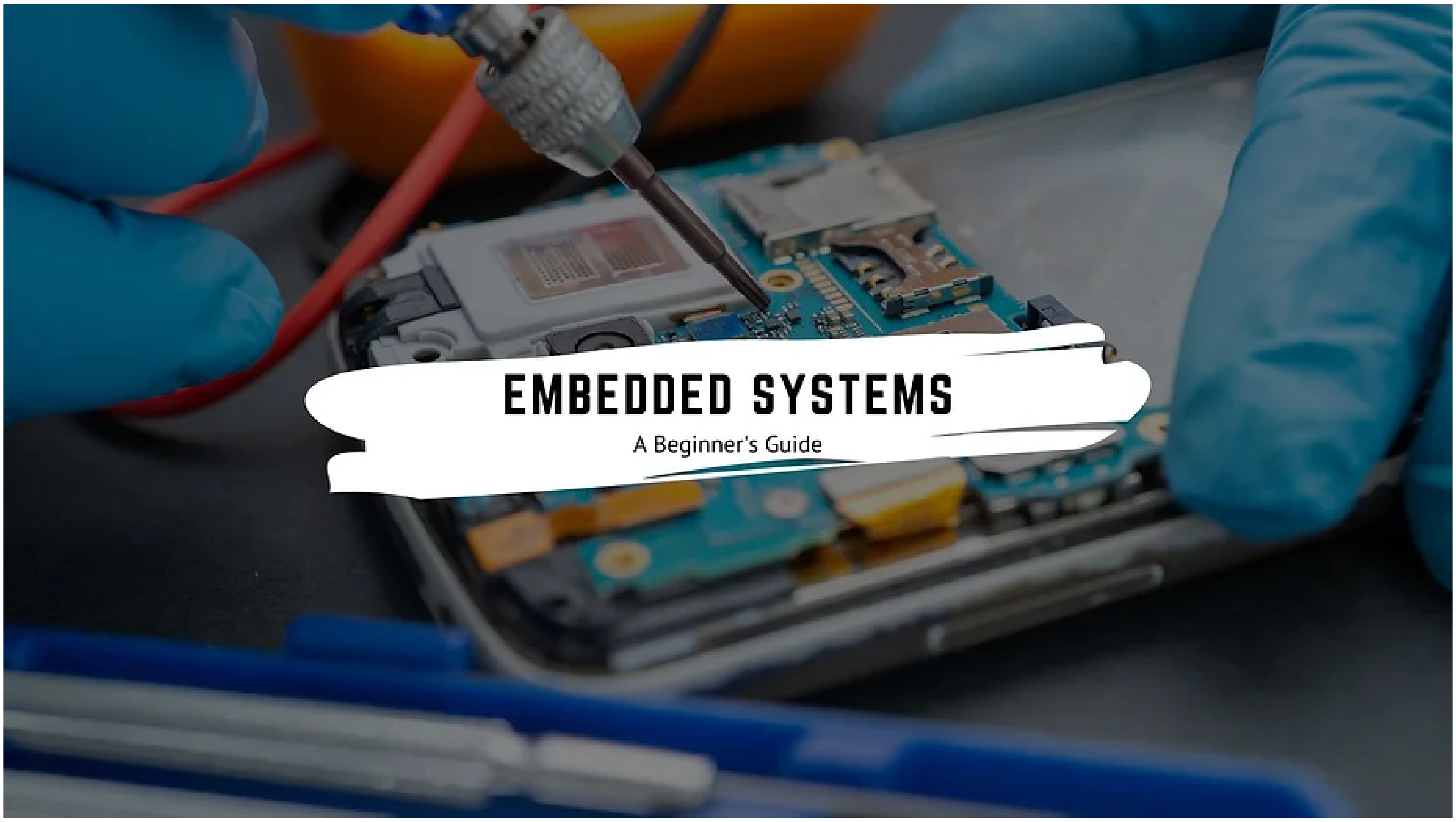
Datt Panchal Mar 22, 2023 9 min read

A Beginner's Guide to Embedded Systems

Updated: 2 days ago

Know Everything About an Embedded System From Hardware to Its Programming and Future Opportunities



Beginner's Guide to Embedded Systems (Photo created with Canva)

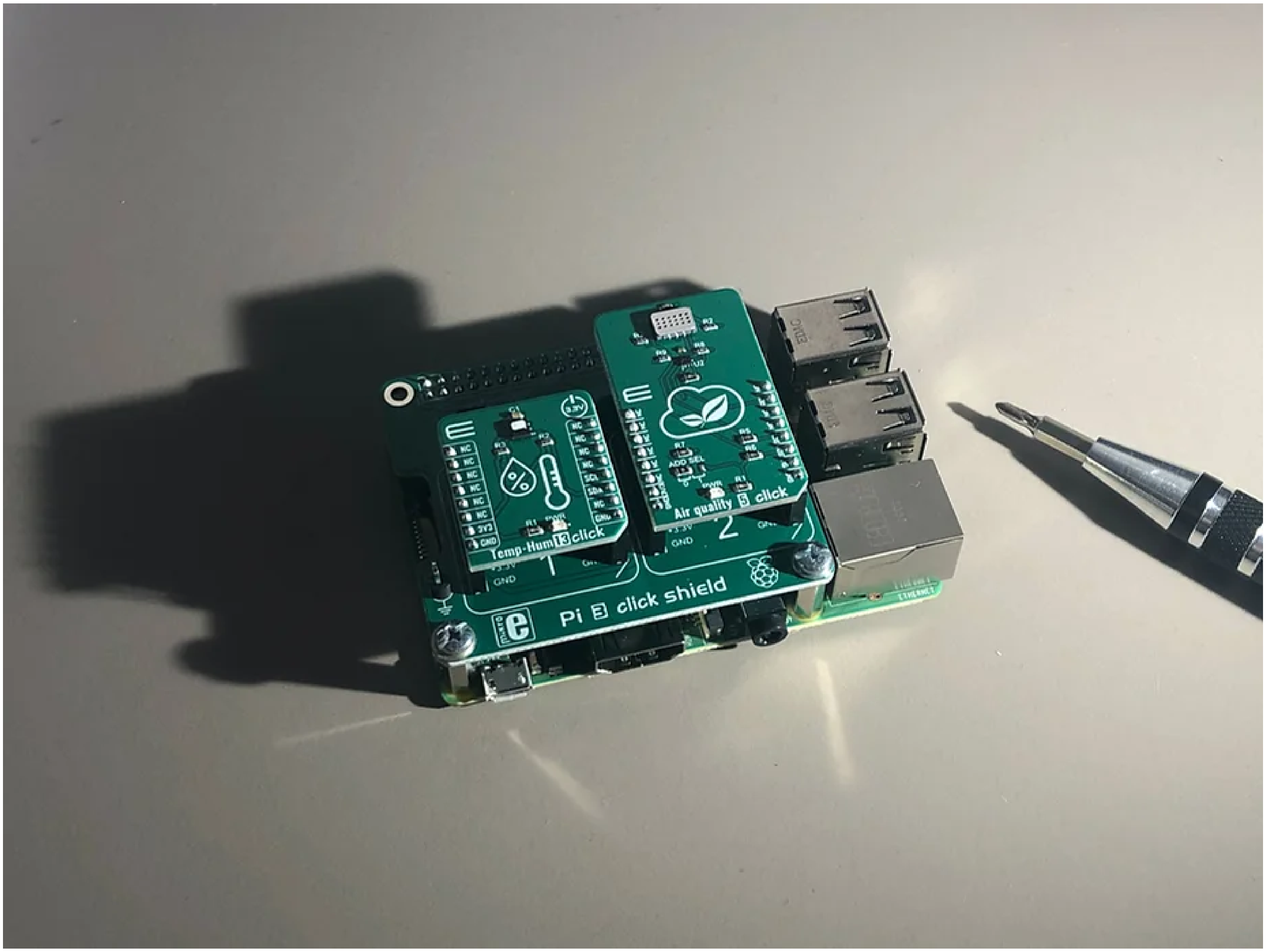
Contents

Introduction

Technology has become so impressive. Every day something new is coming into the tech world such as new electronic devices and gadgets, smart IoT and AI-based devices, wearables, etc. This all technology makes people's lives efficient, convenient, and productive.

But, do you know what is the basic thing behind all of these technologies? Yes, it's an Embedded System. Without an Embedded System, Electronics, IoT, and AI are not possible.

That's why I am coming up with one more interesting and exciting blog post on a **Complete Beginner's Guide to Embedded Systems**. By reading this post, you can clearly understand the Embedded System and if you want, you can also make a career in this area.



A PCB (Source: Unsplash)

I'm so excited about this post, let's start without any more time delay!

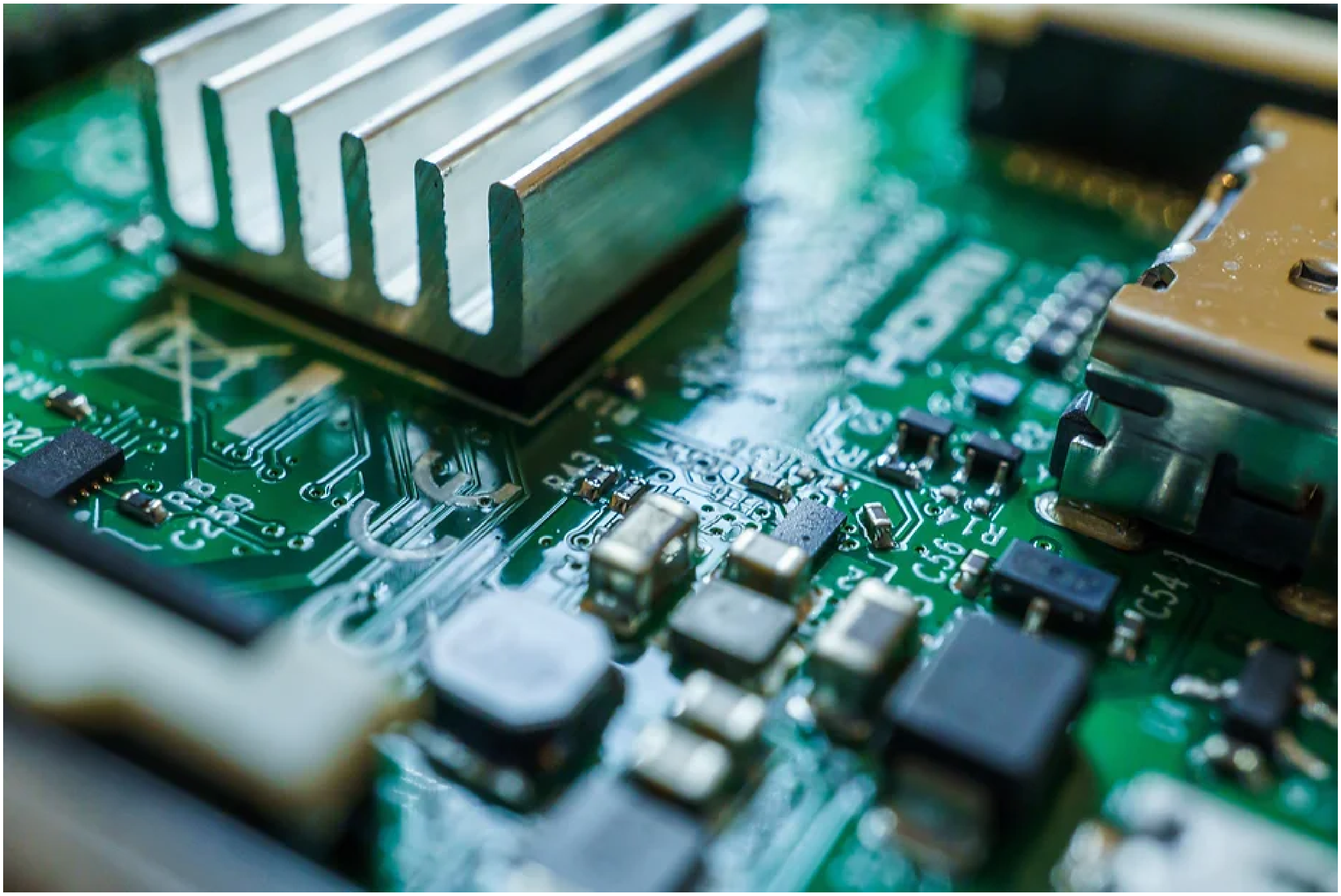
What Are Embedded Systems? A Beginner's Guide to Embedded Systems

Embedded Systems are a **combination of hardware as well as software that performs a specific task repeatedly**. Now, what it means?

In Embedded Systems, we used electronic components and systems as hardware such as Microcontrollers or Microprocessors, Memory, Input-Output Devices, etc. We will discuss it soon.

To control the entire system, make this hardware functional, and connect different electronic systems, we used software. Programming the electronic components/devices and PCB designing is part of the software.

These all parts are combined and worked together to do a particular task continuously means it repeats the same task a number of times. We will see some examples of Embedded Systems after one more point.



Embedded System (Source: Unsplash)

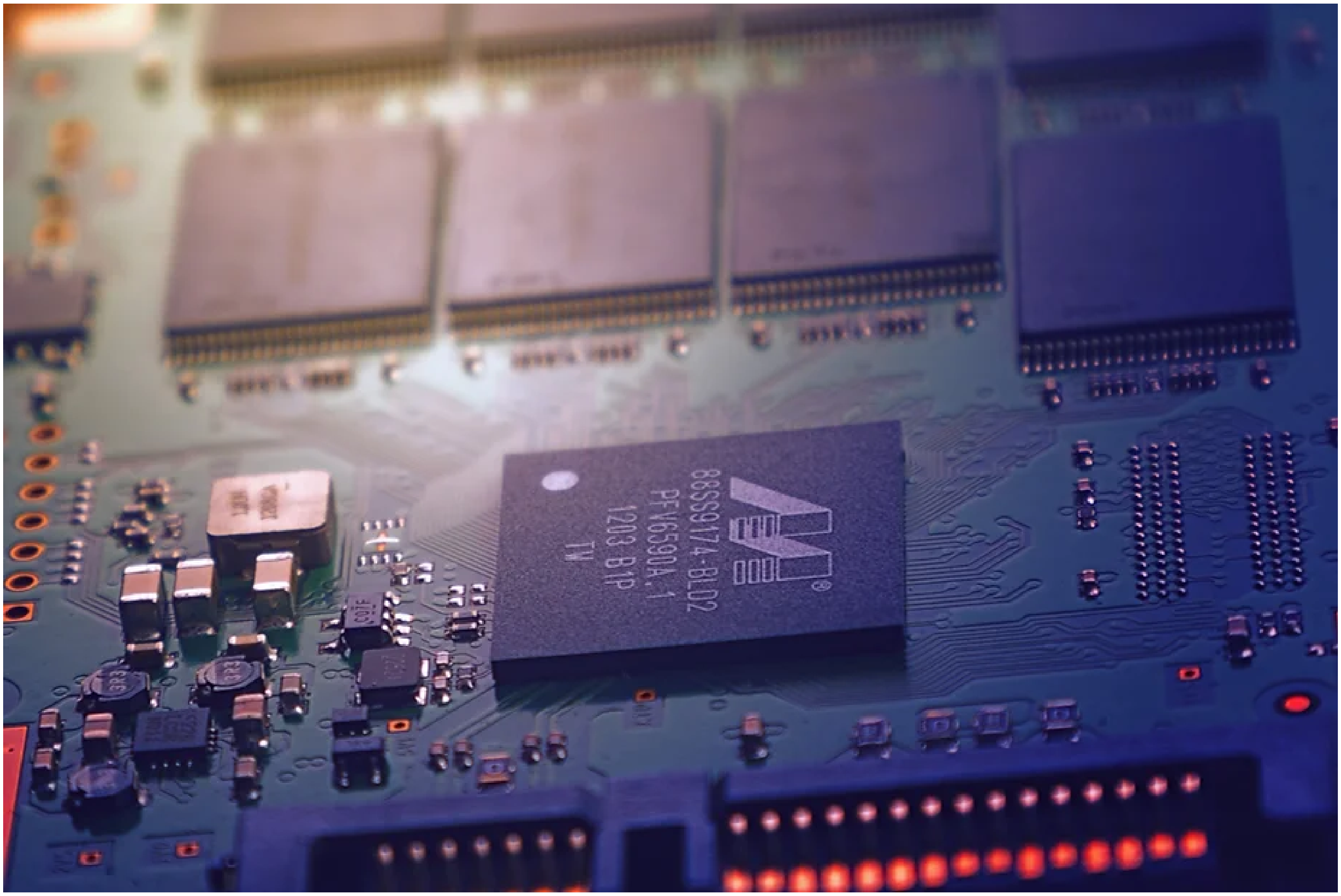
Importance of Embedded Systems

Embedded Systems are very important when we talk about any electronic systems such as mobile phones or computers.

Every person wants their electronic systems should be more efficient, reliable, highly customizable, consume less power, provide flexibility and efficiency, have a longer life span, and be available at low and affordable costs. Right?

Embedded Systems provides these all functions for you and your electronic device. To provide efficiency and make a big difference in people's lifestyles with electronics is the only goal of Embedded Systems.

IoT and AI-based devices are also one kind of Embedded System with some more functionality and features. That's why Embedded Systems are much more important in today's digital world and for the future.



Components of Embedded System (Source: Unsplash)

Examples of Embedded Systems

Here are some examples of Embedded Systems with their primary function:

1. **Mobile Phones and Computers:** Provide a communication link between people, to access and use the Internet and social media, etc.
2. **Refrigerator**: Used to provide cooling to foods and vegetables to keep them fresh for a long time, also used to cold the water, etc.
3. **Air Conditioner**: To control the temperature and humidity in the room or building
4. **Automatic Teller Machine (ATM)**: To provide bank services such as withdrawing cash, deposits, etc.
5. **Digital Camera**: To take pictures and store them in memory in digital format.
6. **Smartwatches**: To show the time with more cool features like a fitness tracker, health checker, GPS, etc.
7. **GPS Navigation System**: To show the actual location and path of a particular destination.

These are only a few examples of embedded systems. Embedded systems are any electronic devices that speed up, facilitate, or otherwise improve human activity.

Components of Embedded Systems

The components or hardware of Embedded Systems are part of a system that works together and makes the task possible. Here are the main components that every Embedded System must have:

1) Microcontroller or Microprocessor

A microprocessor is the **brain of the entire system that controls or manages all tasks** that happen in the system. It is a chip that takes the data from input devices, processes that data according to the program, and gives final data to output devices. It has no input-output devices and memory itself, we have to connect it externally.

Microcontroller is the same as a microprocessor but it has some amount of input-output ports and memory to store the data. It contains a processor, input-output ports, and memory.

Both are used according to different requirements of the system. It speeds up our system and makes it more efficient for people.

Microprocessor (Source: Unsplash)

2) Memory

Memory is a **storage space where we store our data.** The data may be either stored from input devices or stored while processing.

Different types of memory are used for Embedded Systems such as RAM, ROM, EROM, EEPROM, Flash memory, etc. All have different storage capabilities and functions.

3) Input/Output Devices

We used input and output devices in our embedded systems to **take the data and give it to output devices after processing**.

For example, a Keyboard is an input device. When we press any key, the data of that key is sent to the processor or controller that processes that data like which type of key is pressed, and after completing the processing, it gives the data to the output device which is our screen or display.

4) Power Supply

Without a power supply, no one electronic device can operate. The power supply is **used to provide the required power for the device to operate**. DC and AC power supply is used in Embedded Systems.

The basic embedded system architecture is shown below:

Block diagram of basic embedded system (Created with Canva)

Also Read: A Complete Guide on Embedded System Hardware Components

Types of Embedded Systems

Here are the types of embedded systems mentioned which have different use cases with different requirements:

1. **Stand-alone**: Stand-alone embedded systems are independent systems that are not connected to other systems or networks. They are fully working alone. Examples of stand-alone systems are calculators, smart watches, digital cameras, etc.
2. **Real-time System**: Embedded systems can be real-time which means it performs the task in real-time with some small time delay (In microseconds or milliseconds or a few seconds). For example, Industrial machines, automated robots machines, medical devices, etc.
3. **Networked**: A networked embedded system means they work in a network to perform a task. Many Embedded Systems are connected in networks and try to achieve their goal. Examples include IoT devices, traffic controllers, etc.

These are some types of embedded systems you should know!

Programming Embedded Systems

So far we have seen the hardware that is used in embedded systems, now let's discuss the software part where programming and embedded system tools are included.

The embedded system does not work itself, it needs instruction or command to start its process. To give the instruction or commands to the system, we are not using any human language, we are using a programming language. It's a machine!

Using a programming language, we can communicate with a computer system and give commands to control it. In embedded systems, we give instructions to the microprocessor to program it and control the entire system.

Programming (Media from Wix)

Overview of Programming Languages for Embedded Systems

We used many different programming languages for embedded systems, but some of the most used programming languages are listed below:

C/C++

C is a **high-level programming language that boosts a system's speed and effectiveness**. The majority of people favored this language since it has numerous capabilities and functionalities for programming embedded systems. It is used to program development boards with microcontrollers like Arduino, NodeMCU, etc.

Since C++ is an **intermediate language**, it offers greater opportunities for both low-level and high-level system programming.

Start with Arduino if you want to learn C programming and how to work with embedded systems!

Python

Python is a **very high-level, beginner-friendly, and easy-to-learn-and-use programming language**. It is another great option to program the embedded system. Python is used to program high-level machines such as robotic systems, industrial machines, etc.

Python is also used with Rasberry Pi, if you learn how to program embedded systems with Python, start with the Rasberry Pi development board!

Programming (Media from Wix)

Assembly Language

Assembly language is a **low-level language where you can program the system with mnemonics (A short form of a long programming statement).** In industry, Assembly language is absolute, but in any area, it is still used. I recommend starting with C programming or Python, avoid Assembly as much as you can!

These are some programming languages used in embedded systems, now let's see what are programming concepts used in embedded system programming.

Basic Programming Concepts for Embedded Systems

There are many programming concepts are used for embedded systems but some of the most used concepts are mentioned below:

1. **Data Types:** Data types are commonly used programming concepts where you should learn different data types such as Integer, Float, Character, etc.
2. **Loops and Conditional Statements:** Loops and conditional statements such as the If-else ladder are the most used concepts.
3. **Input-output Operation:** You should learn the input and output operations such as how to read data from sensors and give it to the output device, etc.
4. **Interrupt and Even-driven Programming**: Where you learn how to handle any interrupt (An event generated automatically during the execution of a program).
5. **Memory Management:** It is a very important concept you should learn if you working with memories.

These are the most used programming concepts in the embedded system!

Embedded Development Tools

Embedded development tools are mostly used for compiling, debugging, simulating, and testing the embedded system. The most common embedded development tools are:

1. **Compilers:** Tools that compile your system program and warn you if it detects any error
2. **Debugger**: Tools that help you to find the error in the system
3. **Simulators**: Tools that help you to simulate your electronic circuit

Integrated Development Tools (IDE) are also used to write and compile the programs for the system. The most used IDEs are Arduino IDE, Eclipse, and Visual Studio Code.

Coding (Media from Wix)

For hardware testing, you can use tools such as Emulators and JTAG debuggers. If you want to learn more about embedded development tools, you can join online forums such as this and communities of embedded developers such as Dev.io.

Designing an Embedded System

To design any embedded system, you should consider some things, here are some important points that you should keep in mind when developing an embedded system:

Requirements Gathering

Before starting to develop of embedded system, you should spend some time **gathering some requirements of the system**. For example, what is the purpose and goal of the system, and how can achieve it?

After you get the answer, you can move on to collecting hardware and software for your system like which hardware and software are needed. Take some feedback and discuss these requirements with other team members and stakeholders.

Design Consideration

After you gather all requirements and set up your plan, **go for design consideration where you will think about the design or structure of the system.** Choose the required hardware or components, software that you will use, etc.

Other factors like power usage, performance, system cost in its entirety, etc. should be taken into account while designing a system. You must consider the future as well, including questions like whether your design will be scalable, whether modifications are conceivable, and other issues.

Electronic component placement on PCB (Source: Unsplash)

Prototype Development

Your requirements are clear, you have the necessary hardware and software as well as design concepts, and you are now prepared to **construct a system prototype.** Utilizing the system's chosen components, begin prototyping the system.

Develop your prototype while keeping your requirements and design in mind. When everything is done, you are prepared to go on to the next phase.

Testing and Debugging

After you build a prototype for your system, it's time to test it. **Test your system using the embedded development tools** that we discussed already, if you find any error, debug it and try to solve it.

Test whether your system works based on your requirements or not. Improve your system until your system is perfectly working on functions.

So, these are the important steps that you should consider while designing an embedded system.

Applications of Embedded Systems

The applications of embedded systems are listed below:

1. **Automotive**: Embedded systems are mostly used in automotive industries such as manufacturing where robotic and automated systems are widely used. At that time, embedded systems played a major role.
2. **Medical Devices**: Embedded systems are used in medical devices to monitor people's health, so they can identify if there is a health issue before it becomes a big problem.
3. **Consumer electronics**: Embedded systems also play a very big role in consumer electronics products such as televisions, Refridgerators, Mobile phones, Computers, etc.
4. **Industrial Control Systems:** As increasing people demand market products, the industry needs high-tech machinery to do work faster and more efficiently. Embedded systems were used at that time to create control and robotic systems.
5. **Others**: Wearables and Robotic systems are also applications of embedded systems.

These are some real-world applications of Embedded systems.

Robotic hand (Media from Wix)

Future of Embedded Systems

The future scope of Embedded Systems is increasing day by day with increasing electronic technologies. Artificial Intelligence, Machine Learning, The Internet of Things (IoT), 5G Technology, Robotics, Autonomous vehicles, Automation in industries, AR & VR, and ICs, all are based on Embedded Systems.

All the latest technologies consist of embedded systems. If you know and understand the embedded system well, you can develop these types of technologies or you can make your own revolutionary thing with an embedded system.

By seeing the current technology trends, the future is going in the direction of electronics technologies. The future of Embedded Systems is so impressive and stunning.

Also Read: Top 10 Most In-Demand Electronics Fields for Future

Future and Career Image (Media from Wix)

So, why do you wait for? Start learning, exploring, and building an Embedded System, and take your future to the next level!

Final Thoughts

We have discussed the complete beginner guide to embedded systems. We have seen lots of things about Embedded Systems in this post. We know everything about it from a basic introduction to building an embedded system.

We have discussed the embedded system and its importance with some real-world examples. We know the components of embedded systems where we saw different hardware parts. We also mentioned the types of embedded systems including Stand-alone, Real-time, and Networked.

Then we discussed the programming of the embedded system where we have seen the different programming languages that are used, the important programming concepts, Embedded Development Tools, etc.

We also described important steps for designing an embedded system. After that, we mentioned some real-world applications of embedded systems and at the end, we discussed the future of embedded systems.

I appreciate you joining me on this amazing adventure. I hope you enjoyed and learned something from this blog post.

Thank you for giving your valuable time to read!