Lesson 2 – Introduction to Automation

Self Driving Car of the future
Fundamentals you learn in SCRATCH here, apply to this as well

At its simplest, automation means to **make something automatic**.

In manufacturing, or at home, automation refers to **performing one or many tasks autonomously** with minimal or even no human interaction.

By now we have **learnt** all the basics of Scratch.

Let us **apply them** to our daily life & see how machines work for us?

This chapter will be focused on:

- Introduction to **automation.**
- Peep into the **fascinating world of Devices.**
- **Role of sensors** in automation & coding.

These will later be supported with basic sensor based projects**.**
What applies to basic projects, applies to advance multi sensor projects.

What is Automation

The dictionary defines **automation** as “the technique of making an apparatus, a process, or a system operate automatically”

Automation is "the creation and application of technology to **monitor and control** the production and delivery of products and services”

Automation describes technologies that **reduce human intervention** in processes. **Coding sits at its core.**

TODAY's Most Pressing Need - Automation Professionals

Think about it:

- The cell phone and computer you use every day to do your job.
- The car you drive or take to work.
- Think about the television you watch or your music system.
- Anything you can think of **is the result of complex processes.**

Automation professionals are responsible for:

- Solving complex problems of industry and its processes.
- Their work is critically important to safety, security, well-being, health, and to sustain & enhance our quality of life.

Tomorrow, you Can be One of Them.

History of Automation

Prior to nineteenth century almost **everything was manual.** Example:

- If a light had to be switched on, it had to be done manually.
- If a piston had to be moved to pump water, it was done manually.
- If a farmer had to decide when to irrigate his crop, he did it manually.
- If a student did calculations or write, he did it manually.

In the Nineteenth Century things Started Changing

World saw the emergence of automation though **mechanics & pneumatics.**

A pump using steam made it possible **to pump water - automatically.**

Along with **mechanics** it gave us the steam engine. Examples are countless.

World of Sensors, Controllers and Intelligence

With development of sensors, it became possible to switch on and switch off lights **automatically.**

With development of controllers, automation became **programmable.**

With this it became possible:

- For farmers to **irrigate** their crops automatically.
- For students to **do calculations and write** automatically.

Developments are paving the way for machines with intelligence.

Future has no restrain on possibilities.

The Reality is that, children need to learn basics of **Machine intelligence** to:

- Ascertain their interest **using Scratch** in primary school.
- Continue to **develop interest** in technology whilst in middle school.
- **Attain expertise** and **migrate** to higher programming platforms of Python & Embedded or Arduino C whilst still in school.

Enabling them to leave school with **a tested road map of their future.**

Devices - The Enablers of Higher Learning & Coding

Purpose of learning Scratch is to form a foundation for **Doing the real stuff.**

Real stuff is to programme machines **to do what you desire them to do.**

Sprites are suited to 2 D applications like **Games** only.

For 3 D real life applications we require **Devices.**

What do Devices Teach that Sprites do not

Devices **connect** basic engineering, science & math **concepts to the real world.** Children see different engineering & scientific **theories at work.** These include:

- Functioning of **mechanicals.**
- Role of **physics.**
- Experimenting with **electricity & electronics.**
- Applications of **sensors, processors** and communications.
- Influence of **maths in bringing** them to life and in understanding STEM.
- Above all **coding the Device** to achieve the desired results.

Real Life Example of a Device

Typical example is a **cell phone** having:

- **Input sources** like, keys or buttons, microphone, wi-fi etc.
- **Output sources** like, a screen, loudspeaker etc.

In this device each one of these sources or component is individually programmable to perform a given function. For example:

- Mic could be programmed to **talk to Siri.**
- Individual keys programmed to **display results on the screen.**
- Wi-Fi could be programmed to **fetch inputs from the internet.**

This is where the **child's imagination starts taking the driver's seat.**
Our life has examples of automation all around it.

Selection of Devices to learn Coding

A device to learn coding:

- Should be **as close** to the real stuff as possible.
- Should be **simple enough** for the child to understand the theory behind its app.
- This learning should be based on a **single teaching software.**
- Teaching should be as **device neutral** as possible.
- Expandability with low cost of ownership.

mBlock 5 - Software from Makeblock is a unique platform that:

- Enables you to **code** both sprites & devices.
- Enables you to **code** in Scratch, Python & Embedded C.

Makeblock offers an A to Z array of world class devices for project based learning.

Choice of Devices. For serious learning, you require to **invest in** devices.

Projects in this book **make use of** devices from Makeblock.

Principles that apply to these devices, **apply** to any others you may use.
Procedures could vary.

We introduced you to some earlier. We shall keep introducing more, as we go ahead.

Our First Device

As a start, we started **making use of mBot,** a programmable robotics kit of mBot family of Makeblock.

This kit **keeps growing** with the child:

- **From primary school to college** through add-on packs.
- **From teaching** Scratch to teaching Python & Embedded C.

What you buy for primary school, will **teach you for years.** Possibilities are endless.

Coding and the World of Sensors.

A **sensor is an electronic** device that senses the environment to create an **input,** processed to **generate an output** required to **trigger specified actions**.

Att photo shows **a gesture sensor**.
It senses the environment to **pick up changes** as input to be processed and displayed on the **goggle to enable VR gaming.**

Thus, 2D gaming you learn today, will help you become a VR gaming professional tomorrow.

How do sensors work?

A **sensor** converts stimuli such as
sound, moisture, wind, pressure etc into **electrical** signals.

Talking to Siri, we convert sound to an electrical signal which then interacts with Siri to give us an audible output that we hear.

These signals are then converted into a **binary code.**
The binary code then goes to a computer to be **processed.**

Modern **sensors having integral processors** are very potent devices.

Arduino – The Consolidated Chip Set

Chip sets are a **hardware device.**
Remember the words of the famous ad **“Intel Inside”.**
They depend on **external software** sources to do the processing.

Automation & coding is all about **processing one input to give one output.**
Devices working on a chipset using external software power, to do a function as simple as ‘one input – one output’ is an **overkill** for most applications.

Inventors of Arduino came up with a tiny chip that does its own **processing.**
Due to this capability, Arduino sits at the very core of current & future automation.
The ad of tomorrow would be **“Arduino Inside”.**

Sensors and Robotics

Sensors & Arduino **sit at the heart** of robotic industry.
They allow the robot to be **informed** of the surrounding environment
and so **facilitate** it to go with the necessary operations.

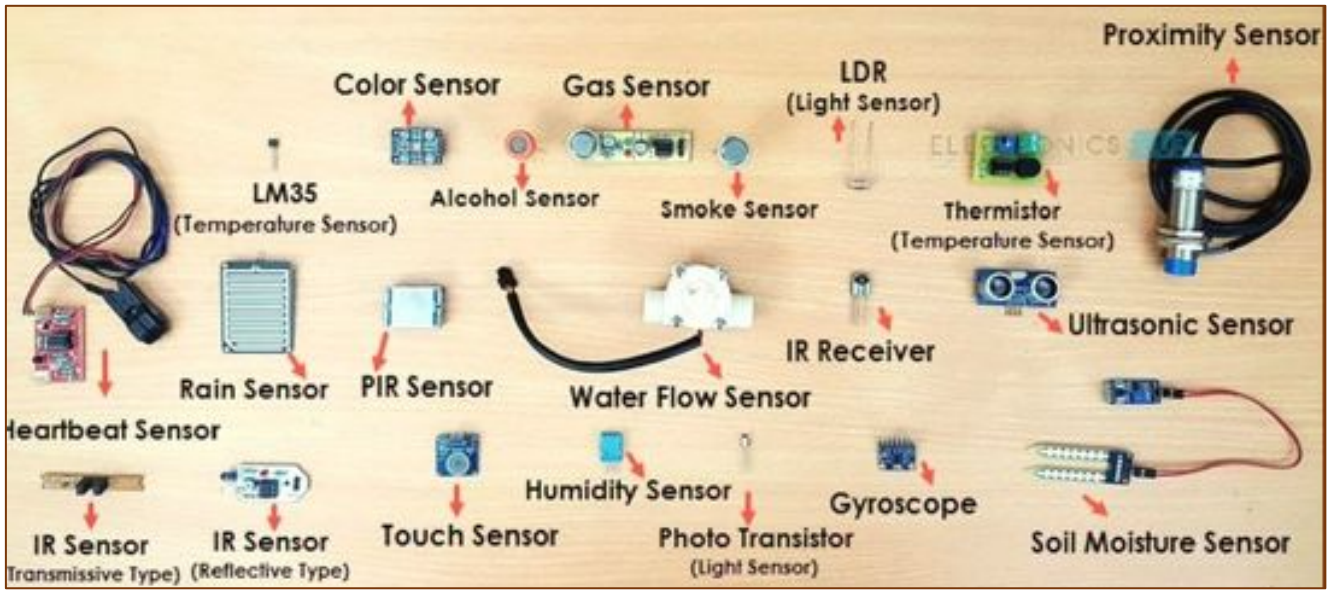
Without sensors, robots can perform only a few **monotonous activities.**
With sensors they can perform many **high-level** operations.

There is **no limit to the number of sensors** in a robot.
The principles that apply to one **apply to all.**

A good site for sensors is -
https://www.elprocus.com/types-of-sensors-with-circuits/

Types of Sensors. Our life is **full of sensors** implemented in different forms. These are broadly classified as **analogue sensors and digital sensors.**
While the list is huge, some common examples include:

- Temperature Sensor.
- IR Sensor.
- Ultrasonic Sensor.
- Gyroscope.
- Touch Sensor.
- Moisture Sensor.
- Gas Sensors.
- Colour Sensor.
- Microphone.
- Wi-Fi module.
- Bluetooth Module.



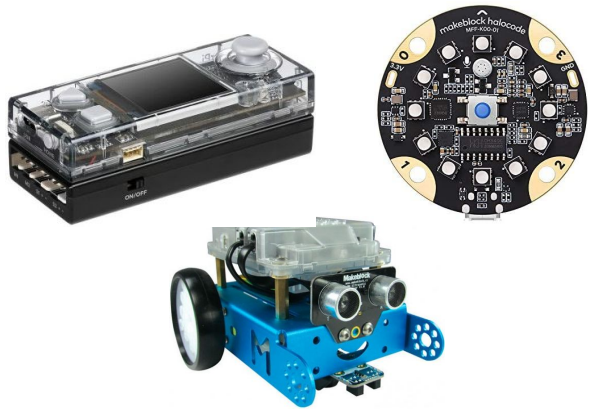
Devices for working with Sensors for learning Automation Fundamentals

We will be using devices of Makeblock.
The three prime devices used by us are
mBot, HaloCode & Cyber Pi.

They go hand in glove for experimentation & learning.
They make a **great learning platform.**

Cyber Pi enable you to **use third party sensors.**
Cyber pi and HaloCode will be your
companion to learn Python.

mBot will be your **buddy to migrate to Arduino C.**



Code Karega India Badhega