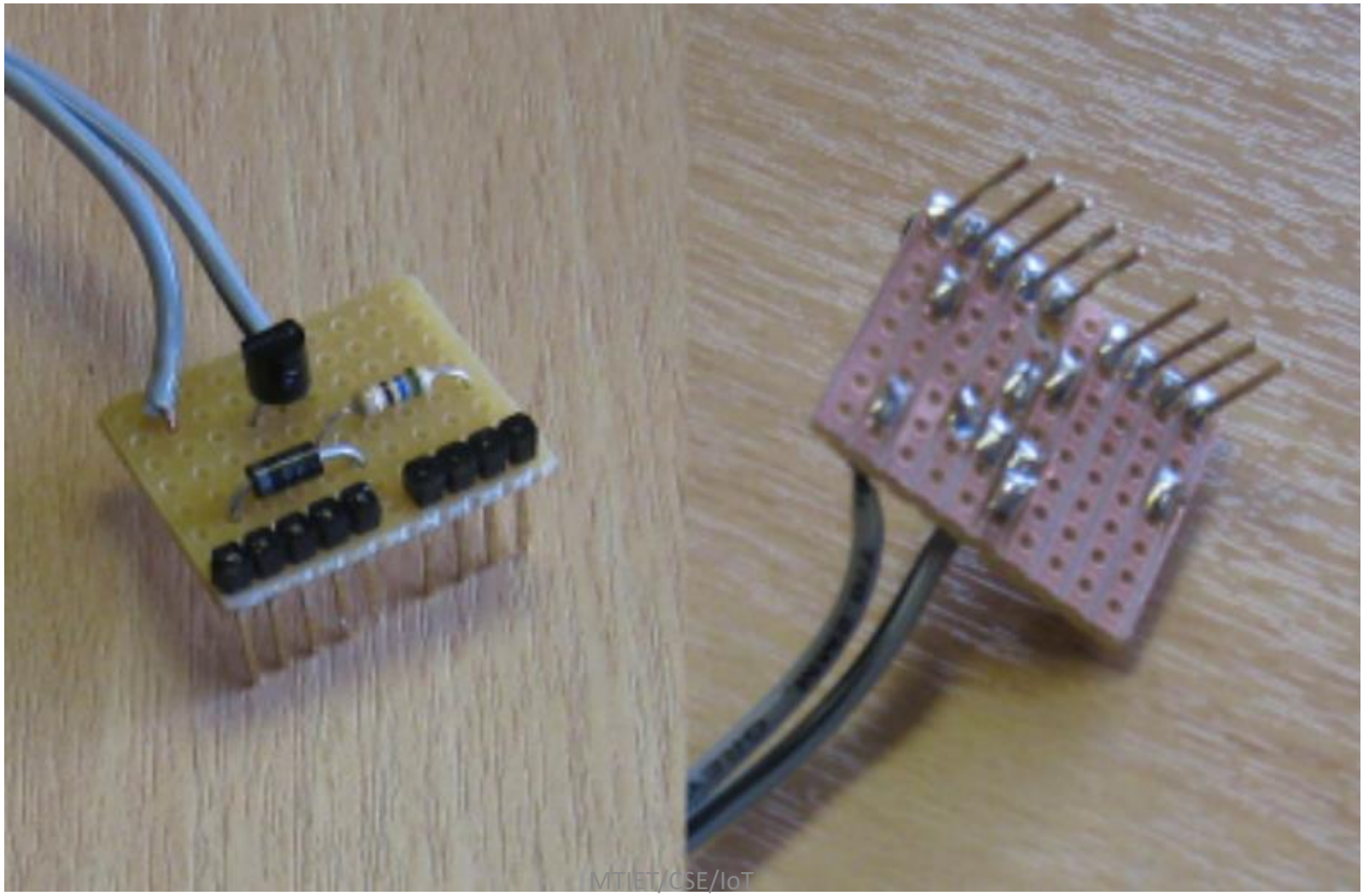
- Most of the prototyping can be done on what are called solderless breadboards.
- They enable you to build components together into a circuit with just a push-fit connection, which also means you can experiment with different options quickly and easily
- When it comes to thinking about the electronics, it's useful to split them into two main categories:
  - **Sensors:** Sensors are the ways of getting information into your device, finding out things about your surroundings.
  - **Actuators:** Actuators are the outputs for the device—the motors, lights, and so on, which let your device do something to the outside world.
- Within both categories, the electronic components can talk to the computer in a number of ways.
  - digital I/O, which has only two states: a button can either be pressed or not; or an LED can be on or of
  - general-purpose input/output (GPIO) pins
  - the voltage that the processor is using to run (commonly 5V or 3.3V)

- Some sensors may be analog for some IoT applications
- computers are purely digital devices, we need a way to translate between the analogue voltages in the real world and the digital of the computer.
- An analogue-to-digital converter (ADC) lets you measure varying voltages.
- The Arduino has 10-bit ADCs, which by default measure voltages between 0 and 5V. A voltage of 0 will give a reading of 0; a voltage of 5V would read 1023
- The flipside of an ADC is a DAC, or digital-to-analogue converter. DACs let you generate varying voltages from a digital value
- pulse-width modulation (PWM), which gives an approximation to a DAC by rapidly turning a digital signal on and off so that the average value is the level we require.
- For more complicated sensors and modules, there are interfaces such as Serial Peripheral Interface (SPI) bus and Inter-Integrated Circuit (I2C).

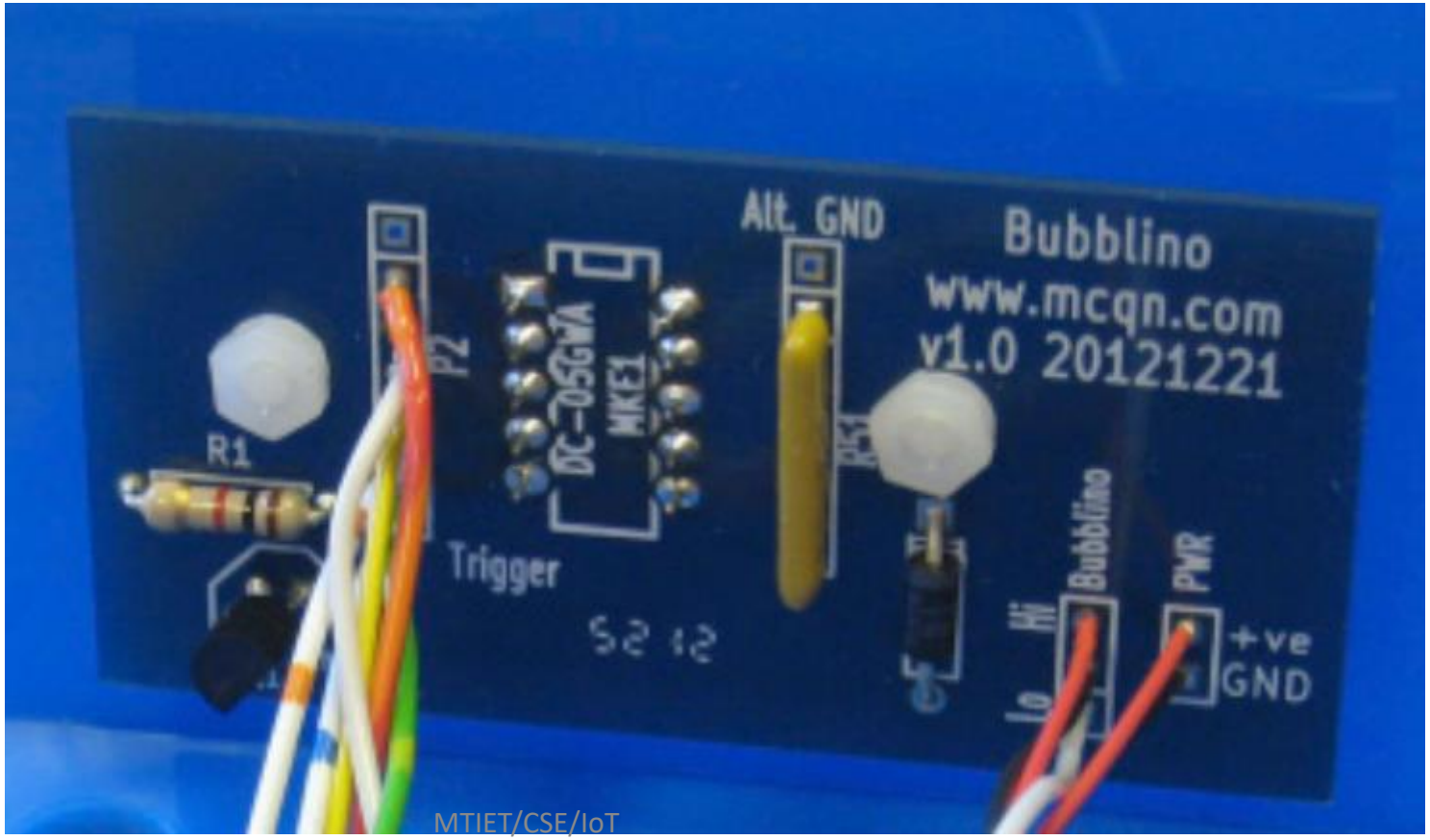
- **Journey to a Circuit Board**
- The first step in creating your circuit is generally to build it up on a breadboard.
- Here we can reconfigure the connections based on the results



- In second step When we are happy with how the circuit works, soldering it onto a stripboard will make the layout permanent and this type of boards are called as The stripboard.



- In third step If you need to make many copies of the circuit, or if you want a professional finish, you can turn your circuit into a PCB (Program Control Board).
- it easier to build up the circuit because the position of each component will be labelled, there will be holes only where the components go, and there will be less chance of short circuits because the tracks between components will be protected by the solder resist.



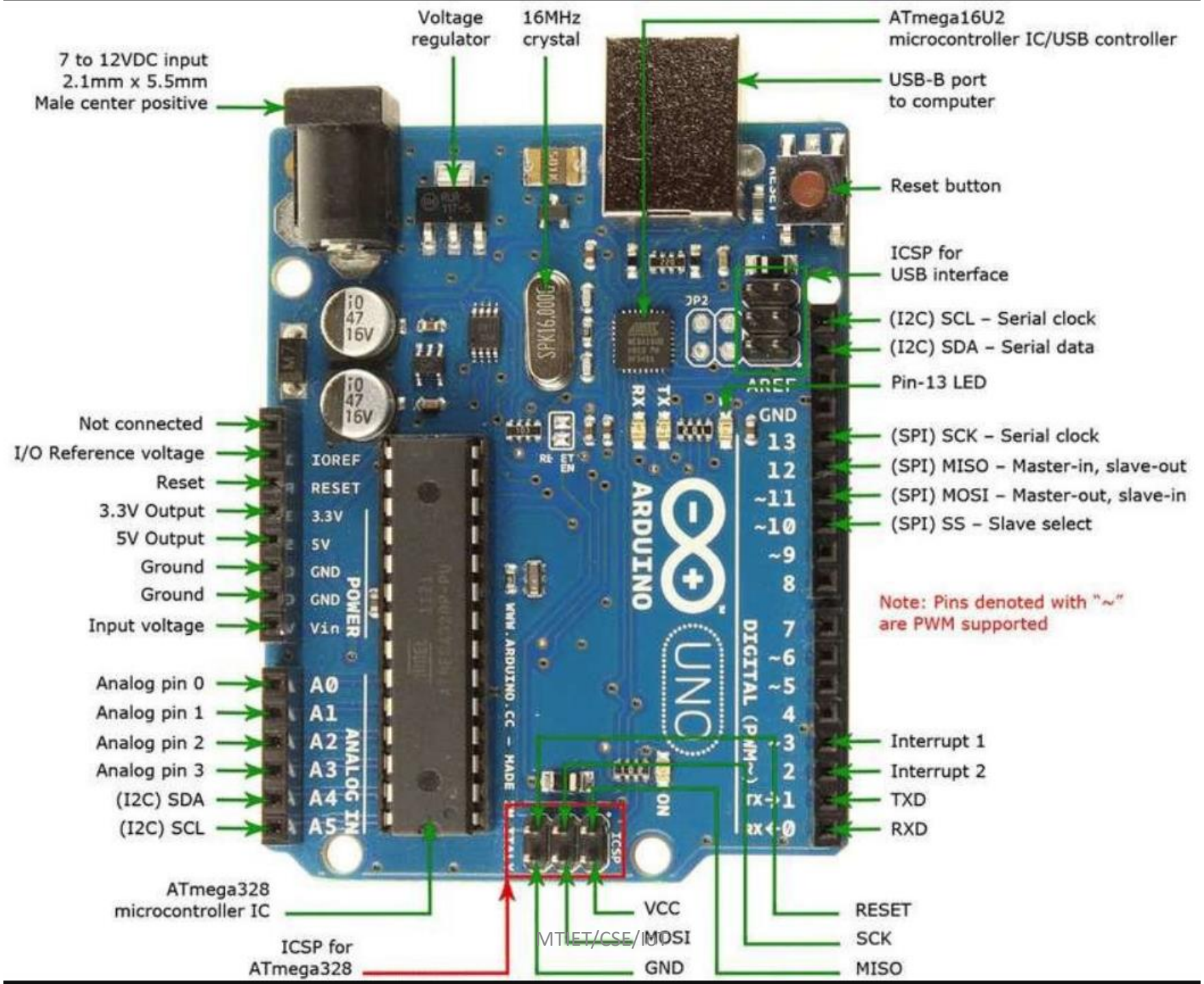
EMBEDDED COMPUTING BASICS

- **MICROCONTROLLERS**
- Internet of Things devices take advantage of more tightly integrated and miniaturized devices
  - from the most basic level of microcontrollers to more powerful system-on-chip (SoC) modules.
- These systems combine the processor, RAM, and storage onto a single chip, and they are much more specialized, smaller than t PC
- Microcontrollers are very limited in their capabilities since
  - Limited RAM capabilities (KB),
- the microcontroller market consists of many manufacturers. (Atmel, Microchip, NXP, Texas Instruments, etc)
- **SYSTEM-ON-CHIPS**
- the SoC comes in between the low-end microcontroller and a PC (for example, the BeagleBone or the Raspberry Pi).
- Like Microcontrollers, SoCs contain a processor and a number of peripherals onto a single chip but usually have more capabilities.
  - RAM measured in megabytes rather than kilobytes, Storage for SoC modules are SD cards

- The greater capabilities of SoC require some sort of operating system to operate their resources
- **CHOOSING YOUR PLATFORM**
- How to choose the right platform for your Internet of Things device**?**
- The platform you choose depends on the particular blend of price, performance, and capabilities that suit what our application require
- The important factors which influence the IoT device platform are
  - Processor Speed
  - RAM
  - Networking
  - USB
  - Power Consumption
  - Interfacing with Sensors and Other Circuitry
  - Physical Size and Form Factor

ARDUINO

- A group from the Interaction Design Institute Ivrea (IDII) wanted a board for its design students to use to build interactive projects in 2005 from northern Italy
- The Arduino team's focus on simplicity rather than raw performance
- The "standard" Arduino board has gone through a number of iterations: Arduino NG, Diecimila, Duemilanove, and Uno.
- The Uno features an ATmega328 microcontroller and a USB socket for connection to a computer. It has 32KB of storage and 2KB of RAM and 14 GPIO pins (of which 6 can also provide PWM output) and 6 10-bit resolution ADC pins
- For more space or a greater number of inputs or outputs, go for the Arduino Mega 2560 providing 256KB of Flash storage, 8KB of RAM, three more serial ports, a massive 54 GPIO pins (14 of those also capable of PWM) and 16 ADCs.
- Arduino Due has a 32-bit ARM core microcontroller, Its configurations are similar to the Mega's, with RAM size od 96KB

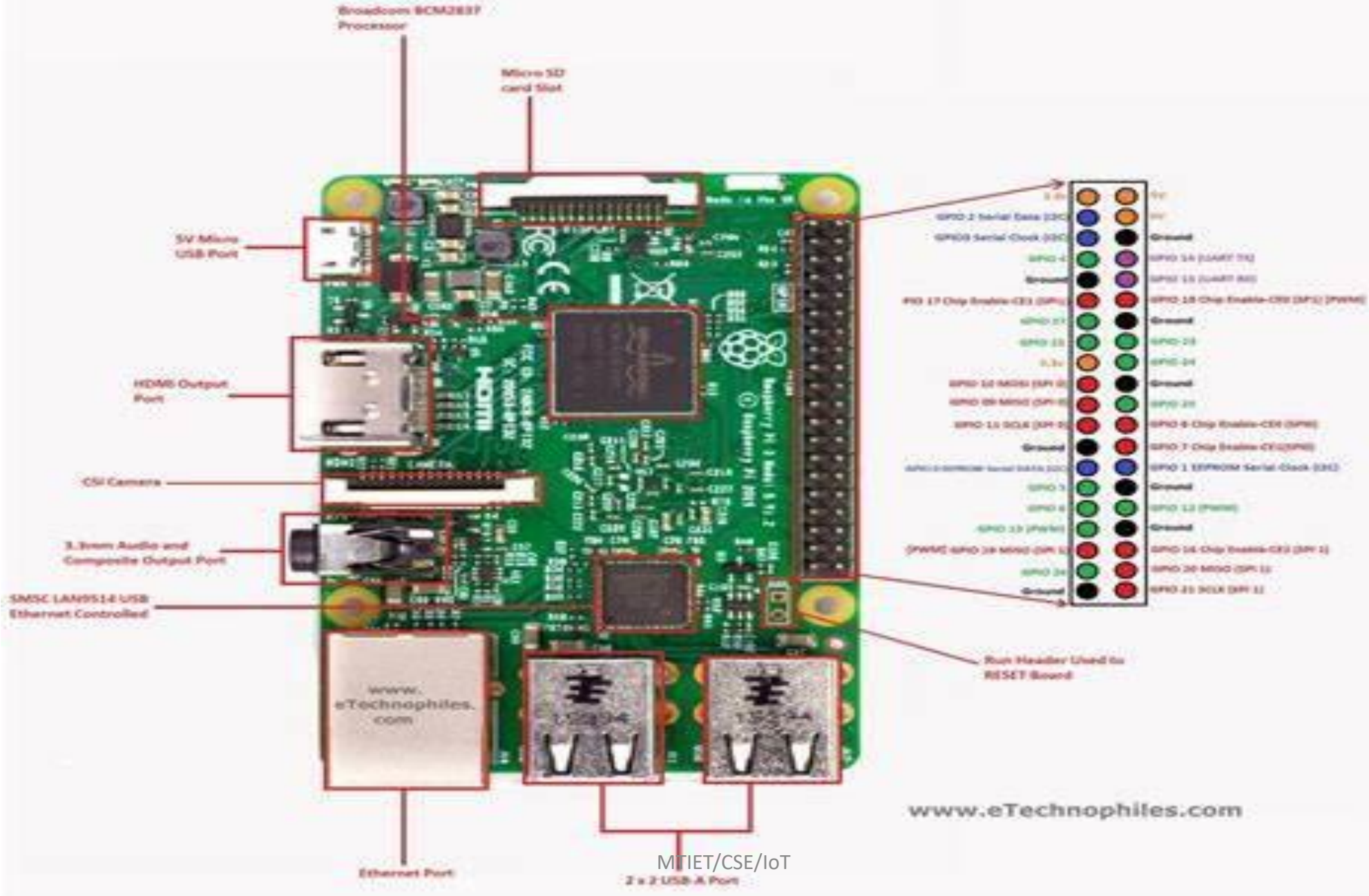


- **DEVELOPING IoT PROJECTS ON THE ARDUINO**
- the Arduino is optimized for simplicity
- Using a single USB cable, you can not only power the board but also push your code onto it, and (if needed) communicate with it
- Integrated Development Environment (IDE)
  - We can develop projects with the Arduino using the integrated development environment (IDE) that the team supply at http://arduino.cc.
  - it is very simple to use
- Pushing Code
  - we need to choose the correct serial port (which you can discover from system logs or select by trial and error) and the board type
  - When your setup is correct, then first, the code is checked and compiled, If the code compiles successfully, it gets transferred to the Arduino and stored in its flash memory.
- Operating System
  - The Arduino doesn't, by default, run an OS
  - When we switch on the board, it simply runs the code that we have compiled until the board is switched off again

- Language
  - The language usually used for Arduino is a slightly modified of C++ derived from the Wiring platform
  - It includes some libraries used to read and write data from the I/O pins provided on the Arduino
  - The code needs to provide only two routines:
    - setup(): This routine is run once when the board first boots. You could use it to set the modes of I/O pins to input or output or to prepare a data structure which will be used throughout the program.
    - loop(): This routine is run repeatedly in a tight loop while the Arduino is switched on. Typically, you might check some input, do some calculation on it, and perhaps do some output in response.
- Debugging
  - debugging experience to be able to identify certain compiler errors
  - Runtime programming errors may be difficult to identify and Arduino platform doesn't use try... catch... Logic
  - , the Arduino allows you to write information over the USB cable using Serial.write()

RASPBERRY PI

- The Eben Upton, trustee and cofounder of the Raspberry Pi Foundation, was to build a computer that was small and inexpensive and designed to be programmed
- which featured an exceptionally powerful graphics processing unit (GPU), capable
- Raspberry Pi is effectively a computer that can run a real, modern operating system, communicate with a keyboard and mouse, talk to the Internet, and drive a TV/monitor with high-resolution graphics.
- The Raspberry Pi has 8 GPIO pins, which are exposed along with power and other interfaces in a 2-by-13 block of male header pins
- The block of pins provides both 5V and 3.3V
- the Raspberry Pi doesn’t have any analogue inputs (ADC), which means that options to connect it to electronic sensors are limited,


Broadcom BCM2835
Processor
Micro SD
card Slot
5V Micro
USB Port
HDMI Output
Port
CSI Camera
3.5mm Audio and
Composite Output Port
SMSC LAN9514 USB
Ethernet Controlled
Run Header Used to
RESET Board
www.eTechnophiles.com
Ethernet Port
2 x 2 USB A Port

- **DEVELOPING ON THE RASPBERRY PI**
- Raspberry Pi is SoC type, hence it functionalities are not limited like Arduino.
- the number of variables on the Raspberry Pi are much greater, and enables do works in alternative ways.
- Operating System
  - many operating systems can run on the Pi, we recommend using a popular Linux distribution, such as
  - Raspbian: Released by the Raspbian Pi Foundation, Raspbian is a distro based on Debian. This is the default "official" distribution and is certainly a good choice for general work with a Pi.
  - Occidentalis: This is Adafruit's customized Raspbian. Unlike Raspbian, the distribution assumes that you will use it "headless"—not connected to keyboard and monitor—so you can connect to it remotely by default.
- Programming Language
  - Python is a good language for educational programming for Raspberry Pi (the name "Pi" comes initially from Python)

- The features of python language in Raspberry Pi are
  - Handling strings of character data
  - Completely avoiding having to handle memory management (and bugs related to it)
  - Making calls to Internet services and parsing the data received
  - Connecting to databases and more complex processing
  - Abstracting common patterns or complex behavior

- Differences between Arduino and Raspberry Pi:-->

	Arduino Due	Raspberry Pi Model B
CPU Speed	84 MHz	700 MHz ARM11
GPU	None	Broadcom Dual-Core VideoCore IV Media Co-Processor
RAM	96KB	512MB
Storage	512KB	SD card (4GB +)
OS	Bootloader	Various Linux distributions, other operating systems available
Connections	54 GPIO pins 12 PWM outputs 4 UARTs SPI bus I²C bus USB 16U2 + native host 12 analogue inputs (ADC) 2 analogue outputs (DAC)	8 GPIO pins 1 PWM output 1 UART SPI bus with two chip selects I²C bus 2 USB host sockets Ethernet HDMI out Component video and audio out

MOBILE PHONES AND TABLETS

- Mobile Phones and Tablets are other notable platforms for developing IoT applications.
- Modern phones and tablets, with OS like iOS (iPhone/iPad), Android, Blackberry, or even Windows.
- These devices also contains sensors like gyroscopes to thermometers, one or more cameras capable of capturing still images and video, microphones, GPS, WiFi, Bluetooth, USB, NFC, buttons, and a multitouch sensitive screen.
- Phones and tablets have always-on Internet connectivity, via WiFi or the phone network.
- These devices also have several appealing outputs: high-fidelity audio output, HD-quality video, and one or more vibrating elements.
- They also have processing power similar to the Raspberry Pi's
- But the Mobile Phones and Tablets are designed for human necessities to perform various applications, there isn't an easy way to wire a phone up to additional circuits or inputs for IoT applications.

PLUG COMPUTING: ALWAYS-ON INTERNET OF THINGS

- Every IoT device must require power. Even the smallest board may have a power pack.
- The idea of a plug computer is to include the circuitry within the same enclosure as the plug and adaptor
- The benefits of Plug computing is
  - You don’t need to place the computer somewhere.
  - It does not have a cable to get dislodged or pulled.
  - Being plugged in, tight against a plug socket, the device is unobtrusive (not attracting).
- One of the most popular plug computing platforms is the SheevaPlug , which typically runs a Debian distribution tailored for its ARM chip. It doesn’t have many input/output capabilities, usually only an Ethernet socket and USB

A SheevaPlug, looking more like a power adapter than a tiny computer.

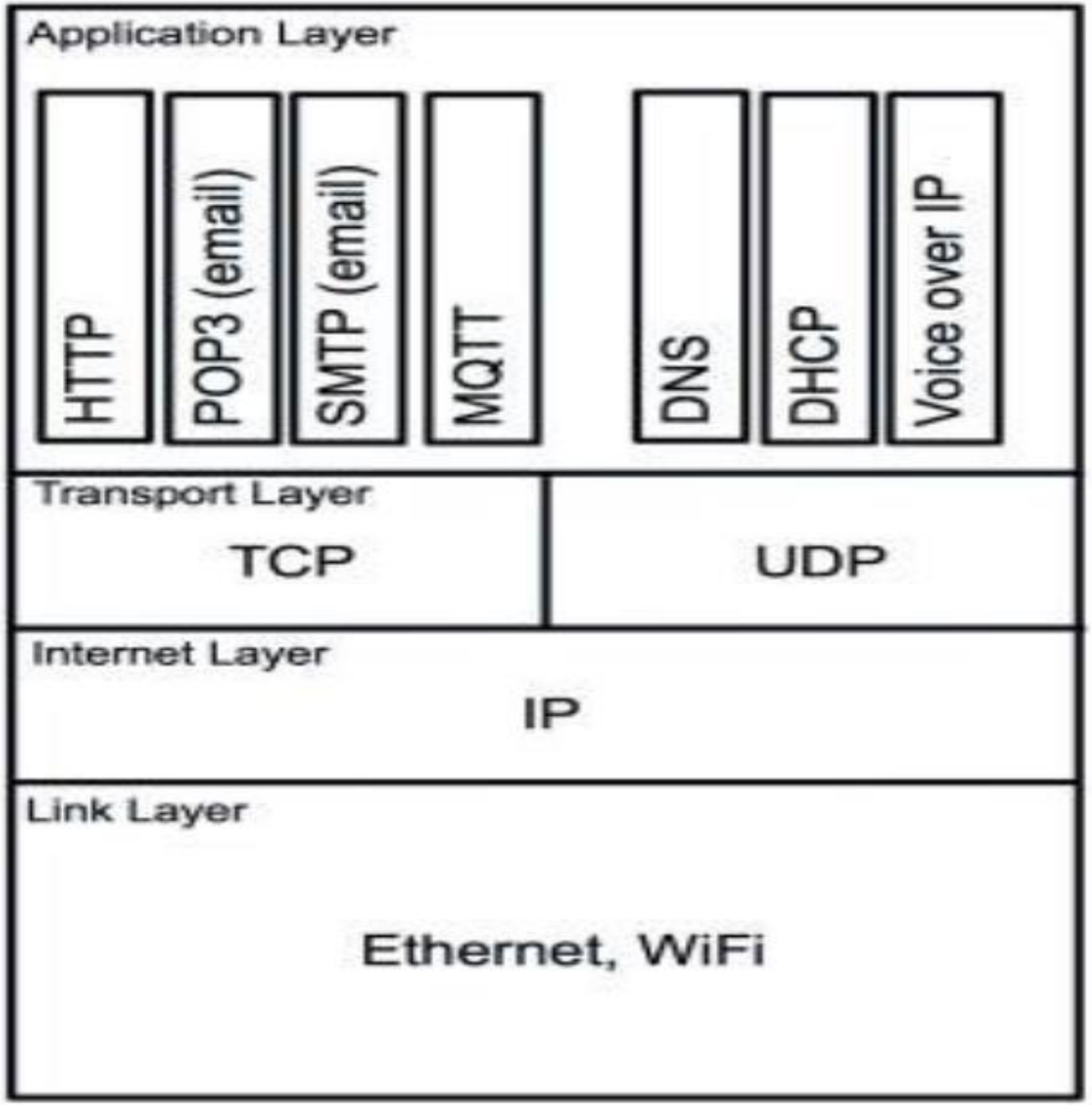
Communication in the IoT
Prototyping online components

Unit 3

INTERNET COMMUNICATIONS: AN OVERVIEW

- For IoT Technology internet is an important communication medium
- For internet, the communication protocols plays a vital role, the important protocols are
  - IP
  - TCP
  - UDP
- **IP**
  - The main objective of Internet Protocol (IP) is sending the data packets from source to destination by using respective IP address
  - the sending machine doesn't always know the best route to the destination in advance, hence the IP protocol identifies the best route
  - the packets of data have to go through a number of intermediary machines, called routers, to reach their destination.
  - an IP packet is a block of data contains the information such as the name and address of the server, and so on.
  - IP packet ever gets transmitted across the local wired network via an Ethernet cable—the cable that connects home broadband router or office local area network (LAN) to a desktop PC—then the whole packet will get bundled up into another type of envelope, an Ethernet Frame, which adds additional information to reach the destination computer
  - The challenge with IP is it works with single packets and not guarantees the delivery

- **TCP**
  - The simplest transport protocol on the Internet, TCP is built on top of the basic IP protocol and adds sequence numbers, acknowledgements, and retransmissions.
  - A message sent with TCP can be long and give the sender some assurance that it arrived at the destination.
  - the combination of TCP and IP is so useful for many services
- **THE IP PROTOCOL SUITE (TCP/IP)**
  - The combination of TCP and IP is so ubiquitous
  - There are 4 important layers are present in TCP/IP
  - The low-level protocols at the link layer manage the transfer of bits of information across a network link throw wired or wireless mediums.
  - The Internet layer ensures the packets must be transferred to destination with IP address.
  - Then TCP, which lives in the transport layer, sits on top of IP and provides end to end communication (application) using port number
  - Finally, the application layer contains the protocols that deal with fetching web pages, sending emails, and Internet telephony
- **UDP**
  - User Datagram Protocol is a connection less and unreliable protocol
  - UDP useful for applications such as streaming data, which can cope with minor errors but doesn't like delays
  - **Ex :** Voice over IP (VoIP)—computer-based telephony, such as Skype



The Internet Protocol suite.

IP ADDRESSES

- the Internet Protocol knows the addresses of the destination and source devices.
- IP addresses are numbers. In Internet Protocol version 4 (IPv4 32bit size), almost 4.3 billion IP addresses are possible—4,294,967,296 to be precise, or 2^{32}
- IP addresses are written as four 8-bit numbers separated by dots (from 0.0.0.0 to 255.255.255.255)—Ex: 192.168.0.1
- Every machine on the Internet has at least one IP address. That means every computer, every network-connected printer, every smartphone, and every Internet of Things device has one
- For Interne of Things (IoT) technology the IPv4 addresses are will not be sufficient, A better solution to this problem is the next generation of Internet Protocol, IPv6
- **DNS**
- IP addresses are easy for most humans to forget. The Domain Name System (DNS) helps human navigate the Internet instead IP address
- **Ex:** google.com, bbc.co.uk, wiley.com, arduino.cc

- Each domain name has a top-level domain (TLD), like .com or.uk, which further subdivides into .co.uk and .gov.uk, and so on.
- The domains then have information about where to direct calls to individual machines or services. Ex: www.google.com, mail.google.com, calendar.google.com
- **STATIC IP ADDRESS ASSIGNMENT**
- How do you get assigned an IP address?
- If you have bought a server-hosting package from an Internet service provider (ISP)
- separated into “classes” of 8 bits, 16 bits, or 24 bits:
  - Class A — From 0.x.x.x
  - Class B — From 128.0.x.x
  - Class C — From 192.0.0.x
- The rigid separation of address ranges into classes was not very efficient , that is all the addresses may not be used in the allocated addresses for an organization, to overcome this problem in 1993 Classless Inter-Domain Routing (CIDR), which allows you to specify exactly how many bits of the address are fixed

- the system administrator assigns server. Then the administrator makes a note of the addresses and updates DNS records. We call this kind of address static because once assigned it won't change again without human intervention.
- **DYNAMIC IP ADDRESS ASSIGNMENT**
- Getting the IP address instantly when a device is connected to a newtwork is known as dynamic IP address assignment
- In this process the device is requested for an IP address from the network itself using the Dynamic Host Configuration Protocol (DHCP).
- When the device tries to connect, instead of checking its internal configuration for its address, it sends a message to the router asking for an address. The router assigns it an address. This is not a static IP address and the address is assigned dynamically according to which addresses are currently available.
- **IPv6**
- IPv6, which uses 128-bit addresses, usually displayed to users as eight groups of four hexadecimal digits—for example, 2001:0db8:85a3:0042:0000:8a2e:0370:7334.
- The IPv6 address space (2^128) is so huge
- the number of Internet of Things devices will almost certainly need IPv6

MAC ADDRESSES

- Like IP address, every network-connected device also has a MAC address
- It is used to differentiate different machines on the same physical network so that they can exchange packets.
- This relates to the lowest-level “link layer” of the TCP/IP stack.
- MAC stands for Media Access Control. It is a 48-bit number, usually written as six groups of hexadecimal digits Ex: 01:23:45:67:89:ab
- Most devices, such as your laptop, come with the MAC address assigned into their Ethernet chips
- There is a special protocol ARP (address resolution protocol) that is used for to identify the MAC address of a certain device in the network.
- One of the applications of MAC addresses is in the filtering process on wireless networks. In order to prevent strangers from accessing a network, the router is set to accept only specific MAC addresses
- MAC addresses can also be used in data recovery to connect to a wireless device

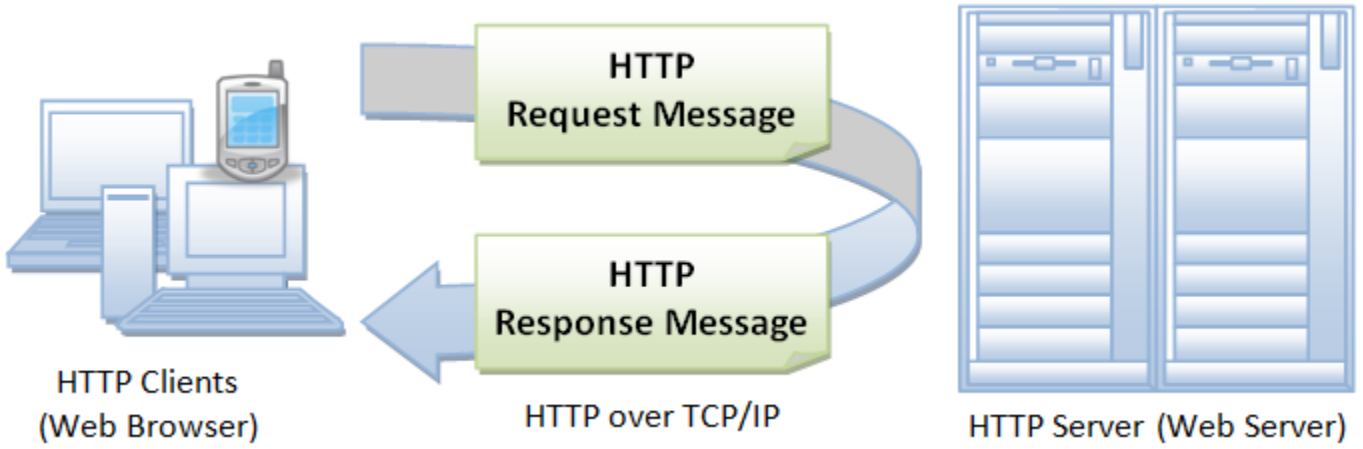
TCP AND UDP PORTS

- when a sender sends a TCP/IP message over the Internet, it must be sent to the right port.
- TCP ports, are referred to by numbers (from 0 to 65535).
- The Internet Assigned Numbers Authority (IANA) is responsible for registering the numbers in these ranges.
- Ports 0–1023 are "well-known ports", and only a system process or an administrator can connect to them
- Ports 1024–49151 are "registered", so that common applications can have a usual port number
- Ex: http://www.example.com:8080
- OTHER COMMON PORTS
  - 80 HTTP
  - 8080 HTTP (for testing servers)
  - 443 HTTPS
  - 22 SSH (Secure Shell)
  - 23 Telnet
  - 25 SMTP (outbound email)
  - 110 POP3 (inbound email)
  - 220 IMAP (inbound email)

APPLICATION LAYER PROTOCOLS

- This is the layer you are most likely to interact with while prototyping an Internet of Things project
- A protocol is a set of rules for communication between computers. It includes rules about how to initiate the conversation and what format the messages should be in. It determines what inputs are understood and what output is transmitted. It also specifies how the messages are sent and authenticated and how to handle (and maybe correct) errors caused by transmission
- The application layer protocols are
- **HTTP (**Hypertext Transfer Protocol)
- HTTP is a simple protocol which follow request and response model for communication between client and server
- In the case of HTTP, a web browser requests HTML files from a web server, which then display in the browser with text, images, hyperlinks, and related assets.
- In this protocol, the client requests a resource by sending a command to a URL, with some headers.
- Ex: http://book.roomofthings.com/hello.txt
- Then the DNS will resolve the name to an IP address.

- HTTP Works with three message types
  - **HTTP GET**: Messages sent to a server contain only a URL
  - **HTTP POST**: Messages place any optional data parameters in the body of the request message
  - **HTTP HEAD**: the server sends back only the header information (contained inside the HTML section).
- **HTTPS: ENCRYPTED HTTP**
  - The HTTPS protocol is actually a mix-up of HTTP over the Secure Socket Layer (SSL) protocol.



- An HTTPS server listens to a different port (usually 443) and on connection sets up a secure, encrypted connection with the client

- Other Application Layer Protocols for IoT
  - HTTP
  - CoAP
  - WebSocket
  - MQTT
  - XMPP
  - DDS
  - AMQP

PROTOTYPING ONLINE COMPONENTS

GETTING STARTED WITH AN API

- IoT devices will most likely use standardized protocols to speak to each other and to other apps and computers on your local or personal network.
- The most important part of a web service, with regards to an Internet of Things device, is the Application Programming Interface, or API.
- An API defines the messages that are sent from client to server and from server to client.
- An API is a way of accessing a service that is targeted at machines rather than people.
- The idea of "mashing up" multiple APIs to get a effective result for user and can be used to powerful effect Ex: Using a mapping API to plot properties to rent or buy using twitter account , google maps and other social media apps
- Screen scraping is the process of copying information that shows on a digital display so it can be used for another purpose
- LEGALITIES: Screen-scraping may break the terms and conditions of a website. For example, Google doesn't allow you to screen-scrape its search pages but does provide an API

- **WRITING A NEW API**
- By using new API we can create an entirely new source of information or services and we can also assemble the data from free or licensed material
- APIs plays an important role in webservices
- The backend API are used to associate the data with the web application through the web browsers
- A new web application API can be created by decoupling the model (core data structure) from the view (HTML/JavaScript) and controller (widgets, form interaction, and so on)
- The tools to create a new web application by using the popular MVC frameworks (Ruby on Rails, Django, Catalyst, and so on)
- CLOCKODILLO
- Clockodillo is an Internet-connected task timer.
- The user can set a dial to a number of minutes, and the timer ticks down until completed.

- It also sends messages to an API server to let it know that a task has been started, completed, or cancelled.
- A number of API interactions deal precisely with those features of the physical device:
  - Start a new timer
  - Change the duration of an existing timer
  - Mark a timer completed
  - Cancel a timer
  - View and edit the timer's name/description
- And, naturally, the user may want to be able to see historical data:
  - Previous timers, in a list
  - Their name/description
  - Their total time and whether they were cancelled
- **SECURITY**
- Security is an important concern with APIs, its related to the how sensitive the information is and for whom it is being transferred? so we need to keep it in mind while designing the API
- In API , the application should also authenticate that request. A password is "good enough" authentication

- Passwords or other devices related information would be transferred through APIs
- CLOCKODILLO maintains the record of all API transaction as shown below with credential exchange among the various application tasks.

Task	Inputs	Outputs
1. Create a new timed task	User, Timer duration	Timer ID
2. Change duration of timed task	User, Timer ID, New duration	OK
3. Mark timer complete	User, Timer ID	OK
4. Cancel timer	User, Timer ID	OK
5. Describe the timed task	User, Timer ID, Description	OK
6. Get list of timers	User	List of Timer IDs
7. Get information about a timer	User, Timer ID	Description, Create time, Status

- One solution to the problem of sending passwords would be to encrypt the whole request, including the authentication details.
- In general For a web API, you can simply do this by targeting https:// instead of http://.

- For IoT Encryption requires solving very large equations and takes CPU and memory. The current Arduino platform doesn't have an HTTPS library
- If you are defining your own API, there are cryptography libraries for Arduino, so there's scope for using them for a custom form of secure communications.
- The OAuth 1.0 protocol—used by services such as Twitter to allow third party applications to access your account without requiring your password— is a good example of providing strong authentication without using HTTPS.
- Here's a revised table of CLOCKODILLO ; we also added some requests to add and check the MAC address for a user

Task	Auth	Inputs	Outputs
1. Create a new timed task	MAC or User/Pass	Timer duration	Timer ID
2. Change duration of timed task	MAC or User/Pass	Timer ID, New timer duration	OK
3. Mark timer complete	MAC or User/Pass	Timer ID	OK
4. Cancel timer	MAC or User/Pass	Timer ID	OK

- **IMPLEMENTING THE API**
- There are four standards for communication through APIs and the standers provides convenient libraries for both client and server to produce and understand the required messages
  - Representational State Transfer (REST): Access a set of web URLs using HTTP methods such as GET and POST, but also PUT and DELETE. The result is often XML or JSON but can often depend on the HTTP content-type negotiation mechanisms
  - JSON-RPC: Access a single web URL, passing a JSON string and The return value would also be in JSON
  - XML-RPC: This standard is just like JSON-RPC but uses XML instead of JSON.
  - Simple Object Access Protocol (SOAP): : This standard uses XML for transport like XML-RPC but provides additional layers of functionality, which may be useful for very complicated systems

REAL-TIME REACTIONS

- A traditional sort of API, where you make an HTTP request to the server and receive a response
- This method has some disadvantages if you want a very responsive system. To establish an HTTP request requires several round-trips to the server
- The IoT devices are constrained devices with low configuration, hence the HTTP method is complex once to execute
- The solution for above concern is POLLING and COMET
- **POLLING**
- The above situation can be handled using HTTP API requests was to make requests at regular intervals. This is called polling.
- In polling process
  - The server: If the device takes off, and there are thousands of devices, each of them polling regularly, you will have to scale up to that load.
  - The client: This is especially important if, as per the earlier Arduino example, the microcontroller blocks during each connect

- **COMET**
- Comet is an umbrella name for a set of technologies developed to get around the inefficiencies of polling
- Various types of Polling mechanisms are:
  1. Long Polling (Unidirectional) : the client making a polling request, long poll waits until there is some response from the server. This means that the server must regularly send a keep-alive to the client to prevent the Internet of Things device, and it is unidirectional
  2. Multipart XMLHttpRequest (MXHR) (Unidirectional) : which allows the server to send subsequent versions of a document via XHR so that the client can able to receive multiple messages from the server.
  3. HTML5 WebSockets (Bidirectional) : the API used to talk directly to the TCP layer is known as the sockets API. When the web community was looking to provide similar capabilities at the HTTP layer, they called the solution WebSockets. WebSockets have the benefit of being bidirectional. After a socket is established, the timer can simply send information down it about tasks being started, modified, or cancelled, and can read information about changes made in software, too.

OTHER PROTOCOLS

- Apart from HTTP there are many protocols better suited to Internet of Things applications.
- **MESSAGE QUEUE TELEMETRY TRANSPORT (MQTT)**
- MQTT (http://mqtt.org) is a lightweight messaging protocol, designed specifically for scenarios where network bandwidth is limited or a small code footprint is desired.
- Rather than the client/server model of HTTP, MQTT uses a publish/ subscribe mechanism for exchanging messages via a message broker
- senders publish messages to a specific topic on the message broker. Recipients subscribe to whichever topics interest them, and whenever a new message is published on that topic, the message broker delivers it to all interested recipients.
- This makes it much easier to do one-to-many messaging
- A sister protocol, MQTT for Sensors (MQTT-S), is also available for extremely constrained platforms or networks, it allows MQTT's reach to extend to sensor networks such as ZigBee.

- **EXTENSIBLE MESSAGING AND PRESENCE PROTOCOL (XMPP)**
- XMPP is a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.
- it uses XML to format the messages. This choice of XML makes the messaging relatively define in many forms
- **CONSTRAINED APPLICATION PROTOCOL (CoAP)**
- The Constrained Application Protocol (CoAP) is designed to solve the same classes of problems as HTTP but, like MQTT-S, for networks without TCP.
- There are proposals for running CoAP over UDP, SMS mobile phone messaging, and integration with 6LoWPAN.

BUSINESS MODELS and MANUFACTRING

IOT UNIT 4

BUSINESS MODELS

A SHORT HISTORY OF BUSINESS MODELS

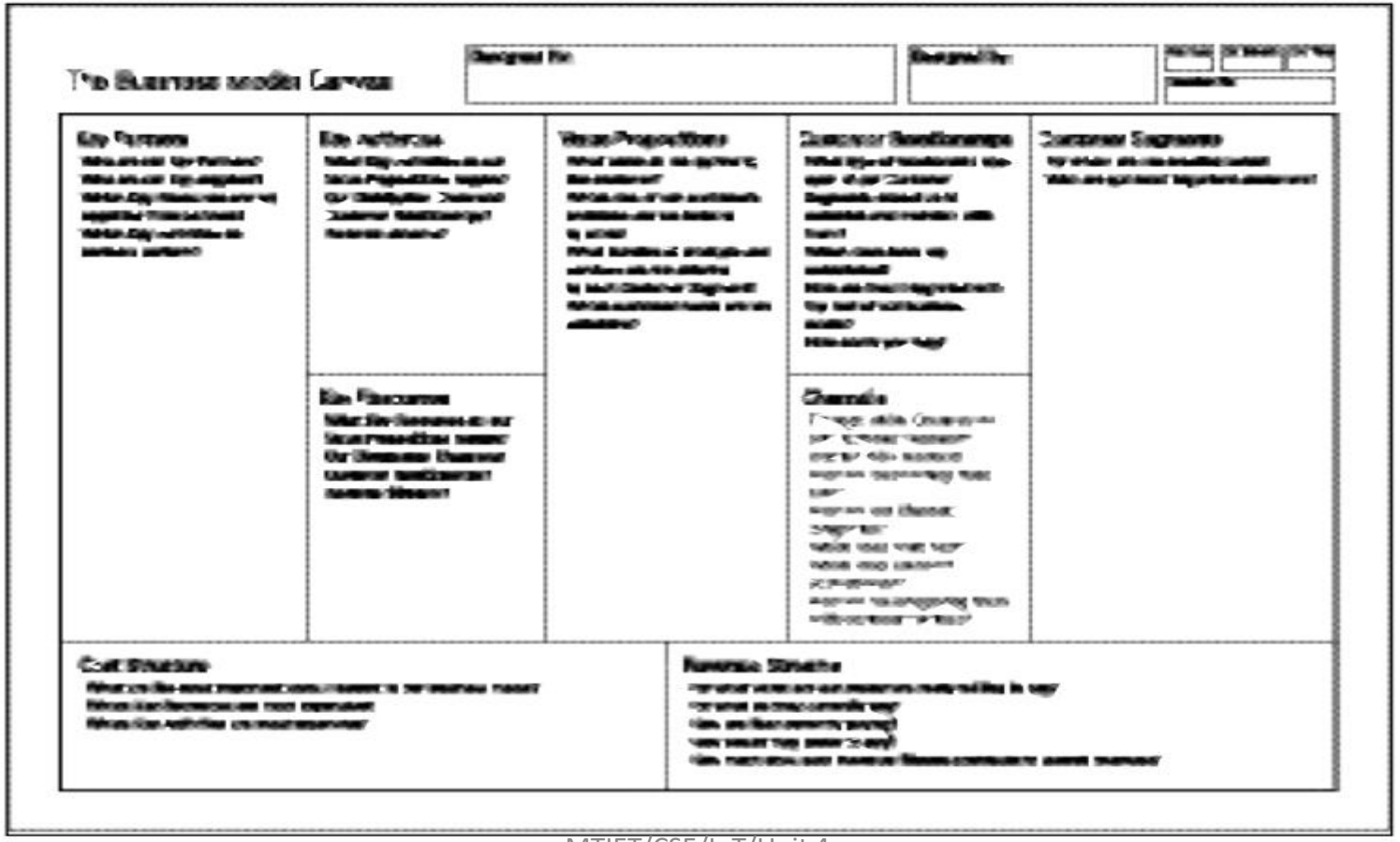
- Business makes money, But there is more to a business model than just money
- what customers want ?, how they want it?, and how an enterprise can organize to best meet those needs?
- **SPACE AND TIME**
- the technology required to move products through space and time.
- In olden days it takes more space and time to transport the goods from one place to another place
- **FROM CRAFT TO MASS PRODUCTION**
- Business has developed from craft to developing mass products
- **In 1450** Gutenberg printing press invention the priceless **handcrafted** books changed to a commodity that could be produced
- Gutenberg Bible, the first mass-produced book.
- In 1884, the British company Lever Brothers launched Sunlight Soap, the first household soap packaged in bars and branded with a logo. This was an innovation in mass production.
- Mass production, perfected by Ford Motor Company, was another major change in business model
- The fact that mass production also drove down the costs to produce these cars also helped keep them affordable
- Toyota is, for example, able to produce a single car of a given colour, configuration of wheels, and so on,

- In 1930 mass production resulted in new business models such as supermarkets, which provides both “self-service shopping” and the sale of a whole range of products under one roof.
- Fast-food franchising began in the 1930s
- In 1950 McDonald’s and Burger King started Standardized menus, pre-prepared ingredients so that you could now eat exactly the same meal in any of a chain restaurant’s stores in your country
- **THE LONG TAIL OF THE INTERNET**
- Huge changes in business practice due to the technological changes
- One of the greatest technological paradigm shifts in the twentieth century was the Internet. From Tim Berners-Lee’s first demonstration of the World Wide Web in 1990
- **In 1995,** eBay and Amazon to open up shop using and internet and started purchasing products
- Chris Anderson of Wired magazine popularized the phrase “long tail” to explain the mechanism behind the shift.
- Long tail Internet giants help by aggregating products from smaller providers, as with Amazon Marketplace or eBay’s sellers
- Newer business models have been created as when Google overturned the world of search engines

- **LEARNING FROM HISTORY**
- From the history of business model what have we learnt that we could apply to an Internet of Things project that we want to turn into a profitable business?
- We have to observe few points from past business models
  - First, we've seen that some models are ancient, such as Make Thing Then Sell It
  - Second, we've seen how new technologies have inspired new business models
  - Third, although there are recurring (designed from previous) patterns and common models, there are countless variations.
  - Finally, new business models have the power to change the world

THE BUSINESS MODEL CANVAS

- One of the most popular templates for working on a business model is the Business Model Canvas by Alexander Osterwalder and his startup
- The canvas is a Creative Commons–licensed single-page planner



The Business Model Canvas.

- Canvas layout gives a meaning and context to each item
- In the canvas, each box is simply an element in a form and 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
- This canvas contains following content like
- Revenue Streams : (bottom right box)Describes about “how are you going to make money?”
- Value Propositions (central box) : Describes about “what you will be producing?”
  - Ex: our Internet of Things project produce either product, service, or platform.
- Customer Segments: the people you plan to deliver the product
- Customer Relationships :might involve a lasting communication between the company and its most passionate customers via social media.
- Channels : are the ways of reaching the customer segments for advertising and distributing the products
- Key Activities : are the things that need to be done
- Key Resources: include the raw materials that you need to create the product
- Key Partners: businesses that are better placed to supply
- Cost Structure : requires you to put a price on the resources and activities

WHO IS THE BUSINESS MODEL FOR?

- the reason to model a business is to have some kind of information about whether it might deliver what you want from it?
- For any start ups, The model is also useful if you want to get other people involved
- The people who are planning to start a business must be able to anser the following questions from the customers
  - 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?
  - 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

- the Business Model Canvas as a tool for generating and analysing models.
- some of the models that Internet of Things companies have used or might use are
- **MAKE THING, SELL THING**
- The simplest category of models in IoT is, “make a Thing and sell it,”
- Many small-scale projects take the option of selling the product in “kit” form, with some assembly required.
- **SUBSCRIPTIONS**
- A subscription model might be appropriate, allowing you to recoup these costs (development, maintenance of servers, hosting costs)and possibly make ongoing profit by charging fees for your service
- In this model, a smaller or larger part of your product is free, while the users are also encouraged to pay a premium to get additional features

- **CUSTOMISATION**
- In this model the IoT product would be customized based on the customer requirements
- Many Internet of Things products have some possibility of customisation
- **BE A KEY RESOURCE**
- In this kind of business, you are positioning yourself as a “key resource” or a “partner” in somebody else’s business model. (like a consultancy)
- **PROVIDE INFRASTRUCTURE: SENSOR NETWORKS**
- Sensor data is important for the Internet of Things
- In this model third-party data sensor platforms will be provide to perform actions related to the sensor
- **TAKE A PERCENTAGE**
- In this model of sensor networks, if the value of the data gathered exceeds the cost of the physical sensor device, you might be able to provide that physical product for free.
- You could also link devices to advertising to reduce the price

FUNDING AN INTERNET OF THINGS STARTUP

- If you have enough personal money to concentrate on your new Internet of Things startup then no problem
- For a new startup, The problem of how to get initial funding ? is a critical one
- Making sure that you don’t need to spend huge amounts on the startup is key
- Many people try to combine a startup with a consultancy business, planning to take short
- **HOBBY PROJECTS AND OPEN SOURCE**
- If your project is also your hobby, you may have no extra costs than what you would spend anyway 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.
- After you have open-sourced a project, you can’t close-source it again

- 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
- **VENTURE CAPITAL**
- getting funding for a project from an external investor
- Startups often concentrate their fundraising activities into rounds
- friends, family, and fools (FFF) round. This stage may be the one in which you've contributed your life savings
- angel round: angels are usually individual investors, often entrepreneurs themselves, who are willing to fund some early-stage startups which a more formal investor
- venture capital (VC) round is similar, but instead of your courting individual investors, the investor is a larger group with significant funds, whose sole purpose is to discover and fund new companies with a view to making significant profit.

- venture capital (VC) are also known as accelerators
- Current accelerators that may be specialized in the Internet of Things (in US)
  - HAXLR8R (http://haxlr8r.com/program/hack-what)
  - PCH Accelerator (www.pchintl.com/accelerator/ accelerator.aspx)
  - Berlin Hardware Accelerator (http://www.berlinhardware accelerator.com/)
  - Bolt (http://www.bolt.io)
  - Lemnos Labs (http://lemnoslabs.com)
  - Y Combinator (http://ycombinator.com)
- **GOVERNMENT FUNDING**
- 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
- **CROWDFUNDING**
- Getting many people to contribute to a project
- the main options for crowdfunding are Kickstarter (www. kickstarter.com) and Indiegogo (www.indiegogo.com).

LEAN STARTUPS

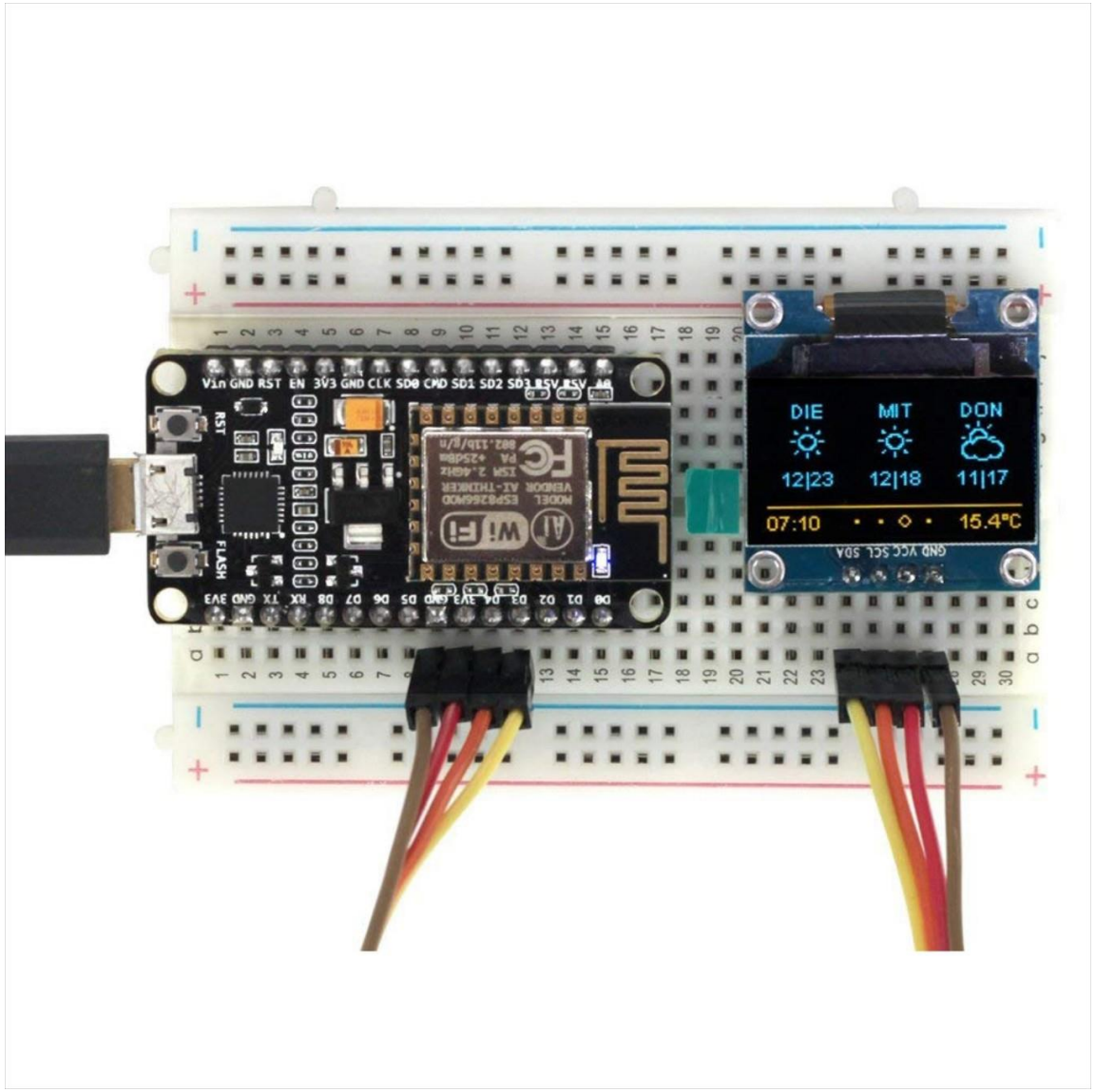
- running a startup on a low budge
- The concept of a “lean startup,” pioneered by Silicon Valley entrepreneur Eric Ries, springs
- The characteristics of LEAN startups are as follows
  - Zoom-in pivot: Focus on what was only a part of the value proposition, and turn that into the whole Minimum Viable Product.
  - Customer segment pivot: Realise that the people who will actually buy your product aren’t the ones you were originally targeting. While you can continue to make exactly the same product, you have been marketing it to the wrong people.
  - Technology pivot: Accomplish the same goals as before, but change the implementation details. While prototyping will almost certainly involve many changes in technology while you establish the best way to make the product from an engineering perspective, this pivot would be a business decision, made to improve manufacturing costs, speed, or quality

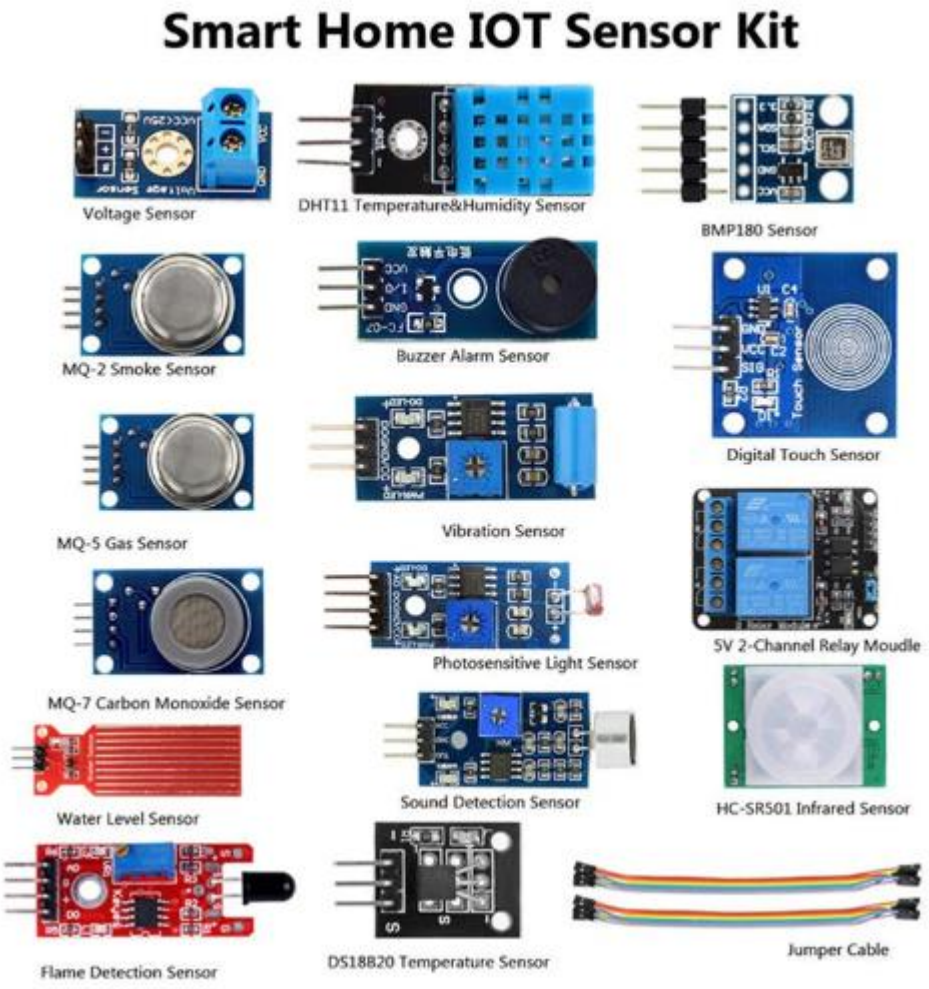
MANUFACTURING

WHAT ARE YOU PRODUCING?

- We can generally split our choices of production into three categories:
  - A kit that your customers can assemble themselves
  - A ready-made electronics board which users can use as a sub-assembly in their own projects
  - A fully fledged product, complete with its housing, instructions, and even packaging, just waiting to be put on the shelf at your local department store
- **DESIGNING KITS**
- The first step towards selling your idea as a product is to provide it as a kit
- Most kits tend to provide only the electronics and software for a particular application
- However, with the growing accessibility of 3D printers and laser-cutters, it is becoming vastly more feasible to provide housing and other physical components even for kits.
- The Instructible website (www.instructables.com) has a number of guides to help

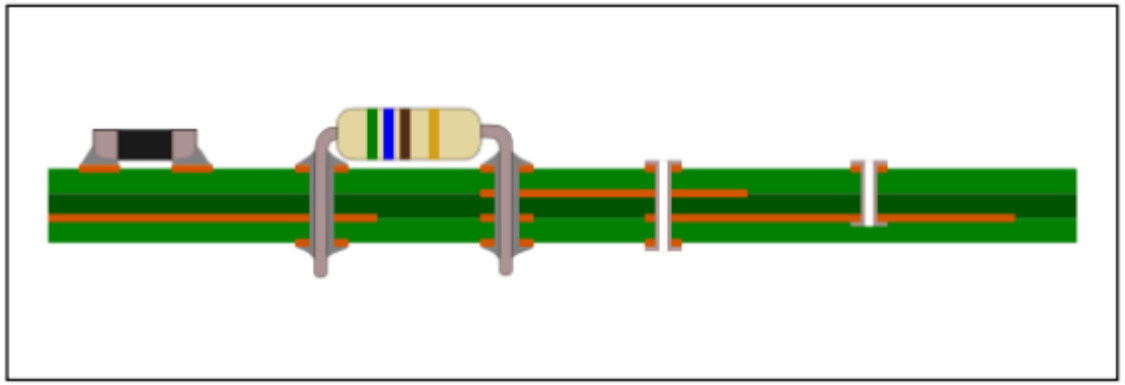
- Once the prototype of a kit is designed and it is working well, next step is making that prototype in this form of KIT using PCBs
- A kit may contain assembled PCBs and populated with all the components.
- Once the PCB is ready to price it we have to create a spread sheet which must contain the list of every single electronic component, connector, cable, PCB, case, and so on, This list is called the bill of materials (BOM)
- you want to sell many of your kits, the most important cost to drive down is that of the BOM
- the final step from kit to consumer product is to manufacture and assemble the housings, linkages, and whatever else is part of the finished device


DIE
MIT
DON
12|23
12|18
11|17
07:10
15.4°C


Smart Home IOT Sensor Kit
Voltage Sensor
DHT11 Temperature&Humidity Sensor
BMP180 Sensor
MQ-2 Smoke Sensor
Buzzer Alarm Sensor
Digital Touch Sensor
MQ-5 Gas Sensor
Vibration Sensor
5V 2-Channel Relay Moudle
Photosensitive Light Sensor
MQ-7 Carbon Monoxide Sensor
Water Level Sensor
Sound Detection Sensor
HC-SR501 Infrared Sensor
Flame Detection Sensor
DS18B20 Temperature Sensor
Jumper Cable

DESIGNING PRINTED CIRCUIT BOARDS

- prototype a bundle of components and wires plugged into a number of breadboards.
- The next step is soldered it up onto protoboard or stripboard and make the prototype more robust with permanent connections
- The multilayer PCB will let you cross connections, so that we can connect other component to the PCB to do IoT application
- The PCB is made up of a number of layers of fibreglass and copper, sandwiched together into the board, the fiberglass will act as insulator to avoid current shock
- Single-sided boards have only one layer of copper and Double-sided boards, have two layers of copper: one on the top and one on the bottom
- Three- or five-layer boards aren’t uncommon, and even some seven-layer boards are used for really complicated circuits

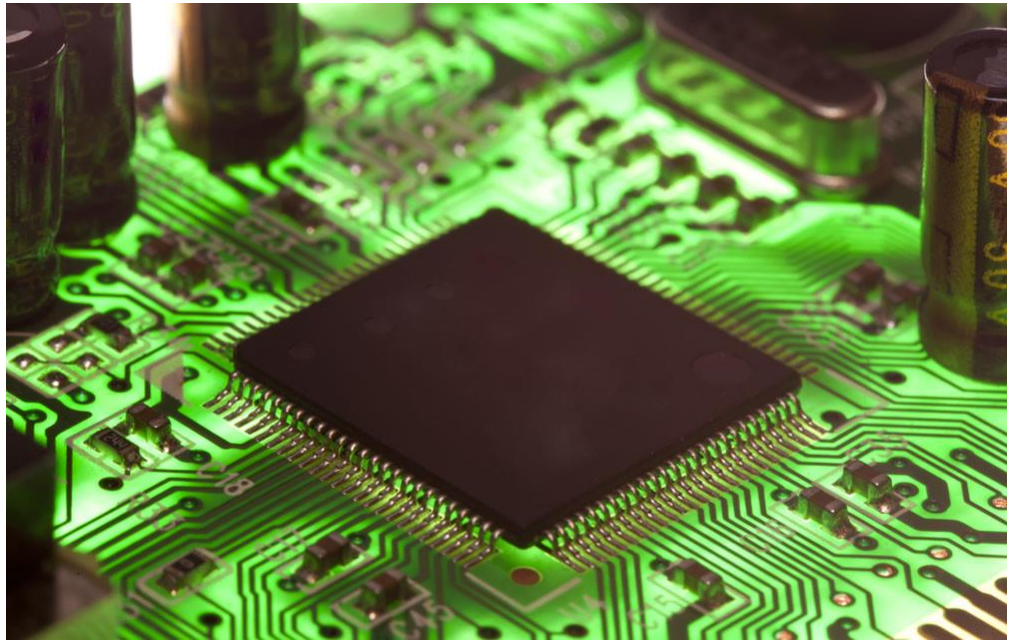


PCB features (left to right): surface-mount component; through-hole component; via; blind via.

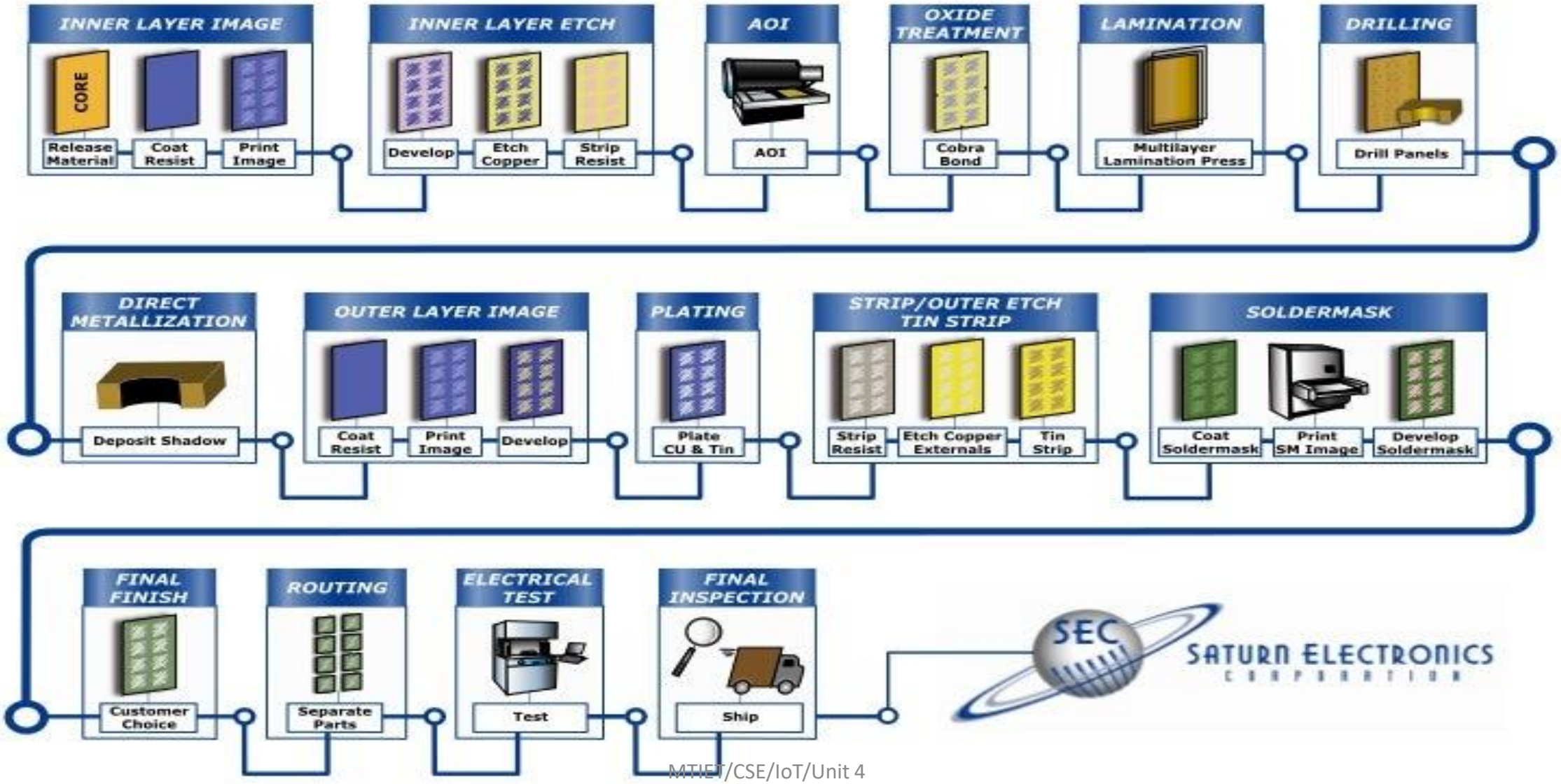
- most of the layer of the board to that connection and just leave it as a filled area of copper. This is known as a plane
- These planes consists of electromagnetic signal lines
- The surfaces of professionally manufactured PCBs undergo processes to apply two other finishes which make them easier to use.
  - First, all the parts of the board and bare copper which aren't the places where component legs need to be soldered are covered in solder mask.
  - Then, on top of the solder mask is the silkscreen. This is a surface finish of paint applied via silkscreen printing

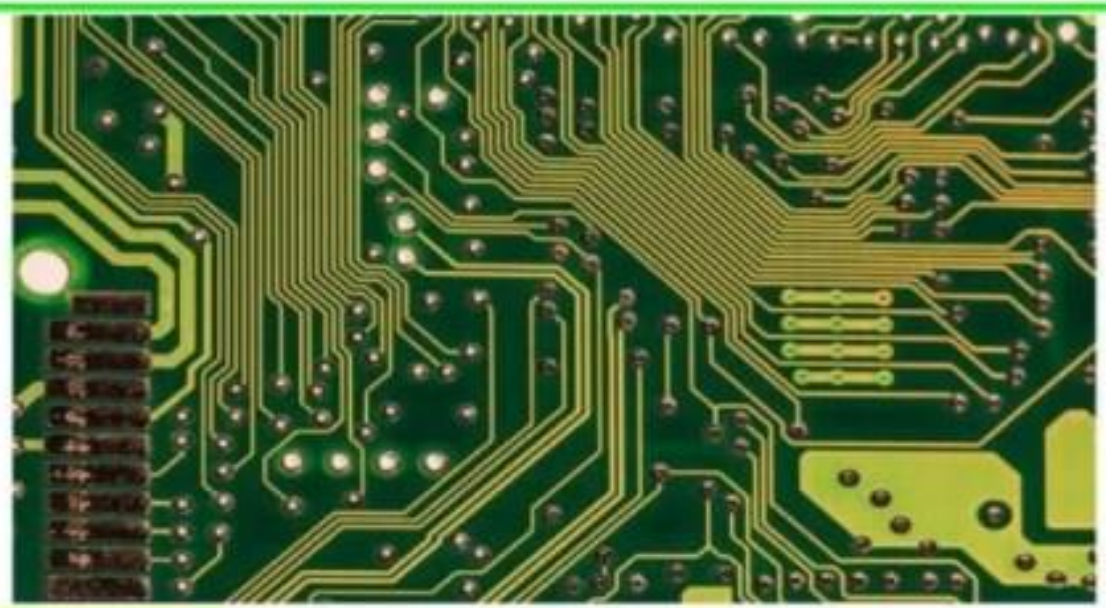
- **Steps to design printed circuit boards**
- Step-1: Patterning or Etching
  - printed circuit boards are manufactured by applying a layer of copper over the entire surface of the PCB substrate material either on one side or both sides. This creates a blank printed circuit board, with the copper everywhere on the surface. From here the unwanted copper is removed by subtractive methods.
- Step-2: Photoengraving
  - The photoengraving process uses a mask or photomask combined with chemical etching to subtract or remove the copper areas from the circuit board substrate.
- Step-3: Lamination
  - Many printed circuit boards are made up of multiple layers; these are referred to as multi-layer printed circuit boards. They consist of several thin etched boards or trace layers and are bonded together through the process of lamination
- Step-4: Drilling
  - Each layer of the printed circuit board requires the ability of one layer to connect to another, this is done through drilling small holes called “VIAS“.
  - Blind Vias: When the holes connect a layer to the outside surface
  - Buried Vias: When the holes only connect interior layers and not to the outside surface.
- Step-5: Solder Plating (Solder Resist)
  - Pads and lands which will require electronic components to be mounted on are plated to allow solderability of the components. Bare copper is not readily solderable and requires the surface to be plated with a material that facilitates soldering.

- Step-6: Silk Screen
  - When visible information needs to be applied to the board such as company logos, part numbers or instructions, silk screening is used to apply the text to the outer surface of the circuit board.
- Step-7: Testing
  - Unassembled circuit boards are subjected to a bare board test where each circuit connection is verified as correct on the finished circuit board. In high volume circuit board production, a bed of nails tester or fixture is used to make contact with the copper lands or holes on one or both sides of the board to facilitate testing



PCB Flow Chart





Green Solder Mask
Substrate with Copper Layer on Both
Green Solder Mask

Double Sided PCB

@electronicsandyou

Multilayer PCB

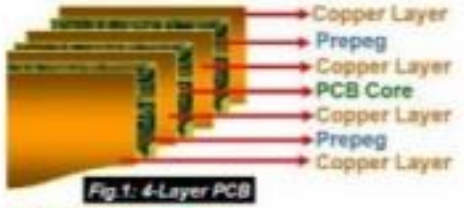
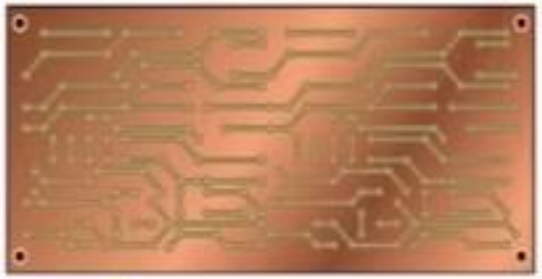
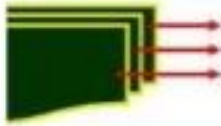


Fig.1: 4-Layer PCB

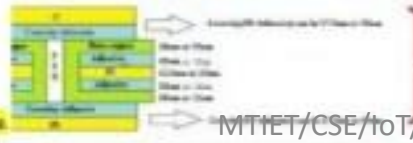




Rigid-Flex PCB

2-Layer Rigid-Flex PCB

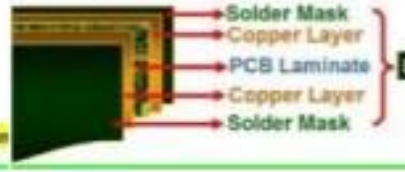
Flexible PCB



2-Layer FPC

Rigid PCB

Rigid PCB



Double Sided Rigid PCB

Manufacturing Cntd..
&
Ethics

IoT Unit 5

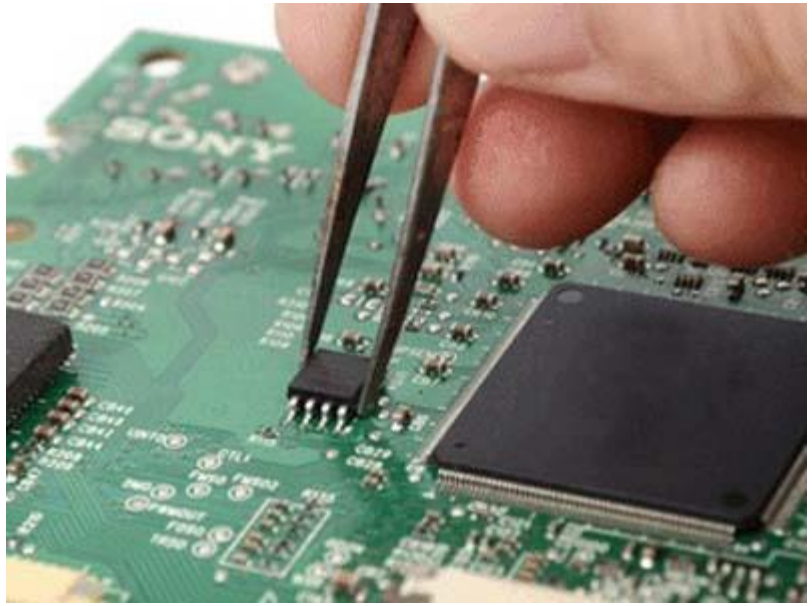
MANUFACTURING PRINTED CIRCUIT BOARDS

- If we want to manufacture more PCBs the important step is ETCHING BOARDS
- **ETCHING BOARDS**
- The most common PCB-making technique for home use is to etch the board**.** This process generally involves printing out the design from your PCB design software onto a stencil (a thin sheet used to produce the cut design on the surface below by the application)
- If you're using photo-resist board, it will be onto a stencil which masks off the relevant areas when you expose it to UV light;
- if you're using the toner-transfer method, it will be for your laser printer to print onto glossy paper ready to transfer.
- For photo-resist board, you will expose it under a bright lamp for a few minutes; and for the toner-transfer method, you'll use a super-hot iron
- Once board is prepared, you can immerse it into the etching solution, where its acidic make-up eats away the exposed copper, leaving the tracks behind.
- The last step is to drill the holes for any mounting points or through-hole components.
- We can drill the board by using hand or CNC machine (Computerized Numerical Control)

- **MILLING BOARDS**
- CNC mill machine is used to drill the holes in your PCB, and also used it to route out the copper from around the tracks themselves. To do this, you need to export the copper layers from your PCB software as Gerber files
- To translate your Gerber file into the G-code that your mill needs requires another piece of software (Line Grinder)
- The mill effectively cuts a path round the perimeter of each track to isolate it from the rest of the copper
- **THIRD-PARTY MANUFACTURING**
- We can also use THIRD-PARTY MANUFACTURING to produce more boards
- **ASSEMBLY**
- once PCBs have been manufactured, you still need to get the components soldered onto them

CNC Machine

- For assembling surface-mount boards, you need one more item from your PCB design Gerber collection: the solder paste layer
- The solder for surface-mount work comes as a paste, supplied in tubs or tubes
- Using tweezers and ideally a loupe or magnifying glass, place each component onto the relevant spot on the PCB. The paste holds the parts

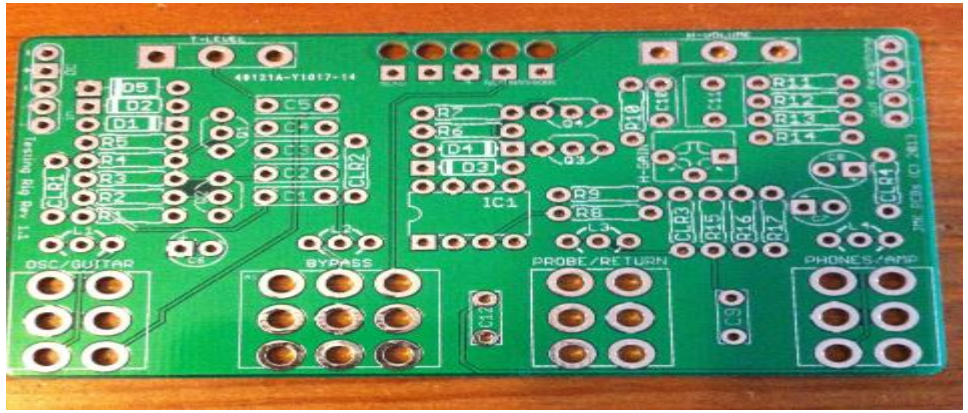


- When you have all the components on the board, you need to melt the solder to fix everything in place
- We can solder all the connections at once if you use a reflow oven.

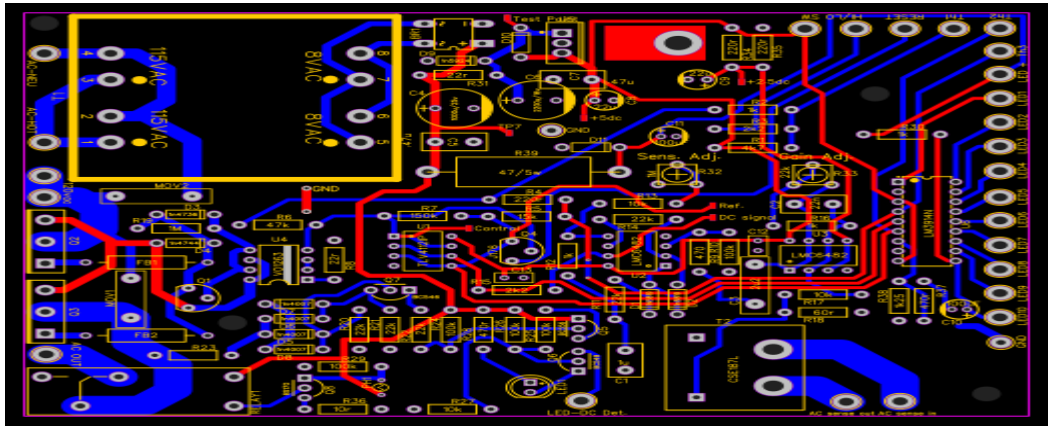
- Instead of hand assembly, we can also robots
- robots that can pick up components using a tiny vacuum nozzle, rotate and place them in the right location on the PCB, and then repeat that process at a rate of tens of thousands of components per hour. These robots are known as pick-and-place assembly machines.
- Robots are used for mass manufacturing, the components to feed into the machine are supplied in a form known as tape and reel

- **TESTING**
- Testing through the automated assembly process,
- This process had some testing steps included already. Assembly lines can include automatic optical inspection (AOI).
- In this process, a high-resolution camera inspects some aspect of the board and its components
- After the boards pass the AOI, the next step is to run them through a functional test

- the functional test just involves powering up the board as it will be used in the finished product and ensuring that it does what it is supposed to
- This functional test ensures the PCB and its components are soldered correctly, that none of the components are faulty, and that there aren't any manufacturing defects in the PCB itself.
- A better approach is to build a specific test rig to exercise the different parts of the circuit and measure the voltages at set points on the board and this process is known as device under test (DUT)
- The test program can then run through its tests and measure voltages at different pogo pins at the relevant time in the test to see how the board being tested performs

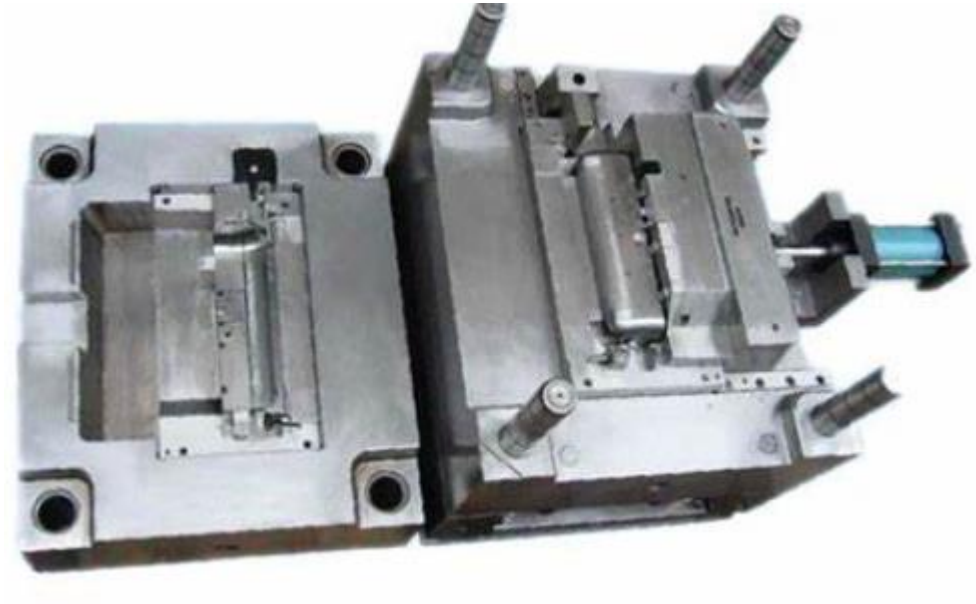


Test Rigs

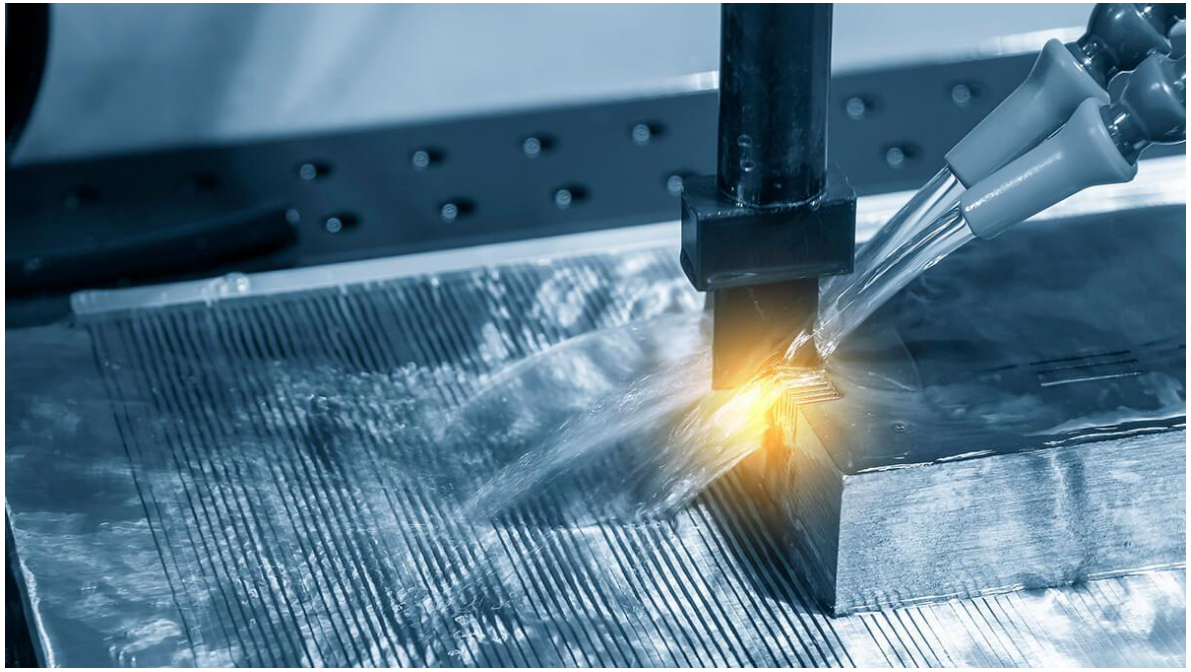


MASS-PRODUCING THE CASE AND OTHER FIXTURES

- the most common method of mass production: injection moulding of plastic
- the process involves injecting molten plastic into a mould to form the desired shape. After the plastic has cooled sufficiently, the mould is separated and the part is popped out by a number of ejection pins and falls into a collection bin
- The expensive part of injection moulding is producing the mould in the first place; this is known as tooling up



- for a super-smooth surface, the moulds are finished with a process called electrical discharge machining (EDM), which uses high-voltage sparks to the surface of the metal and gives a highly polished result
- The simplest moulds are called straight-pull and consist of the mould split into two halves.
- In a process known as multishot moulding, you can even share parts of different colours on the same mould.



CERTIFICATION

- After manufacturing the final product of IoT must be certified from regulated authorities
- The certification ensures the product is safer for use and the product doesn't emit lots of unwanted electromagnetic radiation and interfere with the correct operation of other devices
- Some of the certificate signatures which would be present in PCBs are : the CE mark for meeting European standards; FCC for US Federal Communications Commission regulations; and UL for independent testing laboratory UL's tests.
- The regulations that the device needs to pass vary depending on its exact functionality, target market (consumer, industrial, and so on), and the countries
- Testers check over the materials specifications to ensure you're not using paint containing lead and doesn't go up in flames;
- electromagnetic compatibility, or EMC, testing : This tests both how susceptible your device is to interference from other electronic devices, power surges on the main's electricity supply, and so on, and how much electromagnetic interference your product itself emits.
- The EMC testing is to overcome the problem arises when a circuit emits a sufficiently strong signal unintentionally which disrupts the desired radio frequencies

- The resultant test report of EMC test is added to your technical file
- In addition to the test report, we need to gather together PCB layouts, assembly certificates, the certificates for any precertified modules that you have used, and datasheets for critical components
- In Europe, we must also register for the Waste Electrical and Electronic Equipment Directive (WEEE Directive).
- In the UK, the Environment Agency maintains the certificates for the devices which are eco friendly
- producers and retailers of electronic and electrical products must encourage recycling in which the unwanted electronic devices must be recycled

COSTS

- The cost of the final product of IoT technology depends on the materials and the components which are used in the board or kit
- The cost would also be influenced by the mass production, ie if we are going to manufacture more the cost would be reduced
- And to get through certification, we need to spend some amount , hence this also must be included in the final cost of the product
- BOM is an optimized cost by including all the prices spent while manufacturing and certification

SCALING UP SOFTWARE

- In IOT technology along with the physical devises , software is also important
- To scale up the software we need to be familiar with the certain programming languages and technologies.
- **DEPLOYMENT**
- Copying software from a development machine (your laptop or the team's source code repository) to where it will be run from in production is typically known as deployment
- In the case of the firmware running on the microcontroller, this will be done once only, during the manufacture of the device
- The device like the Arduino which usually only runs a single program, the process will be identical to that in development.
- For a device like a BeagleBone or Raspberry Pi which runs an entire operating system, you will want to "package" the various program code, libraries, and other files
- The software for an online service will tend to run in a single location. However, it will generally be visible to the whole of the Internet

- **CORRECTNESS AND MAINTAINABILITY**
- The designed software must produce correct results with respect to all the inputs
- Testing your code before it is deployed is an important step in helping to produce correctness
- Through internet the server software is easy to update, either to fix bugs or to introduce new features (maintainability)
- In IoT, the embedded code in the device must also be updated through the internet
- **SECURITY**
- Security is an impotent issue regarding IoT software
- Following are some of the more important guidelines for security\
  - Make sure that your servers are kept up-to-date with the latest security patches, are hardened with the appropriate firewalls, and detect and mitigate against password hacking attempts and rootkit attacks
  - User passwords should never be stored in plain text
  - Never simply trust user input. Check that anything that is entered into a web application fits the type of data you expect, and refuse or clean anything which doesn't.
  - Be aware of cross-site request forgery (CSRF) attacks from other malicious or compromised websites

- **PERFORMANCE**
- when considering scaling up software is whether it will be fast enough and handle a large number of users **(performance)**
- If your web service is running on a modern framework, it should be easy to scale up by deploying the code onto a more powerful machine
- To enhance the performance of any software optimize it using following algorithm
  1. Identify that you actually have a problem.
  2. Measure and profile the tasks which are slow and identify the problem.
  3. Fix that problem. The web community does a good job in sharing best practice in how to address such problems, so you often find that others have trodden the path before and documented their tried-and-tested solutions
- **USER COMMUNITY**
- Any IoT project will be successful only if people actually use it
- The designers should think about the user experience
- does our website have documentation, introductory videos, and tutorials?
- , some form of forum, mailing list, or chat room lets users support each other, which reduces your workload, but more importantly helps to build up a community and expertise around the product.

ETHICS

- The technology will have pros and cons on human life, some of the points which will give negative effects are
  - Destroys jobs
  - Is intrusive and will enslave us to technology
  - Disconnects humanity from ancient traditions
  - Encourages us to become lazy and unhealthy
  - Tempts us into thinking we are like gods
- The address all these cons of technology some body expects some rules or ethics about regarding technology development and its utilization.
- To give awareness on ethics there are many traditional disciplines, there are courses on ethics—(for example, in university courses on engineering or computer science.)

CHARACTERIZING THE INTERNET OF THINGS

- what the Internet of Things is, to get a handle on what particular changes it can bring to humanity's relationship to the "good life"?
- This availability of IoT technology brings certain abilities to society and for individual
- the Internet of Things are also primarily related to communication, but now allow the publication and transmission of vast streams of data, from the social to the environmental
- The IoT technology applies primarily to the physical usage of the "Thing", its affordances, sensors, and actuators, and its digital communications. And it provides potentially unexpected capabilities for the thing of IoT
- Connecting the Internet to the real world allows both your physical actions to be made public
- Applying the bidirectional communication to Things can lead to features that interact with the concept of privacy
- the Internet of Things is made up of
  - Physical object + controllers, sensors, and actuators + Internet service
  - Each of these aspects has a part to play in the ethical issues specific to the Internet of Things

PRIVACY

- The Internet, as a massive open publishing platform, has been a disruptive force as regards the concept of privacy.
- Everything we write might be visible to anyone online
- Some important aspects regarding data privacy in internet
  - we may not want your data being visible to an abusive ex-spouse.
  - we might be at risk of assassination by criminal, terrorist, or state organizations.
  - we might belong to a group which is targeted by your state (religion, political party, journalists).
- IoT privacy refers to the issue where the manufacturer of the device has the ability to monitor and to use the data that they get from these devices
- The ability to limit privacy is necessary to establish trust with an IoT system device.
- Questions to Consider for IoT Data Privacy
  - Is my IoT device collecting personal data from others? If yes then what kind of data is it?
  - Where is the data being sent?
  - How is this data being used?
  - Who has access to the data?
  - Where is the data being stored?
  - How long is this data being kept?
  - Should we build a time frame for data expiration? Should the data be deleted after the time frame ends?
  - Is the data secure during transfer? How secure is it?
  - How will the data breaches be handled and which entities will be informed in the case of a data breach?

- **IOT SECURITY AND PRIVACY CONCERNS**
- Even though IoT is growing rapidly, it has some security and privacy concerns.
- SECURITY RISKS:
- IoT devices are connected to the internet and laptops/computers. Lack of security can transmit the user's personal information to IoT devices.
- IoT devices are interconnected to the consumer's internet and if some vulnerability occurs in the device, it can harm the entire infrastructure.
- Unauthorized access to the internet can exploit a security vulnerability and create potential risks.
- PRIVACY CONCERNS:
- As mentioned earlier, IoT devices are interconnected with each other over the same network, so the chances of sensitive information leaking through unauthorized access are present.
- Data is not encrypted during transfer from one end to another.
- Personal information such as usernames, passwords, date of birth, address, health care information, and credit card credentials can be viewed by others.

- SECURITY ISSUES OF IOT DEVICES

There are some issues that can compromise IoT security:

- Unpatched vulnerabilities.
- Weak authentication.
- Vulnerable APIs.
- Use of outdated software and application.
- Weak login credentials.

- TIPS TO ENSURE THE SECURITY OF IOT DEVICES

The following measures can help in ensuring the security of IoT devices:

- Being vigilante on smartphones and tablets. Use a strong password so that bad actors don’t access your private information.
- Regularly checks for antivirus updates.
- Using strong login credentials. Don’t use a generic username and password.
- Deploying an end-to-end encryption model.
- Making sure that mobile devices and their operating systems software are up to date.
- Avoiding the sharing of credentials with others.
- Consulting an expert when needed.

CONTROL

- While the technology itself doesn't cause any controlling behaviour, it could easily be applied by other things
- As with questions about privacy, there are almost always good reasons for giving up some control.
- Companies have the expertise and the technology to interact with the Internet, hence for better results some control should require
- **DISRUPTING CONTROL**
- When we refer to a technology as "disruptive", we mean that it affects the balance of power.
- one of the fears about the Internet of Things is that it will transfer power away from the citizens, the subjects of technology
- One extreme example of this would be how surveillance and fears of the Big Brother state (a TV program, CCTV cameras, remote-controlled helicopter-drones) might be mitigated by "sousveillance".

- **CROWDSOURCING**
- One fascinating feature of modern Internet life is “crowdsourcing”, from knowledge (Wikipedia, et al.) to funding projects to work
- When getting the information or knowledge from IoT projects five critical requirements for a sensor commons project.
  - Gain trust:Trust is largely about the way that an activist project handles itself beyond the seemingly neutral measurements
  - Become dispersible: Becoming dispersible means spreading the sensors throughout the community
  - Be highly visible: Being visible involves explaining why the project’s sensors are occupying a public space.
  - Be entirely open: Being open is perhaps what distinguishes the sensor commons from a government project the most
  - Be upgradable: Finally, the project should be designed to be upgradable, to enable the network to remain useful as the needs change or hardware gets to the end of its working life

ENVIRONMENT

- The important environmental concerns about the production and running of the Thing itself.
- **PHYSICAL THING**
- We need to consider environmental factors, such as emissions produced during normal operation or during disposal of the object.
- Nowadays, most consumer electronics do indeed conform to it; health benefits are obtained both at the point of manufacture and at waste disposal/recycling
- **ELECTRONICS**
- Buying PCBs locally or from a foreign manufacturer affects the carbon cost of shipping the completed units.
- many electronic components rely on "rare earth minerals" (REMs) which have been extracted in China or from other locations worldwide, hence the mining process must be managed properly
- Refining the electronics involves the use of toxic acids
- **INTERNET SERVICE**
- In the digital world, moving data rather than physical objects is faster, is safer, and has a lower environmental cost.
- As well as the cost of transferring the data across the Internet, running your own web server uses power. Many server hosting specialists now offer carbon-neutral hosting, where you pay extra to offset your emissions.

SOLUTIONS

- An instrumented Internet of Things device does seem to use vastly more resources in its production, daily use, and waste disposal
- the number of Internet-connected devices will be exploding in the coming years is spurring massive research into low-power efficient chips and communications.
- **THE INTERNET OF THINGS AS PART OF THE SOLUTION**
- If community-led sensor networks can help supplement government and international science measurements, then we should be doing everything we can to help.
- Instrumenting production lines, home energy usage, transport costs, building energy efficiency, and all other sources of efficiency might seem extreme, but it may be a vital, imperative task.
- Other technologies which aren't principally linked with the Internet of Things will also be important.
- Projects such as a carbon score for every company in the UK will help change attitudes, perhaps simply by gamifying the process of improving one's emissions
- —collective sensor networks and massive business process engineering not for profit but for environmental benefits

- When designing the Internet of Things, or perhaps when designing anything, you have to remember two contrasting points:
  - Everyone is not you. Though you might not personally care about privacy or flood levels caused by global warming, they may be critical concerns for other people in different situations.
  - You are not special. If something matters to you, then perhaps it matters to other people too.
- **THE OPEN INTERNET OF THINGS DEFINITION**
- We can summarize the main goals of the definition as follows
- Accessibility of data: As a stated goal, all open data feeds should have an API which is free to use, both monetarily and unrestricted by proprietary technologies with no alternative open source implementation.
- Preservation of privacy: The Data Subjects should know what data will be collected about them and be able to decide to consent or not to that data collection. This is a very strong provision (and most likely unworkable for data which is inherently anonymous in the first instance) but one which would provide real individual protection if it were widely followed. As with any information gathering, “reasonable efforts” should be made to retain privacy and confidentiality.
- Transparency of process: Data Subjects should be made aware of their rights—for example, the fact that the data has a licence—and that they are able to grant or withdraw consent