IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 1, January 2021
DOI: 10.17148/IARJSET.2021.8108

# Integration of Mechanical Systems with IoT

**Md Rakibul Islam**[1], **Sifat Hasan**[2]
Department of Mechatronics Engineering, CUMT[1]
Department of Mechatronics Engineering, CUMT[2]

**Abstract**: This research paper presents a holistic approach towards integrating mechanical systems with IoT, unlike Industrial Internet of Things(IIoT) which connects every system and process of a industry to data collection, this research paper will give a brief to install IoT on Mechanical systems for small and micro businesses.

**Keywords**: IoT, IIoT, Mechanical Systems, Manufacturing Industry.

## I. INTRODUCTION

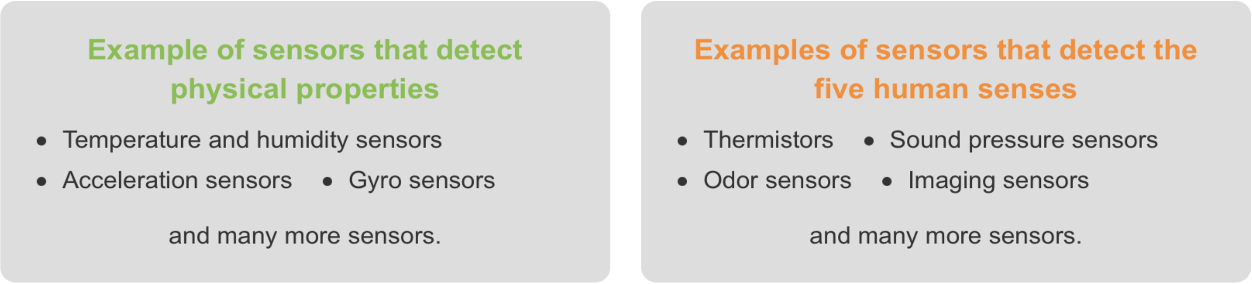
Globally there are millions of small and micro mechanical workshops which uses Lathe machine, Turner, Drill machine, CNC(Computer numerical Control) and VMC(Vertical machining Centre). This machines are costly and their maintenance costs another hefty amount on the owners, thus resulting into more downtime and costlier maintenance which directly effects on its efficiency. A scheduled maintenance is a viable option that will check these systems, in order to conduct scheduled maintenance the systems must be turned off at every appointment thus resulting in more downtime. A viable and more cost effective way of predictive maintenance model must be implemented with help of IoT.

With the advent of internet of things and sophisticated sensors, the data collection of any system has become accessible with greater accuracy and reliability. The forecast of IoT is expected to grow to $1.6 Trillion by 2025. These devices often share data and information in order to provide added convenience and control to consumers and, in some cases, even allow users automate simple processes such as ordering supplies. Tens of billions of these IoT connected devices already exist around the world and this number will only grow as internet connectivity begins to become a standard feature for a great number of electronics devices. Although heavily integrated into the consumer electronics market, IoT extends far beyond handheld devices and home appliances; IoT subsystems such as industrial internet and connected cities aim at automating factories and urban areas rather than just households. Digital virtual assistants such as Amazon's Alexa and Google Assistant serve as the bridge between this network of interconnected devices and there human users.

## II. SENSORS

IoT innovation made it possible to connect everyday things to the internet. Nowadays, almost all entities, such as houses, office buildings, factories, and even cities are connected to the network to collect data and utilise the information for various purposes.The importance of data has increased greatly, with many experts saying "data is the new oil." Sensors play an important role in creating solutions using IoT. Sensors are devices that detect external information, replacing it with a signal that humans and machines can distinguish. Sensors made it possible to collect data in most any situation and are now used in various fields — medical care, nursing care, industrial, logistics, transportation, agriculture, disaster prevention, tourism, regional businesses and many more. With the expansion of the fields in which sensors play an important role, the market is still growing with a variety of sensors.

There are a wide range of IoT sensors used to detect and measure various physical phenomena such as heat and pressure as well as the five human senses: sight, hearing, touch, taste and smell.



**Example of sensors that detect physical properties**

- Temperature and humidity sensors
- Acceleration sensors
- Gyro sensors

and many more sensors.

**Examples of sensors that detect the five human senses**

- Thermistors
- Sound pressure sensors
- Odor sensors
- Imaging sensors

and many more sensors.

**Micro Electromechanical Sensors:** A MEMS (micro-electromechanical system) is a miniature machine that has both mechanical and electronic components. The physical dimension of a MEMS can range from several millimetres to less than one micrometer, a dimension many times smaller than the width of a human hair.

The label MEMS is used to describe both a category of micromechatronic devices and the processes used when manufacturing them. Some MEMS do not even have mechanical parts, yet they are classified as MEMS because they miniaturize structures used in conventional machinery, such as springs, channels, cavities, holes and membranes. Because some MEMS devices convert a measured mechanical signal into an electrical or optical signal, they may also be referred to as transducers. In Japan, MEMS are more commonly known as micromachines, and in European countries, MEMS are more commonly referred to as microsystems technology (MST).

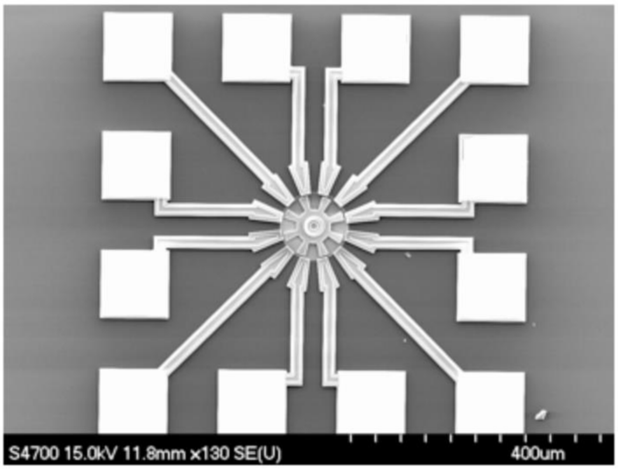
The small system on a chip (SOC) that automatically adjusts screen orientation on a smartphone is an example of a MEMS many people interact with each day. As MEMS become smaller, require less power and are less expensive to manufacture, they are expected to play an important part in the wireless internet of things (IoT) and home automation.

Other commercial applications of MEMS include:

- Sensor-driven heating and cooling systems for building management systems.
- Micro-mirror arrays for high definition projection systems.
- Smart dust for the detection of environmental changes in molecular manufacturing (nanotechnology) clean rooms.
- Micronozzles to control the flow of ink in inkjet printers.
- Tiny gyroscopes, barometers, accelerometers and microphones to support Mobile apps.
- Disposable pressure sensors for use in healthcare.
- Optical switching devices that allow one optical signal to control another optical signal.

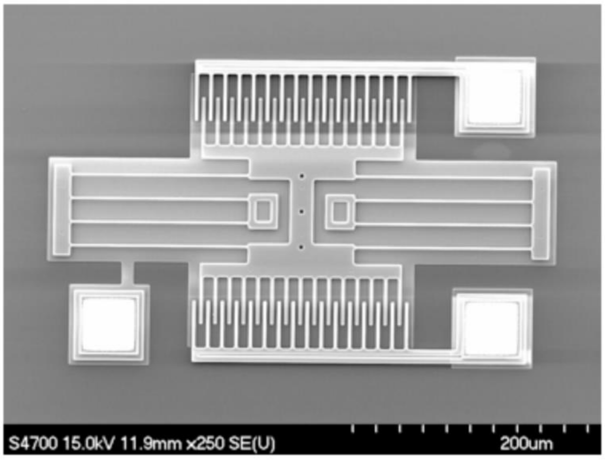
More recently, the MEMS research and development community has demonstrated a number of microactuators including: microvalves for control of gas and liquid flows; optical switches and mirrors to redirect or modulate light beams; independently controlled micromirror arrays for displays, microresonators for a number of different applications, micropumps to develop positive fluid pressures, microflaps to modulate airstreams on airfoils, as well as many others. Surprisingly, even though these microactuators are extremely small, they frequently can cause effects at the macroscale level; that is, these tiny actuators can perform mechanical feats far larger than their size would imply. For example, researchers have placed small microactuators on the leading edge of airfoils of an aircraft and have been able to steer the aircraft using only these microminiaturized devices.

The real potential of MEMS starts to become fulfilled when these miniaturized sensors, actuators, and structures can all be merged onto a common silicon substrate along with integrated circuits (i.e., microelectronics). While the electronics are fabricated using integrated circuit (IC) process sequences (e.g., CMOS, Bipolar, or BICMOS processes), the micromechanical components are fabricated using compatible "micromachining" processes that selectively etch away parts of the silicon wafer or add new structural layers to form the mechanical and electromechanical devices. It is even more interesting if MEMS can be merged not only with microelectronics, but with other technologies such as photonics, nanotechnology, etc. This is sometimes called “heterogeneous integration.” Clearly, these technologies are filled with numerous commercial market opportunities.



A surface micromachined electro-statically-actuated micromotor fabricated by the MNX. This device is an example of a MEMS-based microactuator.

While more complex levels of integration are the future trend of MEMS technology, the present state-of-the-art is more modest and usually involves a single discrete microsensor, a single discrete microactuator, a single microsensor integrated with electronics, a multiplicity of essentially identical microsensors integrated with electronics, a single microactuator integrated with electronics, or a multiplicity of essentially identical microactuators integrated with electronics. Nevertheless, as MEMS fabrication methods advance, the promise is an enormous design freedom wherein any type of microsensor and any type of microactuator can be merged with microelectronics as well as photonics, nanotechnology, etc., onto a single substrate.



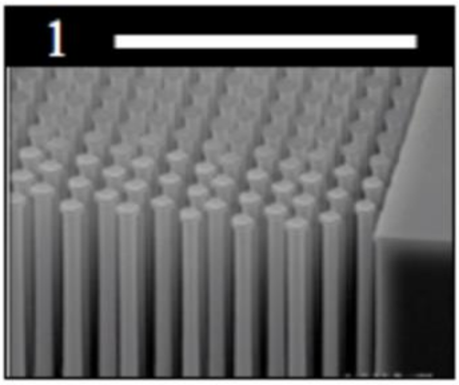
A surface micromachined resonator fabricated by the MNX. This device can be used as both a microsensor as well as a microactuator.

This vision of MEMS whereby microsensors, microactuators and microelectronics and other technologies, can be integrated onto a single microchip is expected to be one of the most important technological breakthroughs of the future. This will enable the development of smart products by augmenting the computational ability of microelectronics with the perception and control capabilities of microsensors and microactuators. Microelectronic integrated circuits can be thought of as the "brains" of a system and MEMS augments this decision-making capability with "eyes" and "arms", to allow microsystems to sense and control the environment. Sensors gather information from the environment through measuring mechanical, thermal, biological, chemical, optical, and magnetic phenomena. The electronics then process the information derived from the sensors and through some decision making capability direct the actuators to respond by moving, positioning, regulating, pumping, and filtering, thereby controlling the environment for some desired outcome or purpose. Furthermore, because MEMS devices are manufactured using batch fabrication techniques, similar to ICs, unprecedented levels of functionality, reliability, and sophistication can be placed on a small silicon chip at a relatively low cost. MEMS technology is extremely diverse and fertile, both in its expected application areas, as well as in how the devices are designed and manufactured. Already, MEMS is revolutionising many product categories by enabling complete systems-on-a-chip to be realised.

**MEMS vs NEMS:** While MEMS stands for micro-electromechanical system, NEMS stands for nano-electromechanical system. NEMS would be used in nanotechnology, which is a technology that can manipulate matter at a nanoscale (around the atomic or molecular level). A top-down approach to nanotechnology uses devices that share many similar techniques to MEMS. MEMS and NEMS are sometimes referred to as separate technologies but can be considered as dependent on one another as NEMS technologies are required for NEMS. As an example, a scanning tunnelling-tip microscope (STM), which can detect atoms, is a MEMS device.

Nanotechnology is the ability to manipulate matter at the atomic or molecular level to make something useful at the nano-dimensional scale. Basically, there are two approaches in implementation: the top-down and the bottom-up. In the top-down approach, devices and structures are made using many of the same techniques as used in MEMS except they are made smaller in size, usually by employing more advanced photolithography and etching methods. The bottom-up approach typically involves deposition, growing, or self-assembly technologies. The advantages of nano-dimensional devices over MEMS involve benefits mostly derived from the scaling laws, which can also present some challenges as well.
Some experts believe that nanotechnology promises to: a). allow us to put essentially every atom or molecule in the place and position desired — that is, exact positional control for assembly, b). allow us to make almost any structure or material consistent with the laws of physics that can be specified at the atomic or molecular level; and c). allow us to have

manufacturing costs not greatly exceeding the cost of the required raw materials and energy used in fabrication (i.e., massive parallelism).



An array of sub-micron posts made using top-down nanotechnology fabrication methods.

Although MEMS and Nanotechnology are sometimes cited as separate and distinct technologies, in reality the distinction between the two is not so clear-cut. In fact, these two technologies are highly dependent on one another. The well-known scanning tunnelling-tip microscope (STM) which is used to detect individual atoms and molecules on the nanometer scale is a MEMS device. Similarly the atomic force microscope (AFM) which is used to manipulate the placement and position of individual atoms and molecules on the surface of a substrate is a MEMS device as well. In fact, a variety of MEMS technologies are required in order to interface with the nano-scale domain.

Likewise, many MEMS technologies are becoming dependent on nanotechnologies for successful new products. For example, the crash airbag accelerometers that are manufactured using MEMS technology can have their long-term reliability degraded due to dynamic in-use stiction effects between the proof mass and the substrate. A nanotechnology called Self-Assembled Monolayers (SAM) coatings are now routinely used to treat the surfaces of the moving MEMS elements so as to prevent stiction effects from occurring over the product’s life.

Many experts have concluded that MEMS and nanotechnology are two different labels for what is essentially a technology encompassing highly miniaturised things that cannot be seen with the human eye. Note that a similar broad definition exists in the integrated circuits domain which is frequently referred to as microelectronics technology even though state-of-the-art IC technologies typically have devices with dimensions of tens of nanometers. Whether or not MEMS and nanotechnology are one and the same, it is unquestioned that there are overwhelming mutual dependencies between these two technologies that will only increase in time. Perhaps what is most important are the common benefits afforded by these technologies, including: increased information capabilities; miniaturisation of systems; new materials resulting from new science at miniature dimensional scales; and increased functionality and autonomy for systems.

## III. CENTRAL PROCESSING UNIT

Connected devices are central to IoT. Devices gather data and monitor parameters. They are found on everything from industrial equipment, buildings, and cars, to animals, cargo shipments, pipelines, and people. Hardware and software components are designed for IoT applications via a standard design protocol: specification development, conceptual design, prototype, test, and ultimately rollout of hardware and software integrated into a network. Some platforms, such as Arduino and Raspberry Pi, may expedite design and allow rapid prototyping without involved customisation, thus expediting the time required to implement an IoT configuration. Design will require identification of performance requirements, the necessary hardware and software to achieve such requirements, followed by specifications for components – either commercially off-the-shelf (COTS) or customised designs — with due consideration for the operating environment and the application to be used within it.

In the context of IoT, a device refers to anything that attains information and transmits it. For example, a pressure sensor within a pipeline, a temperature sensor within a refrigerated railcar, or tiny chips inserted beneath the skin of a cow in a scattered heard — are all devices. Devices may stand alone, or they work in sync with other devices.

**IoT Device Characteristics:** As an IoT landscape develops, new devices and platforms will be introduced. There are key characteristics that are common to IoT devices that provide a basis of comparison when selecting hardware and software to configure a new IoT network or to develop and expand an existing one.

IoT devices may be characterised by capabilities:

- Data acquisition and control
- Data processing and storage
- Connectivity
- Power management

**Data acquisition and control:** *Data acquisition* (DAQ) gathers analog information at a fixed time interval (the data sample rate), and transmits it as a digital signal to a remote output device for a digital readout. DAQ may include signal conditioning (to manipulate and scale raw sensor readings), and analog-to-digital converters to convert the analog sensor readings into digital values so that they can be processed and analysed.

*Sensors* are components that measure physical variables and convert them to electrical signals (voltages). Sensors are available, commercially, off-the-shelf, and specified to measure a range of variables which include: temperature, humidity, pressure, smoke, gas, light, sound, vibration, air-flow, water-flow, speed, acceleration, proximity, GPS position, altitude, and a multitude of other variables.

Sensors vary in their functionality and may be used to measure a wide assortment of conditions. For example, proprioceptive sensors monitor the internal state of the device. Sensors such as push buttons, sliding controls, or touchscreens may be used to interact directly with a device (Human-Machine Interface).

Each type of sensor has many options. Manufacturers and their sensor product specifications vary. The sensor’s accuracy, precision, and required operating condition may vary from catalog to catalog and from manufacturer to manufacturer. Sensors may be used within an engine, underwater, embedded in humans or animals, and even in space. Environments require different sensor characteristics and are designed for different uses accordingly.

An important characteristic of sensor components is resolution. The resolution of a sensor represents the smallest amount of change that the sensor can reliably read and is related to the size of the numeric value that is used to represent raw sensor readings. For example, an analog temperature sensor with 10 bits of resolution represents a temperature reading using a numeric value between 0 and 1023. Bits are binary, so 10 bits provides two to the power of 10, or 1024 possible values in total. However, in practice, sensors are affected by electrical noise which reduces the actual resolution.

While sensors convert a physical variable like temperature to an electrical signal, output devices are the inverse: they convert an electrical signal to a physical outcome. Output devices include LEDs, speakers, and screens. This may require *actuators* such as motors, relays, and solenoids that physically actuate something. Actuators are common in industrial IoT applications. For example, pneumatic linear actuators are widely adopted in manufacturing to move and grip products during the assembly process. An actuator on an axle of a school bus may actuate braking when adverse motion is detected. A high temperature in a refrigerated container with produce inside, may actuate an alarm when a temperature rises to a certain unacceptable level.

**Data Processing and Storage:** IoT devices require specific data processing and storage capability. This helps achieve data aggregation, transmission, and analysis. Some IoT devices may process data directly, while others transmit this data to other devices, gateway devices, or cloud applications for further aggregation and analysis.

Edge analytics performs data analysis at the edges of a network rather than in a centralised location. Data can be analysed in realtime on the devices themselves, or it can be analysed on a nearby gateway device (like a router) connected to the IoT devices, in lieu of transmitting large volumes of data upstream to a cloud server or data centre for further analysis. Processing data at the edge aggregates and filters the data as it is collected, with only the most salient data sent upstream. Edge analytics reduces upstream processing and storage requirements alleviates network load.

The processing power and storage used by an IoT application depends on processing required by the services or apps that consume the data. Available memory and processor specifications, clock speed, and number of cores, all of which subsequently determine the devices rate of data processing. The capacity of the non-volatile flash, which is used to persist data until transmitted upstream, determines how much data can be stored on the device. Devices performing edge analytics require substantially more processing capabilities than devices that perform only basic data processing like validating, normalising, scaling, or converting readings.

**Connectivity:** Network connectivity is a defining characteristic of any IoT device. Devices communicate with other devices locally. They then publish data via cloud-based services. Some devices communicate wirelessly, by using 802.11 (Wi-Fi), Bluetooth, RFID, cellular networks, or Low Power wide area network (LPWAN) technologies like LoRa, SigFox or NB-IoT. Wired communication is suited to stationary devices. Such devices may be installed in smart buildings, home automation, and industrial control applications, for example, and connected with Ethernet, or retrofitted with Ethernet over power. Serial communication is a form of wired connectivity between devices, using standard protocols like *Universal Asynchronous Receiver Transmitter* (UART), or the *Controller Area Network* (CAN) protocol, which has its origins in the automotive industry.

**Power Management:** Power management is a critical factor for portable and wearable IoT devices that rely on a wireless power source, such as batteries or photovoltaic cells (solar). Depending on usage patterns and the power requirements of

DOI: 10.17148/IARJSET.2021.8108

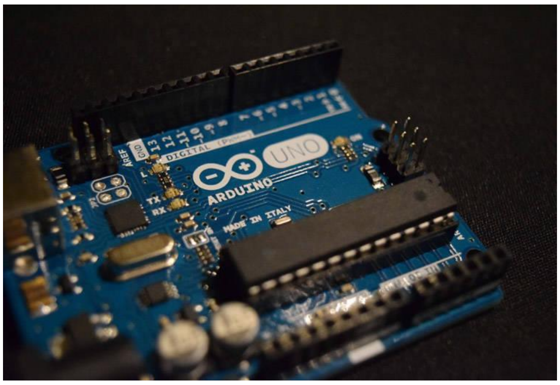
the attached sensors, actuators, or integrated circuits (ICs), a device may be put into sleep mode or into low-power mode periodically to conserve power.
For example, a single-board computer like the Raspberry Pi 4 requires around 700 – 1000mA of current to operate under typical usage. If you were transmitting data constantly over a wifi network, or if the device under heavy load was performing a lot of data processing, the power usage would be high, but then it would drop when the device became idle. If you connect a camera module, the required amperage increases by about 250mA when the camera is in use. Also, sensors normally require power to operate; the GPIO pins on the Raspberry Pi supply 3.3V or 5V, up to a total of 50mA current across all of the pins. The power consumption of the device increases as you increase the number of components that are attached to the pins.

**Hardware and prototyping:** Developing IoT applications is more accessible with the growing availability of low-cost, commercially available off-the-shelf hardware development boards, platforms, and prototyping kits. Modular hardware designs provide great flexibility. With a greater selection of components, designers may substitute new sensors with different specifications. Alternatively, you can independently upgrade the networking, data processing, or storage modules of a device for evolving requirements.
Many commercial off-the-shelf hardware devices, including microcontrollers and single-board computers, are designed around *System-on-a-Chip* (SoC) integrated circuits. SoCs bundle capabilities such as data processing, storage, and networking onto a single chip. This configuration means that you sacrifice some flexibility for the sake of convenience, but, fortunately, there are a huge number of commodity devices available with a range of configurations to choose from.

**Microcontroller Development Boards:** A *microcontroller* is a SoC that provides data processing and storage capabilities. Microcontrollers contain a processor core (or cores), memory (RAM), and *erasable programmable read-only memory* (EPROM) for storing the custom programs that run on the microcontroller. *Microcontroller development boards* are PCBs with additional circuitry to support the microcontroller to make it more convenient to prototype with and program the chip.
Sensors and actuators connect to the microcontroller through digital or analog *General Purpose Input/Output* (GPIO) pins or through a hardware bus. Standard communication protocols like I2C and SPI are used for intra-device communication with the components that are connected with the bus. Adopting standards makes it easier to add or swap out components that are connected with the bus.
Arduino (http://arduino.cc/en/Main/) is an open source device platform, with an active community who are creating compatible development boards and tooling. Device capabilities vary across the official Arduino models (https://www.arduino.cc/en/Products/Compare)), and also between the dozens of third-party compatible boards. All of the devices in are Arduino-compatible microcontrollers, including the ubiquitous Arduino Uno, Particle's Electron, which includes an integrated cellular modem, and Espressif Systems' (https://espressif.com/en/products/hardware/esp8266ex/overview) ESP8266-01, a low cost, low-power microcontroller with integrated wifi.

Like Arduino, the ESP8266, has an active community of adopters. Notable development boards that are based around the ESP8266, include NodeMCU, WeMos D1, and AdaFruit's Feather Huzzah (https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/overview). A number of alternative firmware options for ESP8266-based boards have been developed by the open source and maker community, enabling IoT developers to program for these boards using Lua, Python, and JavaScript, and to support over-the-air (OTA) updates.

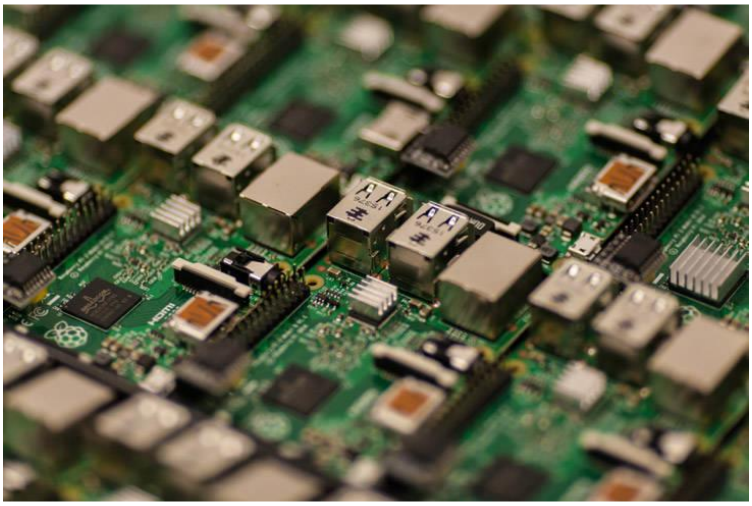
The standard approach for developing the software to run on Arduino-compatible microcontrollers is to use C or C++ and the Arduino IDE, however community-developed language bindings and visual programming tools also exist. Arduino-compatible boards that share common pin layouts are able to be expanded by using optional third-party shields,



IARJSET
ISSN (Online) 2393-8021
ISSN (Print) 2394-1588
International Advanced Research Journal in Science, Engineering and Technology
Vol. 8, Issue 1, January 2021
DOI: 10.17148/IARJSET.2021.8108

for example, to add an Ethernet port or Bluetooth to an Arduino Uno. Arduino is the most widely adopted hobbyist microcontroller development environment, but others like Tessel (https://tessel.io/) and Particle.io (http://particle.io) natively support JavaScript, while Python is supported for boards like MicroPython's PyBoard (http://micropython.org) and by WeIO (http://we-io.net/hardware/).
Selecting an Arduino-compatible microcontroller makes it easier to port programs that are developed using the cross-platform Arduino libraries and Arduino IDE to run on other Arduino-compatible devices. Working through and around subtle difference is still required.
For example, the Arduino Uno uses 5V logic on digital I/O pins (where 0 volts equals LOW or OFF and 5 volts equals HIGH or ON), but the ESP8266 and Particle boards use 3.3V logic (HIGH is 3.3V). This might affect your choice of sensor or actuator components, as some components only work with one or the other. Swapping sensors that are designed for 5V to 3.3V logic might result in unpredictable results and possibly damage the pins that are intolerant to higher voltages, and so you'd need to add a logic-level converter to make this work. When you get down to implementing low-level hardware features like enabling deep sleep mode or reading from connected sensors by using specific protocols, you'll likely need to rely on device or component-specific libraries that will make your code less portable.

Single Board Computers: *Single board computers* (SBCs) are a step up from microcontrollers. They allow you to attach peripheral devices like keyboards, mice, and screens, as well as offering more memory and processing power (for example, a 1.2 GHz 32-bit ARM microprocessor from compared to an 8-bit 16KHz microcontroller from ).



The distinction between microcontrollers and single-board-computers is somewhat arbitrary. Some devices, like the Onion Omega 2 (https://docs.onion.io/omega2-docs/omega2.html#omega2), fall somewhere in between, with almost as much on-board memory and processing capability as a low-end SBC. There are also a number of hybrid devices, like the UDOO Quad (http://www.udoo.org/docs/Introduction/Introduction.html) that integrate an ARM-based Linux system with an Arduino-compatible micro-controller.

As with microcontrollers, SBC device capabilities can be expanded through the addition of stackable expansion boards known as *hats* on Raspberry Pi and *capes* on BeagleBone Black, and through the addition of external modules, such as motor controllers or analog-to-digital converters, to mitigate limitations with the built-in device capabilities.
Many SBC devices are more like a mini-PC, and run an embedded operating system, typically a streamlined Linux distribution. As a result, there are many more development tools and language choices that are available for developing embedded applications that work with the attached sensors and actuators on these devices than on microcontroller boards. However, SBCs are more complex to set up, larger, more power hungry, and more prone to problems like corruption of the SD card or flash memory where applications are stored.

Choosing between Microcontroller and Single Board Computer: Although off-the-shelf microcontroller development boards and single-board computers might only be part of an IoT solution, they are ideal for bootstrapping the development of one.
One way to get started is to consider the key IoT device characteristics in light of your application's requirements, and then work through the following design decisions:

- Determine the type and number of peripheral sensors and output components that you need, and, if necessary, any design circuits for these components

- Select a microcontroller or single-board device to co-ordinate reading from and controlling the peripheral components
- Decide on the data communication protocols that you need to use for intra-device communication (for example, using I2C for communication between the microcontroller and any attached sensors)
- Select the networking hardware and protocols that you need to use to communicate with cloud services and apps

For example, to set up a home automation system on a budget, you may choose the Raspberry Pi Zero W. It is a small and very low-cost SBC device with ample processing power and memory (1GHz ARM6 processor and 512 MB RAM) to perform data processing and analytics on the device. It supports microSD card flash memory expansion up to 64GB for storing programs and data. And, it is equipped with a full 40-pin GPIO header, just like the Raspberry Pi 4, which allows connecting multiple sensors and supports both SPI and I2C protocols. It has on-board wifi for connecting with a home network, and it can be powered with micro-USB off a portable power pack or wall power supply.

As you progress further with your IoT landscape design you may stop and compare your anticipated performance with design intent. Conduct device design and prototype, embedded software selection, and the selection of upstream services and apps, but then stop to assess it. You can periodically assess your prototypes against your functional and non-functional requirements, including performance, reliability, and security, and revisit these choices as necessary.

## IV. CONCLUSION

Every application and situation is different. There is no “one-size-fits all” approach to selecting hardware for IoT projects. Adopting standards-based, commodity hardware like microcontrollers, or single-board-computers, can save time and expense in the early stages of development, without sacrificing flexibility. What you learn in the prototyping phase can help make critical hardware design decisions later in deploying your IoT solution. The research paper has successfully listed critical components and methodologies to implement IoT in Mechanical systems. The sensors play an important role to collect data and processing boards will run the required algorithms to give the output.

**Future Development:** The IoT systems can be used to run Machine learning algorithms and Artificial Intelligence for accuracy, data is generated from different sources with specific data types, it is important to adopt or develop algorithms that can handle the data characteristics. Second, the great number of resources that generate data in real-time are not without the problem of scale and velocity. Finally, finding the best data model that fits the data is one of the most important issues for pattern recognition and for better analysis of IoT data. These issues have opened a vast number of opportunities in expanding new developments. Big data is defined as high-volume, high-velocity, and high variety data that demands cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.

## REFERENCES

1. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. Computer Networks, 54(1), 52787—52805..
2. Bradley, J., Barbier, J., & Handler, D. (2013). Embracing the Internet of Everything to capture your share of $14.4 trillion. Cisco White Paper. Retrieved from http://www.cisco.com/ web/about/ac79/docs/innov/IoE_Economy.pdf
3. Chui, M., Lo¨ffler, M., & Roberts, R. (2010). The Internet of Things. McKinsey & Company. Retrieved from http://www.mckinsey.com/insights/high_tech_telecoms_internet/the_internet_ of_things
3. Fichman, R., Keil, M., & Tiwana, A. (2005). Beyond valuation: ‘‘Options Thinking’’ in IT project management. California Management Review, 47(2), 74—96.
4. Gartner. (2014, March 19). Gartner says the Internet of Things will transform the data center. Retrieved from http://www.gartner.com/newsroom/id/2684616
5. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013).
6. Hewlett Packard. (2014, July 29). HP study reveals 70 percent of Internet of Things devices vulnerable to attack. Retrieved from http://www8.hp.com/us/en/hp-news/press-release. html?id=1744676#.VOTykPnF-ok
7. Li, X., & Johnson, J. (2002). Evaluate IT investment opportunities using real options theory. Information Resources Management Journal, 15(3), 32—47.
8. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., et al. (2011, May). Big data: The next frontier for innovation, competition, and productivity. McKinsey & Company. Retrieved from http://www.mckinsey.com/ insights/business_technology/big_data_the_next_frontier_for_innovation
9. Sundmaeker, H., Guillemin, P., Friess, P., & Woelffle´, S. (2010). Vision and challenges for realising the Internet of Things. Accessible at http://www.researchgate.net/publication/ 228664767_Vision_and_challenges_for_realising_the_ Internet_of_Things
10. TRUSTe. (2014). TRUSTe Internet of Things privacy index——US edition. Retrieved from http://www.truste.com/resources/ privacy-research/us-internet-of-things-index-2014/
11. White, C. C., III, & Cheong, T. (2012). In-transit perishable product inspection. Transportation Research Part E: Logistics and Transportation Review, 48(1), 310—330.